Security Amplification against Meet-in-the-Middle Attacks Using Whitening

Pierre-Alain Fouque, Pierre Karpman

Security Amplification against Meet-in-the-Middle Attacks Using Whitening*

Pierre-Alain Fouque[1,2] and Pierre Karpman[3]

[1] Université de Rennes 1, France
[2] Institut universitaire de France, France
pierre-alain.fouque@irisa.fr
[3] École normale supérieure de Cachan, antenne de Bretagne, France
pierre.karpman@gmail.com

Abstract. In this paper we introduce a model for studying meet-in-the-middle attacks on block ciphers, and a simple block cipher construction provably resistant to such attacks in this model. A side-result of this is a proper formalization for an unproven alternative to DESX proposed by Kilian and Rogaway; this construction can now be shown to be sound in our model. Meet-in-the-middle attacks exploit weaknesses in key schedule algorithms, and building constructions resistant to such attacks is an important issue for improving the security of block ciphers. Our construction is generic so that it can be used on top of any block cipher, and it does not require to increase the key-length. We use an *exposure resilient function* (or ERF) as a building block and we propose a concrete and efficient instantiation strategy based on compression functions.

Keywords: Block cipher, meet-in-the-middle attack, provable security, exposure resilient function.

1 Introduction

In the area of block cipher design, much work up to now has been devoted to proving resistance to classical statistical attacks like standard linear and differential cryptanalysis (see *e.g.* [32,37,13]). However, resistance to attacks that exploit weaknesses of the key schedule has remained mainly unaddressed. These attacks consist principally in meet-in-the-middle (MiTM) [15] and related-key attacks [4].

A typical good design criterion for key schedules is to have a high minimal distance between expanded keys; performance is also often another issue, and key schedules are expected to be fast, so as not to impact too much the encryption of small messages. An additional criterion could be for the key schedule to be non-linear, although many (good) key schedules are in fact linear. These design principles, however, do not really amount to a theory comparable to the one devoted to resistance to statistical attacks. Nonetheless, a few works study the security of key schedules with respect to related-key attacks from a more theoretical perspective [30,29,12].

Meet-in-the-middle attacks are an important technique available to the cryptanalysts studying symmetric primitives. It is important to avoid such attacks since unlike statistical attacks they usually have low data requirements. Beyond the classical result on double-encryption (see *e.g.* [15]), MiTM attacks are effective at exploiting weaknesses in the key schedule algorithms of block ciphers or the message expansion of hash functions. In the context of block ciphers, MiTM attacks are the most efficient attacks on many ciphers: round-reduced version of IDEA [5], round-reduced version of AES [14,17,9], or the full GOST [20]. Furthermore, a MiTM phase is usually used to extend the number of rounds reached by a statistical attack, as seen for instance in the attacks of Biham and Shamir [6] and of Matsui [28] on the DES. Finally, the recent biclique attack is a MiTM-related technique useful to speed up exhaustive search. Bicliques were found to be successful against AES and IDEA [7,22]. In the context of hash functions, MiTM techniques may be used to find preimages; this was for instance the case for attacks on MD4 [25,2], MD5 [36,2], or AES in a hash function setting [35].

* A short version of this paper is to appear in the proceedings of IMA CC 2013. This is the full version, with the complete proof of Thm. 2.

Our Contributions. In this paper, we develop a simple model for meet-in-the-middle attacks and propose a generic block cipher construction that is provably resistant to such attacks in this model. The idea behind our model is simple and based on the fact that many MiTM attacks on block ciphers can be seen as decomposing the cipher into two sub-ciphers, and then applying the classical MiTM attack on double-encryption (in more recent variations, the cryptanalyst may also guess part of the key or part of the intermediate state [20,16]). Hence we argue that studying a construction in the sole context of double-encryption is actually meaningful for studying many types of MiTM attacks on a single cipher vulnerable to such attacks. However our goal is not to study constructions *actually based on double-encryption* (such as for instance the double XOR-cascade [19], *cf.* below). This is because such constructions already lend themselves to meet-in-the-middle attacks even when the underlying cipher(s) does not; our objective is different and consists in obtaining *a construction resistant to MiTM when the underlying cipher is not.* Studying the construction with a composition of two ciphers as an underlying primitive is therefore only *a mean of simulating the construction applied to a single cipher that is vulnerable to meet-in-the-middle attacks.*

Our construction relies on a core (or internal) cipher and on a form of whitening. Let $\mathscr{E}_k$ be the core cipher of key k, and f a function with good enough properties, then define a new cipher

$$\mathrm{EF}_k(p) \triangleq \mathscr{E}_k(p \oplus f(k)) \oplus f(k).$$

The main idea behind this construction is to force an attacker to commit to a value for the whole key before being able to exploit any data he may have access to, thereby making it impossible to work separately on parts of the key[1]. A similar idea can be found for instance in the operation mode of the SHA-3 candidate SIMD [26]: the goal in this context was to make message modification techniques *à la* Wang impossible by forcing the attacker to commit to a specific value of the message, before the message expansion phase. We prove that meet-in-the-middle attacks are not effective against the EF cipher; this is achieved by upper-bounding the maximum advantage of an attacker of the above construction in a double-encryption setting (when $\mathscr{E}$ is a cascade of two ciphers), and showing that it is less than the advantage of a meet-in-the-middle adversary. We do this with a method adapted from Kilian and Rogaway's proof on DESX, and justify formally how this is relevant when the construction is applied to a single cipher on which MiTM attacks may be performed. We also discuss how the construction can be instantiated efficiently in practice.

Related Work. We can distinguish two kinds of works on provable security for block ciphers: proving a property for some high-level and generic construction, or proving the resistance of an actual cipher to more specific attacks. Our work clearly belongs to the first category, whereas from the second we can cite *e.g.* the provable resistance of block ciphers to classical linear and differential cryptanalysis. Our approach is generic so that it can be used on top of any cipher; it is for instance similar to some proposals for building tweakable block ciphers [27].

Similar-looking constructions have already been proposed in the literature, but with a distinct motivation of extending the *equivalent key length* of the core primitive. One such construction is the DESX (or its more generic name 'FX') construction, proposed by Rivest and formally proved by Kilian and Rogaway [24]. It can be described as taking a cipher $\mathscr{E}_k$ under the key k, and defining a new cipher

$$\mathrm{EX}_{k,k_1,k_2}(p) \triangleq \mathscr{E}_k(p \oplus k_1) \oplus k_2.$$

A more recent development is the aforementioned double XOR-cascade of Gaži and Tessaro [19]. This construction is based on a cipher $\mathscr{E}$ and defines a new cipher 2XOR as:

$$2\mathrm{XOR}_{k,k_1}(p) \triangleq \mathscr{E}_{\tilde{k}}(\mathscr{E}_k(p \oplus k_1) \oplus k_1),$$

[1] Or alternatively to force the attacker to guess the value of the whitening independently of the key.

where $\tilde{k}$ is a key related to k (the authors suggest flipping one bit of k). However, this requires two calls to the cipher $\mathcal{E}$, and therefore cannot readily be used in our context. The main difference between the above constructions and ours is that we do not aim for a bigger equivalent key length, and derive all the whitening keys from the original key to $\mathcal{E}$. We also study our construction specifically in the context of resistance to meet-in-the-middle attacks.

Interestingly, Kilian and Rogaway briefly mention a construction that can be seen as an instantiation of ours. Their purpose was to define an alternative to the FX construction that gives more flexibility in the choice of the key length to the user. Instead of using independent keys k, k_1, k_2, they suggest deriving them from an arbitrarily-long key $\hat{k}$, as the (truncated) output of $f(\hat{k})$, $f_1(\hat{k})$, and $f_2(\hat{k})$ respectively, where f, f_1, f_2 are defined as `SHA-1` prefixed with three different constants. Once again the motivation is different from ours, and no proof nor formalization is given for this construction. Note that as a consequence of our results, it is possible to prove that this construction is sound.

Outline of the Paper. We present our model for studying meet-in-the-middle attacks in §2 and our construction in §3. We prove the resistance of the construction to the attacks captured by our model in §4. We discuss our result and its implications on advanced meet-in-the-middle techniques in §5, and instantiation issues in §6.

2 A Model for Meet-in-the-Middle Attacks

2.1 Generic Constructions

The aim of our work is to define constructions resistant to MiTM attacks. We define here what we mean by *construction* and what kinds of constructions we specifically consider. We first recall the definition of a block cipher.

Definition 1 *A block cipher is a mapping* $\mathcal{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ *such that* $\forall k \in \{0,1\}^\kappa$, $\mathcal{E}(k,\cdot)$ *(also noted* $\mathcal{E}_k(\cdot)$*) is a permutation. The first and second arguments of* $\mathcal{E}$ *are called the* key *and* message (block) *respectively. The variables* κ *and* n *denote the* key size (or length) *and* block size (idem) *of* $\mathcal{E}$.

