A New Mode of Operation for Block Ciphers and Length-Preserving MACs

Yevgeniy Dodis[1], Krzysztof Pietrzak[2], and Prashant Puniya[1]

[1] New-York University, USA, {dodis,puniya}@cs.nyu.edu
[2] CWI Amsterdam, The Netherlands, pietrzak@cwi.nl

Abstract. We propose a new mode of operation, *enciphered CBC*, for domain extension of length-preserving functions (like block ciphers), which is a variation on the popular CBC mode of operation. Our new mode is twice slower than CBC, but has many (property-preserving) properties not enjoyed by CBC and other known modes. Most notably, it yields the first constant-rate Variable Input Length (VIL) MAC from any length preserving Fixed Input Length (FIL) MAC. This answers the question of Dodis and Puniya from Eurocrypt 2007. Further, our mode is a secure domain extender for PRFs (with basically the same security as encrypted CBC). This provides a hedge against the security of the block cipher: if the block cipher is pseudorandom, one gets a VIL-PRF, while if it is "only" unpredictable, one "at least" gets a VIL-MAC. Additionally, our mode yields a VIL random oracle (and, hence, a collision-resistant hash function) when instantiated with length-preserving random functions, or even random permutations (which can be queried from both sides). This means that one does not have to re-key the block cipher during the computation, which was critically used in most previous constructions (analyzed in the ideal cipher model).

1 Introduction

Modes of operation allow one to build a Variable Input Length (VIL) primitive from a given Fixed Input Length (FIL) primitive. Currently, variants of two popular modes of operation are used to implements almost all known VIL primitives: the CBC mode, which operates on length preserving functions (like a block cipher), and the Merkle-Damgård (MD, aka as "cascade") mode, which operates on a compression function. In practice, the latter compression function h is often implemented out of a block cipher E via the Davies-Meyers transform: $h(x, y) = E_x(y) \oplus y$. Thus, one way or another, many useful primitives are built from a block cipher in practice. Unfortunately, we argue that neither the CBC nor the MD mode are entirely satisfactory and a new block cipher mode of operation is needed.

CBC MODE. Cipher Block Chaining (CBC) is a popular mode of operation for domain extension of pseudorandom functions (PRFs) [3], thus allowing one to build a MAC (recall that a PRF is a MAC) on roughly $n\ell$ bits by making ℓ calls to an n-bit block cipher E. However, here one must assume that E is a PRF, even if finally one is only interested in getting a MAC. Pseudorandomness is a much stronger assumption than unpredictability (which is all we need from a MAC). Thus, it is natural to ask if the CBC-MAC is secure if the block cipher is "only" *unpredictable*, in other words, if CBC is a good domain extender for MACs. Aside from being of great theoretical importance, a positive answer to this question would provide a hedge against the security of the block cipher E: if E is pseudorandom, one gets a VIL-PRF, while if it is only unpredictable, one at least gets a VIL-MAC. Unfortunately, An and Bellare [1] showed that this is not the case.[3] This motivates the following central question of this work:

Question 1. Is there a simple and efficient way to build a VIL-MAC from a length-preserving MAC (like an unpredictable block cipher)?

This question was recently explicitly addressed by Dodis and Puniya [14]. They argued that none of the existing techniques (as opposed to just CBC) give a satisfactory answer to this question (see [14] for a list of many failed approaches). They also presented the best-known-to-date solution. The idea is to use the Feistel network for $\omega(\log \lambda)$ rounds (where λ is the security parameter) to get a MAC from $2n$ to $2n$ bits. Then one can safely chop half of the output bits, getting a $2n \mapsto n$ bit MAC, after which one can apply any of the known efficient techniques to extend the domain of a "shrinking MAC" [1, 16]. While elegant, this solution evaluates the given FIL-MAC $\omega(\ell \log \lambda)$ times to extend the domain of the FIL-MAC by a factor of ℓ. In contrast, the solution we present shortly will only use 2ℓ calls.

Coming back to CBC, another drawback of this mode is that it does not appear to be useful for building collision-resistant hash function (CRHFs) or random oracles (ROs) from block ciphers, even if the block cipher is modeled as an ideal cipher. Indeed, if the key to the block cipher is fixed and public, it is trivial to find collisions in the CBC mode, irrespective of the actual cipher.

MD MODE. Unlike the CBC mode, the MD mode seems to be quite universal, and variants of it were successfully used to argue domain extension results for many properties, including collision-resistance [11, 18, 21, 8], pseudorandomness [5, 6], unforgeability [1, 16], indifferentiability from a random oracle [10], randomness extraction [12] and even "multi-property preservation" [7]. However, when using a block cipher, we will first have to construct a compression from the block cipher before we can apply MD.

One trivial way of doing this would to simply chop part of the output of E. However, this is very unsatisfactory on multiple levels. First, to achieve constant efficiency rate for the cascade construction, one must chop a constant fraction of the output bits. However, already chopping a super-logarithmic number of bits will not, in general, preserve the security of E as a MAC, making it useless for answering Question 1. Second, even for the case of PRFs and ROs, where chopping a linear fraction of bits does preserve the corresponding property, one loses a lot in exact security, since the output is now much shorter. For example, dropping half of the bits would give a VIL-PRF with efficiency rate 2 and security $\mu^2/2^{n/2}$ (where μ is the total length of queried messages), compared to efficiency rate 1 and security $\mu^2/2^n$ achieved by CBC.

As another option, which is what is done in practice, one could construct the compression function via the Davies-Meyers transform $h(x, y) = E_x(y) \oplus y$. For one thing, this is not very efficient, as it requires one to re-key the block cipher for every call, which is quite expensive for current block ciphers. (For example, eliminating this inefficiency was explicitly addressed and left as a challenge by Black, Cochran and Shrimpton [9].) More importantly, however, using the Davies-Meyers construction requires very strong assumptions on the block cipher to prove security. Namely, one can either make an

[3] Their attack was specific to a two-block CBC, but it is not hard to extend it to more rounds.

ad hoc assumption that the Davies-Meyers compression function satisfies the needed domain extension property (such as being a PRF or a MAC), or formally prove the security of the construction in the ideal cipher model. Both of these options are unsatisfactory. The first option is provably not substantiated even is the block cipher E is assumed to be a pseudorandom permutation (PRP): for example, one can construct (artificial) PRPs for which the Davies-Meyers construction is not even unpredictable. As for the second option, it might be acceptable when dealing with strong properties, like collision-resistance or indifferentiability, when it is clear that the basic PRP property of the block cipher will not be enough [25]. However, to get pseudorandomness, or even unpredictability, going through the ideal cipher argument seems like a very (and unnecessarily) heavy hammer.

NEW MODE OF OPERATION. The above deficiencies of the CBC and the MD mode suggest that there might be a need to design a new mode of operation based on block ciphers, or, more generally, length-preserving (keyed or unkeyed) functions. We propose such a mode which will satisfy the following desirable properties:

- The mode is efficient. If the message length is ℓ blocks, we evaluate the block cipher at most $c\ell$ times (c is called the *efficiency rate*; we will achieve $c = 2$).
- The mode uses a small, constant number of (secret or public, depending on the application) keys for the block cipher. In particular, one never has to re-key the block cipher with some a-priori unpredictable value.
- It gives a provably secure VIL-MAC from length-preserving FIL-MAC, answering Question 1.
- It gives a provably secure VIL-PRF from a length-preserving FIL-PRF, therefore providing the hedge against the security of the block cipher E: if E is pseudorandom, the mode gives a PRF; if E is only unpredictable, one at least gets a MAC.
- It gives a way to build a VIL-RO (and, hence, a VIL-CRHF) from several random permutations.
- The mode is elegant and simple to describe.

Of course, simply being a "secure" domain extension for PRF/MAC/RO is not enough: the exact security achieved by the reduction is a crucial parameter, and we will elaborate on this later in this section.

ENCIPHERED CBC. The mode, *enciphered CBC*, we present in this paper is a relatively simple variant of the CBC mode. We first describe our "basic" mode, which works for domain-extension of MACs, PRFs and ROs, and later show the changes needed to make it work with (random) permutations as well.[4] The basic mode, depicted in Figure 1, consists of three independent length-preserving functions f_1, f_2, f_3 (either keyed or not, depending on whether we are in the secret key setting, or in the random oracle model). First, we define an auxiliary compression function $g(x, y) = f_1(x) \oplus f_2(y)$. Intuitively, the key property of this function — which will hold in all our applications — is that it is weakly collision-resistant (WCR) [1]. This means that, given oracle access to f_1 and f_2, it is infeasible to find a collision for g. Then we use $g(x, y)$ as the compression function in the usual MD mode with strengthening: namely, we apply the Merkle-Damgård

[4] In the random permutation model (where there are no secret keys) we need to worry about the inverse queries of the attacker. In contrast, in the secret key setting, a PRF is also a PRP, so the simpler mode already works for the domain extension of MACs and PRFs.

chaining to the message $(x_1 \dots x_\ell, \langle \ell \rangle)$, where $(x_1 \dots x_\ell)$ is our original message, and get output z. Finally, we output $f_3(z)$ as the value of our (basic) enciphered CBC.[5]

As we argue, if f_1, f_2, f_3 are three independent (keyed) MACs, then the above construction is a (three-keyed) VIL-MAC, answering Question 1. Also, although about twice less efficient than CBC, enciphered CBC also preserves the PRF property. On the other hand, if f_1, f_2, f_3 are random oracles, then the construction is indifferentiable from a VIL-RO. Finally, if we *assume* that f_1 and f_2 are such that $g(x, y)$ above is collision-resistant, then the mode which outputs the value z (and not $f_3(z)$) above is trivially collision resistant, since this is simply the usual MD transform with strengthening applied to a FIL-CRHF. Thus, if f_3 is "collision-resistant" (either trivially if it is a permutation, or even computationally), then enciphered CBC gives a VIL-CRHF. Of course, the assumption on g is not entirely satisfactory, but we argue that it is meaningful in the standard model.

