

# Modes of Operation of a Block Cipher

B. Preneel, K.U.Leuven, Belgium

A  $n$ -bit block cipher with a  $k$ -bit key is a set of  $2^k$  bijections on  $n$ -bit strings. A block cipher is a flexible building block; it can be used for encryption and authenticated encryption, to construct MAC algorithms and hash functions.

When a block cipher is used for confidentiality protection, the security goal is to prevent a passive eavesdropper with limited computational power to learn any information on the plaintext (except for maybe its length). This eavesdropper can apply the following attacks: known plaintext attacks, chosen plaintext attacks and chosen ciphertext attacks.

Applications need to protect the confidentiality of strings of arbitrary length. A mode of operation of a block cipher is an algorithm which specifies how one has to apply an  $n$ -bit block cipher to achieve this. One approach is to pad the data with a padding algorithm such that the bit-length of the padded string is a multiple  $t$  of  $n$  bits, and to define a mode which works on  $t$   $n$ -bit blocks. For example, one always appends a '1'-bit followed by as many '0' bits as necessary to make the length of the resulting string a multiple of  $n$ . An alternative is to define a mode of operation that can process data in blocks of  $j \leq n$  bits.

We first discuss the five modes of operation which have been defined in the FIPS [12] (see also [22]) and ISO/IEC [16] standards: the ECB mode, the CBC mode, the OFB mode, the CTR

mode, and the CFB mode. Next we discuss some alternative modes that have been defined for triple-DES and modes which allow to encrypt values from finite sets.

We use the following notation:  $E_K(p_i)$  denotes the encryption with a block cipher of the  $n$ -bit plaintext block  $p_i$  with the key  $K$ ; similarly  $D_K(c_i)$  denotes the decryption of the ciphertext  $c_i$ . The operation  $\text{rchop}_j(s)$  returns the rightmost  $j$  bits of the string  $s$ , and the operation  $\text{lchop}_j(s)$  returns the leftmost  $j$  bits. The symbol  $\parallel$  denotes concatenation of strings and  $\oplus$  denotes addition modulo 2 (exor).

## 1 The Electronic Code Book (ECB) Mode

The simplest mode is the ECB (Electronic Code-Book) mode. After padding, the plaintext  $p$  is divided into  $t$   $n$ -bit blocks  $p_i$  and the block cipher is applied to each block; the decryption also operates on individual blocks (see Fig. 1):

$$c_i = E_K(p_i) \quad \text{and} \quad p_i = D_K(c_i), \quad i = 1, \dots, t.$$

Errors in the ciphertext do not propagate beyond the block boundaries (as long as these can be recovered). However, the ECB mode is the only mode covered in this article which does not hide patterns (such as repetitions) in the plaintext. Usage of this mode should be strongly discouraged. In the past the ECB mode was sometimes

recommended for the encryption of keys; however, authenticated encryption would be much better for this application (or the AES key wrapping algorithm proposed by NIST).

## 2 The Cipher Block Chaining (CBC) mode

The most popular mode of operation of a block cipher is the CBC (Cipher Block Chaining) mode. The plaintext  $p$  is divided into  $t$   $n$ -bit blocks  $p_i$ . This mode adds (modulo 2) to a plaintext block the previous ciphertext block and applies the block cipher to this result (see Fig. 2):

$$\begin{aligned} c_i &= E_K(p_i \oplus c_{i-1}) \\ p_i &= D_K(c_i) \oplus c_{i-1} \quad i = 1 \dots t. \end{aligned}$$

Note that in the CBC mode, the value  $c_{i-1}$  is used to randomize the plaintext; this couples the blocks and hides patterns and repetitions. To enable the encryption of the first plaintext block ( $i = 1$ ), one defines  $c_0$  as the initial value  $IV$ , which should be randomly chosen and transmitted securely to the recipient. By varying this  $IV$ , one can ensure that the same plaintext is encrypted into a different ciphertext under the same key, which is essential for secure encryption. The  $IV$  plays a similar role in the OFB, CTR and CFB modes.