Definition 2 *A single-cipher construction is a block cipher* $\mathrm{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ *that can be described as the composition* $\mathrm{Post} \circ \mathcal{E} \circ \mathrm{Pre}$ *of functions* Pre, $\mathcal{E}$, *and* Post, *where* $\mathcal{E} : \{0,1\}^{\kappa'} \times \{0,1\}^n \to \{0,1\}^n$ *is a block cipher. The functions* Pre *and* Post *may take both of* $\mathcal{E}$*'s inputs as arguments, that is the key and the message. A* v*-cipher construction is a block cipher that can be described as the composition* $\mathrm{Post}^{(v)} \circ \mathcal{E}^{(v)} \circ \mathrm{Pre}^{(v)} \circ \mathrm{Post}^{(v-1)} \circ \cdots \circ \mathcal{E}^{(1)} \circ \mathrm{Pre}^{(1)}$ *where the ciphers* $\mathcal{E}^{(i)}$ *use independent keys, and where the* $\mathrm{Pre}^{(i)}$ *and* $\mathrm{Post}^{(i)}$ *functions may take any of these keys as inputs. A* v*-cipher construction of the specific form* $\mathrm{Post} \circ \mathcal{E}^{(v)} \circ \mathcal{E}^{(v-1)} \circ \cdots \circ \mathcal{E}^{(1)} \circ \mathrm{Pre}$ *is called a* v^**-cipher construction. Any single-cipher construction can be extended to a* v^**-cipher construction in a straightforward way.*

In this paper, we consider the EF construction, defined in §3, which is a single-cipher construction. It is thus generic, and can be used both with already-existing algorithms, and as a basis to design a cipher *ex nihilo*.

2.2 The Model

Our goal in this section is to give a formal model for MiTM attacks that allows to prove security properties. We later use this model to prove the resistance of the EF construction to such attacks. This model does not capture the concept of *any* MiTM attack, but it does nonetheless take into account a significant class.

The idea behind our model is that a MiTM attack on, say, cipher $\mathscr{E}$ often performs a conceptual decomposition of $\mathscr{E}$ into two sub-ciphers, with separate key bits. We can thus model such attacks as being performed on a double-encryption construction with two independent ciphers, seen as black-boxes. This allows us to consider MiTM attacks against generic ciphers. We detail this argument in the remainder of this section.

The Classical Meet-in-the-Middle Attack on Double Encryption. Let us consider the cipher $\mathscr{E}$, defined as the composition of the two independent ciphers $\mathscr{F}$ and $\mathscr{G}$, operating on independent keys k_1 and k_2 respectively. We denote by $\mathscr{E}_k(p)$ the action of encrypting the plaintext p with $\mathscr{E}$ and key k and producing the ciphertext c. By definition of $\mathscr{E}$, we have $\mathscr{E}_k(p) \triangleq \mathscr{G}_{k_2}(\mathscr{F}_{k_1}(p))$, with k_1 and k_2 uniquely defined by k.

The MiTM attack on double-encryption exploits the fact that k_1 and k_2 are used independently in their respective ciphers; in its simplest form, it can be described as follows:

Get a known plaintext p and its corresponding ciphertext c.
for every possible candidate k_1^i for key k_1 **do**
  Compute $y^i \triangleq \mathscr{F}_{k_1^i}(p)$ and store the result in memory.
end for
for every possible candidate k_2^j for key k_2 **do**
  Compute $y'^j \triangleq \mathscr{G}^{-1}_{k_2^j}(c)$ and store the result in memory.
end for
for every $y^i = y'^j$ **do**
  Output (k_1^i, k_2^j) as a candidate for (k_1, k_2).
end for

This procedure may be repeated until only one candidate for k_1 and k_2 remains, using many plaintext/ciphertext pairs. If we call κ the size of the keys k_1 and k_2 in bits, the cost of the attack is then of the order 2^κ in time and memory, which is much lower than the $2^{2\kappa}$ time that brute-force search on k would cost. If k_1 and k_2 are of different size κ_1 and κ_2, one needs only to store the candidates for the smaller size, *e.g.* in a hash table, and the candidates for the bigger key size can be computed on the fly. The general cost of this attack is thus of the order of $\max(2^{\kappa_1}, 2^{\kappa_2})$ in time, and $\min(2^{\kappa_1}, 2^{\kappa_2})$ in memory.

This attack can still be applied when k_1 and k_2 are not independent but have only some of their bits in common. In that case, one just needs to guess the common bits and repeat the above procedure for every guess.

A Model for Meet-in-the-Middle Attacks on a Single Cipher. MiTM attacks are in no way limited to double-encryption; in fact they are well-suited to iterated ciphers with weak key schedules. However, the ideas involved in a MiTM attack on a single cipher are essentially the same as for attacking double-encryption.

Let us consider an iterated cipher $\mathscr{E}$ with round function ρ^i for round i: we define $\mathscr{E}_k(p)$ as the composition $\rho^r_{k_r} \circ \cdots \circ \rho^1_{k_1}(p)$, where r is the number of iterations of the round function, and the k_i's are round keys generated from k by a key schedule. The idea behind a MiTM attack on $\mathscr{E}$ is to find two sets k_α and k_β of consecutive round keys such that they involve strictly different bits of k. In other words we have $k_\alpha \triangleq \{k_i, \cdots, k_{i+j}\}$, $k_\beta \triangleq \{k_{i+j+1}, \cdots, k_{i+j+1+k}\}$, with $k_\alpha \cap k_\beta = \emptyset$ (when the intersection is taken over the bits of k found in k_α and k_β). Once these sets are found, it is possible to guess independently the bits of k present in k_α and the ones present in k_β, in a way exactly similar as for double-encryption. That is, finding the sets is equivalent to conceptually decompose (a part of) $\mathscr{E}$ in two sub-ciphers with independent keys, on which double-encryption is performed: we have $\mathscr{E}_k = (\rho^{i+j+1+k} \circ \cdots \circ \rho^{i+j+1})_{k_\beta} \circ (\rho^{i+j} \circ \cdots \circ \rho^i)_{k_\alpha}$ (although this equality is true only if $i = 1$ and $i + j + 1 + k = r$. This constraint can easily be waved, however, if we restrict ourselves to finding sub-ciphers for a round-reduced version of $\mathscr{E}$).

We are now ready to formalize our model. We start by stating the security of double-encryption with ideal ciphers, thanks to a theorem of Aiello *et al.* [1][2].

Theorem 1 ([1]) *Let $\mathscr{F} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be an ideal block cipher. For any $\kappa, n, t, q \geq 1$, where t and q denote the number of oracle access to $\mathscr{F}$ and $\mathscr{F}^{-1}$, and $\mathscr{F} \circ \mathscr{F}$ respectively, then the maximum advantage of any adversary $\mathrm{A}_{t,q}$ trying to distinguish $\mathscr{F} \circ \mathscr{F}$ from a random permutation with resources t and q is upper-bounded by $t^2/2^{2\kappa}$. This bound is tight up to a constant factor as long as q is not too small.*

We allow ourselves to use a more general expression of this result when considering double-encryption of not necessarily equal ciphers $\mathscr{F}$ and $\mathscr{G}$, of not necessarily equal key lengths κ_1 and κ_2, where t_1 and t_2 denote the number of oracle access to $\mathscr{F}$ and $\mathscr{G}^{-1}$ respectively. In this case, we use the upper-bound $t_1 \cdot t_2 / 2^{\kappa_1+\kappa_2}$[3]. We now define the notion of constructions resistant to MiTM attacks.

Definition 3 *Let $\mathscr{F} : \{0,1\}^{\kappa_1} \times \{0,1\}^n \to \{0,1\}^n$ and $\mathscr{G} : \{0,1\}^{\kappa_2} \times \{0,1\}^n \to \{0,1\}^n$ be two block ciphers. A two-cipher construction $\mathrm{E}(\mathscr{F},\mathscr{G})$ is said to be* resistant to the meet-in-the-middle attack *if the maximum advantage of any adversary A trying to distinguish $\mathrm{E}(\mathscr{F},\mathscr{G})$ from a random permutation is:*

$$\max_{\mathrm{A}} Adv_{\mathrm{E}(\mathscr{F},\mathscr{G})}(\mathrm{A}_{t_1+t_2,q}) \quad < \quad t_1 \cdot t_2 / 2^{\kappa_1+\kappa_2},$$

up to constant factors.