OPTIMIZATIONS. We also show several optimizations of our mode which, while slightly less efficient, also work for two, or even one length-preserving round function. We only mention the two-key mode, since the one-key mode is a bit less "elegant" and intuitive to describe. The solution we propose (using two functions f and f') is to view $\{0, 1\}^n$ as the finite field $\mathbb{GF}(2^n)$, and then use the three-key solution with functions $f_1(x) = f(x)$, $f_2(y) = \alpha \cdot f(y)$ and $f_3(z) = f'(z)$, where α is any constant in $\mathbb{GF}(2^n)$ different from 0 and 1.[6] Then, we show that the resulting function $g(x, y) = f(x) \oplus \alpha \cdot f(y)$ is still WCR in all our applications.

Finally, we show how to extent the basic enciphered CBC mode to the case of random permutations. As already mentioned in Footnote 4, this is only the issue in the results concerning the random permutation model, since there the attacker can try to invert the random permutation. Indeed, the function $g(x, y) = f_1(x) \oplus f_2(y)$ is obviously *not* collision-resistant (which is crucial for our proof) if the attacker can invert f_1 or f_2. Our solution is to use the Davies-Meyers transform, but without the key. Namely, if π_1 and π_2 are random permutations, we essentially apply the previous mode to functions $f_1(x) = \pi_1(x) \oplus x$ and $f_2(y) = \pi_2(y) \oplus y$. This ensures that the function $g(x, y) = \pi_1(x) \oplus x \oplus \pi_2(y) \oplus y$ is still WCR, even with the oracle access to π_1^{-1} and π_2^{-1}. As for the function f_3, it really must look like a random oracle, so we use a slightly more involved construction $f_3(z) = \pi_3(z) \oplus \pi_3^{-1}(z)$.[7] With these definitions of f_1, f_2 and f_3 using π_1, π_2 and π_3, we get our final enciphered CBC mode on block ciphers. (As we mentioned, though, the simplified mode already works for the case of PRFs and MACs.) We believe that optimizations similar to those made to the simplified mode, might also reduce the number of random permutations below three, but we leave this question to future work.

---

[5] One can also describe enciphered CBC as "enciphering" the input and the output of the standard CBC mode applied to f_1: we encipher all the input blocks (except the first) with f_2, and the output block — with f_3. This (less useful) view explains the name of the mode.

[6] We recommend the constant corresponding the the "polynomial" X in $\mathbb{GF}(2^n)$, since multiplication by this polynomial in $\mathbb{GF}(2^n)$ corresponds to one right shift and one XOR (the latter only if there is a carry), which is very efficient.

[7] This construction is of independent interest since it shows an indifferentiable construction of an n-to-n-bit random oracle from an n-to-n bit random permutation.

SECURITY. We will now discuss how the security of our mode for MAC/PRF/RO compares to known constructions. Recall that a mode of operation has rate c if it makes $c\ell$ calls to the underlying primitive when given an ℓ-block message. We achieve $c = 2$.

We will say that a domain extension for MACs has security d, if the security of the mode is $\epsilon \cdot \mu^d$ where ϵ is the security of the underlying FIL MAC and μ denotes the total length of the messages an adversary is allowed to query. Our mode achieves security $d = 4$, and this is the first constant-rate construction to achieve any security at all. For *shrinking* MACs $\{0,1\}^{n+k} \to \{0,1\}^n$, An and Bellare [1] show that a version of Merkle-Damgård gives a secure domain extension with security $d = 2$ (and rate $c = n/k$, which is constant if $k = \Omega(n)$). This security is much better than what we achieve, but it is unclear how to build a shrinking MACs with good security and compression efficiency (i.e., $k = \Omega(n)$) from a length-preserving MAC. Indeed, prior to this work, the best known construction of Dodis and Puniya builds a shrinking MAC with rate $c = \omega(\log \lambda)$ (where λ is the security parameters) and security $d = 6$, which is inferior to our $c = 2$ and $d = 4$.

As for PRFs, our mode achieves basically the same security $\mu^2/2^n$ as encrypted CBC, which is the best security known for constructions which are iterated, stateless and deterministic. In fact, as discussed in Section 3.3,we will achieve even better exact security when using PRPs (i.e., block ciphers) in place of length-preserving PRFs.

Similarly to MACs, we will say that a construction of a VIL-RO has security d, if it is $\mu^d/2^n$ indifferentiable from a random oracle when instantiated with FIL-ROs or RPs. With this convention, our construction has security $d = 4$. Recently, Maurer and Tessaro [17] give a pretty involved construction with the optimal security rate $d \to 1$ (at the expense of large efficiency rate $c = O(1)$), while the results of Coron et al. [10] for domain extension of "shrinking ROs" easily imply (by chopping some output bits of the length-preserving RO) a range of constructions with efficiency c and security $\mu^2/2^{(1-1/c)n}$. Although approaching security $d = 2$ for a large constant c, for $c = 2$ this gives poorer security $\mu^2/2^{n/2}$ than the security $\mu^4/2^n$ of enciphered CBC.

In the context of building VIL-CRHFs from length-preserving ROs or RPs, Shrimpton and Stam [24] give a simple construction from ROs with $c = 3$ and optimal $d \approx 2$, while Rogaway and Steinberger [23] recently reported a more complicated construction from RPs with $c = 3$ and optimal $d \approx 2$. Additionally, in a companion paper [22] they showed the necessity of non-trivial efficiency/security tradeoffs for any construction of VIL-CRHF in the random permutation model. This suggests the existence of similar (or worse) tradeoffs for the related question of building VIL-RO from length-preserving FIL-RO (or RP).

To summarize this discussion, we designed the first mode of operation *simultaneously* satisfying several demanding properties, some of which were never satisfied before (even in isolation). We conjecture that *any such mode must require some non-trivial tradeoff between efficiency and security*. Our specific mode, while simple and elegant, might not give such optimal tradeoffs. In particular, its security of "only" $\mu^4/2^n$ for the case or ROs and $\epsilon \cdot \mu^4$ for MACs is particularly unsatisfying to make it useful in practice (where $n = 128$; note that $\epsilon \geq 2^{-n}$). It is an interesting open question to understand the optimal efficiency/security tradeoffs, and to potentially improve upon our specific enciphered CBC mode of operation.

2 Preliminaries

We assume that the reader is familiar with the basic security definitions for MACs, PRFs, CRHFs and indifferentiability from RO. We use exact security definitions for each of these primitives.

MACS AND PRFS. The security of a MAC is measured via its resistance to existential forgery under *chosen message attack* (see [3]). A function family F is a (t, q, μ, ϵ)-secure MAC if the success probability of any attacker with running time t, number of queries q and total message length μ is at most ϵ. Similarly, the security of PRFs is measured in terms of its indistinguishability from a truly random function under a chosen message attack, and a (t, q, μ, ϵ)-secure PRF is similarly defined.

INDIFFERENTIABILITY FROM RANDOM ORACLE. We follow the definitions of [10] for indifferentiability of a construction from an ideal primitive $\mathcal{F}$ (which will be a random oracle in this paper). A construction C, that has oracle access to ideal primitive $\mathcal{G}$, is $(t_D, t_S, q, \mu, \epsilon)$-indifferentiable from another ideal primitive $\mathcal{F}$, if there is a $\mathcal{G}$ simulator S that runs in time at most t_S, such that any attacker D with running time t_D, number of queries q and total query length μ can distinguish the $\mathcal{F}$ model (with access to $\mathcal{F}$ and S) from the $\mathcal{G}$ model (with access to C and $\mathcal{G}$) with advantage at most ϵ.

COLLISION RESISTANCE. A function family F is (t, ϵ)-secure CRHF family, if the advantage of any attacker running in time t to find a collision for an f sampled at random from F, is at most ϵ.

3 Three-key enciphered CBC construction

In this section, we will define the three-key enciphered CBC mode of operation and analyze its security under various notions.