The CBC decryption has a limited error propagation: errors in the  $i$ th ciphertext block will garble the  $i$ th plaintext block completely, and will be copied into the next plaintext block. The CBC decryption allows for parallelism and random access: if necessary, one can decrypt only a small part of the ciphertext. However, the encryption mode is a serial operation. To overcome this restriction, ISO/IEC 10116 [16] has defined

a variant of the CBC mode which divides the plaintext into  $r$  parallel streams and applies the CBC mode to each of these streams. This requires however  $r$  different  $IV$  values.

A security proof of the CBC mode (with random and secret  $IV$ ) against an adversary who has access to chosen plaintexts has been provided by Bellare *et al.* [3]; it shows that if the block cipher is secure in the sense that it is hard to distinguish it from a random permutation, the CBC mode offers secure encryption in the sense that the ciphertext is random (which implies that it does not provide the opponent additional information on the plaintext). The security result breaks down if the opponent can obtain approximately  $q = 2^{n/2}$  plaintext/ciphertext pairs due to a matching ciphertext attack [18]. This can be seen as follows. Note that the ciphertext blocks  $c_i$  are random  $n$ -bit strings. After observing  $q$   $n$ -bit ciphertext blocks, one expects to find approximately  $q^2/2^{n+1}$  pairs of matching ciphertexts that is, indices  $(v, w)$  with  $c_v = c_w$  (see also the birthday paradox). As a block cipher is a permutation, this implies that the corresponding plaintexts are equal, or  $p_v \oplus c_{v-1} = p_w \oplus c_{w-1}$  which can be rewritten as  $p_v \oplus p_w = c_{v-1} \oplus c_{w-1}$ . Hence, each pair of matching ciphertexts leaks the sum of two plaintext blocks. To preclude such a leakage, one needs to impose that  $q \ll 2^{(n+1)/2}$  or  $q = \alpha \cdot 2^{n/2}$  where  $\alpha$  is a small constant (say  $10^{-3}$ , which leads to a collision probability of 1 in 2 million). If this limit is reached, one needs to change the key. Note that the proof only considers security against chosen plaintext attacks; the CBC mode is not secure if chosen ciphertext attacks are allowed. The security against these attacks can be obtained by using authenticated encryption.

For some applications, the ciphertext should have exactly the same length as the plaintext,