This definition is made meaningful by the tightness of the bound of theorem 1. Essentially, it says that a two-cipher construction is resistant to meet-in-the-middle attacks if no adversary can distinguish it with an advantage that is *at least as good as the best one it could get if only composition of the two ciphers were used instead.*

Definition 4 *Let $\mathscr{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. A single-cipher construction $\mathrm{E}(\mathscr{E})$ is said to be* resistant to the meet-in-the-middle attack *if for any two block ciphers $\mathscr{F} : \{0,1\}^{\kappa_1} \times \{0,1\}^n \to \{0,1\}^n$ and $\mathscr{G} : \{0,1\}^{\kappa_2} \times \{0,1\}^n \to \{0,1\}^n$ such that $\mathscr{E} = \mathscr{G} \circ \mathscr{F}$ and $\kappa = \kappa_1 + \kappa_2$, the maximum advantage of any adversary A trying to distinguish $\mathrm{E}(\mathscr{E})$ from a random permutation is:*

$$\max_{\mathrm{A}} Adv_{\mathrm{E}(\mathscr{E})}(\mathrm{A}_{t_1+t_2,q}) \quad < \quad t_1 \cdot t_2 / 2^{\kappa_1+\kappa_2},$$

up to constant factors.

In other words, this means that the best attack on the construction $\mathrm{E}(\mathscr{E})$ *is strictly worse than the best meet-in-the-middle attack on $\mathscr{E}$*. Our model is justified by the next proposition:

Proposition 1 *Let E be a two*-ciphers construction resistant to the meet-in-the-middle attack. Then the restriction of E to a single cipher is a single-cipher construction resistant to the meet-in-the-middle attack.*

Proof. This is a direct consequence of definitions 3 and 4. □

Hence, the resistance of a single-cipher construction to MiTM attacks can be studied by analyzing its two*-ciphers variant. In practice, we perform this analysis in the ideal block cipher model [3,24].

[2] The result is stated in the specific case where the two ciphers are equal.

[3] Although in practice, we actually study our construction in the case of $\kappa_1 = \kappa_2$, and therefore we will really be using exactly the same bound as in the main result of [1].

3 A Construction Resistant to Meet-in-the-Middle Attacks

We now formally define our construction. We start by introducing the notion of *Exposure-Resilient Functions* (or ERF), as defined by Canetti *et al.* [11]. ERFs are similar to *all or nothing transforms*, which were introduced by Rivest [34].

Definition 5 *An ℓ-ERF is a mapping $f : \{0,1\}^\alpha \to \{0,1\}^\beta$ such that for random values r, R in $\{0,1\}^\alpha$, $\{0,1\}^\beta$; for any $L \in \binom{\alpha}{\ell}$, the distributions $\langle [r]_{\bar{L}}, f(r)\rangle$ and $\langle [r]_{\bar{L}}, R\rangle$ are indistinguishable one from another; where $\binom{\alpha}{\ell}$ denotes the set of subsets of $\{1 \dots \alpha\}$ of size ℓ, and for $x \in \{0,1\}^\alpha$, $[x]_{\bar{L}}$ denotes x restricted to its bits not in L.*

Here, we will consider particularly weak ERFs, in the sense that when less than ℓ bits of r are unknown to the adversary, he is then supposed to be able to predict the value of $f(r')$ for any r' that fixes the unknown bits of r to some value. However, we will mostly consider degenerate cases where ℓ is zero. That is, the output of the ERF is indistinguishable from random until all of its input is revealed. Constructions of ERFs are known to exist in the standard model [11], and it is trivial to see that a random oracle meets the definition of a zero-ERF. Hence, from a practical point of view, a zero-ERF can be instantiated by a hash function, in the random oracle model.

We now define our construction, which we will note EF for short.

Definition 6 *Let $\mathcal{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. We write $\mathcal{E}_k(p) \triangleq c$ the action of encrypting the plaintext p with $\mathcal{E}$ under the key k, to produce the ciphertext c. Let $f : \{0,1\}^\kappa \to \{0,1\}^n$ be an ℓ-ERF. Then we define the* EF *construction with core cipher $\mathcal{E}$ as $\mathrm{EF}_k(p) \triangleq \mathcal{E}_k(p \oplus f(k)) \oplus f(k)$, where '$\oplus$' denotes bitwise exclusive or.*

Our goal is to prove the resistance of this construction to MiTM attacks. According to our model, we will study this construction as a two-ciphers construction. That is, we instantiate $\mathcal{E}_k$ by $\mathcal{G}_{k_2} \circ \mathcal{F}_{k_1}$. In this case, the EF construction applied to $\mathcal{E}$ can be written as:

$$\mathrm{EF}_{(k_1,k_2)}(p) \triangleq \mathcal{G}_{k_2}(\mathcal{F}_{k_1}(p \oplus f(k_1,k_2))) \oplus f(k_1,k_2).$$

4 Resistance of the EF Construction to Meet-in-the-Middle Attacks

In this section, we prove an upper-bound on the advantage of an adversary trying to distinguish the EF two-ciphers construction from a random permutation, in function of the number of queries made to different oracles. The bound we obtain shows that our construction significantly improves the resistance of double-encryption to generic attacks such as the classical MiTM, and hence is resistant to the MiTM attack, in the terminology of definitions 3 and 4.

4.1 Security Model

We consider the EF two-ciphers construction applied to $\mathcal{G} \circ \mathcal{F}$, where $\mathcal{F} : \{0,1\}^{\kappa_1} \times \{0,1\}^n \to \{0,1\}^n$ and $\mathcal{G} : \{0,1\}^{\kappa_2} \times \{0,1\}^n \to \{0,1\}^n$ are ideal block ciphers [3,24]: for each key k_1 (resp. k_2), the map $\mathcal{F}_{k_1}$ (resp. $\mathcal{G}_{k_2}$) is a permutation randomly chosen from the set Π_n of all $(2^n)!$ permutations operating on words of size n. For ease of presentation, and without loss of generality, we assume that $\kappa_1 = \kappa_2 \triangleq \kappa$. The '$f$' function used in EF is an ℓ-ERF with ℓ small. We consider an adversary who is given access to four oracles:

– Two of them are $\mathcal{F}$ and $\mathcal{G}$; when provided with a key k_1' (resp. k_2') of size κ and an input x (resp. y) of size n, they return the result y (resp. z) of encrypting x (resp. y) with $\mathcal{F}$ (resp. $\mathcal{G}$) with key k_1' (resp. k_2'). Queries to the inverse oracles $\mathcal{F}^{-1}$ and $\mathcal{G}^{-1}$ are permitted, and are not distinguished from regular queries. That is, if $\mathcal{F}(x) \triangleq y$ has been queried, we consider that $\mathcal{F}^{-1}(y) \triangleq x$ has also been queried, and conversely.

– One oracle gives an access to f, and when provided with an input of size 2κ bits, returns the result of size n of the evaluation of f on this input.
– The last oracle, which we call $\mathscr{U}$, takes as input a plaintext p and returns either $\mathrm{EF}(p)$, with EF instantiated with $\mathscr{F}$, $\mathscr{G}$, and f with fixed, randomly chosen keys k_1 and k_2; or the image of p from a fixed permutation π randomly selected from the set Π_n. Again, queries to $\mathscr{U}^{-1}$ are permitted.

Each access to an oracle will be counted and expressed by the following variables:

– The number of accesses to $\mathscr{U}$ is denoted by D. They represent the amount of data available to the adversary.
– The number of accesses to $\mathscr{F}$ and $\mathscr{F}^{-1}$ (resp. $\mathscr{G}$ and $\mathscr{G}^{-1}$) is denoted by q_1 (resp. q_2).
– The number of accesses to f is denoted by q_f.

The goal of the adversary is to distinguish between $\mathscr{U}$ being π or the EF construction. We define the *advantage* of an adversary as its probability of successfully distinguishing the two instantiations of $\mathscr{U}$. More formally:

Definition 7 *Let Π_n be the set of permutations on words of size n; let us note $x \xleftarrow{R} \mathscr{S}$ the action of randomly choosing an object x from the set $\mathscr{S}$; let us denote by A^{EF} and A^{π} the answer, 0 or 1, of an adversary* A *with access to the aforementioned oracles, when $\mathscr{U}$ is the* EF *construction and a randomly chosen permutation respectively. Then the advantage Adv_{A} of the adversary* A *is defined as:*

$$Adv_{\mathrm{A}} \triangleq$$
$$\Pr\left[\forall k'_1 \in \{0,1\}^{\kappa}, \mathscr{F}_{k'_1} \xleftarrow{R} \Pi_n; \forall k'_2 \in \{0,1\}^{\kappa}, \mathscr{G}_{k'_2} \xleftarrow{R} \Pi_n; \pi \xleftarrow{R} \Pi_n : \mathrm{A}^{\pi} = 1\right] -$$
$$\Pr\left[\forall k'_1 \in \{0,1\}^{\kappa}, \mathscr{F}_{k'_1} \xleftarrow{R} \Pi_n; \forall k'_2 \in \{0,1\}^{\kappa}, \mathscr{G}_{k'_2} \xleftarrow{R} \Pi_n; k_1 \xleftarrow{R} \{0,1\}^{\kappa}; k_2 \xleftarrow{R} \{0,1\}^{\kappa} : \mathrm{A}^{\mathrm{EF}} = 1\right].$$

Our objective is to bound this advantage for any adversary A in function of the number of queries he has made to the oracles. We note $Adv(\ell, D, q_1, q_2, q_f)$ the advantage of any adversary having made less than D, q_1, q_2, and q_f queries to the oracles, when f is an ℓ-ERF.