First, we make some auxiliary definitions. Given two length-preserving functions $f_1, f_2 : \{0,1\}^n \to \{0,1\}^n$, we define the shrinking **XOR compression function**, $g[f_1, f_2]$, from $2n$ bits to n bits by $g[f_1, f_2](x_1 \parallel x_2) \stackrel{\text{def}}{=} f_1(x_1) \oplus f_2(x_2)$, where $x_1, x_2 \in \{0,1\}^n$. Given this function, we define the **XOR hash function** $G[f_1, f_2]$ to be simply the cascade construction applied to the XOR compression function. Namely, given input $x = x_1 \parallel \ldots \parallel x_\ell$, where $x_i \in \{0,1\}^n$, we let

$$G[f_1, f_2](x_1 \parallel \ldots \parallel x_\ell) \stackrel{\text{def}}{=} g[f_1, f_2](x_\ell \parallel g[f_1, f_2](\ldots g[f_1, f_2](x_2 \parallel x_1) \ldots))$$

THE CONSTRUCTION. The new mode of operation, $H[f_1, f_2, f_3]$, uses three length-preserving functions $f_1, f_2, f_3 : \{0,1\}^n \to \{0,1\}^n$ and takes a variable-length input $x = x_1 \parallel \ldots \parallel x_\ell$ (wlog, we assume the length to be a multiple of n; if not, then a suitable encoding scheme can be used to achieve this, such as appending a 1 followed by 0s). It simply applies the XOR hash function $G[f_1, f_2]$ described above to a suffix-free encoding of the input, followed by the third length-preserving function f_3. The particular suffix-free encoding we use is *Merkle-Damgård (MD) strengthening* [11, 18], where one simply appends the input length $\langle \ell \rangle$ to the input. The resulting mode, depicted in Figure 1, is called *enciphered CBC mode*, and it is defined as:

$$H[f_1, f_2, f_3](x_1 \parallel \ldots \parallel x_\ell) = f_3(G[f_1, f_2](x_1 \parallel \ldots \parallel x_\ell \parallel \langle \ell \rangle))$$

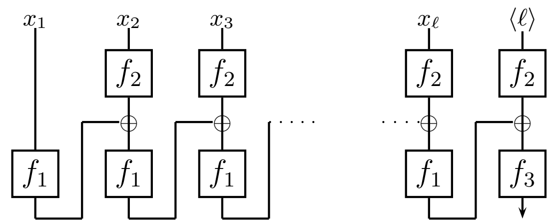

Fig. 1. The basic three-key enciphered CBC construction $H[f_1, f_2, f_3]$.

3.1 VIL-MAC from length-preserving FIL-MAC

In this section we will prove, that unlike plain CBC, the enciphered CBC (cf. Figure 1) does give a secure VIL-MAC when instantiated with a length preserving MACs (here denoted $f_{k_1}, f_{k_2}, f_{k_3}$ to emphasize the secret keys k_1, k_2, k_3). We will use an elegant methodology of An and Bellare [1] which they used to analyze their *NI Construction* of a VIL-MAC from a shrinking FIL-MAC. However, we will see that it will be useful in our setting as well. In brief, the methodology introduced a notion of *weak collision-resistance* (WCR) and essentially reduced the construction of a VIL-MAC to that of a FIL-WCR. Details follow.

WEAK COLLISION-RESISTANCE (WCR). Consider a keys family of functions $F = \{f_k\}$, and the following attack game involving this function family. An attacker A gets oracle access to f_k (for random k) and returns a pair of messages $m \neq m'$ in the domain of F. The attacker A wins if these message collide: $f_k(m) = f_k(m')$. The function family F is said to be a (t, q, μ, ε)-*secure WCR* function family if the success probability of any attacker with running time t, number of queries q and total message length μ is at most ε.

FROM WCR TO MAC. The methodology of An and Bellare [1] utilized the notion of WCR via the following reasoning (which we immediately attempt to apply to the case of enciphered CBC).

Step 1: The composition of a FIL-MAC f_k and a WCR function $h_{k'}$ is a secure MAC $f_k(h_{k'}(\cdot))$ (Lemma 4.2 [1]). Applied to enciphered CBC, where f_{k_3} is a FIL-MAC, it means that it suffices to show that the XOR hash function $G[f_{k_1}, f_{k_2}]$, with suffix-free inputs, is a VIL-WCR.

Step 2: The cascade construction, with suffix-free inputs, applied to a FIL-WCR function gives a VIL-WCR function (Lemma 4.3 [1]). In our case, the XOR hash function is exactly the required cascade construction applied to the XOR compression function $g[f_{k_1}, f_{k_2}]$. Thus, it suffices to show that the latter is FIL-WCR.

Step 3: Build a FIL-WCR. In the case of the NI Construction of [1], one needed to build a FIL-WCR from a shrinking MAC, which was easy to do: any shrinking FIL-MAC is FIL-WCR (Lemma 4.4 [1]). Applied to our setting, it would suffice to show that the XOR compression function $f_{k_1}(x_1) \oplus f_{k_2}(x_2)$ is a FIL-MAC. However, this is easily seen to be false: for example, the XOR of its outputs applied to inputs $(x_1 \parallel x_2)$, $(x_1 \parallel x_2')$, $(x_1' \parallel x_2)$ and $(x_1' \parallel x_2')$ is always 0^n, which easily leads to a forgery.

Despite this "setback", we give a direct proof that the XOR compression function is a FIL-WCR, despite not being a FIL-MAC. And this is all we need.

Lemma 1. Let $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a family of functions. Define the function family $g[f_{k_1}, f_{k_2}](x_1 \parallel x_2) \stackrel{\text{def}}{=} f_{k_1}(x_1) \oplus f_{k_2}(x_2)$. If the function family f is a (t, q, qn, ϵ)-secure MAC family, then $g[f_{k_1}, f_{k_2}]$ is a $(t', q, 2qn, \epsilon \cdot q^4/2)$-secure WCR family, where $t' = t - O(qn)$.

Proof: Let A be an adversary which finds a collision for $g[f_{k_1}, f_{k_2}]$ with probability ϵ' (if k_1, k_2 are uniformly random). From such an A we will construct a new adversary B which is basically as efficient as A, and which forges f with probability at least $2\epsilon'/q^4$. Instead only giving A access to $g[f_{k_1}, f_{k_2}]$, we allow A to make q queries to f_{k_1} and f_{k_2} respectively, but we require this queries are made alternately, i.e. after a query to f_{k_b}, A must make a query to $f_{k_{3-b}}$ (note that such an A can trivially simulate q queries to $g[f_{k_1}, f_{k_2}]$). Moreover we assume that if $x_1\|x_2, x_1'\|x_2'$ is A's final output, then A always made the f_{k_1} queries x_1, x_1' and the f_{k_2} queries x_2, x_2' (this can be done wlog. if we allow A two extra queries to f_{k_1} and f_{k_2} respectively). Assume A is successful, and finds a collision $x_1\|x_2 \neq x_1'\|x_2'$ for $g[f_{k_1}, f_{k_2}]$. We say that a query x (say to f_{k_1}) is a winner query, if it is the first query where for some b, c, d, the pair $x\|b \neq c\|d$ is a collision for $g[f_{k_1}, f_{k_2}]$ and A already knows (i.e. made the queries) $f_{k_2}(b), f_{k_1}(c), f_{k_2}(d)$. Note that if A found a collision, then it must have made a winner query. Our attacker B, which must forge f_k (for some random unknown k) is now defined as follows. First B flips a random coin $r \in \{1,2\}$, and samples a random key k' for f. Now B lets A attack f_{k_1}, f_{k_2}, where $f_k = f_{k_r}$ and $f_{k'} = f_{k_{3-r}}$. During the attack, for a random $i, 2 \leq i \leq q$, B stops when A makes the i'th query x to f_{k_r} and "guesses" that this will be the winning query. Then B randomly chooses three already made queries b, c, d, conditioned on $x\|b \neq c\|d$ (hoping that $x\|b, c\|d$ is a collision), and guesses the forgery $\rho := f_{k_{3-r}}(b) \oplus f_{k_r}(c) \oplus f_{k_{3-r}}(d)$ for $f_{k_r} = f_k$ for the message x. Note that ρ is a good forgery for $f_k = f_{k_r}$, if $x\|b, c\|d$ is indeed a collision for $g[f_{k_r}, f_{k_{3-r}}]$. Thus B will be successful if A makes a winning query (which happens with probability ϵ'), and moreover B correctly guesses r (i.e. whether the winning query will be a f_1 or f_2 query), the index i of the winning query and also the three other queries involved. The probability of all that guesses being correct is at least $2\epsilon'/q^4$. By assumption (on the security of f as a MAC) we have $2\epsilon'/q^4 \leq \epsilon$, thus the success probability of B must be at most $\epsilon \cdot q^4/2$ as claimed. $\square$

Combining this result with the Lemmas 4.2 and 4.3 from [1], we immediately get

Theorem 1. *Let $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a (t, q, qn, ε)-secure length-preserving FIL-MAC. Then $H[f_{k_1}, f_{k_2}, f_{k_3}](.)$ (where k_1, k_2, k_3 is the secret key) is a $(t', q, qn, \varepsilon \cdot q^4)$-secure variable input-length MAC, where $t' = t - O(qn)$.*

3.2 VIL-RO from length-preserving FIL-RO

In this section we show that the enciphered CBC mode provides a domain extension for length-preserving ROs (in the sense of [10]).