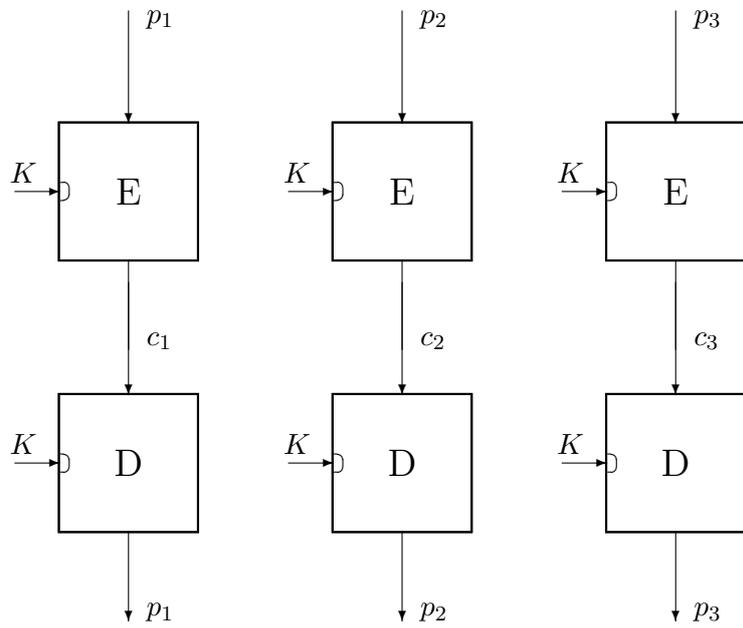


Figure 1: The ECB mode of a block cipher

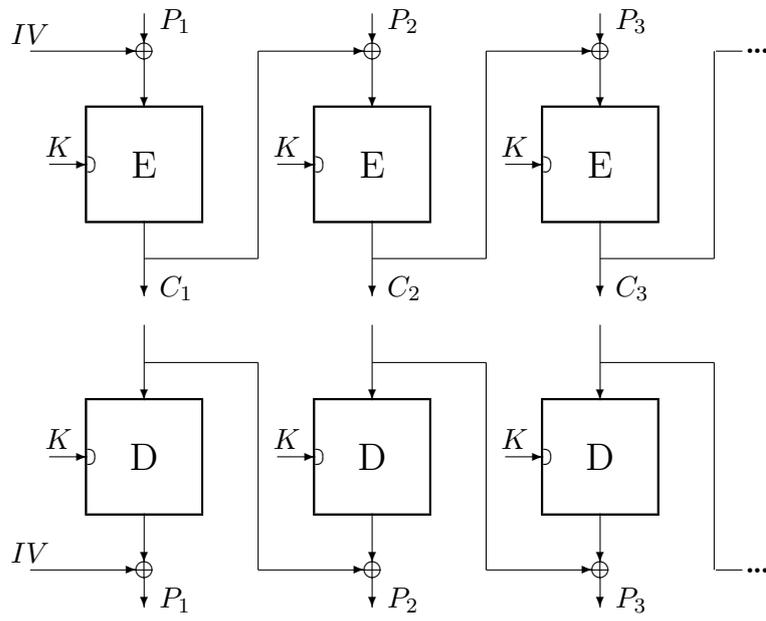


Figure 2: The CBC mode of a block cipher

hence padding methods cannot be used. Two heuristic constructions have been proposed to address this problem; they are not without problems (both leak information in a chosen plaintext setting). A first solution encrypts the last incomplete block  $p_t$  (of  $j < n$  bits) in OFB mode (cf. Sect. 3):

$$c_t = p_t \oplus \text{rchop}_j(E_K(c_{t-1})).$$

A second solution is known as *ciphertext stealing* [21]: one appends the rightmost  $n-j$  bits of  $c_{t-1}$  to the last block of  $j$  bits  $p_t$ , to obtain a new  $n$ -bit block:

$$\begin{aligned} c_{t-1} &= E_K(p_{t-1} \oplus c_{t-2}) \\ c_t &= E_K(p_t \parallel \text{rchop}_{n-j}(c_{t-1})). \end{aligned}$$

For the last two blocks of the ciphertext, one keeps only the leftmost  $j$  bits of  $c_{t-1}$  and  $n$  bits of  $c_t$ . This variant has the disadvantage that the last block needs to be decrypted before the one but last block.

It turns out that the common padding methods are vulnerable to side channel attacks that require chosen ciphertexts: an attacker who can submit ciphertexts of her choice to a decryption oracle can obtain information on the plaintext by noting whether or not an error message is returned stating that the padding is incorrect. This was first pointed out for symmetric encryption by Vaudenay in [24]; further results on concrete padding schemes can be found in [8, 9, 23]. The specific choice of the padding rule makes a difference: for example, the simple padding rule described in the introduction seems less vulnerable. Moreover, the implementation can to some extent preclude these attacks, for example by interrupting the session after a few padding errors. However, the preferred solution is the use of authenticated encryption.

### 3 The Output FeedBack (OFB) mode

The OFB mode transforms a block cipher into a synchronous stream cipher. This mode uses only the encryption operation of the block cipher. It consists of a finite state machine, which is initialized with an  $n$ -bit initial value or  $s_0 = IV$ . The state is encrypted and the encryption result is used as key stream and fed back to the state (see also Fig. 3):

$$s_i = E_K(s_{i-1}) \quad \text{and} \quad c_i = p_i \oplus s_i, \quad i = 1, 2, \dots$$

Treating an incomplete last block in the OFB mode is very simple: one selects the leftmost  $m$  bits of the last key word. The OFB mode can also be applied when the strings  $p_i$  and  $c_i$  consist of  $m < n$  bits; in that case one uses only the  $m$  leftmost bits of each key word  $s_i$ . This results in a performance penalty with a factor  $n/m$ .

It is essential for the security of the OFB mode that the key stream does not repeat. It can be shown that the average period equals  $n \cdot 2^{n-1}$  bits [14] and that the probability that an  $n$ -bit state lies on a cycle of length  $< c$  is equal to  $c/2^n$ . This implies that after  $2^{n/2}$   $n$ -bit blocks one can distinguish the output of the OFB mode from a random string (in a random string one expects to see repetitions of  $n$ -bit blocks after  $2^{n/2}$  blocks as a consequence of the birthday paradox, but it is highly unlikely that such repetitions occur in an OFB key stream). This suggests that one should rekey the OFB mode after  $\alpha \cdot 2^{n/2}$   $n$ -bit blocks for a small constant  $\alpha$ . A repetition could also be induced in a different way: if  $IV$  is chosen uniformly at random for every message, the birthday paradox implies that  $IV$  values will repeat with high probability after approximately  $2^{n/2}$  messages. The impact of such a repetition