Notations.

– A *key candidate* for $\mathscr{F}$ (resp. $\mathscr{G}$) is denoted by k'_1 (resp. k'_2). Those are keys for which the adversary makes queries to the oracles awaiting a key as part of their inputs.
  Moreover, we may count the accesses to those oracles when queried with a specific key. The number of accesses to $\mathscr{F}$ (resp. $\mathscr{G}$, f) with key k'_1 (resp. k'_2, (k'_1, k'_2)) is written $q_1(k'_1)$ (resp. $q_2(k'_2)$, $q_f(k'_1, k'_2)$).
  Furthermore, the number of queries to the $\mathscr{F}$ (resp. $\mathscr{G}$) oracle with key k'_1 (resp. k'_2) and for a specific message p is noted $q_1(k'_1, p)$ (resp. $q_2(k'_2, p)$).
– If $\mathscr{U}$ has been instantiated with EF, the key used in the instantiation for $\mathscr{F}$ (resp. $\mathscr{G}$) is denoted by k_1 (resp. k_2). For ease of presentation, we will consider it valid to talk about those keys even when it is not clear if $\mathscr{U}$ is instantiated with EF.
– We denote by x, y, z, the intermediate values $p \oplus f(k_1, k_2)$, $\mathscr{F}_{k_1}(p \oplus f(k_1, k_2))$, and $\mathscr{G}_{k_2}(\mathscr{F}_{k_1}(p \oplus f(k_1, k_2)))$ respectively.
– We make use of an *indicator function*, written $\mathbb{1}$. We have $\mathbb{1}(x) = 0$ if and only if x is zero, and it is one otherwise. This may be extended to sets, where $\mathbb{1}(x) = 0$ if and only if x is the empty set, and is one otherwise.
– The concatenation of words x and y is noted $x||y$, the Hamming weight of a word x is denoted by $\mathrm{hw}(x)$.

4.2 The Result

Our main result about the security of the EF construction is summarized by the following theorem.

Theorem 2 *The advantage $Adv(\ell, D, q_1, q_2, q_f)$ of an adversary trying to distinguish* EF *from a random permutation is upper-bounded by:*

$$2^{-2\kappa} \max\left(2^{\ell}\binom{n}{\ell} \cdot q_f, \quad 2^{-n} \cdot D \sum_{k_1', k_2'} \min(q_1(k_1'), q_2(k_2'))\right). \tag{1}$$

One can see that to gain an advantage of one, an adversary with D available data needs at least $\frac{2^{2\kappa}}{2^{\ell}\binom{n}{\ell}}$ or $\frac{2^{\kappa+n}}{D}$ queries to the oracles, whichever is the smallest. For $\ell = 0$, $n = \kappa$, and $D = 1$, these two terms are equal to $2^{2\kappa}$; in the case of double-encryption without the EF construction, one would only need of the order of 2^{κ} queries. This result immediately implies the following corollary:

Corollary 1 *The two* -ciphers and single-cipher* EF *construction is resistant to meet-in-the-middle attacks, in the terminology of definitions 3 and 4,* as long as $D < 2^n$ (*i.e.* the adversary does not have access to the whole codebook).

Proof. For small values of ℓ, and when $D < 2^n$, $Adv(\ell, D, q_1, q_2, q_f)$ is strictly smaller than the bound of definitions 3 and 4. □

The restriction that $D < 2^n$ is important, and comes from more general properties of FX-like constructions. In particular, such a construction cannot be used in all generality in order to increase the equivalent key length of a block cipher, as the key length cannot be shown to be more than the one of the core cipher when all the codebook is available to the adversary. This is the case for the original construction of DESX, and remains true in our modified setting when the core cipher is a 2-cascade (*i.e.* a composition of two ciphers). In the latter case, the double XOR-cascade of Gaži and Tessaro *does* increase the effective key length of the (modified) 2-cascade. However, as it has already been noted, we want our construction to be applicable to a single cipher as well, and therefore cannot use one similar to theirs. The above restriction notwithstanding, we believe that our construction is still interesting in practice. The first reason is that attacks where the adversary uses the whole codebook would not only be of limited interest to the attacker, they can also be made asymptotically as expensive as the designer of the cipher wishes to; so big an amount of data is also fairly unrealistic for many ciphers with big block sizes (*e.g.* 128 bits). The second reason is that raising the data complexity of a MiTM attack from the information-theoretically lower-bound to the whole codebook, for an adversary to get the same time complexity, is in itself a huge improvement of the resistance to MiTM attacks.

4.3 Proof Sketch

We outline here the strategy used for proving theorem 2, while the full proof is given in the appendix.

Given the similarities between the EF and FX constructions and their security models, our proof has a structure close to the one of Kilian and Rogaway [24]. In particular, we use games for each oracle to define the situations where an adversary may distinguish the instantiation of the $\mathscr{U}$ oracle (we do not fully redefine the games in this paper, though, and refer to [24] for a more detailed description). For each such situation, we then bound the probability of the distinction being possible in function of the past queries made by the adversary. We bound separately the advantage of an adversary trying to find distinguishing situations for the three oracles, $\mathscr{F}$, $\mathscr{G}$, and $\mathscr{U}$, and then combine these bounds together with the bound of the advantage over f to produce a general result.

It should be noticed that we do not need to consider situations for the "inverse" oracles (such as *e.g.* $\mathscr{F}^{-1}$), as the constraints possibly put on the input/output pairs of queries to those oracles are simply swapped when compared to the ones for the "forward" oracles (an input of an inverse oracle being an output for the corresponding forward oracle). Therefore, the advantage when distinguishing inverse oracles is not different from when considering forward oracle.

The main difference between our proof and the one of [24] is that, from the structure of the construction, an adversary has essentially the choice of guessing the output of the f function directly (thereby seeing $f(k_1,k_2)$ as a third independent key), or via its inputs and the properties of f. This shows in the bound of theorem 2 as the two arguments of the maximum function. Because our construction is more complex than FX, we also have more oracles to consider.

5 Discussion

In the previous section, we have shown that the EF construction increases the resistance of a cipher to "classical" MiTM attacks. We now argue that this improved resistance carries on to more advanced MiTM techniques, which further increases the relevance and interest of the construction. We also address the relevance of our ideal-cipher-based model.

5.1 About Ideal Ciphers

As pointed above, our proof uses the ideal cipher model. This could be seen as limiting the relevance of the applications we claim —building constructions resistant to MiTM attacks— as we use a setting where the best attack on the sub-ciphers is basically brute-force.

We claim that this is not a limitation. Using ideal ciphers allows one to express bounds in terms of number of queries to the relevant oracles: the cost of each query is of little importance in so far as it is bounded by a constant. In other words, our results show that the EF construction increases the security of double-encryption by ensuring that an adversary needs *to perform more queries to the oracles* to gain an advantage comparable to the one he would get when the construction is *not* used. Whether the queries are expensive or not (*i.e.* whether the best attacks on the sub-ciphers is brute-force or not) does not change the asymptotic increase of security that one can expect by choosing a bigger key or block size. This is as much valid for the sub-ciphers decomposition of a MiTM attack on single cipher as for double-encryption.

Another concern might be that actual MiTM attacks do not typically perform a full decomposition of the attacked cipher in two sub-ciphers with independent keys. This is indeed an ideal case for the attacker which is seldom met in practice. However, real attacks that do not conform to this ideal case are typically less powerful, while still needing a decomposition in two sub-ciphers at some point; hence our construction increases the security against these real attacks as well.

5.2 The Splice-and-Cut Exception

Before going on to the expected advantages of the EF construction, we should mention one situation where it does not seem to be useful, *i.e.* when protecting against splice-and-cut MiTM attacks.

