Storage media encryption

Mart Sõmermaa

September 30, 2005

1 Introduction

Recently, new algorithms have been proposed for encryption of storage media at sector level. Main motivation for the new schemes is the IEEE effort to determine a standard architecture for encrypted shared storage media [22].

This article demonstrates that the classic encryption modes standardized by NIST [23] are unsuitable for designing storage media encryption schemes; gives an overview of the new encryption modes and proposes a general model for implementing storage media encryption at sector level.

◀ FIXME: discuss the benefits of sector level encryption over filesystem level encryption if necessary ▶

2 Preliminaries

We will consider a *storage medium* to be a finite totally ordered set $Z \in \mathcal{Z}$ of $N \in \mathbb{N}$ sectors, where $\mathcal{Z}$ is the storage media space. A *sector* $S \in \mathcal{S}$ is a sequence of 4096 bits, where $\mathcal{S} = \{0,1\}^{4096}$ is the sector space.

Sector level cryptosystems consist of a hierarchy of transforms. At the highest level is the mapping from plaintext media to encrypted media and its inverse, $\mathcal{E} : \mathcal{K} \times \mathcal{Z} \to \mathcal{Z}$ and $\mathcal{D} : \mathcal{K} \times \mathcal{Z} \to \mathcal{Z}$, where $\mathcal{K}$ is the key space and $\mathcal{D}(\mathcal{E}(Z)) = Z$, $Z \in \mathcal{Z}$.

$\mathcal{E}$ and $\mathcal{D}$ make use of a *tweakable enciphering scheme* as defined in [5, 7] to transform individual sectors. A tweakable sector enciphering scheme is a function $E : \mathcal{K} \times \mathcal{T} \times \mathcal{S} \to \mathcal{S}$ that maps a plaintext sector S^p to the corresponding ciphertext sector $E_K^T(S^p)$, and its inverse $D : \mathcal{K} \times \mathcal{T} \times \mathcal{S} \to \mathcal{S}$ that maps a ciphertext sector S^c to the plaintext sector $D_K^T(S^c)$, given a key $K \in \mathcal{K}$ and a *tweak* $T \in \mathcal{T}$. In our application domain the tweak space $\mathcal{T}$ is the index set of Z — the sector index (sector location on storage media) is used as the tweak. Using the tweak ensures that encryption depends not only on the plaintext and the key, but also on the sector location.

The enciphering schemes usually divide sectors into $m = 4096/n$ blocks and use a n-bit block cipher to transform the individual blocks. Thus, the enciphering scheme is a *mode of operation* for the block cipher.

Considering application performance and ease of implementation, it is usually required that sector transform operations should be independent in the sense that they should not utilize other sectors during transformation. Also, E, D should be lenght-preserving, i.e. $|S| = |E(S)| = |D(S)|$. These requirements exclude authenticated encryption modes as candidate algorithms for sector enciphering schemes.

There is also a hidden assumption that $\mathcal{E}, \mathcal{D}$ should be order-preserving, i.e. $S_i \xrightarrow{\mathcal{E}} E(S)_i$ for any index $i \in \mathcal{T}$. However, in practice there are cryptosystems that reorder sectors [8] to achieve better security properties.

2.1 Security notions

We use the following notions originally given in [10, 7, 5] throughout the text.

Given a random secret key $K \in \mathcal{K}$, plaintext P and ciphertext c, $p, c \in \{0,1\}^n$, $n \in \mathbb{N}$, we say that a transform $f : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ is a *pseudorandom permutation* (*prp*), if an oracle that maps p into $f_K(p)$ is indistinguishable from an oracle that outputs random permutations.

We say that a transform f is a *strong pseudorandom permutation* ($\pm prp$), if f and its inverse are both pseudorandom permutations.

Lastly, given a tweak $T \in \mathcal{T}$, we say that a transform $f : \mathcal{T} \times \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and its in-

verse $g : \mathcal{T} \times \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ is a *tweakable strong pseudorandom permutation* ($\pm\widetilde{prp}$), if an oracle that maps (T, p) into $f_K^T(p)$ and maps (T, c) into $g_K^T(c)$ is indistinguishable from an oracle that realizes an T-indexed family of random permutations and their inverses.

◀ FIXME: it may be necessary to define what is meant by indistinguishability ▶

2.2 Modes of operation

We will consider the modes in the context of the application domain as candidate algorithms for sector enciphering schemes.

Let e signify encryption and d decryption mode of a $\pm prp$ block cipher and p_i signify i-indexed plaintext and c_i ciphertext blocks.

◀ FIXME: elaborate more ▶

3 The standard modes of operation

The standard modes of block cipher operation are specified in the NIST standard [3]. The security analysis of CBC and CTR modes is given in [1] and of ECB and CBC in [9].

We will demonstrate by concrete attacks that none of the standard modes is $\pm\widetilde{prp}$-secure.

3.1 Electronic Codebook mode

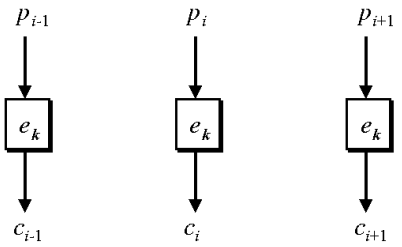

Figure 1: Encryption in Electronic Codebook (ECB) mode

Encryption. Input: K, e, $S^p = p_1, \ldots, p_m$.

$$e_K(p_i) \to c_i, \quad i = 1, \ldots, m.$$

Decryption. Input: K, d, $S^c = c_1, \ldots, c_m$.

$$d_K(c_i) \to p_i, \quad i = 1, \ldots, m.$$

Security. The mode is trivially not prp-secure as equal plaintext blocks are transformed to equal ciphertext blocks. Knudsen illustrates the vulnerability with images that are clearly identifiable when encrypted in ECB mode [9].