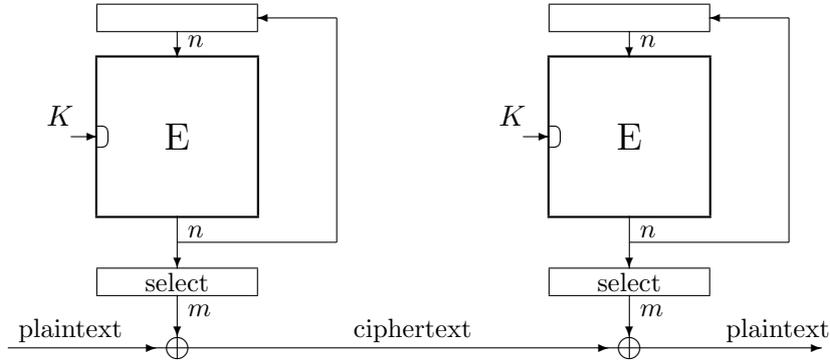


Figure 3: The  $m$ -bit OFB mode of an  $n$ -bit block cipher

is dramatic, since it will leak the sum of all the plaintext blocks of the two messages encrypted with this  $IV$  value (for simplicity it is assumed here that all messages have equal length).

The main advantage of the OFB mode is that it has no error propagation: errors in the  $i$ th ciphertext bit will only affect the  $i$ th plaintext bit. The OFB mode does not allow for parallelism or random access.

It can be shown that the OFB mode is secure against chosen plaintext attacks if the block cipher is secure in the sense that it is hard to distinguish it from a random permutation. The proof requires that one changes the key after  $\alpha \cdot 2^{n/2}$   $n$ -bit blocks for small  $\alpha$  (say  $10^{-3}$ ).

Note that an early draft of [12] included a variant of the OFB mode where only  $m < n$  bits were fed back to the state, which acted as a shift register. However, this variant of the OFB mode has an average period of about  $n \cdot 2^{n/2}$  bits [11]. This variant was removed because of this weakness.

## 4 The Counter (CTR) mode

The CTR mode is another way to transform a block cipher into a synchronous stream cipher. As the OFB mode, this mode only uses the encryption operation of the block cipher. It consists of a finite state machine, which is initialized with an  $n$ -bit integer  $IV$ . The state is encrypted to obtain the key stream; the state is updated as a counter  $\text{mod } 2^n$  (see also Fig. 4):

$$c_i = p_i \oplus E_K(\langle (IV + i) \text{ mod } 2^n \rangle), \quad i = 1, 2, \dots$$

The mapping  $\langle . \rangle$  converts an  $n$ -bit integer to an  $n$ -bit string. The processing of an incomplete final block or of shorter blocks is the same as for the OFB mode.

The period of the key stream is exactly  $n \cdot 2^n$  bits. This implies that after  $2^{n/2}$   $n$ -bit blocks one can distinguish the output of the CTR mode from a random string (as for the OFB mode). This suggests that one should rekey the CTR mode after  $\alpha \cdot 2^{n/2}$   $n$ -bit blocks for a small constant  $\alpha$ . A repeating value of  $IV$  has the same risks as for the OFB mode.

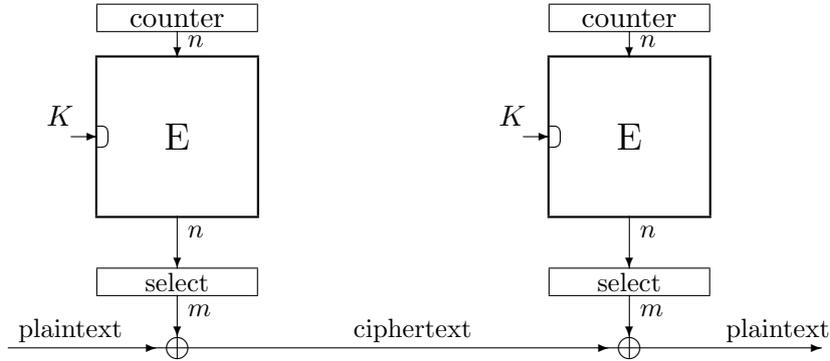


Figure 4: The  $m$ -bit CTR mode of an  $n$ -bit block cipher

As the OFB mode, the CTR mode has no error propagation. Moreover the CTR mode allows for parallelism and for random access in both encryption and decryption.