A splice-and-cut MiTM attack uses a conceptual decomposition of a cipher with sub-ciphers that may be defined by considering the first and last round of the attacked cipher as consecutive. That is, we consider decompositions of say, $\mathscr{E}$, that can be written as, say $\mathscr{G}_2 \circ \mathscr{F} \circ \mathscr{G}_1$. In order to perform this variant of the MiTM attack, the attacker typically guesses one intermediate value at the boundary between the two sub-ciphers using different key subsets, and then queries the plaintext or ciphertext corresponding to the encryption or decryption of this value for a given sub-key candidate. With data obtained this way, it is then easy to perform a MiTM attack.

It is possible to adapt our model to the conceptual decomposition used in a splice-and-cut attack. However, because this attack typically requires the whole code-book to be performed, the bound that would be obtained by a generalization of theorem 2 would not show any improved resistance. It might be that improved resistance to the splice-and-cut MiTM could be shown by leaving the information-theoretic view of the ideal cipher model, but this does not seem to be an easy task.

In the end, because of their huge data requirements, splice-and-cut MiTM are more suited to attacking hash functions, and are seldom used against block ciphers. Consequently, we think that the impact of the EF construction not being efficient against them is somewhat limited in its targeted applications.

5.3 Taking Advanced Attack Techniques into Account

We now discuss the issue of including some more advanced MiTM techniques used by cryptanalysts in our proof that the EF construction increases the resistance of a single cipher to MiTM attacks. We discuss this for two important techniques: the *initial structure* of Sasaki and Aoki [36], and its later generalization to *bicliques* by Khovratovich *et al.* [23].

Initial Structure. This is an advanced techniques that may be used in order to increase the number of rounds reached by an existing MiTM attack. It consists in finding an *initial structure* between two sub-ciphers of the MiTM attack that use, say, key subsets k_1 and k_2 respectively. The structure consists itself in a sub-cipher that can be computed thanks to the key subsets $k'_1 \subseteq k_1$ and $k'_2 \subseteq k_2$, but with the key bits from k'_2 being used *before* the bits of k'_1 in the structure (otherwise the relevant parts of the initial structure can be included in the two sub-ciphers of the MiTM). Finding initial structures gives more flexibility to the attacker in the matching phase, leading to more powerful attacks. For instance, they were key in finding the first preimage attack on the full `MD5` [36].

In order to be applicable, the initial structure technique still needs the cipher under attack to be decomposed in two sub-ciphers. The added sophistication of the matching phase typically allows the attacker to define a decomposition that covers more rounds of the full cipher than what he would obtain without using the initial structure. Yet this does not change the fact that the attacker needs to test the key candidates for both sub-ciphers. What the bounds on the EF construction say is that it is impossible to use a MiTM technique to do this efficiently, because of the increased number of queries that have to be made to the sub-ciphers (and possibly to the f function) in order to test all possible keys and get an advantage of one; this is completely independent of the number of rounds covered by the sub-ciphers.

In essence, the initial structure technique allows an attacker to find better decompositions, but it does not improve the key-testing phase *per se*. Therefore we claim that the EF construction still improves the security of a cipher against adversaries using initial structures: however good the decomposition of the initial cipher they may get, the construction layer will prevent efficient use of it.

Bicliques. This is a generalization of the previous technique that allows for a more systematic way to construct initial structures, instead of searching them manually as was done originally. In its simplest form, the biclique technique can be seen as a way to extend an existing (splice-and-cut) MiTM attack by constructing *bicliques* between intermediate states in order to cover the rounds not included in the existing attack. Again, this leads to more powerful attacks, and this technique has successfully been used to analyze several hash functions and block ciphers such as `Skein` [23], `AES` [7], or `IDEA` [22].

Again, for this technique to be applicable, finding a decomposition in sub-ciphers is necessary. Hence, even if the biclique parts themselves are not something that is captured in our

analysis, it is still the case for the decomposition: for a given decomposition of given parameters size, the bound on the number of queries to the sub-ciphers necessary to gain a given advantage is not changed by the presence of bicliques. The interest of the EF construction in this case is again to ensure that no decomposition can be efficiently used, either alone or as part of a wider attack.

This analysis and its arguments is similar to the one that was performed on the recent PRINCE cipher in order to assess its resistance to bicliques [8].

Summary. We did not formalize the arguments presented in this section, and it does not seem to be easy to do so. In addition, although we think that an increased resistance to the MiTM phase is an important step towards some sort of provable security against these techniques, the interest of that in the case of actual designs might be somewhat more limited. The reason is that if these techniques allow to efficiently use a decomposition with small parameters size as part of a bigger attack, the increased resistance to the MiTM phase might not impact the overall complexity significantly. This might be a concern when resistance to *e.g.* bicliques is considered.

In the end, much work still needs to be done in order to better understand how to resist advanced MiTM techniques, and this is far beyond the scope of this paper. Yet we believe that an increased resistance to even just the basic MiTM attack should be an important part of this work, much as resisting standard statistical attacks is an important part of modern cipher design.

5.4 Alternatives for the Construction

We end this section by outlining three alternatives for the EF construction that differ with the main proposal in the way the output of the f function is combined with the input of the core cipher.

A first obvious variation is to use modular addition instead of XOR.

Definition 8 *Let $f : \{0,1\}^\kappa \to \{0,1\}^n$ be an ℓ-ERF. Then we define the* $\mathrm{EF}^{\boxplus}$ *construction with core cipher $\mathscr{E}$ as* $\mathrm{EF}^{\boxplus}_k(p) \triangleq \mathscr{E}_k(p \boxplus f(k)) \boxplus f(k)$*, where $\boxplus$ denotes addition in $\mathbb{Z}/2^n\mathbb{Z}$.*

This variant may be useful in practice, when the core cipher of the construction is no longer seen as a black box. Because a block cipher typically performs key whitening with sub-keys derived from k, if it is performed with an XOR operation, it may be better to combine the output of f with modular addition, in order to make it non-linear with respect to this whitening[4]. Conversely, the original construction is maybe to be preferred when the whitening is done with modular addition. Note that adding whitening keys with modular addition instead of XOR was already proposed by Dunkelman *et al.* as a generalization of the Even-Mansour construction [18].

A second variation exploiting a similar idea is to replace XOR with multiplication in a finite field.

Definition 9 *Let $f : \{0,1\}^\kappa \to \{0,1\}^n$ be an ℓ-ERF. Then we define the* $\mathrm{EF}^{\boxtimes}$ *construction with core cipher $\mathscr{E}$ as* $\mathrm{EF}^{\boxtimes}_k(p) \triangleq \mathscr{E}_k(p \boxtimes f(k)) \boxtimes f(k)$*, where $\boxtimes$ denotes multiplication in $\mathbb{F}_{2^n}$.*

Although quite slower than XOR or modular addition, multiplication in a finite field mixes its inputs very thoroughly, which makes it attractive when performance is not critical.

A last variation we suggest is to use even stronger mixing with a decorrelation module [37].

Definition 10 *Let $f, g : \{0,1\}^\kappa \to \{0,1\}^n$ be two ℓ-ERFs. Then we define the* $\mathrm{EFG}^{\mathrm{DC}}$ *construction with core cipher $\mathscr{E}$ as* $\mathrm{EFG}^{\mathrm{DC}}_k(p) \triangleq \mathscr{E}_k(p \boxtimes f(k) \oplus g(k)) \boxtimes f(k) \oplus g(k)$.

Finally, we can also propose an obvious variation orthogonal to the three above, consisting in using two (four in the case of $\mathrm{EFG}^{\mathrm{DC}}$) different functions to derive the two whitening keys.

[4] Alternatively, a concrete instantiation could use $f(k)$ as the unique whitening key.

It is worthwhile noting that all these variants directly benefit from the proofs for the EF construction, as these did not rely on any specific property of XOR not shared by the alternative operations used here (in particular they are all invertible). However, the different constructions are likely to give different levels of security when used in practice, especially when other attacks than MiTM are considered. For instance, decorrelation modules are expected to provide some additional protection against classical differential attacks, which XOR or modular addition do not by themselves.

6 Practical Instantiation

We conclude this work by discussing how to efficiently instantiate the EF construction in practice. We start by showing how it is possible to use a hash function h as the f function. This is justified by the fact that such a function is a zero-ERF in the random-oracle model. We note EH the resulting construction.

Corollary 2 *With notations adapted from §§3 and 4, the advantage $Adv(D, q_1, q_2, q_h)$ of an adversary trying to distinguish $\mathrm{EH}_k(p) \triangleq \mathscr{E}_k(p \oplus h(k)) \oplus h(k)$ from a random permutation, where h is a hash function, is upper-bounded in the random oracle model by:*

$$2^{-\kappa} \max\left(q_h, \quad 2^{-n} \cdot D \sum_{k'_1, k'_2} \min(q_1(k'_1), q_2(k'_2)) \right) \tag{2}$$

for queries q_1 and q_2 to any two ciphers $\mathscr{F}_{k_1}$ and $\mathscr{G}_{k_2}$ such that $\mathscr{E}_k = \mathscr{G}_{k_1} \circ \mathscr{F}_{k_2}$.