3.2 Cipherblock Chaining mode

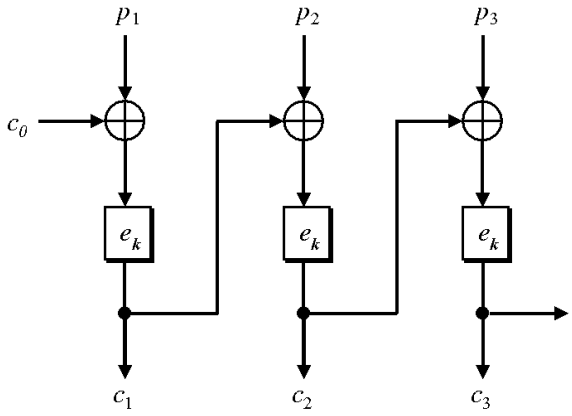

Figure 2: Encryption in Cipherblock Chaining (CBC) mode

Encryption. Input: K, e, T, $S^p = p_1, \ldots, p_m$.

$$e_K(p_i \oplus c_{i-1}) \to c_i, \quad i = 1, \ldots, m, \ c_0 = T.$$

Decryption. Input: K, d, T, $S^c = c_1, \ldots, c_m$.

$$c_{i-1} \oplus d_K(c_i) \to p_i, \quad i = 1, \ldots, m, \ c_0 = T.$$

Security. The mode is malleable (see vulnerability 3.2.1) and vulnerable to copy-paste attacks (vuln. 3.2.2), hence not $\pm prp$-secure. Also, it is vulnerable to watermarking (vuln. 4.1.1) if the adversary can compute the tweak (initialisation vector c_0) and choose plaintext.

Vulnerability 3.2.1 (malleability)**.** Modifications in the ciphertext block c_{i-1} corrupt the corresponding plaintext block p'_{i-1} and enable the adversary

to fully control the contents of the next plaintext block p'_i, as $p'_i = c'_{i-1} \oplus d_K(c_i)$.

Attack.
Input: $i \in \mathbb{N}$, a plaintext block p_i known to the adversary and ciphertext blocks c_i, c_{i-1}; a replacement block $p^{\#}$ chosen by the adversary.
Result: p'_{i-1} will be pseudorandom, $p'_i = p^{\#}$.
As

$$d_K(c_i) = c_{i-1} \oplus p_i \quad \text{and} \quad p^{\#} = p^{\#} \oplus p_i \oplus p_i,$$

then if the adversary choses $c'_{i-1} = p^{\#} \oplus c_{i-1} \oplus p_i$, then

$$p'_i = c'_{i-1} \oplus d_K(c_i) = c'_{i-1} \oplus c_{i-1} \oplus p_i = \\ = p^{\#} \oplus c_{i-1} \oplus p_i \oplus c_{i-1} \oplus p_i = p^{\#}.$$

As $d_K(c'_{i-1})$ is pseudorandom, $p'_{i-1} = c_{i-2} \oplus d_K(c'_{i-1})$ will also be pseudorandom. □

Vulnerability 3.2.2 (reordering)**.** Sequences of ciphertext blocks can be copied and pasted to a new location. The first and the next after last block will be corrupted, the intermediate blocks will be decrypted correctly.

Attack.
Input: $i, j, k \in \mathbb{N}$, plaintext blocks $p_i, \ldots, p_{i+k}$, that the adversary wants to replace with $p_j, \ldots, p_{j+k}$; corresponding ciphertext blocks $c_i, \ldots, c_{i+k}$ and $c_j, \ldots, c_{j+k}$.
Result: $p'_{i-1} = c_{j-2} \oplus c_{i-2} \oplus p_{i-1}$, $p'_i = p_j, \ldots, p'_{i+k} = p_{j+k}$, $p'_{i+k+1} = c_{j+k} \oplus c_{i+k} \oplus p_{i+k+1}$.
If the blocks are replaced,

$$\ldots,\ c_{i-2},\ c'_{i-1} = c_{j-1},\ c'_i = c_j,\ \ldots, \\ c'_{i+k} = c_{j+k},\ c_{i+k+1},\ \ldots$$

then trivially

$$p'_{i-1} = c_{i-2} \oplus d_K(c'_{i-1}) = \\ c_{i-2} \oplus d_K(c_{j-1}) = c_{i-2} \oplus c_{j-2} \oplus p_{j-1}, \\ p'_i = c'_{i-1} \oplus d_K(c'_i) = c_{j-1} \oplus c_{j-1} \oplus p_j = p_j, \\ \ldots, \\ p'_{i+k} = c'_{i+k-1} \oplus d_K(c'_{i+k}) = p_{j+k}, \\ p'_{i+k+1} = c'_{i+k} \oplus d_K(c_{i+k+1}) = \\ c_{j+k} \oplus c_{i+k} \oplus p_{i+k+1}.$$ □

3.3 Output Feedback mode

This and the following modes contain an intermediate cipherblock layer $z_1, \ldots, z_m$ that is XORed with plaintext to get final ciphertext. The values $z_1, \ldots, z_m$ should never repeat. Accordingly it is required that the tweak is a nonce for a given key. However, in a sector enciphering scheme the tweak is a simple integer index that is reused whenever new data is written to a particular sector. Hence all these modes are trivially vulnerable (vuln. 3.3.1) when used in a sector enciphering scheme.

