Attacks on single-pass confidentiality modes of operation

Olivier Markowitch , Jorge Nakahara Jr*

[1]Departement d'Informatique, Université Libre de Bruxelles, Brussels, Belgium,

{olivier.markowitch, jorge.nakahara}@ulb.ac.be

Abstract. *The main contributions of this paper are efficient distinguishing attacks against block ciphers that are conventionally modeled as pseudorandom permutations (PRP). Formally, block ciphers operate on fixed-length blocks of n bits, for example, $n = 128$ for the Advanced Encryption Standard (AES). Our analysis takes place in the setting in which the messages are m bits long, representing the entire input plaintext, where m is variable and unrelated to n. We show distinguish-from-random attacks for any n-bit block cipher in the standard modes of operation for confidentiality: ECB, CBC, CFB, OFB, CTR and XTS. We demonstrate that in all these 1-pass modes* **any** *n-bit block cipher leaves 'footprints' that allows an adversary to efficiently (in time and memory) distinguish them from a random permutation. We claim that two passes (in opposite directions) over the m-bit message, with text-dependent feedforward (chaining) and in streaming mode are sufficient to circumvent the presented attacks.*

Keywords: left-to-right diffusion, distinguishing attacks, modes of operation, (super-)pseudorandom permutations, IND-KPA, IND-CPA.

1. Introduction

Block ciphers are length-preserving cryptographic primitives that operate on finite, fixed-length text blocks. More precisely, block ciphers are keyed permutations, denoted $E_K : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, where n is a fixed integer denoting the size of one text block, and the secret key K is chosen uniformly at random from a sufficiently large key space $\mathcal{K}$. In general, n is a small integer value such as $n = 32$ for KATAN32, $n = 64$ for DES, $n = 96$ for BKSQ [Daemen and Rijmen 2000] and $n = 128$ for the AES [FIPS197 2001]. Larger values such as $n = 4096$ were adopted by the Mercy cipher [Crowley 2000]. The value of n is arbitrary and set up by convenience according to design considerations and for specific applications; n does not need to be an even integer nor a power of two. For instance, in the CTC cipher, $n = 255$ bits [Courtois 2006].

We assume the size of the key K to be large enough, say $|K| \geq 128$ bits, and subkeys to be generated efficiently and securely. Our analysis is independent of the key size or its value. Also, we do not exploit the existence (or not) of equivalent keys, weak keys [Menezes et al. 1997], complementation property (DES) or other weaknesses in the key schedule algorithm. Moreover, our attacks are in the single-key model (no related keys).

Traditionally, secure n-bit block ciphers are modeled as pseudorandom permutations (PRP) [Luby and Rackoff 1988]. It means that computationally bounded adversaries A, allowed a polynomial number q of queries, may distinguish a given block cipher E from a random permutation π, chosen uniformly at random from the set RP^n of $2^n!$ permutations, with

*Research funded by INNOVIRIS, the Brussels Institute for Research and Innovation, under the ICT Impulse program CRYPTASC.

negligible advantage

$$\text{Adv}_A(q) = |\text{Pr}(k \xleftarrow{\$} \mathcal{K} : A^{E_K} = 1) - \text{Pr}(\pi \xleftarrow{\$} \text{RP}^n : A^{\pi} = 1)|,$$

where $y \xleftarrow{\$} \mathcal{Y}$ means y is selected uniformly at random from the set $\mathcal{Y}$, and A^X returns '1' if A believes it is dealing with oracle X; otherwise, A returns '0'. Therefore, '1' means 'success' while '0' means 'failure'; 'negligible' means that the advantage grows slower than the inverse of any polynomial (in n). If the advantage is negligible even if the adversary is allowed decryption queries ($D_K = E_K^{-1}$) then, the block cipher is called a strong pseudorandom permutation (SPRP).

For negligible advantage, if the (encryption) queries are only **known** by the adversary, then the block cipher is deemed indistinguishable under known-plaintext attacks (IND-KPA); if the (encryption) queries are **chosen** by the adversary, then the block cipher is deemed indistinguishable under a chosen-plaintext attack (IND-CPA); if the decryption queries are **chosen non-adaptively** by the adversary, then the block cipher is deemed indistinguishable under a (non-adaptive) chosen-ciphertext attack (IND-CCA1); if the decryption queries are adaptively chosen by the adversary, then the block cipher is deemed indistinguishable under an adaptively chosen-ciphertext attack (IND-CCA2). If the advantage is non negligible for a single adversary, then if the queries are known to the adversary, the block cipher is not IND-KPA. Analogously, for chosen queries the block cipher is not IND-CPA, and so on.

In practice, real messages are m bits long, with m variable and unrelated to n. A naive solution to provide confidentiality in all cases would be to have block ciphers defined for every possible value of m, but this is not realistic. Rather, modes of operation [Dworkin 2001, IEEE 2008] are defined to extend the domain of application of E_K from $\mathbb{Z}_2^n$ (one text block) to $\mathbb{Z}_2^m$ (the full message), where m may be arbitrarily large but is always finite. Informally, a secure mode of operation should not disclose the fact that it is using an n-bit block cipher E_K as a building block. In summary, a secure mode should turn an n-bit (S)PRP into an m-bit (S)PRP. Consequently, issues such as padding, ciphertext expansion, blockwise or bitwise diffusion, unidirectional diffusion should be avoided, that is, weaknesses in the underlying n-bit block cipher should not propagate to the larger m-bit block.

Standard confidentiality modes of operation nowadays include: Electronic CodeBook (ECB), Cipher Block Chaining (CBC), Output FeedBack (OFB), Cipher FeedBack (CFB), Counter (CTR) and XEX Tweakable block cipher with ciphertext Stealing (XTS) [Dworkin 2001, Dworkin 2010a, IEEE 2008, Rogaway 2004]. The XTS mode in [IEEE 2008] was explicitly instantiated with the AES as the underlying block cipher. Moreover, the maximum size m of a message allowed to be encrypted via XTS-AES was upperbounded to 2^{20} 128-bit blocks. The fact that these modes of operation are linked to the AES [Dworkin 2001, Dworkin 2010a] makes this analysis quite relevant nowadays[1].

We consider random permutations operating directly on m-bit strings, and not n-bitwise like E_K, whether n is even, odd, a power of two, a divisor of m or otherwise. Moreover, random permutations are not structured transformations that require modes of operation or Feistel or SPN structures like E_K. A random permutation, in our context, is a bijective mapping chosen at random from the set of all $2^m!$ permutations of the space $\{0,1\}^m$ of m-bit strings. We

[1] FIPS standards, such as NIST SP800-38E and SP800-38A, do not include any analysis at all of any mode of operation, not even about any limitations of these modes.

denote a random permutation as $\pi \xleftarrow{\$} \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2^m$, a mapping selected at random from the set $\text{RP}^m = \{\pi_i^m : 1 \leq i \leq 2^m!\}$.