This is very similar to an alternative to the FX construction proposed by Kilian and Rogaway, and already mentioned in §1.

From an efficiency point of view, using a call to a (small) hash function as part of the encryption process could be expensive. Therefore, the EH construction might be of little interest when computational power is limited or when the key has to be regularly changed (for instance because the cipher is itself used in a hashing mode). However, we believe that there are meaningful applications for block ciphers where none of these restrictions apply, making this type of instantiation still of interest. It is also worth noting that the input to h/f is of fixed size, and thus only a "one-shot" compression function with a fixed IV $\hat{h} : \{0,1\}^\kappa \to \{0,1\}^n$ is actually needed, and not a full-fledged hash function $h : \{0,1\}^* \to \{0,1\}^n$. Finally, it is likely that a smart instantiation would use synergy between the core of the cipher and the h / $\hat{h}$ function; using two completely unrelated functions for both components would probably not be the simplest way to proceed. In particular, it seems an interesting option to build the hash function by using the block cipher $\mathscr{E}$ itself in a hashing mode[5]. If we can make it so that only one call to the compression function is needed (this is the case *e.g.* when the key and block sizes of $\mathscr{E}$ are equal), then it is even possible to build the compression function from $\mathscr{E}$ used as a *fixed permutation* $\mathscr{E}'$, with all its round keys independently set to a constant. This can be achieved by *e.g.* using $\mathscr{E}$ in Matyas-Meyer-Oseas mode, with $\hat{h}(x)$ then defined as $\mathscr{E}'(x) \oplus x$. Note that such a construction can be performed with many current block ciphers, including `AES-128`. Even though care should be taken before using any cipher in a hashing mode (this is the case for `AES` too [35]), the fact that in this case it may be used with independent round keys may significantly improve its security in that setting (not least because it rules out meet-in-the-middle attacks such as [35]). Therefore, we believe that this instantiation strategy is sound, and that it can be applied to many existing ciphers, as well as being usable for future designs.

[5] Obviously, one may also consider reduced-round variants of the same cipher in order to make this step faster. It is a designer's role to find a good tradeoff between efficiency and security, and this instantiation strategy makes no exceptions.

References

1. Aiello, W., Bellare, M., Crescenzo, G.D., Venkatesan, R.: Security Amplification by Composition: The Case of Doubly-Iterated, Ideal Ciphers. In Krawczyk, H., ed.: CRYPTO. Volume 1462 of Lecture Notes in Computer Science., Springer (1998) 390–407
2. Aoki, K., Sasaki, Y.: Preimage Attacks on One-Block MD4, 63-Step MD5 and More. In Avanzi, R.M., Keliher, L., Sica, F., eds.: Selected Areas in Cryptography. Volume 5381 of Lecture Notes in Computer Science., Springer (2008) 103–119
3. Bellare, M., Krovetz, T., Rogaway, P.: Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In Nyberg, K., ed.: EUROCRYPT. Volume 1403 of Lecture Notes in Computer Science., Springer (1998) 266–280
4. Biham, E.: New Types of Cryptanalytic Attacks Using Related Keys. J. Cryptology **7**(4) (1994) 229–246
5. Biham, E., Dunkelman, O., Keller, N., Shamir, A.: New Data-Efficient Attacks on Reduced-Round IDEA. IACR Cryptology ePrint Archive **2011** (2011) 417
6. Biham, E., Shamir, A.: Differential Cryptanalysis of the Full 16-Round DES. [10] 487–496
7. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. In Lee, D.H., Wang, X., eds.: ASIACRYPT. Volume 7073 of Lecture Notes in Computer Science., Springer (2011) 344–371
8. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In Wang, X., Sako, K., eds.: ASIACRYPT. Volume 7658 of Lecture Notes in Computer Science., Springer (2012) 208–225
9. Bouillaguet, C., Derbez, P., Fouque, P.A.: Automatic Search of Attacks on Round-Reduced AES and Applications. In Rogaway, P., ed.: CRYPTO. Volume 6841 of Lecture Notes in Computer Science., Springer (2011) 169–187
10. Brickell, E.F., ed.: Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings. In Brickell, E.F., ed.: CRYPTO. Volume 740 of Lecture Notes in Computer Science., Springer (1993)
11. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-Resilient Functions and All-or-Nothing Transforms. In Preneel, B., ed.: EUROCRYPT. Volume 1807 of Lecture Notes in Computer Science., Springer (2000) 453–469
12. Choy, J., Zhang, A., Khoo, K., Henricksen, M., Poschmann, A.: AES Variants Secure against Related-Key Differential and Boomerang Attacks. In Ardagna, C.A., Zhou, J., eds.: WISTP. Volume 6633 of Lecture Notes in Computer Science., Springer (2011) 191–207
13. Daemen, J., Rijmen, V.: The Wide Trail Design Strategy. In Honary, B., ed.: IMA Int. Conf. Volume 2260 of Lecture Notes in Computer Science., Springer (2001) 222–238
14. Demirci, H., Selçuk, A.A.: A Meet-in-the-Middle Attack on 8-Round AES. [31] 116–126
15. Diffie, W., Hellman, M.: Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard. Computer **10** (1977) 74–84
16. Dinur, I., Dunkelman, O., Keller, N., Shamir, A.: Efficient Dissection of Composite Problems, with Applications to Cryptanalysis, Knapsacks, and Combinatorial Search Problems. In Safavi-Naini, R., Canetti, R., eds.: CRYPTO. Volume 7417 of Lecture Notes in Computer Science., Springer (2012) 719–740
17. Dunkelman, O., Keller, N., Shamir, A.: Improved Single-Key Attacks on 8-Round AES-192 and AES-256. In Abe, M., ed.: ASIACRYPT. Volume 6477 of Lecture Notes in Computer Science., Springer (2010) 158–176
18. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in Cryptography: The Even-Mansour Scheme Revisited. [33] 336–354
19. Gaži, P., Tessaro, S.: Efficient and Optimally Secure Key-Length Extension for Block Ciphers via Randomized Cascading. [33] 63–80
20. Isobe, T.: A Single-Key Attack on the Full GOST Block Cipher. [21] 290–305
21. Joux, A., ed.: Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers. In Joux, A., ed.: FSE. Volume 6733 of Lecture Notes in Computer Science., Springer (2011)
22. Khovratovich, D., Leurent, G., Rechberger, C.: Narrow-Bicliques: Cryptanalysis of Full IDEA. [33] 392–410
23. Khovratovich, D., Rechberger, C., Savelieva, A.: Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family. In Canteaut, A., ed.: FSE. Volume 7549 of Lecture Notes in Computer Science., Springer (2012) 244–263
24. Kilian, J., Rogaway, P.: How to Protect DES Against Exhaustive Key Search. In Koblitz, N., ed.: CRYPTO. Volume 1109 of Lecture Notes in Computer Science., Springer (1996) 252–267
25. Leurent, G.: MD4 is Not One-Way. [31] 412–428
26. Leurent, G.: Design and Analysis of Hash Functions. PhD thesis, Université Paris 7 (2010)
27. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable Block Ciphers. J. Cryptology **24**(3) (2011) 588–613
28. Matsui, M.: The First Experimental Cryptanalysis of the Data Encryption Standard. In Desmedt, Y., ed.: CRYPTO. Volume 839 of Lecture Notes in Computer Science., Springer (1994) 1–11
29. May, L., Henricksen, M., Millan, W., Carter, G., Dawson, E.: Strengthening the Key Schedule of the AES. In Batten, L.M., Seberry, J., eds.: ACISP. Volume 2384 of Lecture Notes in Computer Science., Springer (2002) 226–240
30. Nikolic, I.: Tweaking AES. In Biryukov, A., Gong, G., Stinson, D.R., eds.: Selected Areas in Cryptography. Volume 6544 of Lecture Notes in Computer Science., Springer (2010) 198–210

31. Nyberg, K., ed.: Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers. In Nyberg, K., ed.: FSE. Volume 5086 of Lecture Notes in Computer Science., Springer (2008)
32. Nyberg, K., Knudsen, L.R.: Provable Security Against Differential Cryptanalysis. [10] 566–574
33. Pointcheval, D., Johansson, T., eds.: Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. In Pointcheval, D., Johansson, T., eds.: EUROCRYPT. Volume 7237 of Lecture Notes in Computer Science., Springer (2012)
34. Rivest, R.L.: All-or-Nothing Encryption and the Package Transform. In Biham, E., ed.: FSE. Volume 1267 of Lecture Notes in Computer Science., Springer (1997) 210–218
35. Sasaki, Y.: Meet-in-the-Middle Preimage Attacks on AES Hashing Modes and an Application to Whirlpool. [21] 378–396
36. Sasaki, Y., Aoki, K.: Finding Preimages in Full MD5 Faster Than Exhaustive Search. In Joux, A., ed.: EUROCRYPT. Volume 5479 of Lecture Notes in Computer Science., Springer (2009) 134–152
37. Vaudenay, S.: Provable Security for Block Ciphers by Decorrelation. In Morvan, M., Meinel, C., Krob, D., eds.: STACS. Volume 1373 of Lecture Notes in Computer Science., Springer (1998) 249–275

A Proof of Theorem 2

We use the notations of §4.1.