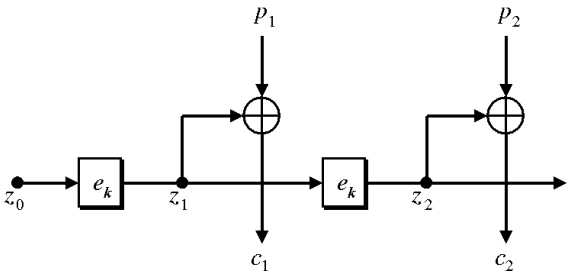

Figure 3: Encryption in Output Feedback (OFB) mode

Encryption. Input: K, e, T, $S^p = p_1, \ldots, p_m$.

$$e_K(z_{i-1}) \to z_i, \quad p_i \oplus z_i \to c_i, \quad i = 1, \ldots, m,\ z_0 = T.$$

Decryption. Input: K, d, T, $S^c = c_1, \ldots, c_m$.

$$e_K(z_{i-1}) \to z_i, \quad c_i \oplus z_i \to p_i, \quad i = 1, \ldots, m,\ z_0 = T.$$

Security. The mode is not *prp*-secure, if the tweak T is reused (vuln. 3.3.1). Even if the tweak is not reused, the mode is still malleable (same attack as in vuln. 3.2.1 with even worse impact), hence not $\pm$*prp*-secure.

Vulnerability 3.3.1 (repeating XOR operand)**.** If a block z_i from the cipherblock layer is used twice, then the adversary gains information about the plaintext.

Attack.
Input: $i \in \mathbb{N}$, ciphertext blocks $c_i^a = p_i^a \oplus z_i$ and $c_i^b = p_i^b \oplus z_i$.
Result: adversary will know $p_i^a \oplus p_i^b$.
Trivially,

$$c_i^a \oplus c_i^b = p_i^a \oplus z_i \oplus p_i^b \oplus z_i = p_i^a \oplus p_i^b.$$

If either p_i^a or p_i^b is zero, the other plaintext block will be revealed to the adversary. □

3.4 Counter mode

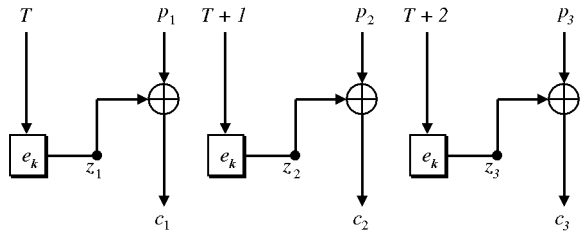

Figure 4: Encryption in Counter (CTR) mode

Encryption. Input: K, e, T, $S^p = p_1, \ldots, p_m$.

$$e_K(T + i - 1) \to z_i, \quad p_i \oplus z_i \to c_i, \quad i = 1, \ldots, m,$$

Decryption. Input: K, d, T, $S^c = c_1, \ldots, c_m$.

$$e_K(T + i - 1) \to z_i, \quad c_i \oplus z_i \to p_i, \quad i = 1, \ldots, m,$$

Security. The mode is not *prp*-secure, if the tweak T is reused (vuln. 3.3.1).

3.5 Cipher Feedback mode

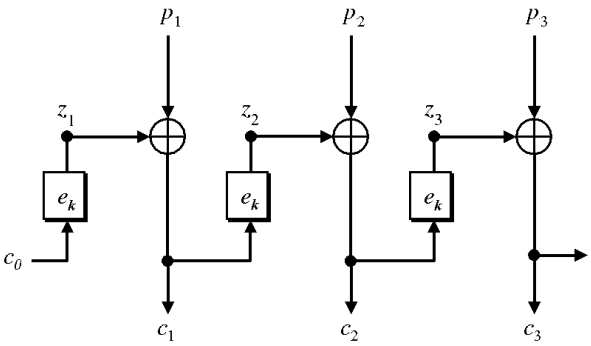

Figure 5: Encryption in Cipher Feedback (CFB) mode

Encryption. Input: K, e, T, $S^p = p_1, \ldots, p_m$.

$$e_K(c_{i-1}) \to z_i, \quad p_i \oplus z_i \to c_i, \quad i = 1, \ldots, m,\ c_0 = T.$$

Decryption. Input: K, d, T, $S^c = c_1, \ldots, c_m$.

$$e_K(c_{i-1}) \to z_i, \quad c_i \oplus z_i \to p_i, \quad i = 1, \ldots, m,\ c_0 = T.$$

Security. Like OFB and CTR, the first block is subject to vulnerability 3.3.1 if the tweak T is reused. The mode is not *prp*-secure in this case. Additionally, like CBC, the mode is malleable (vuln. 3.2.1) and vulnerable to copy-paste attacks (vuln. 3.2.2), hence not $\pm$*prp*-secure even if the tweak is a nonce.

3.6 Conclusion

As demonstrated above, none of the standard modes is suitable for constructing a length-preserving sector enciphering scheme.

4 Case study: a CBC-based sector level cryptosystem

The sector level cryptosystem described in this section was used in the Linux kernel block device encryption modules *loopAES* and *dm_crypt*. The enciphering scheme described below is considered to be deprecated, but serves well as a case study. We will refer to it as cryptosystem $\mathbb{A}$.

The encryption was (and is) implemented as a ciphering loop filter between filesystem and device driver layers in the storage stack (see fig. 6).

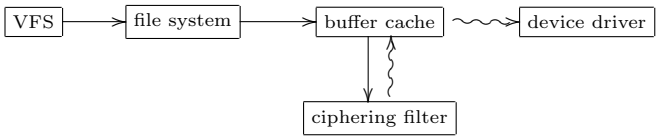

Figure 6: Storage stack with a ciphering filter

The following CBC-based sector enciphering scheme was utilized in the cryptosystem:

$$E(K, T = i, S_i^p) \to S_i^c, \quad D(K, T = i, S_i^c) \to S_i^p,$$

$$E : e_K(p_j \oplus c_{j-1}) \to c_j, \quad j = 1, \ldots, m,$$
$$D : c_{j-1} \oplus d_K(c_j) \to p_j, \quad j = 1, \ldots, m,$$
$$c_0 = T.$$

The sector index is directly used as the CBC mode initialisation vector (IV), which means that the IV is a counter. As noted in [1] and demonstrated

in vulnerability 4.1.1, counter IV should never be used in CBC mode, as this considerably weakens the mode's security.