In this paper, instead of forcing random permutations to operate on n-bit strings, abiding to a block cipher domain size, we look at how block ciphers fare when forced to operate on m-bit strings for arbitrary, variable m, which is unrelated to n. In other words, instead of 'downsizing' the random permutation to always operate on fixed n-bit blocks, we work the other way around: we operate on m-bit blocks from the start because m represents the real size of an entire input message. Consequently, the queries made by an adversary are m bits long, which may be smaller, equal or larger than n bits. Before each attack starts, we set a value for m and do not change it until the attack ends.

In a block cipher setting, both an n-bit block and a full m-bit message are usually called plaintext. To make the distinction clear for our attacks, n is bound to a block cipher domain space, like $n = 64$ bits for the DES, while m is bound to a full input text message, for instance, the Project Gutenberg (ASCII) copy of the King James Bible (Old and New testaments) is 4.13 Mbytes or $m = 34,663,312$ bits long. To avoid extreme cases such as $m = O(2^n)$, we restrict our analysis to m being a polynomial in n: $m = O(n^t)$ for t a fixed constant unrelated to n. Otherwise, the adversary could cheat by using a single 2^n n-bit long message that contains all n-bit values. Further, depending on the mode of operation used, this single, long message could provide the entire codebook, which allows one to encrypt and decrypt any n-bit block without knowing the key.

This paper is organized as follows: Sect. 2 lists our contributions; Sect. 3 briefly describes the confidentiality modes under analysis; Sect. 4 describes distinguishing attacks in a PRP setting that apply to any block cipher; Sect. 5 discusses 2-pass modes and how they counter the attacks described in the previous section. Sect. 6 lists our conclusions.

2. Contributions

Our contributions address real **limitations/shortcomings of standard single-pass confidentiality modes of operation**. We explain systematically and constructively, for all these modes, how to perform efficient distinguishing attacks in a PRP setting. We describe attacks that

(i) work in a black-box setting, which in our case means the attacks work for any block cipher and any key schedule algorithm,
(ii) are very efficient concerning time, data and memory complexities, and thus violate any reasonable security thresholds whether in theory or in practice,
(iii) have very high success rate,
(iv) do not depend on (and cannot be countered by changing) the key size, key value, number of rounds, IV or nonces.

3. Brief Description of Confidentiality Modes

We briefly summarize the modes of operation under analysis in this paper. Let $P = (P_1, P_2, \ldots, P_t)$ denote an m-bit plaintext message and $C = (C_1, C_2, \ldots, C_t)$ denote the corresponding ciphertext, where $m = \sum_{i=1}^{t} |P_i|$. Some modes require random n-bit initial values, denoted IV.

- ECB: in the Electronic CodeBook mode each ciphertext block C_i depends only on P_i according to the formula $C_i = E_K(P_i)$ for $i \geq 1$. Diffusion in ECB is blockwise, which

means limited error propagation and independent (parallel) encryption of blocks. There may be a need for padding, if $m \not\equiv 0 \bmod n$.

- CBC: in Cipher Block Chaining mode, each ciphertext block is chained to the previous cipher block, and thus, depends on all previous plaintext blocks, according to the formula $C_i = E_K(P_i \oplus C_{i-1})$, for $i \geq 1$, and $C_0 =$ IV. Diffusion is better compared to the ECB mode, due to text chaining in left-to-right direction. On the other hand, parallel processing is hindered because of chaining. Error propagation is limited: if C_i is damaged, only P_i and P_{i+1} are affected.
- b-bit CFB: b-bit Cipher FeedBack mode turns a block cipher into a self-synchronous stream cipher according to the formula $C_i = P_i \oplus \text{lsb}_b(E_K(C_{i-1}))$, for $i \geq 1$ and $C_0 =$ IV, where $\text{lsb}_b(x)$ denotes the least significant b bits of x. Due to the ciphertext chaining C_{i-1}, this mode is non parallelizable. On the other hand, the CFB mode only needs E_K for both the encryption and decryption operations. Moreover, there is no need for padding, since it is a streaming mode operating on variable-length blocks.
- b-bit OFB: b-bit Output FeedBack mode turns a block cipher into a synchronous stream cipher according to the formula $C_i = P_i \oplus X_i$, where $X_1 = \text{lsb}_b(E_K(\text{IV}))$ and $X_i = E_K(X_{i-1})$ for $i > 1$. In this paper, we assume $b = n$. Note that the key stream exclusive-ored to P_i is text independent, i.e. unlike the CFB mode, X_i only depends on the IV and the key K. Therefore, OFB is a parallelizable mode since X_i can be precomputed (and stored) beforehand. It also means independent blocks P_i can be (re)encrypted without affecting C_j whether $j < i$ or $j > i$. The propagation of bit-flipping errors is limited: a bit flipped in C_i only affects P_i. This error propagation is the same as in the One-Time Pad (OTP) [Menezes et al. 1997]. Note that only E_K is enough for both the encryption and decryption modes.
- CTR: in counter mode each ciphertext block is computed as $C_i = P_i \oplus E_K(X_i)$ where $X_1 =$ IV and $X_i = E_K(f(X_{i-1}))$ for $i > 1$, with f a simple counter function or a Linear Feedback Shift Register (LSFR). CTR is a stream mode, thus, there is no need for padding, and error propagation is limited (like in a OTP). Just like in OFB, only E_K is enough for both the encryption and decryption operations in CTR mode.
- XTS: in XEX Tweakable with ciphertext Stealing mode, each ciphertext block C_i is computed as $C_i = X_i \oplus E_{K_2}(P_i \oplus X_i)$, where $X_i = E_{K_1}(i) \otimes \alpha$. According to [Dworkin 2010a], $n = 128$ bits, E_K is AES, α is a primitive element of GF(2^{128}), and $\otimes$ is multiplication in GF(2^{128}). XTS is a parallelizable mode, operating blockwise like in ECB, but requiring double encryption per block. XTS uses ciphertext stealing [Menezes et al. 1997] when $m \not\equiv 0 \bmod n$.

4. Distinguishing Attacks

The weakest goal of an adversary is to be able to distinguish a ciphertext from a random string. If a cipher does not leak information on the plaintext through to the ciphertext, then adversaries cannot distinguish the given cipher from a random permutation (over the same plaintext space). In this paper, we focus exclusively on this type of distinguishing attack. A modern trend is to complement the confidentiality property with an authentication tag, such as in IACBC (Integrity-Aware CBC) and IAPM (Integrity Aware Parallelizable Mode) [Jutla 2001, Jutla 2000]. There are several authenticated-encryption (AE) modes such as CCM (CBC-MAC with Counter Mode) [Whiting et al.], EAX (uses OMAC) [Bellare et al. 2004], CWC (Carter-Wegman-Counter) [Kohno et al.] and GCM

(Galois-Counter Mode) [McGrew and Viega 2004]. They perform two (or more) passes over the input message, but one pass is for encryption while the other passes are for computing an authentication tag. Our focus is on confidentiality modes only.