Theorem 2. *Consider three length-preserving ROs $f_1, f_2, f_3 : \{0,1\}^n \to \{0,1\}^n$. Then the enciphered CBC construction $H[f_1, f_2, f_3]$ is $(t_D, t_S, q, \mu, \epsilon)$-indifferentiable from a VIL-RO. Here $t_S = \mathcal{O}(q^2)$, $\epsilon = \mathcal{O}((q+\mu)^4/2^n)$ and t_D is arbitrary.*

One might hope that the proof of this theorem can be given by using the corresponding indifferentiability result of Coron et al [10] for the NMAC construction. However, this intuition turns out to be incorrect since in order to use the result of [10], we will need to show that the *XOR compression function* $g[f_1, f_2]$ is indifferentiable from a FIL-RO from $2n$ bits to n bits. But this is clearly false, since for three n-bit input blocks x, y, y', we can see that $g[f_1, f_2](x \parallel y) \oplus g[f_1, f_2](x \parallel y')$ is independent of the n-bit block x which is certainly not true for an ideal FIL-RO!

Hence we give a direct proof for this result. In the proof, we need to construct a FIL-RO simulator that responds to the queries made by the indifferentiability attacker A to the FIL-ROs f_1, f_2 and f_3 in the VIL-RO model. Roughly speaking, the simulator responds to f_1 and f_2 queries at random and hopes that no collisions occur for the input to f_3 in the last round of the enciphered CBC construction. If no such collisions occur, then it can adjust its response to f_3 queries to match the VIL-RO output on the corresponding variable-length input (which it finds by searching through its previous responses).

Proof: We will prove the indifferentiability of the enciphered CBC mode of operation $H[f_1, f_2, f_3]$ from a variable input-length random oracle (VIL-RO) $F : \{0,1\}^* \rightarrow \{0,1\}^n$, in the random oracle model for the underlying fixed input-length functions $f_1, f_2, f_3 : \{0,1\}^n \rightarrow \{0,1\}^n$. The proof consists of two parts: the description of the FIL-RO simulator and the proof of indifferentiability.

The Simulator. The simulator S responds to queries of the form (i, x), where $i \in \{1, 2, 3\}$ and $x \in \{0,1\}^n$. In particular, the response $y \in \{0,1\}^n$ of the simulator S to a query (i, x) will be interpreted as the output $f_i(x)$ by the distinguisher, i.e. $y = f_i(x)$. The simulator also maintains a table $\mathcal{T}$ consisting of entries of the form (i, x, y), for each query (i, x) that it responded to with the output y.

f_1 QUERIES. In response to a query of the form $(1, x)$, the simulator S looks up its table for an entry of the form $(1, x, y)$. If it finds such an entry, then it responds with the output y recorded in this tuple, otherwise it responds to this query by choosing an output y that is uniformly distributed over $\{0,1\}^n$ and records the tuple $(1, x, y)$ in its table $\mathcal{T}$.

f_2 QUERIES. The simulator responds to queries of the form $(2, x)$ in the same way as it responds to f_1 queries, i.e. first looking up its table for a matching tuple $(2, x, y)$, else responding with a fresh uniformly distributed output y.

f_3 QUERIES. In response to queries of the form $(3, x)$, the simulator needs to check if there is a variable length input X, such that it needs to be consistent with the VIL-RO output $F(X)$ on this input. It firsts looks up its table $\mathcal{T}$ to find out if there is a matching tuple $(3, x, y)$ corresponding to a duplicate query, in which case it responds with y. Otherwise, it looks up the table $\mathcal{T}$ for a sequence of tuples $(1, x_1^1, y_1^1) \ldots (1, x_i^1, y_i^1)$ and $(2, x_1^2, y_1^2) \ldots (2, x_i^2, y_i^2)$, that satisfy the following conditions:

(a) For $j = 2 \ldots i$, it holds that $x_j^1 = y_{j-1}^1 \oplus y_{j-1}^2$.
(b) For the last tuples $(1, x_i^1, y_i^1)$ and $(2, x_i^2, y_i^2)$, it holds that the current f_3 input $x = y_i^1 \oplus y_i^2$.
(c) The bit string $x_1^1 \parallel x_1^2 \parallel \ldots \parallel x_i^2$ is such that $x_i^2 = \langle i \rangle$. That is, it should be the output of Merkle-Damgård strengthening applied to a valid input.

If the simulator finds such a sequence of tuples, then it queries the VIL-RO F to find out the output $y = F(x_1^1 \parallel x_1^2 \parallel \ldots \parallel x_{i-1}^2)$ and responds to the query $(3, x)$ with the output y, and records the tuple $(3, x, y)$ in its table $\mathcal{T}$. If it does not find such a sequence of tuples then it responds with a uniformly random output $y \in \{0,1\}^n$ and records $(3, x, y)$ in $\mathcal{T}$.

The proof of indifferentiability is postponed to the full version of this paper [13]. □

3.3 VIL-PRF from length-preserving FIL-PRF

If we remove the f_2 boxes in our enciphered CBC mode of operation (cf. Figure 1), we get a well known mode of operation called *encrypted CBC*, which is known to be a good domain extension for PRFs [19, 20]. The security of encrypted CBC (i.e. the distinguishing advantage from a uniformly random function, URF) when instantiated with two PRFs is $(\mu^2/2^n + 2\epsilon)$, where μ is the total length (in n bit blocks) of the messages queried and ϵ is a term that accounts for the insecurity of the underlying PRF. It is not surprising that our enciphered CBC mode is almost as secure, as the application of f_2 (not present in the usual encrypted CBC mode) does not affect the security by much.

Theorem 3. *Let $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a $(t, \mu, \mu n, \epsilon)$-secure FIL-PRF family. Then $H[f_{k_1}, f_{k_2}, f_{k_3}](.)$ is a $(t', q, \mu n, 2\mu^2/2^n + 3\epsilon)$-secure VIL-PRF family where $t' = t - O(qn)$.*

We will not formally prove this theorem, but just explain how it follows from the known $(t', q, \mu n, \mu^2/2^n + 2\epsilon)$ security of the encrypted CBC-MAC (under the same assumption on the PRF like in the theorem). The main observation here is that we can turn any distinguisher D for enciphered CBC into a distinguisher D' for encrypted CBC, by simply sampling some key k_2 at random, and then enciphering with f_{k_2} (except the first block) the queries made by D, before forwarding them to the oracle of D'. If the oracle of D' is *encrypted* CBC, then the oracle's answers look *exactly* as if they were computed by an *enciphered* CBC. In the ideal experiment, where the oracle of D' is a VIL-URF, the oracle's answers still look uniformly random, even if the input is first applied to f_{k_2}, unless D makes two queries containing blocks $x \neq x'$ which collide on f_{k_2}. The probability of that happening can be upper bounded by $\mu^2/2^n + \epsilon$, as f_{k_2} can be distinguished from a URF with advantage at most ϵ, and the probability to find a collision for a URF with range $\{0,1\}^n$ making μ queries is at most $\mu^2/2^n$. This $\mu^2/2^n + \epsilon$ accounts for the gap in the security for enciphered CBC (as in the theorem) and encrypted CBC (as mentioned above).

IMPROVING THE BOUND FOR BLOCK CIPHERS. As just explained, the gap in the security of encrypted and enciphered CBC is bounded by the probability that one can find a collision for the PRF f_{k_2}. Thus, if f_{k_2} is a permutation (where there are no collisions), $(t, q, \mu n, \delta)$-security for encrypted CBC implies basically the same $(t - O(\mu n), q, \mu n, \delta)$ security for enciphered CBC. This observation is useful, as in practice the PRF is usually instantiated by a block cipher, which is a permutation. And further, for the encrypted CBC mode of operation, one can prove much better bounds than $(\mu^2/2^n + 2\epsilon)$ if both f_{k_1} and f_{k_2} are assumed to be pseudorandom permutations

(PRPs) [4, 20] as opposed to PRFs. Thus, this better bounds for encrypted CBC translate directly to our mode of operation. To state the improved bounds, one must assume an upper bound ℓ on the length of *each* message queried by the distinguisher (this should not be a problem in practice, as the bound can be exponential). Let q be the number of queries the adversary is allowed to make, then if no messages is longer than $\ell \leq 2^{n/4}$ (and thus the total length μ is at most ℓq), the security of encrypted CBC instantiated with PRPs is $q^2\ell^{\Theta(1/\ln\ln\ell)}/2^n$ (plus some ϵ term accounting for the insecurity of the PRP). With the stronger condition that $\ell \leq 2^{n/8}$, one gets an even stronger $O(q^2/2^n)$ bound [20], which is tight up to a constant factor. Note that this is much better than the $O(q^2\ell^2/2^n)$ bound implied by Theorem 3, and in particular is independent of the message length ℓ.

3.4 Collision Resistance of Enciphered CBC

Now we discuss the collision-resistance of the enciphered CBC mode of operation. Note that the problem of constructing variable input-length CRHFs from length-preserving collision-resistant (CR) functions does not make much sense, since it is trivial to construct length-preserving CR functions (such as the identity function). However, as discussed in the introduction, we can make the following simple observation about the enciphered CBC mode of operation.