4.1 Vulnerabilities

All CBC mode vulnerabilities (see vuln. 3.2.1 and 3.2.2) apply to cryptosystem $\mathbb{A}$ as well. Additionally, due to the counter IV utilised in the system, it is subject to the "watermarking" vulnerability 4.1.1.

Also, cryptoanalysis of the ciphertext is facilitated as

- known plaintext is always given as filesystem metadata contents and location in ciphertext is known to the adversary,
- all of the ciphertext is encoded with the same key.

Vulnerability 4.1.1 (controllable collisions)**.** It is possible to create collisions in subsequent sectors in cryptosystem $\mathbb{A}$ with a chosen plaintext attack.

The attack is based on the following property of any CBC-based sector level cryptosystem: if the values of IV are known to the adversary, she can choose p_1^i, p_1^j, given sector indexes $i, j \in \mathcal{T}$, such that $IV_i \oplus IV_j = p_1^i \oplus p_1^j$, then

$$IV_i \oplus p_1^i = IV_j \oplus p_1^j \Rightarrow$$
$$e_K(IV_i \oplus p_1^i) = e_K(IV_j \oplus p_1^j) \Rightarrow c_1^i = c_1^j.$$

The attack was first described by Saarinen [18], it follows directly from the CBC-mode collision analysis of Knudsen [9].

Attack.
Input: $i \in \mathcal{T}$, the first plaintext blocks $p_1^{S_i}, p_1^{S_{i+1}}, p_1^{S_{i+2}}$ of three subsequent sectors S_i, S_{i+1}, S_{i+2} chosen by the adversary.
Result: either $c_1^{S_i} = c_1^{S_{i+1}}$ or $c_1^{S_{i+1}} = c_1^{S_{i+2}}$.

Andversary chooses the following bit sequences for the plaintexts:

$$\begin{aligned} p_1^{S_i} &= (y_n, \ldots, y_2, 0), \\ p_1^{S_{i+1}} &= (y_n, \ldots, y_2, 1), \\ p_1^{S_{i+2}} &= (y_n, \ldots, y_2, 0), \end{aligned}$$

where $(y_n, \ldots, y_2)$ is an arbitrary bit sequence.

Let $IV_i = i = (x_{32}, \ldots, x_1)$. If i is even, then

$$p_1^{S_i} \oplus IV_i = (y_n, \ldots, y_2, 0) \oplus (x_{32}, \ldots, x_2, 0) =$$
$$(y_n, \ldots, y_{32} \oplus x_{32}, \ldots, y_2 \oplus x_2, 0) =$$
$$(y_n, \ldots, y_2, 1) \oplus (x_{32}, \ldots, x_2, 1) = p_1^{S_{i+1}} \oplus IV_{i+1}.$$

If i is odd, then as $i + 1 = (x'_{32}, \ldots, x'_1)$ is even, then

$$p_1^{S_{i+1}} \oplus IV_{i+1} = (y_n, \ldots, y_2, 1) \oplus (x'_{32}, \ldots, x'_2, 0) =$$
$$(y_n, \ldots, y_{32} \oplus x'_{32}, \ldots, y_2 \oplus x'_2, 1) =$$
$$(y_n, \ldots, y_2, 0) \oplus (x'_{32}, \ldots, x'_2, 1) = p_1^{S_{i+2}} \oplus IV_{i+2}.$$

If $p_1^r \oplus IV_r = p_1^q \oplus IV_q$ for some sector indexes $r, q \in \mathcal{T}, r \neq q$, then

$$e_K(p_1^r \oplus IV_r) = e_K(p_1^q \oplus IV_q) \Rightarrow c_1^r = c_1^q.$$

Hence there are two equal cipherblocks among blocks $c_1^{S_i}, c_1^{S_{i+1}}, c_1^{S_{i+2}}$.

If $p_2^r = p_2^q$ also, then the next cipherblock pairs will be equal as well,

$$p_2^r = p_2^q,\ c_1^r = c_1^q \Rightarrow$$
$$e_K(p_2^r \oplus c_1^r) = e_K(p_2^q \oplus c_1^q) \Rightarrow c_2^r = c_2^q,$$

etc.

Using a deterministic transform with non-secret input to get a pseudorandom IV does not help — as noted above, if adversary can compute IV and choose plaintext, she can create collisions in ciphertext. □

4.2 Conclusion

The sector enciphering scheme of cryptosystem $\mathbb{A}$ is not *prp*-secure.

5 New $\pm\widetilde{prp}$-secure modes

IEEE Security in Storage Working Group [22] has been evaluating proposals for $\pm\widetilde{prp}$-secure modes as candidate algorithms for a standard sector enciphering scheme since 2002. As of now, the modes described below have been proposed. No standard has been agreed upon yet, the candidate algorithms are currently EME and LRW.

All modes described below are provably $\pm\widetilde{prp}$-secure (given that the underlying block cipher is $\pm prp$-secure). However, the random oracle model used in the proofs has been widely critizised (see e.g. [12], 15.2.6), accordingly it doesn't necessarily follow from the proofs that the modes don't have any practical weaknesses.

The modes are divided into narrow- and wide-block modes. Wide-block modes operate with at least sector granularity, whereas narrow-block modes generally operate with cipher block granularity. Wide-block modes are preferrable as it is possible to detect e.g. database write patterns to the storage media when a narrow block mode is in use [17].

We use $\otimes$ to signify multiplication in the field $GF(2^n)$. Note that multiplication by 2 is much easier to implement and computationally less costly than general multiplication in $GF(2^n)$.

5.1 EME

Authors: Shai Halevi and Phillip Rogaway.

Specification: [5, 6, 21].

Description. EME stands for *ECB-mix-ECB*, the algorithm entails two layers of ECB encryption and a "lightweight mixing" in between.