Our attacks deal with the dichotomy n versus m, that is, the fact that modes of operation using block ciphers E_K are inevitably bound to operate on n-bit blocks, for fixed n, while random permutations can freely operate on m bits, without need to partition the plaintext in n-bit (or smaller) pieces. Our attacks use very few known- or chosen-plaintext (KP or CP) queries and are independent of the key size, the number of rounds, the block size n and the internal cipher components of E_K.

The classical case $n = m$ has already been treated [Bellare et al. 1997, Bellare and Rogaway 2006]. The motivation to move beyond the setting $n = m$ is that it allows us to view the interaction between different n-bit encrypted blocks. The setting $n \neq m$ is powerful since it allows us to exploit peculiar behaviors of modes of operation (padding, blockwise operation, IV, poor diffusion) that set them apart from random permutations when operating on arbitrary-size plaintext messages.

We focus our analyses on two cases:
(i) $n > m$: in this case, for ECB, XTS and CBC modes, some padding scheme is needed because E_K necessarily operates blockwise and cannot be applied to less than n bits. On the other hand, π^m operates smoothly on m-bit inputs without padding, and generates an m-bit output. For E_K, even ciphertext stealing [Dworkin 2010b] is not an option since there are no previous ciphertext block to steal bits from. Even if bits are stolen from an initial value (IV) or from the key K, the end result is ciphertext expansion: while the input block has m bits, the ciphertext output has necessarily $n > m$ bits for E_K. Moreover, the excess $n - m$ bits cannot be removed otherwise decryption will not work. Therefore, the lenght of the ciphertext alone indicates if E_K or π^m was used, and the advantage in distinguishing between the two will be 1. In the XTS and CBC modes, different messages may use different initial values (IVs), but this is not an issue in our attacks. Exceptionally, in this case, we only need a single known-plaintext query.

We assume that IV's, nonces and tweaks are agreed upon between the legitimate parties like the key K. Nonetheless, the former are public values while K is secret. We assume that the former are not accounted for in the input size n nor in m. In other words, these auxiliary values do not consume bandwidth, i.e. they are not transmitted along with the ciphertext. Otherwise, they would lead to ciphertext expansion and we could use them to discriminate between E_K and π^m (since the latter clearly does not need them).

In OFB and CTR modes, only m keystream bits are enough to encode an m-bit message. These stream modes have the same bitwise diffusion as the One-Time-Pad (OTP): if a single bit of the ciphertext flips, only the corresponding bit of the plaintext flips (after decryption). We query a single, known m-bit message P and obtain the corresponding ciphertext C. Next, we flip a single bit of C to get C' and ask for its decryption. In both OFB and CTR modes, the corresponding plaintext P' from C' will differ in a single bit compared to P and in the same position of the bit changed in C. For π^m, the entire plaintext will be garbled, and the probability that a single bit flip in C leads to a single bit flip in P is $1/2^{m-1}$ since $m - 1$ bits have to be equal to both P and P'. The advantage in this case is $1 - 2^{1-m}$. The larger m is, the larger the advantage. To achieve an even larger advantage, another bit of C could be flipped, leading

to C'', and the attack repeated. We cannot use two messages (P, P') differing in a single bit because each of P and P' would necessarily require different IVs. In CTR mode the counter is the IV.

In b-bit CFB mode, typically $b = 1$ or $b = n$ but let us assume $b = m$ is allowed. Then, the attack is similar to the one in OFB and CTR modes. Let P be a known message with $|P| = m$ bits. The m-bit ciphertext is $C = P \oplus E_K(\text{IV})$. Notice that since the message is smaller than a single block there is no chance of ciphertext chaining, since there is no initial ciphertext, just the IV. Thus, the effect is just like in OFB and CTR modes because $E_K(\text{IV})$ is text-independent.

If $b = 1$, then we encrypt P as before and get C. Next, we flip the last bit of C to get C', and ask for its decryption. The flipped bit of C' will be fedback into the state at the latest and the corresponding plaintext P' will differ from P only in the last bit under E_K. The probability for this single bit difference in π^m is $1/2^{m-1}$ i.e. $m - 1$ bits will have to be equal under π^m. Even if b were secret, there are only a finite number of possibilities for b: 1 up to $m - 1$. With at most $2(m-1)$ queries, the previous attack can be repeated for every possible value of b. The advantage is $1 - 2^{1-m}$.

(ii) $n < m$: there are two subcases to consider

- $m \equiv 0 \bmod n$: the ECB mode is straightforward to analyse. Just query repeated blocks (P, P, P) and observe if the ciphertext is a repeated sequence (C, C, C). If so, then the adversary identified a block cipher E_K, otherwise, a random permutation π^m. The advantage is $1-2^{-n}$. Observe that while E_K operates on n-bit blocks, π rather operates on the entire sequence (P, P, P) at once, and there is only a tiny chance 2^{-n} that the result would be (C, C, C).
  In CBC mode, the adversary asks two queries $(P_1,\ P_2,\ P_3)$ and $(P'_1,\ P_2,\ P_3)$ such that $P_1 \oplus \text{IV} = P'_1 \oplus \text{IV}'$, where IV and IV' are the corresponding initial values [Bellare et al. 1997]. Note that we choose P_1 and P'_1, not the IVs. Thus, $C_1 = E_K(P_1 \oplus \text{IV}) = E_K(P'_1 \oplus \text{IV}') = C'_1$. Since the remaining blocks are the same for the rest of the message, and the first ciphertext block fedback in CBC mode is the same in both messages, the remaining ciphertext blocks are also identical for E_K. For a random permutation on m bits, this collision will never happen since π^m is a permutation. The advantage is 1.
  If (ever) the IV happens to be the same, then we query two messages (P_1, P_2, P_3) and (P_1, P_2, P'_3) such that $P_3 \neq P'_3$. Notice that the (ciphertext) chaining in CBC is in the left-to-right direction. Left-to-right chaining means that P_i is processed before P_j for $i < j$. In summary, P_i blocks are encrypted for increasing values of i starting with $i = 1$. Therefore, P_j depends on P_i for all $i < j$, but not the other way around. Thus, only C_3 will differ: C_1 and C_2 will be the same for both messages since the IV is the same. For π^m, in this case, the probability is 2^{-2n} for two consecutive n-bit blocks to be equal, and the advantage is $1 - 2^{-2n}$. For E_K and the given messages, the two n-bit ciphertext blocks C_1 and C_2 will always be the same. The advantage grows for longer messages.
  In OFB, XTS and CTR modes, we make a message query P and obtain C. Further, we flip a single bit of C to get C', and ask for its decryption. For E_K, just a single bit of the resulting plaintext P' will differ from P like in a One-Time Pad (OTP). For π^m, the probability of observing a 1-bit difference in two m-bit plaintexts is $1/2^{m-1}$, and

the advantage is $1 - 2^{-8}$. Note that in this case the adversary is making an adaptively chosen-ciphertext query, and the decrypted ciphertext results in a meaningfull plaintext (except, eventually, for the garbled bit position). Again, notice that in OFB, XTS and CTR modes there is no plaintext-dependent chaining. Likewise, for b-bit CFB mode, the attack proceeds like in OFB mode since the diffusion is in the left-to-right direction only.