Lemma 2. Consider three length-preserving functions f_1, f_2 and f_3 on n bits. If the XOR compression function $g[f_1, f_2]$ and the function f_3 are collision-resistant, then the enciphered CBC mode of operation, $H[f_1, f_2, f_3]$, is collision-resistant as well.

This observation is a simple consequence of the result of Merkle-Damgård [11, 18], since we already use a suffix-free encoding the the enciphered CBC mode. Notice that assuming that a length-preserving function f_3 is a CRHF is a very mild requirement, since any permutation trivially satisfies this property. Thus, we the main assumption we need is that the XOR of functions f_1 and f_2 is a CRHF. Of course, in the random oracle model, it is well known the the XOR of two random oracles is collision-resistant (in fact, in this setting we showed in Section 3.2 that the enciphered CBC mode even gives a VIL-RO, let alone a "mere" VIL-CRHF).

Our point is that it is not essential to make idealized assumptions on the functions f_1 and f_2 to prove collision resistance of the construction $g[f_1, f_2]$. For instance, consider any finite field $\mathbb{F}$ for which the *discrete logarithm* problem is hard, and whose elements can be naturally encoded as binary strings. Define the functions $f_1, f_2 : \{0,1\}^n \to \{0,1\}^n$ as $f_1(x) = gen_1^x$ and $f_2(x) = gen_2^x$, where gen_1 and gen_2 are two generators of $\mathbb{F}$. Further, let us replace the XOR operation in $g[f_1, f_2]$ by a field-multiplication over $\mathbb{F}$. Then we get a new function $g(x \parallel y) = gen_1^x \cdot gen_2^y$ which is provably collision-resistant under the discrete log assumption. Coupled with the RO justification, this example suggests that our assumption on $g[f_1, f_2]$ is not too unreasonable.

We stress, though, that the XOR compression function is definitely *not* collision-resistant when f_1 and f_2 are (public) random *permutations*, as any two pairs (x, y), $(x', f_2^{-1}(f_1(x) \oplus f_2(y) \oplus f_1(x')))$ give a collision. Indeed, as we explain next, our mode has to be slightly modified to handle the case of random permutations.

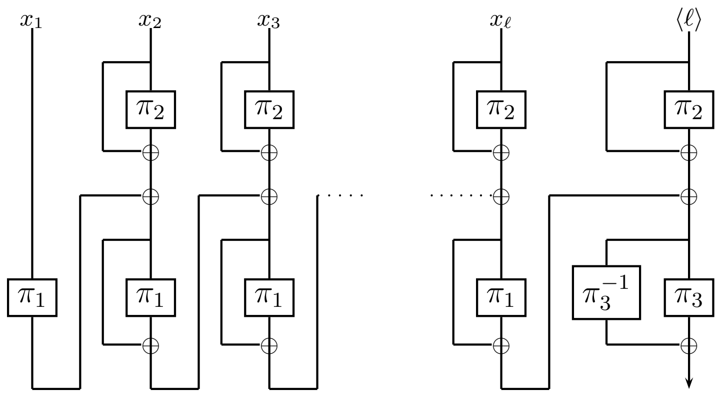

Fig. 2. The "enhanced" three-key enciphered CBC construction $H^*[\pi_1, \pi_2, \pi_3]$ which is a domain extender for random oracles, even if instantiated with random *permutations*.

4 A Block Cipher based Mode of Operation

So far we described the enciphered CBC mode for three length-reserving functions. But, as already mentioned at the end of Section 3.4 and in Footnote 4, we need to modify our basic mode in order for it to work with permutations in "unkeyed" settings, such as indifferentiability from RO and collision-resistance. In the "keyed" settings, i.e. for MACs and PRFs, replacing the functions with permutations does not make a qualitative difference (up to a birthday bound), since a PRP is also a PRF. Thus, the enciphered CBC construction works for domain extension of MACs and PRFs even if one uses a block cipher to implement the these primitives. However, even in these cases the construction may have slightly different (up to a birthday bound) exact security. For instance, as discussed for the case of PRFs in Section 3.3, the enciphered CBC construction has actually *better* exact security if permutations are used instead of functions.

"ENHANCED" ENCIPHERED CBC. We now described the (enhanced) enciphered CBC mode of operation based on three permutations π_1, π_2 and π_3. While this more complicated mode is only needed for the "unkeyed" settings (RO and CRHF), we will see that it still works for the "keyed" settings (PRF and MAC), although under slightly stronger assumptions than before. The mode is depicted in Figure 2 and is denoted $H^*[\pi_1, \pi_2, \pi_3]$. We observe that this enhanced mode is *precisely* the basic enciphered CBC construction $H[f_1, f_2, f_3]$ with length-preserving functions f_1, f_2 and f_3 defined as follows: $f_i(x) = \pi_i(x) \oplus x$ for $i = 1, 2$, and $f_3(x) = \pi_3(x) \oplus \pi_3^{-1}(x)$. The reason for this choice will become clear in the sequel, when we discuss why this "enhanced" mode works for building VIL-RO and VIL-CRHF.

4.1 Collision Resistance from Random Permutations

Using Lemma 2, in order to argue the collision-resistance of the enhanced mode, it suffices to argue the collision resistance of the XOR compression function $f_(x) \oplus f_2(y) =$

$\pi_1(x) \oplus x \oplus \pi_2(y) \oplus y$, and the function $f_3(x) = \pi_3(x) \oplus \pi_3^{-1}(x)$, even if the attacker can invert π_1, π_2 and π_3. In the standard model, we will have to simply make these (unusual but not unreasonable) assumptions for whatever public permutations we end up using. However, we must first justify that these assumption at least hold in the random permutation model. We start with the XOR compression function.

Lemma 3. For two independent permutations π_1, π_2, the XOR compression function $g[f_1, f_2]$ (with f_1 and f_2 as defined above) is (t, ϵ)-collision-resistant in the random permutation model for π_1 and π_2. Here $\epsilon = q^4/2^n$ if the attacker makes at most $q \leq \min(t, 2^{n-1})$ random permutation queries.

Proof: Let A be any collision-finding attacker who outputs a collision $(x_1 \parallel x_2), (x_1' \parallel x_2')$. When the attacker makes its forward query x to π_i (here $i = 1, 2$) or a backward query y to π_i^{-1}, we will record a tuple $(i, x, pi_i(x))$ or $(i, pi_i^{-1}(y), y)$ to a special table T. Wlog, we assume that A does not make redundant queries and that, at the the end of the game, T contains all the "collision-relevant" values $(1, x_1, y_1 = \pi_1(x_1)$, $(1, x_1', y_1' = \pi_1(x_1')$, $(2, x_2, y_2 = \pi_2(x_2)$, $(2, x_2', y_2' = \pi_1(x_2')$. This means that instead of having A output a collision, we can declare A victorious if T contains 4 (not necessarily distinct) tuples, as above, such that $x_1 \oplus y_1 \oplus x_2 \oplus y_2 = x_1' \oplus y_1' \oplus x_2' \oplus y_2'$. To complete the proof, we will argue, by induction on $0 \leq j \leq q$, that after A makes his first j queries, the probability that T will contain the required 4-tuple is at most $j^4/2^n$.

Consider query number $j + 1$. Wlog, assume it is to π_1 or π_1^{-1}. Then, either T already contained the colliding 4-tuple before this query was made (which, by induction, happens with probability at most $j^4/2^n$), or the answer to the current query $j + 1$, together with 3 prior queries, resulted in the colliding equation. Let us fix any one of these at most j^3 choices of 3 prior queries. Once this choice is fixed, it defines a unique answer to query $j + 1$ which will result in collision. Indeed, if the query $j + 1$ is to $\pi_1(x_1)$, and the 3 prior table values are $(1, x_1', y_1')$, $(2, x_2, y_2)$, $(2, x_2', y_2')$, then the only answer y_1 which will result in collision is equal to $y_1 = x_1 \oplus x_1' \oplus y_1' \oplus x_2 \oplus y_2 \oplus x_2' \oplus y_2'$. Similarly, if the query was to $\pi_1^{-1}(y_1)$, then the only answer x_1 resulting in a collision is $x_1 = y_1 \oplus x_1' \oplus y_1' \oplus x_2 \oplus y_2 \oplus x_2' \oplus y_2'$. However, since the total number of queries $j \leq 2^{n-1}$, for each fresh query there are at least $2^n - j \geq 2^{n-1}$ equally likely answers. Thus, the chance that a random such answer will "connect" with a given subset of 3 prior queries is at most $1/2^{n-1}$.

Overall, the get that the probability that there will be a collision in T after $j + 1$ queries is at most $j^4/2^n + j^3/2^{n-1} < (j + 1)^4/2^n$, completing the proof. $\square$

Next, we need to prove the collision resistance of the construction $f_3(x) = \pi_3(x) \oplus \pi_3^{-1}(x)$ in the random permutation model. However, this will trivially follow from a much stronger result we prove in the upcoming Lemma 4, which will be needed to prove the indifferentiability of our mode from a VIL-RO.