Encryption. Input: K, e, T, $S^p = p_1, \ldots, p_m$.

$$
\begin{aligned}
&L \leftarrow 2 \otimes e_K(0^n)\\
&PP_i \leftarrow 2^{i-1} \otimes L \oplus p_i,\\
&\quad PPP_i \leftarrow e_K(PP_i), \quad i = 1, \ldots, m\\
&SP \leftarrow PPP_2 \oplus \cdots \oplus PPP_m\\
&MP \leftarrow PPP_1 \oplus SP \oplus T\\
&MC \leftarrow e_K(MP)\\
&M \leftarrow MP \oplus MC\\
&CCC_i \leftarrow PPP_i \oplus 2^{i-1} \otimes M, \quad i = 2, \ldots, m\\
&SC \leftarrow CCC_2 \oplus \cdots \oplus CCC_m\\
&CCC_1 \leftarrow MC \oplus SC \oplus T\\
&CC_i \leftarrow e_K(CCC_i),\\
&\quad CC_i \oplus 2^{i-1} \otimes L \rightarrow c_i, \quad i = 1, \ldots, m
\end{aligned}
$$

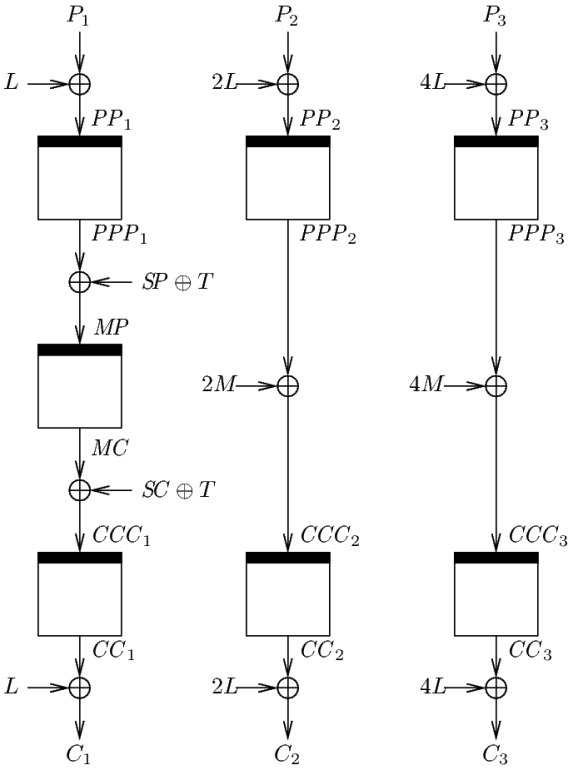

Figure 7: Encryption in EME mode

Decryption. Input: K, d, T, $S^c = c_1, \ldots, c_m$.

$$
\begin{aligned}
&L \leftarrow 2 \otimes e_K(0^n) \\
&CC_i \leftarrow 2^{i-1} \otimes L \oplus c_i, \\
&\quad CCC_i \leftarrow d_K(CC_i), \quad i = 1, \ldots, m \\
&SC \leftarrow CCC_2 \oplus \cdots \oplus CCC_m \\
&MC \leftarrow CCC_1 \oplus SC \oplus T \\
&MP \leftarrow d_K(MC) \\
&M \leftarrow MP \oplus MC \\
&PPP_i \leftarrow CCC_i \oplus 2^{i-1} \otimes M, \quad i = 2, \ldots, m \\
&SP \leftarrow PPP_2 \oplus \cdots \oplus PPP_m \\
&PPP_1 \leftarrow MP \oplus SP \oplus T \\
&PP_i \leftarrow d_K(PPP_i), \\
&\quad PP_i \oplus 2^{i-1} \otimes L \rightarrow p_i, \quad i = 1, \ldots, m
\end{aligned}
$$

Considerations. EME is patented and relatively computationally expensive. Makes use of the operation $2\otimes$ that needs to be implemented separately.

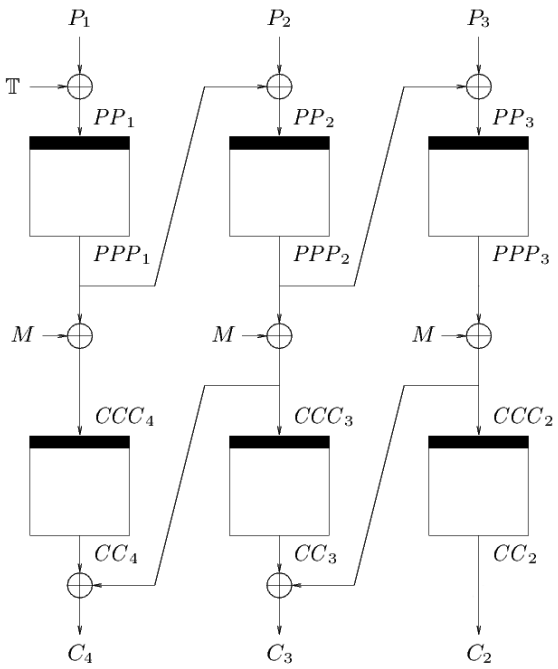

Figure 8: Encryption in CMC mode

5.2 CMC

Authors: Shai Halevi and Phillip Rogaway.

Specification: [7].

Description. CMC stands for *CBC-mix-CBC*, the algorithm makes a pass of CBC encryption, XORs in a mask, and then makes a pass of CBC decryption. The layered structure is similar to EME.

The authors recommend EME over CMC as it is as secure but has several advantages: it is parallelizable, only one key required, utilizes only e in encryption and d in decryption, remains secure for variable length input.