- $m \not\equiv 0 \bmod n$: this case is similar to the case $n > m$, and the focus is on the last message block that contains only $m \bmod n$ bits. The treatment of these trailing bits by each mode of operation allows the adversary to detect whether E_K or π^m was used. For ECB, XTS and CBC modes, ciphertext stealing could be used, and our previous argument in the case $n > m$ do not apply. For ECB and XTS modes, the adversary queries two messages (P_1, P_2, P_3) and (P_1, P_2, P_3') where $|P_3| = |P_3'| = m \bmod n$, but $P_3 \neq P_3'$. For CBC mode, the messages are (P_1, P_2, P_3) and (P_1', P_2, P_3') where $|P_3| = |P_3'| = m \bmod n$, but $P_3 \neq P_3'$. P_1 and P_1' are such that $P_1 \oplus \mathrm{IV} = P_1' \oplus \mathrm{IV}'$, so $C_1 = C_1'$.
  In ECB, XTS and CBC modes, after padding, only C_3 and C_3' will differ while $C_i = C_i'$ for $i < 3$ whatever E_K is used. If the same IVs are ever used, we can just choose different P_3 and P_3'. Thus, the adversary can distinguish between E_K and π^m with advantage $1 - 2^{m \bmod n - m}$ for m-bit messages, since only the last $m \bmod n$ bits differ in both messages.
  For OFB, CTR and CFB modes there is no padding, but the same strategy as in the OTP also apply: we exploit the bitwise diffusion.

In our attacks, we exploited the following facts that are inherent to any block cipher E_K using a confidentiality mode of operation:

- padding and ciphertext stealing: in ECB, XTS and CBC modes, the size of each text block has to be at least n bits, because E_K cannot operate on smaller blocks. To fill in the missing bits, padding is needed. It does not matter which padding scheme is used since there will be ciphertext expansion anyway, and this fact alone is enough to detect that E_K was used instead of π^m. Notice that random permutations π^m never need padding.
- left-to-right (L2R) diffusion and one pass over the message: CBC and CFB modes applied to a message $(P_1, P_2, P_3, \ldots)$ chains values in left-to-right order (and never the other way around), that is, C_i depends on C_j and indirectly on P_j for $j \leq i$, but C_i is independent of C_l and P_l for $l > i$. In a sense, the left-to-right chaining order makes both the CBC and the CFB modes a kind of T-function [Klimov and Shamir 2002]. This unidirectional diffusion is due to the design of these modes: only a single pass is allowed over the message due to efficiency and buffering considerations. We exploited precisely these weaknesses to construct our message queries and attacks. Notice that the attacks work independently of the underlying block cipher E_K or the key size. In comparison, for π^m there is full diffusion across an entire m-bit string. Moreover, the avalanche effect holds for π^m: changing a single bit in any of the m input bits implies all output bits change with 50% chance. For E_K over m-bit messages, the avalanche effect does not hold.
- plaintext-independent chaining: in ECB, XTS, OFB and CTR modes, the dependence between consecutive n-bit blocks (if ever) depends on the key, the tweak or the IV but not on the plaintext nor the ciphertext. This feature is motivated by parallel

processing capabilities of these modes to speed-up encryption. In π^m, we expect full text-dependent diffusion across the entire m-bit string.

- bitwise diffusion in OTP: in streaming modes such as OFB and CTR, the key bit-stream generated simulates a One-Time-Pad in the sense that the ciphertext is simply the message xored to a plaintext-independent key stream. This fact means that diffusion is worse than the left-to-right diffusion pointed out for the CBC and CFB modes: if only a limited set of bits change in the message, the very same isolated set of bits will change in the ciphertext (and vice-versa). This is extremely unlikely to be observed in a random permutation π^m operating on the whole m bits at once, and this phenomenon can be detected for E_K with only two queries: one encryption and one decryption.

The probability of the adversary simply guessing whether the oracle he interacts with is E_K or π^m, is 1/2. In all cases, our attacks have advantage much larger than 1/2, for appropriate and realistic values of m.

In summary, all the modes analysed previously leak information, that is, leave footprints or signatures of their presence in the ciphertext, independently of which block cipher E and key K are used. For instance, a random permutation π^m provides full diffusion across an m-bit string as a monolithic transformation. On the other hand, all modes of operation mentioned necessarily work blockwise, n bits at a time, and in the left-to-right direction, i.e diffusion is unidirectional. Thus, the avalanche effect is compromised.

To fix these problems, we claim that:

- achieving complete diffusion is necessary; it is suggested that modes of operation perform two passes over the m-bit message in both left-to-right (L2R) and right-to-left (R2L) directions. L2R is the natural order in which P_i blocks are presented in the input: P_i before P_j for $i < j$. Therefore, L2R difusion means that P_j depends on P_i for $j > i$, but not the other way around. R2L means the opposite, i.e. block P_i is processed before P_j for $i > j$. Separately, L2R and R2L provide weak diffusion, but combining L2R and R2L results in much stronger diffusion.
  A drawback with two passes over the message is buffering: the intermediate data processed in the first pass should be securely stored[2] for the second pass (in the reverse direction), before ciphertext is output. Well-known modes of operation such as PEP [Chakraborty and Sarkar 2006], CMC [Halevi and Rogaway 2003] and EME [Halevi and Rogaway 2004] already required buffering due to multiple passes over the data. The buffering issue is less critical in settings such as in disk-sector encryption [SISWG] since only 512 bytes need to be stored, which is a small amount and is known beforehand. In general, though, the total size of the input, m, is not known in advance. If the intermediate data, for example, $X_i = E_K(P_i \oplus X_{i-1})$ in a 2-pass mode is leaked, then the n-bit secret intermediate state X_i is exposed and security may be compromised [Biham 1998], for example, by a meet-in-the-middle attack. The fact that hard-disk encryption modes actually use multiple passes means that our attacks are relevant.
- modes should use chaining that is either plaintext or ciphertext dependent, such as in CBC and CFB modes. Multiple passes, in opposite directions, over the data for modes such as ECB, OFB and CTR are void, since these modes have no text-dependent chaining. For instance, 2-pass CTR mode (with or without the same IV or key) still does

[2]We assume some kind of secure storage is available. It is intended to protect the partially (1-pass) intermediate encrypted data from leaking.

not counter the attacks described previously since xoring two key streams (under different counters and keys) are equivalent to applying two OTP keystreams in succession. In other words, diffusion remains bitwise in both 1-pass and 2-pass CTR because the key streams are independent of plaintext and ciphertext. In fact, the same reasoning holds for any number of passes of CTR mode. The same rationale applies to ECB and OFB modes. A drawback of our recommendation is that (chained) modes become non-parallelizable due to text-dependent chaining. Another consequence of the two-pass procedure is that text-dependent chaining causes infinite error propagation across the entire m-bit ciphertext. This effect, though, simply means complete diffusion was achieved.