4.2 Building VIL-RO from Random Permutations

In this section we argue that the enhanced enciphered CBC mode gives a VIL-RO in the *random permutation* model for π_1, π_2, π_3. The actual proof (and the exact security) of this result is quite similar to the proof of Theorem 2. Therefore, instead of repeating the (long) proof of this result, we will only (semi-informally) highlight the key new ingredients of the proof which we must address in the random permutation model. Concentrating on these ingredients will also help us to "de-mystify" why we defined the functions f_1, f_2, f_3 in the way we did.

RANDOM ORACLE FROM RANDOM PERMUTATION. The most modular way to extend Theorem 2 to the random permutation model would be to show how to implement (in the indifferentiability framework) a length-preserving RO from an RP, and then use the general composition theorem in the indifferentiability framework (see [10]). And, indeed, it turns out that this is precisely what we did for the function f_3 (but *not* f_1 and f_2; stay tuned) by defining it as $\pi_3 \oplus \pi_3^{-1}$. Intuitively, f_3 must really look like a full-fledged FIL-RO in the proof of Theorem 2. The security of this construction for f_3 is of independent interest, since it builds a FIL-RO from an RP, and follows from the following Lemma (which also implies that f_3 is collision-resistant in the random permutation model):

Lemma 4. Let $\pi : \{0,1\}^n \to \{0,1\}^n$ be a permutation. Then the construction $f[\pi] \stackrel{\text{def}}{=} \pi \oplus \pi^{-1}$ is $(t_D, t_S, q, \mu, \mathcal{O}(q^2/2^n))$-indifferentiable from a length-preserving FIL-RO on n bits in the *random permutation model* for π (here t_D is arbitrary and $t_S = \mathcal{O}(qn)$).

Proof: We will show that the construction $f[\pi]$ is indifferentiable from a FIL-RO $F : \{0,1\}^n \to \{0,1\}^n$ in the random permutation model for $\pi : \{0,1\}^n \to \{0,1\}^n$. The proof consists of two parts: a description of the RP simulator S and the proof of indifferentiability.

The Simulator. The simulator S responds to queries of the form (i, x), for $i = -1, +1$ and $x \in \{0,1\}^n$. The distinguisher interprets the response of the simulator to a query $(+1, x)$ (resp. $(-1, x)$) as the (resp. inverse) permutation output $\pi(x)$ (resp. $\pi^{-1}(x)$). The simulator maintains a table $\mathcal{T}$ of permutation input-output pairs (x, y) such that, either it responded with y to a query $(+1, x)$ or with x to a query $(-1, y)$. On a query $(+1, x)$ (resp. $(-1, y)$), S first searches its table $\mathcal{T}$ for a pair (x, y') (resp. (x', y)) and if it finds such a pair then it responds with y' (resp. y).

On a new query $(+1, x)$, the simulator searches its table for a pair of the form (x', x) (i.e. x was an earlier RP output). If it finds such a pair, then it queries the FIL-RO F to find the output $F(x)$. It then responds with the output $y = x' \oplus F(x)$, and records the pair (x, y) in its table $\mathcal{T}$.

On a new query $(-1, y)$, the simulator searches its table for a pair of the form (y, y') (i.e. y was an earlier RP input). If it finds such a pair, then it queries the FIL-RO F to find the output $F(y)$. It then responds with the $x = y' \oplus F(x)$ to the query, and records the pair (x, y) in its table $\mathcal{T}$.

The proof of indifferentiability is postponed to the full version of this paper [13]. □

Of course, we could have the above Lemma to define f_1 and f_2 as well, but this would double the efficiency rate of our enhanced mode from 2 to 4. Instead, we observe that in the proof of Theorem 2, we "only" need the functions f_1 and f_2 to be such that the XOR compression function $g[f_1, f_2]$ is what we call *extractable*.[8]

[8] Technically, we need the whole XOR hash function $G[f_1, f_2]$ to be extractable, but it is easy to see that this is implied by the extractability of the compression function $g[f_1, f_2]$. In this case, if the XOR Hash function is extractable and the attacker makes an oracle call $f_3(y)$, the

EXTRACTABILITY. Informally, a hash function g^f built from some oracle f is ϵ-extractable (where ϵ could depend on some other parameters), if there exists and extractor Ext such that no attacker A can "fool" Ext with probability more than ϵ in the following game. A is given oracle access to f and outputs a value y. Ext takes y and the oracle queries that A made to f so far, and attempts to outputs a preimage x of y under g^f. Then A is allowed to run some more (making more calls to f) and outputs its own preimage x' of y. Then A "fools" Ext if $g^f(x') = y$ but $x \neq x'$.

Coming back to our situation, where $f = (f_1, f_2)$ and $g^f = g[f_1, f_2](x_1 \parallel x_2) = f_1(x_1) \oplus f_2(x_2)$, we only need to argue the extractability of this construction in the random permutation model, when we define $f_i(x) = \pi_i(x) \oplus x$. The extractor for this construction is defined naturally: given y, search the list of A's queries for a pair of inputs/outputs (x_1, y_1), (x_2, y_2) to π_1 and π_2, respectively, such that $y = x_1 \oplus y_1 \oplus x_2 \oplus y_2$. If exactly one such pair is found, output $x = x_1 \parallel x_2$, else fail. The security of this extractor is given below.

Lemma 5. For two independent permutations π_1, π_2, the XOR compression function $g[f_1, f_2]$ (with f_1 and f_2 as defined above) is extractable in the random permutation model for π_1 and π_2. In particular, if the attacker makes at most q permutation queries, it can fool the above extractor with probability at most $\mathcal{O}(q^4/2^n)$.

We remark that extractability can be viewed as a slight strengthening of collision-resistance: indeed, finding a collision allows one to trivially fool any extractor with probability at least $1/2$. Not surprisingly, the proof of this Lemma is only marginally harder than the proof of Lemma 3. Omitting details, we use the proof of Lemma 3 to argue that the extractor will never find more than one preimage of y through A's oracle queries. And if at most one such preimage is found, a similar argument can show that the chance of the attacker to find a different preimage x' of y is at most $q^2/2^n$.

This completes our high-level argument why the enhanced enciphered CBC mode yields a VIL-RO (and also explains our definition of f_1, f_2, f_3 in terms of π_1, π_2, π_3).

4.3 Revisiting Security for PRFs and MACs

Although the basic enciphered CBC mode already works for the case of PRFs and MACs, even when permutations are used, we argue that the enhanced mode continues to work for these settings as well. First, note that if π is a PRF (resp. MAC), then the construction $[\pi(x) \oplus x]$ is also a PRF (resp. MAC) with the same exact security. Thus, we do not need to make any stronger assumptions on π_1 and π_2 than what we made on f_1 and f_2. However, in order to prove that $f_3 = \pi_3 \oplus \pi_3^{-1}$ is a PRF (resp. MAC), we will need to make slightly stronger assumption on π_3 than being the "usual" PRF (resp. MAC). In some sense, this is expected since an inverse query to π_3 is used in the construction itself. Luckily, the extra assumptions we need are quite standard and widely believed to hold for current block ciphers. Specifically, for the case of PRFs we require that π_3 is a *(strong) pseudorandom permutation (sPRP)*: i.e., it remains a PRP even if the attacker can make both the forward and the inverse queries. Similarly, for the case of MACs, we need to assume that π_3 is a *(strong) unpredictable permutation*

Simulator can extract the preimage $x = (x_1 \ldots x_\ell)$ of y and "define" $f_3(y) = F(y)$, where F is the VIL-RO.

(sUP): i.e., a permutation for which no attacker can produce a (non-trivial) forgery even if given oracle access to both the forward and the inverse queries. The proof of this simple lemma will be given in the full version.

Lemma 6. Let $\Pi = \{\pi_k\}_k$ be a family of permutations, and define the family $F = \{f_k\}_k$ of length-preserving functions by $f_k(x) = \pi_k(x) \oplus \pi_k^{-1}(x)$. Then F is a

- $(t, q, \mu, \epsilon + \mathcal{O}(q^2/2^n))$-secure PRF if Π is a $(t + \mathcal{O}(qn), 2q, 2\mu, \epsilon)$-secure sPRP.
- $(t, q, \mu, \mathcal{O}(\epsilon \cdot q^2))$-secure MAC if Π is a $(t + \mathcal{O}(qn), 2q, 2\mu, \epsilon)$-secure sUP.

We remark that for the case of MACs, the exact security of $\epsilon \cdot q^2$ might sound alarming, especially when combining this with the statement of Theorem 1, where there is an additional loss of the q^4 factor. However, a closer look at the proof of Theorem 1 reveals that the exact security of the enciphered CBC is actually at most $\epsilon_3 + (\epsilon_1 + \epsilon_2) \cdot q^4$, where ϵ_i is the security of f_i. Coupled with the above Lemma, we get security $\epsilon \cdot q^2 + (\epsilon_1 + \epsilon_2) \cdot q^4$ (where ϵ is the security of sUP π_3, and ϵ_1, ϵ_2 are the securities of MACs π_1 and π_2).