It can be shown that the CTR mode is secure against chosen plaintext attacks if the block cipher is secure in the sense that it is hard to distinguish it from a random permutation [3]. Again it is recommended to change the key after  $\alpha \cdot 2^{n/2}$   $n$ -bit blocks for small  $\alpha$  (say  $10^{-3}$ ).

## 5 The Cipher FeedBack (CFB) mode

The CFB mode transforms a block cipher into a self-synchronizing stream cipher. As the OFB and CTR mode, this mode only uses the encryption operation of the block cipher. It consists of a finite state machine, which is initialized with an  $n$ -bit initial value  $s_0 = IV$ . The state is encrypted and the leftmost  $m$  bits of the result are added to the  $m$ -bit plaintext block; the resulting ciphertext is fed back to the state (see also

Fig. 5):

$$\begin{aligned} c_i &= p_i \oplus \text{lchop}_m(E_K(s_{i-1})), \\ s_i &= \text{lchop}_{n-m}(s_{i-1}) \parallel c_i, \quad i = 1, 2, \dots \end{aligned}$$

Treating an incomplete last block in the CFB mode is very simple: one selects the required number of bits from the output of the block cipher. The CFB mode is a factor  $n/m$  times slower than the CBC mode, since only  $m$  bits are used per encryption operation. In practice one often uses  $m = 1$  and  $m = 8$ ; this results in a significant speed penalty.

It can be shown that the CFB mode is secure against chosen plaintext attacks if the block cipher is secure in the sense that it is hard to distinguish it from a random permutation. A matching ciphertext attack also applies to the CFB mode (cf. Sect. 2) [19]; the analysis is more complex since one can now consider  $n$ -bit ciphertext blocks which are shifted over  $m$  positions. To preclude leakage of information on the plaintexts one needs to impose that the number  $q$  of  $m$ -bit ciphertext blocks to which an opponent has access satisfies  $q \ll 2^{(n+1)/2}$  or  $q = \alpha \cdot 2^{n/2}$  where  $\alpha$  is a small constant (say  $10^{-3}$ ). If this

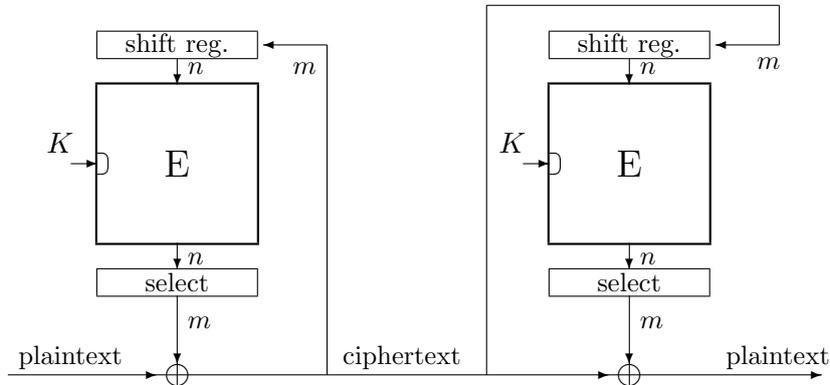


Figure 5: The  $m$ -bit CFB mode of an  $n$ -bit block cipher

limit is reached, one needs to change the key.

The CFB decryption has a limited error propagation: errors in the  $i$ th ciphertext block will be copied into the  $i$ th plaintext block; about  $n$  subsequent plaintext bits will be completely garbled, since the error will stay for  $n/m$  steps in the state register  $s$ . From then on the decryption will recover. Moreover, if a multiple of  $m$  bits of the ciphertext are lost, synchronization will return as soon as  $n$  consecutive correct ciphertext bits have been received. Particularly when  $m = 1$ , this is very attractive, since this allows for a recovery after loss of an arbitrary number of bits. The CFB decryption allows for random access and parallel processing, but the encryption process is serial.