Encryption. Input: K, $\tilde{K}$, e, T, $S^p = p_1, \ldots, p_m$.

$$
\begin{aligned}
&\mathbb{T} \leftarrow e_{\tilde{K}}(T) \\
&PPP_0 \leftarrow \mathbb{T} \\
&PP_i \leftarrow p_i \oplus PPP_{i-1}, \\
&\quad PPP_i \leftarrow e_K(PP_i), \quad i = 1, \ldots, m \\
&M \leftarrow 2 \otimes (PPP_1 \oplus PPP_m) \\
&CCC_i \leftarrow PPP_{m+1-i} \oplus M, \quad i = 1, \ldots, m \\
&CCC_0 \leftarrow 0^n \\
&CC_i \leftarrow e_K(CCC_i), \\
&\quad CC_i \oplus CCC_{i-1} \rightarrow c_i, \quad i = 1, \ldots, m \\
&c_1 \oplus \mathbb{T} \rightarrow c_1.
\end{aligned}
$$

Decryption. Input: K, $\tilde{K}$, e, d, T, S^c =

$c_1, \ldots, c_m$.

$$\mathbb{T} \leftarrow e_{\tilde{K}}(T)$$
$$CCC_0 \leftarrow \mathbb{T}$$
$$CC_i \leftarrow c_i \oplus CCC_{i-1},$$
$$\quad CCC_i \leftarrow d_K(CC_i), \quad i = 1, \ldots, m$$
$$M \leftarrow 2 \otimes (CCC_1 \oplus CCC_m)$$
$$PPP_i \leftarrow CCC_{m+1-i} \oplus M, \quad i = 1, \ldots, m$$
$$PPP_0 \leftarrow 0^n$$
$$PP_i \leftarrow e_K(PPP_i),$$
$$\quad PP_i \oplus PPP_{i-1} \rightarrow p_i, \quad i = 1, \ldots, m$$
$$p_1 \oplus \mathbb{T} \rightarrow p_1.$$

Considerations. CMC is patented and relatively computationally expensive. Needs two keys. Makes use of the operation $2\otimes$ that needs to be implemented separately.

5.3 LRW

Authors: Based on the theoretical construction given in [10] by Moses Liskov, Ronald L. Rivest and Donald Wagner. Instantiated by Ian F. Blake, Cyril Guyot, Clement Kent and V. Kumar Murty.

Specification: [10, 20, 2].

Description. The name is based on the first letters of the surnames of original authors. It is a simple mode utilizing multiplication in $GF(2^n)$.

Encryption. Input: K, $\tilde{K}$, e, T, $S^p = p_1, \ldots, p_m$.

$$\mathbb{T}_i \leftarrow \tilde{K} \otimes (T + i),$$
$$\quad e_K(p_i \oplus \mathbb{T}_i) \oplus \mathbb{T}_i \rightarrow c_i, \quad i = 1, \ldots, m$$

Decryption. Input: K, $\tilde{K}$, d, T, $S^c = c_1, \ldots, c_m$.

$$\mathbb{T}_i \leftarrow \tilde{K} \otimes (T + i),$$
$$\quad d_K(c_i \oplus \mathbb{T}_i) \oplus \mathbb{T}_i \rightarrow c_i, \quad i = 1, \ldots, m$$

Considerations. LRW is effective, but needs two keys and is a narrow-block mode. Makes use of the general operation $\otimes$ that needs to be implemented separately (adds considerable complexity to the otherwise simple implementation).

5.4 XCB

Authors: Based on the theoretical construction given in [11] by Michael Luby and Charles Rackoff. Instantiated by David A. McGrew and Scott R. Fluhrer.

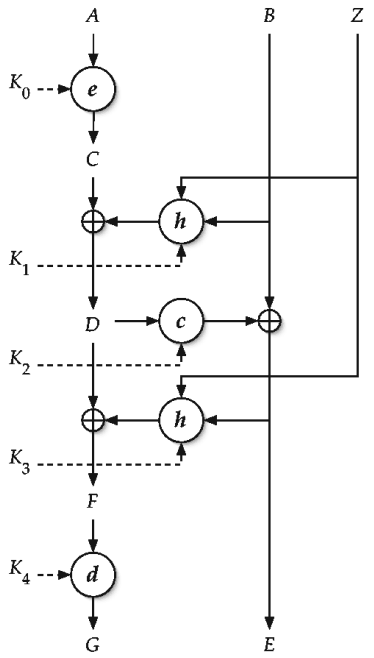

Figure 9: Encryption in XCB mode

Specification: [13, 14, 11].

Description. XCB stands for *Extended Codebook*.

The mode entails two layers of hashing and a CTR-like layer in between that utilises the hash. GHASH [14] is used as the hash function.

The mode seems to be a direct follow-up to the officially unpublished ABL mode by David A. McGrew and John Viega that is dubbed "a Luby-Rackoff cipher" in the informal specification [15].

Encryption. Input: K, e, d, $h = GHASH$, T, $S^p = p_1, \ldots, p_m$.

$$K_i \leftarrow e_K(i), \quad i = 0, \ldots, 4$$
$$B \leftarrow p_2, \ldots, p_m$$
$$D \leftarrow e_{K_0}(p_1) \oplus h_{K_1}(B, T)$$
$$E \leftarrow B \oplus [e_{K_2}(D + 0) || e_{K_2}(D + 1 \bmod 2^n) || \cdots$$
$$\quad || e_{K_2}(D + m - 2 \bmod 2^n)]$$
$$F \leftarrow D \oplus h_{K_3}(E, T)$$
$$G \leftarrow d_{K_4}(F)$$
$$G || E \rightarrow C$$

Decryption. Input: K, e, d, h, T, $S^c = c_1, \ldots, c_m$.

$$K_i \leftarrow e_K(i), \quad i = 0, \ldots, 4$$
$$E \leftarrow c_2, \ldots, c_m$$
$$D \leftarrow e_{K_4}(c_1) \oplus h_{K_3}(E, T)$$
$$B \leftarrow E \oplus [e_{K_2}(D+0)||e_{K_2}(D+1 \bmod 2^n)|| \cdots ||e_{K_2}(D+m-2 \bmod 2^n)]$$
$$A \leftarrow D \oplus h_{K_1}(B, T)$$
$$p_1 \leftarrow d_{K_0}(A)$$
$$p_1||B \rightarrow P$$

Considerations. XCB is a new mode, possibly not scrutinized enought by cryptologists. Patented, utilizes hashing that needs to be implemented separately and is computationally expensive.