5 Two-key enciphered CBC construction

In this section we show that it is possible to instantiate the (basic) enciphered CBC mode using only two independent length-preserving functions.

A first natural idea is to define the function f_2 in the three-key version using the function f_1. For example, we can make $f_2 = f_2$. However, in this case it is easy to see that the resulting mode is insecure for all the security notions considered in this paper. This is because the resulting XOR compression function $g[f_1, f_1]$ becomes a constant function 0^n on any "symmetric" input $(x \parallel x)$. Luckily, we show that this problem can be resolved by instantiating f_2 with a different multiple of f_1!

THE CONSTRUCTION. Consider a function $f : \{0,1\}^n \to \{0,1\}^n$. We can view the inputs/outputs of f as elements of the field $\mathbb{GF}(2^n)$, and the bit-by-bit XOR operation becomes addition over the field $\mathbb{GF}(2^n)$. Let α be any element of this field other than 0 or 1. Then we define the functions f_1 and f_2 in the enciphered CBC mode of operation $H[f_1, f_2, f_3]$ as follows: $f_2(\cdot) \stackrel{\text{def}}{=} f(\cdot)$ and $f_1(\cdot) \stackrel{\text{def}}{=} \alpha \cdot f(\cdot)$. We still use a different FIL function f' as the third function f_3 in the construction $H[f_1, f_2, f_3]$.

This defines the new XOR compression function $g_\alpha[f]$ as $g_\alpha[f](x_1 \parallel x_2) \stackrel{\text{def}}{=} f(x_1) \oplus (\alpha \cdot f(x_2))$. Intuitively, the key point we will repeatedly use in our analyses is that the function $g_\alpha[f]$ is still WCR (or even extractable in the RO model) when $\alpha \notin \{0,1\}$. We also denote the corresponding XOR hash function as $G_\alpha[f]$, and our new mode of operation using two functions f' and f' as $H_\alpha[f, f']$, where:

$$H_\alpha[f, f'](x_1 \parallel \ldots \parallel x_\ell) \stackrel{\text{def}}{=} f'\left(G_\alpha[f](x_1 \parallel \ldots \parallel x_\ell \parallel \langle \ell \rangle)\right)$$

The construction is illustrated in Figure 3. We will now analyze its security for various security notions.

5.1 Two-key enciphered CBC is MAC preserving

In the full version of the paper we prove the following lemma.

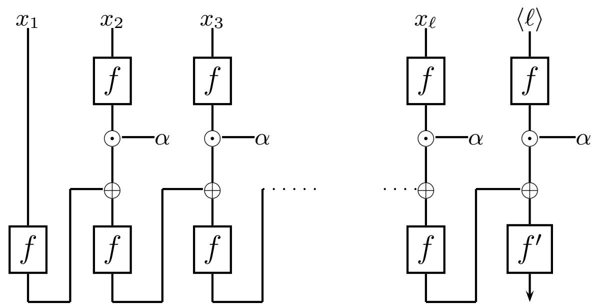

Fig. 3. The two-key enciphered CBC construction $H_\alpha[f, f']$.

Lemma 7. If the function family f is $(t, 2q, 2qn, \epsilon)$-secure MAC family, then $g_\alpha[f]$ is a $(t', q, 2qn, \epsilon \cdot 32 \cdot q^4)$-secure WCR family, where $t' = t - O(qn)$.

As explained in Section 3.1, we can now use Lemma 7 along with Lemmas 4.2 and 4.3 from [1] to get the following Theorem.

Theorem 4. *Let $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a $(t, 2\mu, 2\mu n, \varepsilon)$-secure length-preserving FIL-MAC. Then $H_\alpha[f_k, f_{k'}](.)$ (where k, k' is the secret key) is a $(t', q, \mu n, \varepsilon \cdot 33 \cdot \mu^4)$-secure variable input-length MAC, where $t' = t - O(\mu n)$ and q is arbitrary.*

5.2 VIL-RO using the two-key construction

We now show that given two independent FIL-ROs $f, f' : \{0,1\}^n \to \{0,1\}^n$, the two-key enciphered CBC construction $H_\alpha[f, f']$ is indifferentiable from a VIL-RO F. The proof of indifferentiability for this construction is similar to the corresponding proof for the three FIL-RO enciphered CBC construction. The only difference is in the way the simulator searches for a variable length input where it might need to be consistent with the VIL-RO, when responding to a f' query. We give a proof of this lemma in the full version of this paper [13].

Theorem 5. *Consider two length-preserving functions $f, f' : \{0,1\}^n \to \{0,1\}^n$. Then the new enciphered CBC construction $RO2[f, f']$ is $(t_D, t_S, q, \mu, \epsilon)$-indifferentiable from a random oracle in the FIL-RO model for f and f'. Here $t_S = \mathcal{O}(q^2)$, $\epsilon = \mathcal{O}((q + \mu)^4/2^n)$ and the result holds for any t_D.*

5.3 VIL-PRF using the two-key construction

Recall that to prove that the three-key enciphered CBC $H[f_1, f_2, f_3]$ is a good domain extender of PRFs, we reduced its security to the security of encrypted CBC, by simply simulating the invocations of f_2 (which are present in the enciphered, but not in the encrypted CBC mode). This does not work for $H_\alpha[f, f']$, as we can't simulate f because we do not know its key (in the three key case, f_2 and f_1 used independent keys, so this was possible). So one has to do a direct proof. In the full version of this paper we prove the following Theorem (we give a high level sketch of the proof in Section 6.3).

Theorem 6. *Let $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a $(t, 2\mu, 2\mu n, \epsilon)$-secure FIL-PRF family. Then $H_\alpha[f_k, f_{k'}](.)$ (where k, k' is the secret key) is a $(t', q, \mu n, 4\mu^2/2^n + 2\epsilon)$-secure VIL-PRF family where $t' = t - O(\mu n)$.*

5.4 CRHF using the two-key construction

In order to prove the collision-resistance of the two-function construction $H_\alpha[f, f']$, we essentially need to show that the XOR compression function $g_\alpha[f]$ is collision-resistant, since it is trivial to find a length-preserving collision-resistant function f' and we use MD strengthening in this construction (similar to Lemma 2). If we make a suitably strong assumption (for instance, f is a FIL-RO), then we can show that $g_\alpha[f]$ is a FIL-RO. We give a proof of this lemma in the full version of this paper.

Lemma 8. Let $f : \{0,1\}^n \to \{0,1\}^n$ be a length preserving function. The XOR compression function $g_\alpha[f]$ is (t, ϵ)-secure collision resistant function in the FIL-RO model for f. Here $\epsilon = \mathcal{O}(q^4/2^n)$, where q is the number of FIL-RO queries made by the attacker to f.

6 Single-key enciphered CBC Construction

Finally, we show how to further optimize our mode to use only a single length-preserving function f. The first natural idea is to start with the two-key mode from the previous section, and then simply make the second function $f' = f$. It is easy to see that this does not affect the collision-resistance much (since the "outer function" f' did not do anything there anyway). Unfortunately, this change makes our mode insecure. In essence, the reason is due to the fact that our (suffix-free) encoding is not prefix-free, and so called "extension attacks" become possible. (This is quite analogous to the usual CBC-MAC [3] and cascade constructions [5] which are only secure for prefix-free inputs.)

CONSTRUCTION FOR PRFS AND MACS. Luckily, it turns out that if instead of appending the input length, we prepend it (to get a *prefix-free encoding*) then the resulting construction can be proven secure (with $f' = f$) for the "keyed" setting of MACs and PRFs. The resulting construction, depicted in Figure 4 and denoted $H_\alpha[f]$, is formally defined below:

$$H_\alpha[f](x_1 \parallel \ldots \parallel x_\ell) \stackrel{\text{def}}{=} f\left(G_\alpha[f](\langle \ell \rangle \parallel x_1 \parallel \ldots \parallel x_\ell\right)$$

CONSTRUCTION FOR VIL-RO. Unfortunately, the above construction is still not enough for the question of building a VIL-RO from a single FIL-RO. To handle this case as well, we need to modify the two-key construction as follows:

1. Instead of setting $f' = f$, we use the Davies-Mayers-type construction $f'(x) = f(x) \oplus x$.
2. We still keep the suffix-free encoding (by appending the number of blocks to the input), but now also ensure the prefix-free encoding by prepending the number of blocks to the input.

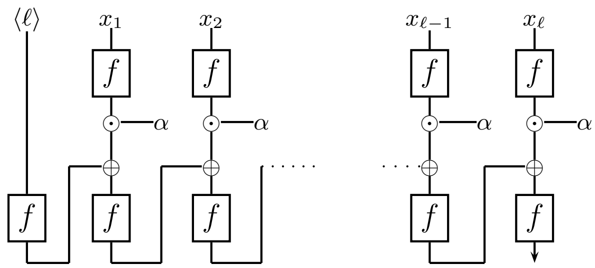

Fig. 4. The single-key enciphered CBC construction $H_\alpha[f]$ for constructing MAC and PRF.