A.1 The $\mathscr{F}$ Game

In this sub-section, we give a bound for the advantage of an adversary when it is trying to distinguish $\mathscr{U}$ thanks to the answers of queries to the $\mathscr{F}$ oracle. We will say that such an adversary is playing the $\mathscr{F}$ game. We first identify these situations when the instantiation of $\mathscr{U}$ makes a difference for the answers to queries to $\mathscr{F}$.

The distinguishing situation. On a query to $\mathscr{F}_{k_1'}(x) \triangleq y$, then:

1. if $k_1' = k_1$;
2. and if $\mathscr{U}(x \oplus f(k_1, k_2)) \triangleq c$ has been previously queried (or 'def', for short);
3. and if $\mathscr{G}_{k_2}^{-1}(c \oplus f(k_1, k_2)) \triangleq y'$ has been previously queried;

then we must have $y = y'$ when $\mathscr{U}$ is instantiated by EF, which makes the distinction possible.

Probability of Distinction in Function of the Number of Queries. Our proof revolves around the combinatorial argument that when the number of oracle queries is known, it is not too hard to bound the probability of encountering the *distinguishing situation* for a random choice of k_1 and k_2. In other words, given the amount of past queries, it is possible to bound the proportion of the key-space of $\mathscr{F}$ and $\mathscr{G}$ that will force at least one specific value for an input/output pair of $\mathscr{F}$ when $\mathscr{U}$ is instantiated by EF.

From the distinguishing situation, it is straightforward to express the probability of obtaining an input/output pair for $\mathscr{F}$ that can lead to distinction. That is, one that makes distinguishing possible if $\mathscr{F}$ is additionally queried with key k_1. This is equal to:

$$\Pr\left[\mathscr{U}^{-1}(z \oplus f(k_1, k_2)) \oplus f(k_1, k_2) \text{ def}, \mathscr{G}_{k_2}^{-1}(z) \text{ def}\right], \tag{3}$$

where z is arbitrary. We can see that the first term in (3) defines the input of a call to $\mathscr{F}$ that may possibly lead to distinction, and the second term defines the corresponding output of such a call. The full probability of distinguishing $\mathscr{U}$ on a query to $\mathscr{F}_{k_1'}(x)$ can then be written as:

$$\Pr\left[k_1' = k_1, \mathscr{U}^{-1}(z \oplus f(k_1, k_2)) \oplus f(k_1, k_2) \text{ def}, \mathscr{G}_{k_2}^{-1}(z) \text{ def}\right]. \tag{4}$$

We first bound the advantage of an adversary when assuming that f is a zero-ERF. That is, the output of a call to f is completely random until all of its input is known.

From (3), we can see that an input/output pair of $\mathscr{F}$ for a pair of key candidates (k_1', k_2') may become defined via two ways: either the adversary made a query to $f(k_1', k_2')$ and corresponding queries to $\mathscr{U}$ and $\mathscr{G}$; either he has guessed sufficiently many values for the output of $f(k_1', k_2')$ without querying f, and made many queries to $\mathscr{U}$ and $\mathscr{G}$ to be able to check and find the right value. We will say that an adversary behaving according to the first (resp. second) way follows the first (resp. second) strategy.

The potential advantage (no pun intended) of the second strategy is that it allows to work separately on the guesses for k_1 and k_2. This is reminiscent of the classical MiTM attack; however, we will see that with the EF construction, this advantage is still considerably reduced because of the presence of f.

Strategy One. We study the case of an adversary following the first strategy exclusively. We start by defining a variable τ_2, function of k_1' and k_2'. We use this variable to bound the probability for input/output pairs of terms from (3) to be defined for a pair of key candidates (k_1', k_2'). We have:

$$\tau_2(k_1', k_2') \triangleq \mathbb{1}(D)\,\mathbb{1}(q_2(k_2'))\,\mathbb{1}(q_f(k_1', k_2')). \tag{5}$$

We then define a variable θ_2, function of k_1', that we use to bound the number of cases defined for a fixed key candidate k_1'. This will be useful to bound the advantage of the adversary when k_1' is the right first key. This variable can be succinctly defined thanks to τ_2, and we have:

$$\theta_2(k_1') \triangleq \mathbb{1}(q_1(k_1'))\sum_{k_2'}\mathbb{1}(\tau_2(k_1', k_2')). \tag{6}$$

It follows that for a right guess k_1 of k_1', the advantage of the adversary can be bounded by $2^{-\kappa}\theta_2(k_1)$. Consequently, we define a final variable Θ_1 as:

$$\Theta_1 \triangleq \sum_{k_1'}\theta_2(k_1'). \tag{7}$$

This allows us to bound the advantage of an adversary by $2^{-2\kappa}\Theta_1$. If we fully develop Θ_1 again, this yields:

$$\Theta_1 \triangleq \mathbb{1}(D)\sum_{k_1'}\mathbb{1}(q_1(k_1'))\sum_{k_2'}\mathbb{1}(q_2(k_2'))\,\mathbb{1}(q_f(k_1', k_2')). \tag{8}$$

What is interesting here is that we need at least one call to f for every pair of candidates (k_1', k_2') for those candidates to contribute to the value of Θ_1. Therefore, we can bound Θ_1 by $\sum_{k_1', k_2'}\mathbb{1}(q_f(k_1', k_2'))$ without losing too much precision. This gives us the final bound we will use for the advantage of an adversary playing the $\mathscr{F}$ game and following strategy one:

$$Adv(0, D, q_1, q_2, q_f) \leq 2^{-2\kappa}\sum_{k_1', k_2'}\mathbb{1}(q_f(k_1', k_2')) = 2^{-2\kappa}\cdot q_f. \tag{9}$$

This means that to gain an advantage of one, an adversary will need at least $2^{2\kappa}$ queries q_f to the f oracle.

Strategy Two. We now turn our attention to an adversary exclusively following the second strategy. In this case, for the guess of a key candidate k_1', in order to check a key candidate k_2', there may be many guesses of the value of $f(k_1', k_2')$ to consider to gain a non-marginal advantage. In the case of strategy two, we redefine the variable τ_2 as:

$$\tau_2(k_1', k_2') \triangleq 2^{-n}\cdot Dq_2(k_2'). \tag{10}$$

We also redefine θ_2 as:

$$\theta_2(k_1') \triangleq \sum_{k_2'}\tau_2(k_1', k_2')\min\left(1, \frac{q_1(k_1')}{q_2(k_2')}\right) = 2^{-n}\cdot D\sum_{k_2'}\min(q_2(k_2'), q_1(k_1')). \tag{11}$$

In this case we do not have indicator functions as in strategy one anymore, as what becomes important is the *proportion* of guesses for $f(k_1,k_2)$ that were made out of how many possible values it can take (whereas in the case of strategy one, we explicitly compute f for the couples (k_1,k_2), and thus only care if the right one was once considered). We also witness the introduction of a *min* term that translates the fact that for a guess of the value of $f(\cdot)$ to be useful in an attack, it needs to be verifiable form both the plaintext and the ciphertext. If we put it in another way, the queries $q_1(k_1')$ (resp. $q_2(k_2')$) define *at most* $D\cdot q_1(k_1')$ (resp. $D\cdot q_2(k_2')$) differences with the plaintexts (resp. the ciphertexts); if more differences are defined with, say, the plaintexts than with the ciphertexts (because *e.g.* of more calls to q_1 than to q_2), some of those differences will not possibly be checked for the whole construction, and thereby will not contribute to the attack.