- finally, to deal with both the cases $n < m$ and $n > m$ a stream mode should be used. For $n < m$, there are padding schemes, but for $n > m$ there is no way out for modes that operate blockwise, such as in ECB and CBC.

These conditions are aimed to make the modes behave closer to a random permutation over m-bit strings.

5. Two-pass modes

Let an m-bit input message be denoted $P = (P_1, P_2, P_3, \ldots, P_t)$, where $|P_i| = n$ bits for $1 \leq i \leq t-1$, $|P_t| = n - m \bmod n$ bits and $t = \lceil m/n \rceil$. We assume randomly chosen, uniformly distributed, publicly-known n-bit initial values, IV_j for $j > 0$, if needed. Conventionally, 2-pass modes have been associated with authentication modes such as AES-CCM [Dworkin 2004], but, in the later, the second pass is aimed at computing an authentication tag on top of the confidentiality service of a mode of operation. Therefore, even in AES-CCM, diffusion is still unidirectional: both passes over the data are in left-to-right direction and thus, do not provide appropriate diffusion across an m-bit block.

There are 2-pass modes of operation that perform R2L diffusion, such as the EMD (Encrypt-Mask-Decrypt) and CMC (CBC-Mask-CBC) modes [Rogaway , Halevi and Rogaway 2003]. But, they use CBC decryption in right-to-left order, and they are aimed at encryption of fixed-size disk sectors, not arbitrary m-bit messages. It means that these modes are allowed only if $m \equiv 0 \bmod n$, since no padding scheme was defined. Moreover, the CMC and EMD modes contain an additional masking layer in between the two CBC layers. The mask, denoted M in [Halevi and Rogaway 2003], mixes intermediate blocks X_i using multiplication over GF(2^n). This mask provides diffusion across n-bit blocks and helps counter exhaustive search attacks on specific n-bit blocks. Without M, diffusion would be much weaker: suppose we have two messages $(P_1, P_2, P_3, P_4, P_5)$ and $(P_1, P_2, P_3, P_4, P'_5)$ that differ only in the last block: $P_5 \neq P'_5$. Then, during the first pass in CBC, only X_5 is different from X'_5. Since there is no mask, X_5 and X'_5 only affect C_5, C'_5 (directly) and C_4, C'_4 (due to the backwards text chaining in the CBC decryption of the second pass). The rest of the ciphertext is not affected and diffusion is limited. See Fig. 1.

An alternative scheme would be to perform two-pass CBC both in left-to-right direction, but chaining X_t as IV to X_1 in the second pass since X_t would depend on all P_i. Thus, $C_1 = E_K(X_1 \oplus X_t)$ and, for $1 < i \leq t$, we have $C_i = E_K(X_i \oplus C_{i-1})$. We call it wrapped-CBC mode. See Fig. 2(a). For simplicity, let us consider the $m \equiv 0 \bmod n$ case. There is full diffusion across an m-bit message due to the chaining of X_t; the intermediate blocks X_i mask the P_i and also protect (P_1, C_1) from exhaustive key search attacks. Nonetheless, the

decryption mode is quite weak because all chainings are only fed forward: changing C_1, for instance, will affect P_1, P_2 and P_3 only. See Fig. 2(b).

To avoid this kind of discrepancy between the encryption and decryption modes, an alternative is 2-pass IGE mode. The 1-pass IGE mode was described by C. Campbell in [Campbell 1978]. IGE means Infinite Garble Extension because a single bit flipping error causes infinite error propagation, that is, all subsequent blocks, including the one with the error, will be garbled due to chaining in encryption/decryption. Still, 2-pass IGE mode cannot deal with the case $n > m$ without causing ciphertext expansion, since encryption is blockwise like in ECB and CBC modes. Therefore, only stream modes can deal with all m values.

For completeness sake, we describe the 2-pass IGE encryption which we call wrapped-IGE mode, and is depicted in Fig. 3(a). We assume $n < m$. There are two sub-cases to consider

- $m \equiv 0 \bmod n$: in this case, all blocks are n bits wide. In the first pass over the m-bit message, the conventional IGE mode is applied and intermediate blocks X_i are generated. Diffusion only occurs in the left-to-right direction: $X_1 = E_K(P_1 \oplus \text{IV}_1)$, and for $1 < i \leq t$ we have $X_i = E_K(P_i \oplus X_{i-1}) \oplus P_{i-1}$. Note the chaining from the previous blocks P_{i-1}, and the unknown X_{i-1} masks the P_i block, hiding it. For the second pass, the last ciphertext block is initially generated as $C_t = E_K(X_t) \oplus \text{IV}_2$, and for $1 \leq i < t$ we have $C_i = E_K(X_i \oplus C_{i+1}) \oplus X_{i+1}$ in the reverse direction. Note the feedforward of C_{i+1} leading to avalanche in the in the right-to-left direction. The intermediate X_i blocks mask the outputs of E_K.
  Unlike wrapped-CBC mode, the decryption of wrapped-IGE mode has the same (complete) diffusion across the m-bit block as encryption. An inconvenience is that the ciphertext blocks must be processed in reverse order: from C_t down to C_1. See Fig. 3(b).
  Unlike CMC and EMD modes, wrapped-IGE does not use nor need multiplication over GF(2^n), since the chaining of X_i and C_i values are enough to provide the necessary diffusion.
- $m \not\equiv 0 \bmod n$: in this case, there is a final block with only $m \bmod n$ bits that will be treated with the ciphertext-stealing technique [Menezes et al. 1997] to avoid ciphertext expansion.
  In the first pass, we proceed as before in IGE mode, but taking care of the last partial block. Again, intermediate data X_i is generated: $X_1 = E_K(P_1 \oplus \text{IV}_1)$ and, for $1 < i < t$ we have $X_i = E_K(P_i \oplus X_{i-1}) \oplus P_{i-1}$. For the last partial block we use ciphertext stealing. Recall that $|P_t| = m \bmod n$ bits, so $X_t = E_K((P_t \| \text{lsb}_{n-m \bmod n}(X_{t-1}) \oplus X_{t-1}) \oplus P_{t-1}$, where $\text{lsb}_v(y)$ denotes the v least significant bits of y. To adjust the size of the output to the same size m of the input message, we rearrange the bits as follows: $X_j = \text{lsb}_{n-m \bmod n}(X_{j-1}) \| (X_j \gg n - m \bmod n)$ for $1 < j \leq t-1$ and $X_t = \text{lsb}_{m \bmod n}(X_t)$, where $x \gg y$ denotes x shifted right by y bits (the y least significant bits of x are dropped). There are two reasons for the rearrangement of bits: (i) the second pass explained in the next paragraph; (ii) the $\text{lsb}_{n-m \bmod n}(X_{t-1})$ was used in (and can be recovered from) X_t, there is no need to keep it in both X_t and X_{t-1}. Notice that $\text{lsb}_{n-m \bmod n}(X_1)$ became redundant.
  For the second pass, we move in the right-to-left direction, guaranteeing full diffusion across the entire m-bit string. Initially, $C_t = E_K(X_t \oplus \text{IV}_2)$. For $1 < i < t$, we have $C_i = E_K(X_i \oplus C_{i+1}) \oplus X_{i+1}$. Finally, for the last block, we have $C_1 =$

$E_K((X_1\|\text{msb}_{m \bmod n}(C_2)) \oplus C_1) \oplus X_1$. Lasty, we adjust the size of the ciphertext: $C_2 = \text{lsb}_{n-m \bmod n}(C_2)$. This way, there is no ciphertext expansion.