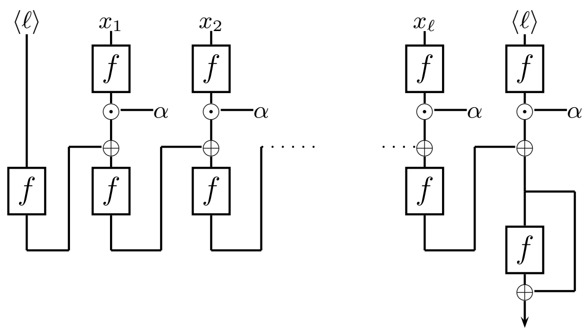

Fig. 5. The "enhanced" single-key enciphered CBC construction $H_\alpha[f]'$ for constructing RO.

We call this final construction $H_\alpha[f]'$ (see Figure 5), and formally define it on input $X = x_1 \parallel \ldots \parallel x_\ell$ as follows:

$$H_\alpha[f]'(X) \stackrel{\text{def}}{=} f(G_\alpha[f](\langle\ell\rangle \parallel X \parallel \langle\ell\rangle)) \oplus G_\alpha[f](\langle\ell\rangle \parallel X \parallel \langle\ell\rangle)$$

We remark that although this final construction $H_\alpha[f]'$ is defined for building VIL-RO (for which the simpler construction $H_\alpha[f]$ is not enough), it is easy to extend the MAC/PRF security of $H_\alpha[f]$ to show that $H_\alpha[f]'$ also works for the case of MACs and PRFs. For the sake of elegance, though, we only analyze the simpler variant $H_\alpha[f]$ when studying the domain extension of PRFs and MACs.

6.1 Single-key VIL-MAC construction

To prove that the one-key enciphered CBC $H_\alpha[f]$ is a good domain extension for MACs, we cannot apply the methodology of An and Bellare (as explained in Section 3.1) that we used for the three and the two key construction. Recall that in this methodology, one first proves that the construction (ignoring the last invocation of f) is weakly collision resistant, and then the final application of f (with an independent key) gives us the MAC property. In $H_\alpha[f]$ there is no final invocation of f with an independent key. Instead, in the full version of the paper, we give a direct reduction to prove the following Theorem.

Theorem 7. *If the function family f is a $(t, 3\mu, 3\mu n, \epsilon)$-secure MAC family, then $H_\alpha[f_k]$, where k is the secret key, is a $(t', q, \mu n, \epsilon \cdot 49 \cdot \mu^4)$-secure MAC where $t' = t - O(\mu n)$ and q is arbitrary.*

6.2 Single-key VIL-RO construction

As discussed above, the single-function RO construction $H_\alpha[f]'$ is slightly different from the MAC and PRF case. We show that this construction is indifferentiable from a VIL-RO. The formal proof of this theorem is more involved than the two/three FIL-RO case. In particular, the proof of indifferentiability crucially uses the "extractability" of the Davies-Meyer construction in the end of the "enhanced" enciphered CBC construction. We defer the formal proof to the full version of this paper [13].

Theorem 8. *Consider a length-preserving function $f : \{0,1\}^n \to \{0,1\}^n$. Then the single-function RO construction $H_\alpha[f]'$ is $(t_D, t_S, q, , \mu, \epsilon)$-indifferentiable from a random oracle in the FIL-RO model for f. Here $t_S = \mathcal{O}(q^2)$, $\epsilon = \mathcal{O}((q + \mu)^4/2^n)$ and the result holds for any t_D.*

6.3 Single-key VIL-PRF construction

We prove that our single-key enciphered CBC construction $H_\alpha[f]$ is a secure domain extension for PRFs by adapting the proof for "plain" prefix-free CBC of Maurer (Theorem 6 in [15]). The situation here is somewhat more complicated than in the three and two key cases considered so far. There, security can be proven using the following high level idea: first one proves that the construction (ignoring the final invocation of f) is (computationally) almost universal (see [2]); i.e. any two *fixed* messages are unlikely to collide. And this is enough to prove security because of a final invocation of an independent PRF. For $H_\alpha[f]$, this proof idea does not directly work, as there is no final invocation with an f using an independent key. Fortunately, one can use a powerful theorem (Theorem 2 from [15]) to still argue security in our setting as well. Details are deferred to the full version [13].

Theorem 9. *Let $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a $(t, 3\mu, 3\mu n, \epsilon)$-secure FIL-PRF family. Then $H_\alpha[f_k](.)$ (where k is the secret key) is a $(t', q, \mu n, 9\mu^2/2^n + 2\epsilon)$-secure VIL-PRF family where $t' = t - O(\mu n)$.*

Acknowledgments: We would like the thank Dan Boneh, Marc Fischlin and Phillip Rogaway for several very useful conversations in the early stages of this work.

References

1. J. H. An, M. Bellare, *Constructing VIL-MACs from FIL-MACs: Message Authentication under Weakened Assumptions*, CRYPTO 1999, pages 252-269.
2. M. Bellare, *New Proofs for NMAC and HMAC: Security without Collision-Resistance*, Advances in Cryptology - Crypto 2006 Proceedings, Springer-Verlag, 2006.
3. M. Bellare, J. Kilian, and P. Rogaway. The Security of Cipher Block Chaining. In *Crypto '94*, pages 341–358, 1994. LNCS No. 839.

4. Mihir Bellare, Krzysztof Pietrzak and Phillip Rogaway. *Improved Security Analyses for CBC MACs.* In *Crypto'05*, pages 527–545, 2005.
5. M. Bellare, R. Canetti, and H. Krawczyk, *Pseudorandom Functions Re-visited: The Cascade Construction and Its Concrete Security*, In Proc. 37th FOCS, pages 514-523. IEEE, 1996.
6. Mihir Bellare, Ran Canetti, Hugo Krawczyk: Keying Hash Functions for Message Authentication. CRYPTO 1996: 1-15
7. M. Bellare and T. Ristenpart, *Multi-Property-Preserving Hash Domain Extension and the EMD Transform*, In Advances in Cryptology - Asiacrypt 2006.
8. J. Black, P. Rogaway, T. Shrimpton, *Black-Box Analysis of the Block Cipher-Based Hash-Function Constructions from PGV*, in Advances in Cryptology - CRYPTO 2002, California, USA.
9. John Black, Martin Cochran, Thomas Shrimpton, *On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions*, EUROCRYPT 2005: 526-541.
10. J.-S. Coron, Y. Dodis, C. Malinaud and P. Puniya, *Merkle-Damgård Revisited: How to Construct a Hash Function*, Advances in Cryptology, Crypto 2005 Proceedings: 430-448, Springer-Verlag, 2006.
11. I. Damgård, *A Design Principle for Hash Functions*, In Crypto '89, pages 416-427, 1989. LNCS No. 435.
12. Y. Dodis, R. Gennaro, J. Håstad, H. Krawczyk, and T. Rabin, *Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes*, Advances in Cryptology - CRYPTO, August 2004.
13. Yevgeniy Dodis, Krzysztof Pietrzak and Prashant Puniya. *A New Mode of Operation for Block Ciphers and Length-Preserving MACs.* Full version of this paper, 2008.
14. Yevgeniy Dodis, Prashant Puniya, *Feistel Networks Made Public, and Applications*, EUROCRYPT 2007: 534-554.
15. Ueli Maurer. *Indistinguishability of Random Systems*, In *Eurocrypt'02*, pages 110–132, 2002.
16. Ueli Maurer and Johan Sjodin. *Single-key AIL-MACs from any FIL-MAC*, In *ICALP 2005*, July 2005.
17. Ueli Maurer and Stefano Tessaro, *Domain Extension of Public Random Functions: Beyond the Birthday Barrier*, Advances in Cryptology - CRYPTO 2007, Lecture Notes in Computer Science, Springer-Verlag, vol. 4622, pp. 187-204, Aug 2007.
18. R. Merkle, *One way hash functions and DES*, Advances in Cryptology, Proc. Crypto'89, LNCS 435, G. Brassard, Ed., Springer-Verlag, 1990, pp. 428-446.
19. Erez Petrank, Charles Rackoff, *CBC MAC for Real-Time Data Sources*, J. Cryptology 13(3): 315-338 (2000).
20. Krzysztof Pietrzak. *A Tight Bound for EMAC.* In *ICALP'06*, volume 2, pages 168–179, 2006.
21. B. Preneel, R. Govaerts and J. Vandewalle, *Hash Functions Based on Block Ciphers: A Synthetic Approach*, in Advances in Cryptology - CRYPTO '93,, Santa Barbara, California, USA.
22. Phillip Rogaway and John Steinberger, How to Build a Permutation-Based Hash Function, Dagstuhl workshop, September 2007.
23. Phillip Rogaway and John Steinberger, *Security/Efficiency Tradeoffs for Permutation-Based Hashing*, in Eurocrypt 2008, April 2008, Istanbul, Turkey.
24. Thomas Shrimpton and Martijn Stam, *Building a Collision-Resistant Compression Function from Non-Compressing Primitives*, Cryptology ePrint Archive: Report 2007/409.
25. Daniel R. Simon, *Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions?*, EUROCRYPT 1998: 334-345.