Now we can use θ_2 and define Θ_1 to bound the advantage of the adversary as previously; the only difference will be the full expression for Θ_1. In this case it will be:

$$\Theta_1 \quad \triangleq \quad 2^{-n}\cdot D\sum_{k_1'}\sum_{k_2'}\min(q_2(k_2'),q_1(k_1')). \tag{12}$$

Hence we can bound the advantage of an adversary playing the $\mathscr{F}$ game and following strategy two by:

$$Adv(0,D,q_1,q_2,q_f) \quad \leq \quad 2^{-2\kappa-n}\cdot D\sum_{k_1',k_2'}\min(q_2(k_2'),q_1(k_1')). \tag{13}$$

This means that to gain an advantage of one, an adversary will need at least $\frac{2^{\kappa+n}}{D}$ queries q_1 and q_2 to the $\mathscr{F}$ and $\mathscr{G}$ oracles, those being possibly made independently for the two oracles.

Mixing the two strategies. In order to bound the advantage of an adversary playing the $\mathscr{F}$ game regardless of the strategy it is following, it suffices to combine the bounds (9) and (13) together. We can do this easily without losing too much precision by taking the maximum of the two. Indeed we can see that the term $q_f(k_1',k_2')$ does not influence the value of (13), and that the number of to queries D, $q_1(k_1')$, and $q_2(k_2')$ does not influence the value of (9) in so far as it is greater than one. This gives the following result:

Lemma 1 *The advantage $Adv(0,D,q_1,q_2,q_f)$ of an adversary trying to distinguish $\mathscr{U}$ by playing the $\mathscr{F}$ game is upper-bounded by:*

$$2^{-2\kappa}\max\left(q_f,\quad 2^{-n}\cdot D\sum_{k_1',k_2'}\min(q_2(k_2'),q_1(k_1'))\right). \tag{14}$$

A.2 The $\mathscr{G}$ Game

This game being exactly symmetrical with the previous one, we only give the result here.

Lemma 2 *The advantage $Adv(0,D,q_1,q_2,q_f)$ of an adversary trying to distinguish $\mathscr{U}$ by playing the $\mathscr{G}$ game is upper-bounded by:*

$$2^{-2\kappa}\max\left(q_f,\quad 2^{-n}\cdot D\sum_{k_1',k_2'}\min(q_1(k_1'),q_2(k_2'))\right). \tag{15}$$

A.3 The $\mathscr{U}$ Game

We now consider the game that identifies the situation where queries to $\mathscr{U}$ itself may lead to distinction.

The distinguishing situation. On a query to $\mathscr{U}(p)\triangleq c$, then:

1. if $\mathscr{F}_{k_1}(p\oplus f(k_1,k_2))\triangleq y$ has been previously queried;
2. and if $\mathscr{G}_{k_2}(y)\triangleq z$ has been previously queried;

then we must have $c=z\oplus f(k_1,k_2)$ when $\mathscr{U}$ is instantiated with EF, which makes distinction possible.

Probability of Distinction in Function of the Number of Queries. Again, we express the probability of obtaining an input/output pair possibly leading to distinction. This is equal to:

$$\Pr\left[\mathscr{F}_{k_1}^{-1}(y) \oplus f(k_1,k_2) \text{ def}, \mathscr{G}_{k_2}(y) \oplus f(k_1,k_2) \text{ def}\right]. \tag{16}$$

In this equation, the first (resp. second) term corresponds to an input (resp. output) of a distinguishing call to $\mathscr{U}$. We can treat this equation in a similar fashion as previously, although we need to redefine the two strategies.

Strategy One. To compute our bound, we may define an intermediate variable τ_U, function of k_1' and k_2', that plays the same role as in the $\mathscr{F}$ game. We have:

$$\tau_U(k_1',k_2') \quad \triangleq \quad \mathbb{1}(q_1(k_1'))\,\mathbb{1}(q_2(k_2'))\,\mathbb{1}(q_f(k_1',k_2')). \tag{17}$$

For a right guess of k_1' (resp. k_2'), it is easy to see that we can bound the advantage of an adversary by $2^{-\kappa}\sum_{k_2'}\tau_U(k_1',k_2')$ (resp. $2^{-\kappa}\sum_{k_1'}\tau_U(k_1',k_2')$). It ensues that the full advantage of an adversary playing the $\mathscr{U}$ game by following strategy one can be bounded by:

$$2^{-2\kappa}\sum_{k_1',k_2'}\tau_U(k_1',k_2') \quad \leq \quad 2^{-2\kappa}\sum_{k_1',k_2'}\mathbb{1}(q_f(k_1',k_2')) \quad = \quad 2^{-2\kappa}\cdot q_f\,. \tag{18}$$

To gain an advantage of 1, an adversary will therefore need at least of the order of $2^{2\kappa}$ queries to the f oracle.

Strategy Two. In the case of strategy two, we need to redefine the variable τ_U. For each query p to $\mathscr{U}$, only joint queries to $\mathscr{F}$ and $\mathscr{G}^{-1}$ with similar differences from p and $\mathscr{U}(p)$ respectively may help to define the terms in 16; and then we have τ_U defined as:

$$\tau_U(k_1',k_2') \quad \triangleq \quad 2^{-n}\cdot D\sum_p q_1(k_1',p)\,q_2(k_2',p). \tag{19}$$

For a right guess of k_1' (resp. k_2') we can use this variable to bound the advantage of an adversary similarly as for strategy one. It ensues that the full advantage of an adversary playing the $\mathscr{U}$ game by following strategy two can be bounded by:

$$2^{-2\kappa}\sum_{k_1',k_2'}\tau_U(k_1',k_2') \quad = \quad 2^{-2\kappa-n}\cdot D\sum_{k_1',k_2',p} q_1(k_1',p)\,q_2(k_2',p). \tag{20}$$

To gain an advantage of 1, an adversary will need a workload at least of the order of $\frac{2^{\kappa+n}}{D}$ queries to the $\mathscr{F}$ and $\mathscr{G}$ oracles each, with those possibly being made independently one from another.

This gives us our result for the $\mathscr{U}$ game:

Lemma 3 *The advantage $Adv(0,D,q_1,q_2,q_f)$ of an adversary trying to distinguish $\mathscr{U}$ by playing the $\mathscr{U}$ game is upper-bounded by:*

$$2^{-2\kappa}\max\left(q_f,\quad 2^{-n}\cdot D\sum_{k_1',k_2',p} q_1(k_1',p)\,q_2(k_2',p)\right). \tag{21}$$

A.4 The f Game

The EF construction is defined with f being an ℓ-ERF, but so far we gave bounds for the advantage of an adversary specifically when ℓ is zero. We claim that taking ℓ greater than zero decreases the bounds by a factor at most $\binom{n}{\ell}2^{\ell}$, making the construction still relevant if ℓ is small (for instance, with $n = 128$, we can take $\ell = 3$ and lose around 22 bits of security). We prove this claim for the $\mathscr{F}$ game, but the proof easily generalizes to the other games.

First we should notice that the only strategy we need to consider is strategy one. Indeed, whether f is an ℓ-ERF or a zero-ERF has no impact if the value of $f(k_1', k_2')$ is guessed independently of (k_1', k_2'), because we expect f to have a full range in any case. With this in mind we can redefine the variable τ_2 for strategy one. We have:

$$\tau_2(k_1', k_2') \triangleq \mathbb{1}(D)\,\mathbb{1}(q_2(k_2'))\,\mathbb{1}\{\, q_f(k_1'', k_2'') \mid \mathrm{hw}(k_1'||k_2' \oplus k_1''||k_2'') \leq \ell\}. \tag{22}$$

We can plug this directly into the value of Θ_1 to find :

$$\Theta_1 \quad \triangleq \quad \mathbb{1}(D) \sum_{k_1'} \sum_{k_2'} \mathbb{1}(q_1(k_1'))\,\mathbb{1}(q_2(k_2'))\,\mathbb{1}\{\, q_f(k_1'', k_2'') \mid \mathrm{hw}(k_1'||k_2' \oplus k_1''||k_2'') \leq \ell\}. \tag{23}$$

As each query to $f(k_1', k_2')$ yields at most $2^\ell \binom{n}{\ell}$ pairs (k_1'', k_2'') such that $\mathbb{1}\{\, q_f(k_1'', k_2'') \mid \mathrm{hw}(k_1'||k_2' \oplus k_1''||k_2'') \leq \ell\}$ is not zero, we get the bound:

Lemma 4 *The advantage $Adv(\ell, D, q_1, q_2, q_f)$ of an adversary trying to distinguish $\mathcal{U}$ when following strategy one is upper-bounded by:*

$$2^{-2\kappa+\ell} \binom{n}{\ell} \cdot q_f\,. \tag{24}$$

A.5 Joining Pieces Together

The proof of theorem 2 is just obtained by combining the results of lemmata 1, 3, 4, and by remarking that the second term in lemma 3 is upper-bounded by the second term in lemma 1.