Now, for a secure streaming mode, we suggest 2-pass b-bit CFB or wrapped b-bit CFB. See Fig. 4. We assume n-bit initial values IV_i, for $i > 0$, as needed. We assume that k-bit keys K_1 and K_2 are dependent, for instance, jointly generated like $K_1 = E_K(S)$ and $K_2 = E_K(S \oplus K_1)$ from a random n-bit seed S and a random k-bit key K, with $|S| = |K| = |K_1| = |K_2|$. For instance, E_K is AES with $n = k = 128$. This requirement on K_1 and K_2 aims to counter meet-in-the-middle attacks.

The encryption proceeds as follows: (i) $n > m$: we cannot use $b = m$, even though that would be more efficient than a smaller b. Recall that the keystream generated by CFB mode is exclusive-ored to P, but if the keystream itself depends on P, then, there is a self-referential issue: $C = P \oplus \text{msb}_m(E_{K_1}(\text{msb}_{n-m}(\text{IV}_1)\|P))$ cannot be decrypted.

Even if we encrypt $P\|0^{n-m}$ i.e. P padded with $n - m$ zero bits, in the first pass, as $X = (P\|0^{n-m}) \oplus E_K(\text{IV}_1)$. Then, in a second pass, $C = \text{msb}_m(X \oplus (E_{K_2}(X)))$. But, again, due to a self-referential result, we cannot decrypt C. Moreover, E_{K_2} depends on n bits of X, and we only have m bits. Therefore, we assume $b = 1$. Let uppercase symbols, such as S denote an n-bit block while lowercase symbols such as s_i denote a single bit. In the first pass, there is an initialization step: $S = E_{K_1}(\text{IV}_1) = (s_1, \ldots, s_n)$. Next, the bits of $P = (p_1, \ldots, p_m)$ are encrypted one by one, and the result is fedback into E_{K_1}: $x_i = p_i \oplus \text{lsb}_1(S)$, $S = E_{K_1}((S \ll 1)\|x_i)$, for $1 \leq i \leq t$, where $x \ll y$ denotes x left-shifted by y bits. There is no need for padding, since encryption operates bitwise. Using $b = 1$ is inefficient, but since $m < n$, the penalty is minimal. The result is an m-bit string $X = (x_1, \ldots, x_m)$.

In the second pass, we wrap around. Initially, $Y = E_{K_2}(\text{lsb}_{n-m}(\text{IV}_2)\|X_t) = (y_1, \ldots, y_m)$. Then, $c_{m-i} = y_i \oplus \text{lsb}_1(Y)$, $Y = E_{K_2}((Y \ll 1)\|c_{m-i})$, for $1 \leq i \leq t$. Note the indexing of the ciphertext bits c_{m-i}, for $1 \leq i \leq t$, indicating 'right-to-left' direction. This case shows how a streaming mode such as CFB can deal efficiently and smoothly with variable-length inputs, while IGE and other blockwise modes could not, particularly the $n > m$ case, even with padding.

(ii) $n < m$: there are two subcases to consider:

- $m \equiv 0 \bmod n$: in this case, we use $b = n$ which means full feedback to improve performance. In the first pass over the m-bit message, $Y = E_{K_1}(\text{IV}_1)$ is initially generated. Diffusion occurs in the left-to-right direction following a blockwise encryption: $X_i = P_i \oplus Y$ and $Y = E_{K_1}(X_i)$, for $1 \leq i \leq t$. For the second pass, we wrap-around: initially, $Y = E_{K_1}(X_t)$. Then, repeatedly $C_i = X_i \oplus Y$ and we update $Y = E_{K_2}(C_i)$, for $1 \leq i \leq t$. We could alternatively have reversed direction: $C_{t-i} = X_{t-i} \oplus Y$ and $Y = E_{K_2}(C_{t-i})$ for $t > i > 0$. Both options are equivalent.
- $m \not\equiv 0 \bmod n$: we have two choices: (i) we can use $b = 1$, which is inefficient for large values of m and fixed values of n; (ii) we can use $b = n$ for $\lfloor m/n \rfloor$ blocks (most of them) and then switch to $b = 1$ for the last partial block of $m \bmod n$ bits. Option (ii) is similar to the item $n > m$ where encryption is bitwise using a single bit from E_{K_j}, $j \in \{1, 2\}$, at a time. Option (i) is a mixture of the items $m \equiv 0 \bmod n$ (for $m > n$) and $n > m$ (for the last $m \bmod n$ bits). Again, there are two passes over the m-bit message in opposite directions: left-to-right and right-to-left.

The wrapped-CFB mode counters all the attacks described against 1-pass modes, for

both $n < m$ and $n > m$. Flipping a single or even multiple bits of the ciphertext C_i (resp. plaintext P_i) will affect all plaintext bits of P_i (resp. ciphertext C_i) due to text chaining and bi-directional diffusion in the intermediate blocks X_i. The double pass in opposite directions guarantees full diffusion for both encryption and decryption, making them equally strong. Concerning meet-in-the-middle (MITM) attacks, the first n-bit block P_0 is the most interesting target: $C_0 = P_0 \oplus E_{K_1}(IV_1) \oplus X_0$ and $X_0 = C_0 \oplus E_{K_2}(C_1)$. The other pairs (P_j, C_j), $j > 0$, contain an unknown quantity X_{j-1}. In a known-plaintext setting P_0 is known. IV_1, C_0 and C_1 are also known. If K_1 and K_2 were independent k-bit keys, then a MITM attack could be applied to the (P_0, C_0) pair with time complexity around 2^k calls to E_K, instead of 2^{2k} [Menezes et al. 1997]. But, by construction, K_1 and K_2 are (nonlinearly) dependent.