5.5 Authenticated encryption modes

As only length-preserving transformations can be used in the sector enciphering scheme, we will not give a thourough treatment to authenticated encryption modes.

Still, we will list some of the issues that must be addressed when using these modes in sector level cryptosystems.

Authentication code storage. Where and how should the MACs be stored to protect against tampering and storage media corruption.

Storage space loss. MACs will consume space.

Storage media corruption. A corrupted MAC sector will render all corresponding ciphertext sectors inacessible.

I/O overhead. Every I/O operation needs to access the corresponding MAC sector(s) additionally to data sectors.

All these issues add considerable complexity to the implementation.

5.6 Conclusion

Comparison of the new modes is given in table 10. The column labels have the following meanings: Gr. — mode granularity, either sector (s) or block (b); e, $\oplus$, $2\otimes$, h — number of block cipher, XOR, $GF(2^n)$ multiplication and hash operations respectively, given m blocks; IP – intellectual property issues, either unencumbered ($-$) or encumbered ($+$) by patents.

Name	Gr.	e	$\oplus$	$2\otimes$	h	IP
CMC	s	$2m+1$	$2m+1$	1	–	+
EME	s	$2m+1$	$5m$	$3m-1$	–	+
LRW	b	m	$2m$	m ($\otimes$)	–	–
XCB	s	$m+1$	$m+1$	–	2	+

Figure 10: Comparison of new modes

A generally usable mode should be provably $\pm\widetilde{prp}$-secure, patent-free and should operate with sector granularity. Currently, there is no such mode.

6 General model for secure sector level cryptosystems

As noted in [8], there are other considerations apart from specifying a mode of operation when implementing a secure sector level cryptosystem — cryptoanalysis will be harder if per-sector unique keys are used and if the ordering of sectors is changed. The following components can be identified in the system:

1. a block cipher,
2. a mode of operation,
3. a function for generating sector keys,
4. a function for sector reordering,

◀ FIXME: siia ainemudeli joonis ▶.

As discussed above, the mode of operation should be $\pm\widetilde{prp}$-secure. Accordingly, the underlying block cipher should be $\pm prp$-secure. We will discuss other components and their requirements below.

6.1 Functions for generating sector keys

Several sources [16, 8] recommend against using a single key for encrypting large amounts of redundant data. Hence it is adviseable to use a *key generation function* $f : \mathcal{T} \times \mathcal{K} \rightarrow \mathcal{K}$ that maps a tweak

$T \in \mathcal{T}$ for sector $S_T \in \mathcal{S}$ to corresponding sector key $K_T \in \mathcal{K}$, using the global key $K \in \mathcal{K}$. The function f should be *prp*-secure. The global key K should not be used in other functions (see e.g. [24]).

The function f has to be deterministic, as domain constraints require sector transformations to be independent: no data expansion (e.g. per-sector random key storage like in FreeBSD GDBE cryptosystem [8]) can occur. Deterministic key transforms do not complicate brute-force or dictionary attacks, but they hinder cryptoanalysis and chosen plaintext/ciphertext attacks. For example, unique per-sector keys protect against the "watermarking" (vuln. 4.1.1) and sector reordering (special case of vuln. 3.2.2) attacks in cryptosystem $\mathbb{A}$.

A $\pm prp$ block cipher e is a natural candidate for the key generation function f,

$$e_K(T) \to K_T.$$

Similar approach for generating round keys is used e.g. in XCB mode (see section 5.4).

Note that even if the same algorithm is used for transforming the tweak in the sector enciphering scheme, the transform will be $e_{K_T}(T)$, not $e_K(T)$, as the use of the global key is not allowed in the sector transform. As $e_{e_K(T)}(T)$ is unrelated to $e_K(T) = K_T$, we claim that there are no security concerns or information leakage in this case. However, this claim needs further verification.

6.2 Functions for sector reordering

Sector reordering is mainly useful for hiding known plaintext — all filesystems contain metadata whose location and contents are either constant or computable[1]. A *sector reordering function* $g : \mathcal{T} \times \mathcal{K} \times \mathbb{Z}_{2^{64}} \to \mathcal{T}$ is a bijection that maps a plaintext sector index (location on disk) $T^p \in \mathcal{T}$ to corresponding ciphertext sector index $T^c \in \mathcal{T}$, given a key $K \in \mathcal{K}$ and the cardinality $N \in \mathbb{Z}_{2^{64}}$ of the set $\mathcal{T}$.

However, the reordering can not be arbitrary as effects on I/O performance have to be considered — the storage stack is optimised for contigous large-block I/O. Accordingly, we propose to divide the sector set into clusters of some architecture-specific size and first reorder the clusters and then reorder sectors within the cluster. That ensures that file data is located in the same proximity (to make use of optimizations like read-ahead etc.) up to the cluster boundary.

Sector reordering is used e.g. in the FreeBSD GDBE cryptosystem [8].

[1] For example, in the *ext2/ext3* filesystems the first 16 bytes of fourth sector in a partition are zero-filled.

6.3 Conclusion

A secure storage media encryption scheme should contain the following components:

1. a $\pm prp$-secure block cipher
$$e : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n,$$
2. a $\pm\widetilde{prp}$-secure sector enciphering scheme
$$E : \mathcal{K} \times \mathcal{T} \times \mathcal{S} \to \mathcal{S},$$
3. a *prp*-secure key generation function
$$f : \mathcal{T} \times \mathcal{K} \to \mathcal{K},$$
4. a sector reordering function
$$g : \mathcal{T} \times \mathcal{K} \times \mathbb{Z}_{2^{64}} \to \mathcal{T}.$$

We also note, that there are other implementantion issues like key management and key backup that need to be addressed.