What about birthday-paradox-type attacks for m-bit messages? For random, unpredictable IVs and keys, there is no collision possible in wrapped-CFB mode because this mode effectively performs an m-bit permutation. It means that two distinct m-bit plaintexts (resp. ciphertexts) always lead to different m-bit ciphertexts (resp. plaintexts). Consider now the case of variable IVs. Since the text-independent parameters consist of $(\mathrm{IV}_1, \mathrm{IV}_2, K_1, K_2)$, then if the keys are fixed and only the IVs change then, a birthday-paradox effect on $(\mathrm{IV}_1, \mathrm{IV}_2)$ would lead to a collision in the m-bit ciphertexts after $\sqrt{2^m}$ encryptions under 2^{2n} different IVs (assuming $|\mathrm{IV}_1| = |\mathrm{IV}_2| = n$). This means $m \approx 2n$, but if we assume $n = k$, then an effort of $\sqrt{2^m}$ encryptions is the same as an exhaustive key search. If $m > 2n$, then collisions are void.

There are several modes of operation that aim to achieve better diffusion than the modes in [Dworkin 2001, IEEE 2008]. Multiples encryption passes over the data cannot be avoided if full diffusion is the objective. A strategy is to achieve full diffusion by employing so called universal hash functions instead of encryption. For instance, the Hash-Encrypt-Hash (HEH) mode [Sarkar 2007] was based on the Naor-Reingold [Naor and Reingold , Naor and Reingold 1999] paradigm. HEH targeted disk-sector encryption where the input message is a single disk sector. The buffering issue is minimized, since for disk-sector encryption the storage needed is only $m = 4096$ bits or 512 bytes. The HEH mode uses the ECB mode between two layers of a universal hash function $H : \mathbb{Z}_2^t \rightarrow \mathbb{Z}_2^t$ (thus, the name HEH). Moreover, m must be a multiple of n, the input size of E_K. Diffusion across an m-bit string is provided by the hash function H, which is invertible and cheaper to compute than several E_K instances. But, if m is not a multiple of n, it is straightforward to distinguish E_K in HEH mode from π^m since HEH cannot be applied due to padding issues.

6. Conclusions

In this paper, we argued about **limitations/drawbacks in six standard confidentiality modes of operation: ECB, CBC, OFB, CFB, CTR and XTS that perform a single pass over the input message**. A pervasive problem is unidirectional diffusion (left-to-right direction) or bitwise diffusion (in stream modes: OFB, CFB and CTR). Similar conclusions hold for the inverse of these modes, whatever the underlying n-bit block cipher E_K and whatever the key K. To compound the problem, it is rare to find text-dependent chaining (only present in CBC and CFB modes). Consequently, these modes behave significantly worse than a random permutation over message spaces larger or smaller than a single n-bit block. Therefore, the 1-pass modes cannot properly model a random permutation over message spaces composed of m-bit strings.

We claim that using the message space of m bits is more relevant as a testing ground,

since in practice messages do not respect n-bit boundaries. This setting has already been discussed and is widely accepted for disk-sector encryption, but in the latter m is limited to 512 bytes, which is a multiple of n for many block ciphers such as the AES. In this paper, we allow m to be unrelated to n, not necessarily a multiple of n, although at most a polynomial in n.

The distinguish-from-random attacks described in this paper can be countered by processing the entire message in two passes in opposite directions: left-to-right and right-to-left, to provide full diffusion across the m-bit input. Moreover, this countermeasure requires text-dependent chaining. Also, to account for the case $n > m$, streaming modes are necessary, since any padding would cause ciphertext expansion. This combination of double-pass and text-dependent chaining guarantees complete diffusion just as random permutations would do and as would be expected of a block cipher aimed at mimicking the behaviour of π^m over large m-bit strings.

In our attacks, the adversary may need black-box access to both encryption and decryption oracles. The queries are small in size (a few n-bit blocks each) and we used at most two message queries. In all cases, the amount and size of queries are polynomial-sized in n. Our attacks have high advantage and are independent of n, or of the cipher structure (Feistel, SPN, IDEA-like, LFSR-like) or the key size or the number of rounds. Therefore, our attacks apply independently of the underlying block cipher. Table 1 summarizes the results in this paper. From this table, we conclude that under the given assumptions on (n, m), no block cipher in ECB mode can be either IND-KPA or IND-CPA. Analogously, no block cipher in CBC mode can be either IND-KPA or IND-CPA; no block cipher in CFB mode can be either IND-KPA or IND-CCA2; no block cipher in CTR mode can be either IND-KPA or IND-CCA2; no block cipher in OFB mode can be either IND-KPA or IND-CCA2; no block cipher in XTS mode can be either IND-KPA or IND-CPA.

References

Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. (1997). A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS'97*, pages 394–403.

Bellare, M. and Rogaway, P. (2006). The security of triple encryption and a framework for code-based game-playing proofs. In Vaudenay, S., editor, *Adv. in Cryptology, Eurocrypt*, volume 4004 of *LNCS*, pages 409–426. Springer.

Bellare, M., Rogaway, P., and Wagner, D. (2004). The eax mode of operation. In *Fast Software Encryption (FSE)*, volume 3017 of *LNCS*, pages 389–407. Springer.

Biham, E. (1998). Cryptanalysis of multiple modes of operation. *Journal of Cryptology*, 11(1):45–58.

Campbell, C. (1978). Design and specification of cryptographic capabilities. In Brandstad, D., editor, *Computer Security and the Data Encryption Standard*, Special Publications 500-27, pages 54–66. National Bureau of Standards, US Dept of Commerce.

Chakraborty, D. and Sarkar, P. (2006). A new mode of encryption providing a tweakable strong pseudorandom permutation. In *Fast Software Encryption (FSE)*, volume 4047 of *LNCS*, pages 293–309. Springer.

Courtois, N. (2006). How fast can be algebraic attacks on block ciphers? IACR ePrint archive 2006/168.

Table 1. 1-pass modes, attack complexities, advantage and weaknesses.

1-pass mode	attack complexity data/memory†	attack complexity time	advantage	issue	comments
ECB	1 KM	1	1	padding	$n > m$
ECB	1 CM	1	$1 - 2^{-n}$	blockwise diffusion	$n < m, m \equiv 0 \bmod n$
ECB	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m, m \not\equiv 0 \bmod n$
CBC	1 KM	1	1	padding	$n > m$
CBC	1 CM	1	1	collision	$n < m, m \equiv 0 \bmod n$
CBC	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m, m \not\equiv 0 \bmod n$
CFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bit diffusion	$n > m$
CFB	1 KM	1	$1 - 2^{1-m}$	L2R diffusion	$n < m, m \equiv 0 \bmod n$
CFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \not\equiv 0 \bmod n$
CTR	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n > m$
CTR	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \equiv 0 \bmod n$
CTR	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \not\equiv 0 \bmod n$
OFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n > m$
OFB	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \equiv 0 \bmod n$
OFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \not\equiv 0 \bmod n$
XTS	1 KM	1	1	padding	$n > m$
XTS	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \equiv 0 \bmod n$
XTS	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m, m \not\equiv 0 \bmod n$

KM: Known Message; CM: Chosen Message; CC: Chosen Ciphertext

†: memory complexity is the space needed to store the given data.