References

[1] Mihir Bellare, Anand Desai, Eron Jokipii ja Phillip Rogaway. *A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation*, kogumikus *Proceedings of 38th Annual Symposium on Foundations of Computer Science*, IEEE, 1997.
URL: `http://www-cse.ucsd.edu/users/mihir/papers/sym-enc.ps`

[2] Ian F. Blake, Cyril Guyot, Clement Kent ja V. Kumar Murty. *Encryption of Stored Data in Networks: Analysis of a Tweaked Block Cipher*, IEEE turvaliste salvestussüsteemide töögrupp, august 2004.
URL: `http://grouper.ieee.org/groups/1619/email/ps00000.ps`

[3] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation — Methods and Techniques*, Ameerika Ühendriikide Rahvusliku Standardi- ja Tehnoloogiainstituudi publikatsioon nr SP800-38a, detsember 2001.
URL: `http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf`

[4] Donald E. Eastlake 3rd, Stephen D. Crocker ja Jeffrey I. Schiller. *Randomness Recommendations for Security*, kommentaarikutse nr 1750.
URL: `http://www.faqs.org/rfcs/rfc1750.html`

[5] Shai Halevi, Phillip Rogaway. *A Parallelizable Enciphering Mode*, kogumikus *Cryptology ePrint Archive*, raport nr 2003/147, 2003.
URL: `http://eprint.iacr.org/2003/147`

[6] Shai Halevi, Phillip Rogaway. *EME*: extending EME to handle arbitrary-length messages with associated data*, kogumikus *Cryptology ePrint Archive*, raport nr 2004/125, 2004.
URL: `http://eprint.iacr.org/2004/125`

[7] Shai Halevi, Phillip Rogaway. *A Tweakable Enciphering Mode*, kogumikus *Cryptology ePrint Archive*, raport nr 2003/148, 2003.
URL: `http://eprint.iacr.org/2003/148`

[8] Poul-Henning Kamp. *GBDE – GEOM Based Disk Encryption*, kogumikus *BSDCon '03 Conference Proceedings*, 2003.
URL: `http://phk.freebsd.dk/pubs/bsdcon-03.gbde.paper.pdf`

[9] Lars R. Knudsen. *Block ciphers – Analysis, Design and Applications*, doktoriväitekiri, kogumikus DAIMI PB–485, Århusi Ülikool, 1994.
URL: `http://www.daimi.au.dk/PB/485/PB-485.pdf`

[10] Moses Liskov, Ronald L. Rivest ja David Wagner. *Tweakable Block Ciphers*, kogumikus *Advances in Cryptology - CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA*, Springer-Verlag, 2002.
URL: `http://www.eecs.berkeley.edu/~daw/papers/tweak-crypto02.pdf`

[11] Michael Luby and Charles Rackoff. *How to Construct Pseudo-Random Permutations from Pseudo-Random Functions*, kogumikus *Lecture notes in Computer Science*, köide 218, Springer-Verlag, 1986.
URL: `http://www.springerlink.com/app/home/contribution.asp?wasp=hb5c1xjkwq5qxjfe1yvm&referrer=parent&backto=issue,34,44;journal,1946,1949;linkingpublicationresults,1:105633,1`

[12] Wengbao Mao. *Modern Cryptography: Theory and Practice*, Prentice Hall PTR, 2003.

[13] David A. McGrew, Scott R. Fluhrer. *The Extended Codebook (XCB) Mode of Operation*, kogumikus *Cryptology ePrint Archive*, raport 2004/278, oktoober 2004.
URL: `http://eprint.iacr.org/2004/278`

[14] David A. McGrew, John Viega. *The Galois/Counter Mode of Operation (GCM)*, jaanuar 2004. Vt. [23].

[15] David A. McGrew, John Viega. *The ABL Mode of Operation*, esitlusslaidid, aprill 2004.
URL: `http://grouper.ieee.org/groups/1619/email/pdf00004.pdf`

[16] Alfred J. Menezes, Paul C. van Oorschot ja Scott A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, 1996.

[17] Adam J. Richter. *LRW has more data modification leakage than CBC?*, sõnum SISWG meililoendisse, 25. detsember 2004.
URL: http://article.gmane.org/gmane.linux.kernel.device-mapper.dm-crypt/663

[18] Markku-Juhani O. Saarinen. *Encrypted watermarks and Linux laptop security*, kogumikus *Proc. of the 5th International Workshop on Information Security Applications (WISA2004)*, 2004 (ilmumata).
URL: http://www.tcs.hut.fi/~mjos/doc/wisa2004.pdf

[19] *DES Modes of Operation*, Ameerika Ühendriikide Rahvusliku Standardi- ja Tehnoloogiainstituudi standard nr 81, detsember 1980.
URL: http://csrc.nist.gov/publications/fips/fips81/fips81.htm

[20] *Draft Proposal for Tweakable Narrow-block Encryption, Draft 1.00:00*, IEEE turvaliste salvestussüsteemide töögrupp, august 2004.
URL: http://siswg.org/docs/LRW-AES-10-19-2004.pdf

[21] *Draft Proposal for Tweakable Wide-block Encryption, Draft 1.00:02*, IEEE turvaliste salvestussüsteemide töögrupp, november 2004.
URL: http://siswg.org/docs/EME-AES-03-22-2004.pdf

[22] IEEE turvaliste salvestussüsteemide töögrupi veebileht,
URL: http://siswg.org/

[23] *Modes of Operation for Symmetric Key Block Ciphers*, Ameerika Ühendriikide Rahvuslik Standardi- ja Tehnoloogiainstituut.
URL: http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/ProposedModesPage.html

[24] *Recommendation for Key Management* Ameerika Ühendriikide Rahvuslik Standardi- ja Tehnoloogiainstituut, standardi kavand, jaanuar 2003. URL: http://csrc.nist.gov/CryptoToolkit/kms/guideline-1-Jan03.pdf