Crowley, P. (2000). Mercy: a fast large block cipher for disk sector encryption. In Schneier, B., editor, *Fast Software Encryption (FSE)*, volume 1978 of *LNCS*, pages 49–63. Springer.

Daemen, J. and Rijmen, V. (2000). The block cipher bksq. In Quisquater, J.-J. and Schneier, B., editors, *Third International Conference on Smart Card Research and Applications (CARDIS)*, volume 1820 of *LNCS*, pages 236–245. Springer.

Dworkin, M. (2001). Recommendation for block cipher modes of operation methods and techniques. National Institute of Standards and Technology NIST Special Publication 800-38A (2001).

Dworkin, M. (2004). Recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality. National Institute of Standards and Technology (NIST). NIST Special Publication 800-38C (2004).

Dworkin, M. (2010a). Recommendation for block cipher modes of operation: The xts-aes mode for confidentiality on storage devices. National Institute of Standards and Technology (NIST). NIST Special Publication 800-38E (2010).

Dworkin, M. (2010b). Recommendation for block cipher modes of operation: Three variants of ciphertext stealing for cbc mode. National Institute of Standards and Technology (NIST). Addendum to NIST SpecialPublication 800-38A (2010).

FIPS197 (2001). Advanced encryption standard (aes). FIPS PUB 197 Federal Information

Processing Standard Publication 197, U.S. Department of Commerce (2001).

Halevi, S. and Rogaway, P. (2003). A tweakable enciphering mode. In Boneh, D., editor, *Adv. in Cryptology, Crypto*, volume 2729 of *LNCS*, pages 482–499. Springer.

Halevi, S. and Rogaway, P. (2004). A parallelizable enciphering mode. In *CT-RSA*, volume 2964 of *LNCS*, pages 292–304. Springer.

IEEE (2008). The xts-aes tweakable block cipher - an extract from ieee std 1619-2007. The Institute of Electrical and Electronics Engineers, Inc.

Jutla, C. (2000). Parallelizable encryption mode with almost free message integrity. http://citeseer.ist.psu.edu/jula00parallelizable.html.

Jutla, C. (2001). Encryption modes with almost free message integrity. In Pfitzmann, B., editor, *Adv. in Cryptology, Eurocrypt*, volume 2045 of *LNCS*, pages 529–544. Springer.

Klimov, A. and Shamir, A. (2002). A new class of invertible mappings. In *Cryptographic Hardware and Embedded Systems (CHES)*, volume 2523 of *LNCS*, pages 470–483. Springer.

Kohno, T., Viega, J., and Whiting, D. Cwc: a high-performance conventional authenticated encryption mode. Cryptology ePrint Archive, report 2003/106 (2003).

Luby, M. and Rackoff, C. (1988). How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386.

McGrew, D. and Viega, J. (2004). The security and performance of the galois/counter mode (gcm) of operation. In Canteaut, A. and Viswanathan, K., editors, *Indocrypt*, volume 3348 of *LNCS*, pages 343–355. Springer.

Menezes, A., van Oorschot, P., and Vanstone, S. (1997). *Handbook of Applied Cryptography*. CRC Press.

Naor, M. and Reingold, O. A pseudorandom encryption mode. Manuscript available at http://www.wisdom.wiezmann.ac.il/~naor.

Naor, M. and Reingold, O. (1999). On the construction of pseudorandom permutations: Luby-rackoff revisited. *Journal of Cryptology*, 12(1):29–66.

Rogaway, P. The emd mode of operation (a tweaked, wide-blocksize strong prp). Cryptology ePrint Archive 2002/148.

Rogaway, P. (2004). Efficient instantiations of tweakable block ciphers and refinements to modes ocb and pmac. In Lee, P., editor, *Adv. in Cryptology, Asiacrypt*, volume 3329 of *LNCS*, pages 16–31. Springer.

Sarkar, P. (2007). Improving upon the tet mode of operation. In Nam, K.-H. and Rhee, G., editors, *Information Security and Cryptology (ICISC)*, volume 4817 of *LNCS*, pages 180–192. Springer.

SISWG. Ieee security in storage working group (siswg). http://www.siswg.com.

Whiting, D., Housley, R., and Ferguson, N. Submission to nist: Counter with cbc-mac (ccm) aes mode of operation. Computer Security Division, Computer Security Resource Center (NIST).

A. Modes of Operation Schematics

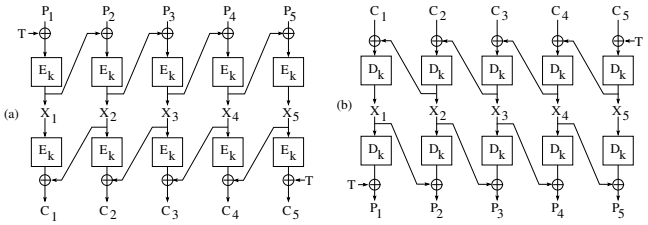

Figure 1. CMC mode without **masking layer (T is tweak): (a) encryption, (b) decryption.**

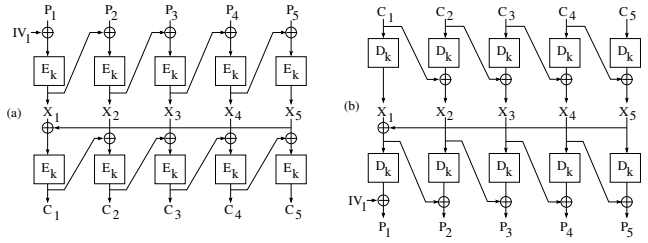

Figure 2. Wrapped CBC mode (2-pass CBC with feedback of last block): (a) encryption, (b) decryption.

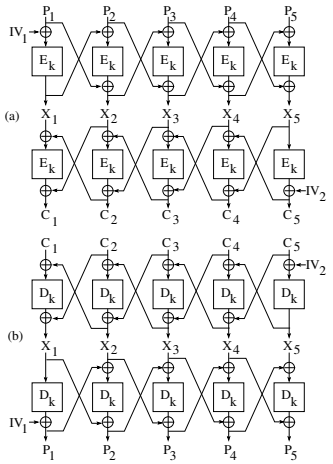

Figure 3. Wrapped IGE mode (two IGE passes): (a) encryption, (b) decryption.

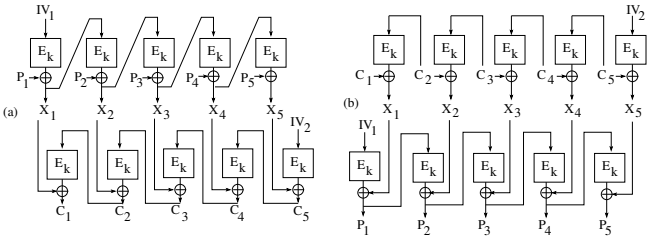

Figure 4. Wrapped CFB mode: (a) encryption, (b) decryption.