ISO/IEC 10116 [16] specifies two extensions of the CFB mode: a first extension allows to encrypt plaintext blocks of length  $m' < m$ ;  $m - m'$  '1' bits are then prepended to the ciphertext  $c_i$  before feeding it back to the state. This mode offers a better speed, but increases the risk of a matching ciphertext attack. For example, if  $n = 64$ ,  $m = 8$ , and  $m' = 7$ , one expects repetitions of the state after  $2^{28}$  blocks, since the 64-bit

state always contains eight '1' bits. A second extension allows for a larger state  $s$  (for example of  $r \cdot n$  bits). This allows for parallel processing (with  $r$  processors) in the CFB encryption, at the cost of  $r$  IVs, a delayed error propagation and a slower synchronization.

Yet another variant of the CFB mode [1] improves the efficiency by using all the bits of  $E_K(s_{i-1})$ . A new encryption is only calculated if all bits of the  $n$ -bit block have been used or if a specific pattern of fixed length is observed in the ciphertext. The latter property allows resynchronization: the shorter the pattern, the faster the resynchronization, but the slower the performance.

## 6 Other Modes of Operation

In the early 1990s, modes for multiple encryption of DES were analyzed. The simplest solution is to replace DES by triple-DES and to use triple-DES in one of the five modes discussed above. For triple-DES, these solutions are known as the *outer modes* [17]. However, their disadvantage is that one

can only encrypt  $\alpha \cdot 2^{n/2}$  blocks with a single key for small  $\alpha$  (for example due to matching ciphertext attacks on CBC and CFB mode). This motivated research on *inner modes*, also known as interleaved or combined modes, where the modes themselves are considered as primitives (*e.g.*, inner-CBC for triple-DES consists of three layers of single-DES in CBC mode). Biham has analyzed all the 36 double and 216 triple interleaved modes [4, 5], where each layer consists of ECB, OFB, CBC, CFB and the inverses of CBC and CFB. His goal is to recover the secret key (total break). He notes that by allowing chosen plaintext and chosen ciphertext attacks, “*all triple modes of operation are theoretically not much more secure than a single encryption.*” The most secure schemes in this class require for DES  $2^{67}$  chosen plaintexts or ciphertexts,  $2^{75}$  encryptions, and  $2^{66}$  storage. Biham also proposes a small set of triple modes, where a single key stream is generated in OFB mode and XORed before every encryption and after the last encryption and a few quadruple modes [4] with a higher conjectured security. However, Wagner has shown that if one allows chosen ciphertext/chosen *IV* attacks, the security of all but two of these improved modes with DES can be reduced to  $2^{56}$  encryptions and between 2 and  $2^{32}$  chosen *IV* texts [25]. A further analysis of the influence of the constraints on the *IV*s has been provided by Handschuh and Preneel [15]. The ANSI X9.52 standard [2] has opted for the outer modes of triple-DES. Coppersmith *et al.* propose the CBCM mode [10], which is a quadruple mode; this mode has also been included in ANSI X9.52. Biham and Knudsen present a certification attack on this mode with DES requiring  $2^{65}$  chosen ciphertexts and memory that requires  $2^{58}$  encryptions [6]. In conclusion, one can state

that it seems possible to improve significantly over the matching ciphertext attacks. However, the security results strongly depend on the model, security proofs have not been found so far and the resulting schemes are rather slow. It seems more appropriate to upgrade DES to AES.

A second area of research is on how to encrypt plaintexts from finite sets, which are not necessarily of size  $2^n$ ; this problem is partially addressed by Davies and Price in [11]; a formal treatment has been developed by Black and Rogaway in [7].

## References

- [1] A. Alkassar, A. Gerald, B. Pfitzmann, A.-R. Sadeghi, “Optimized self-synchronizing mode of operation,” *Fast Software Encryption, Lecture Notes in Computer Science 2355*, M. Matsui, Ed., Springer-Verlag, 2002, pp. 78–91.
- [2] ANSI X9.52, “*Triple Data Encryption Algorithm Modes of Operation*,” 1998.
- [3] M. Bellare, A. Desai, E. Jorjipii, P. Rogaway, “A concrete security treatment of symmetric encryption,” *Proceedings 38th Annual Symposium on Foundations of Computer Science, FOCS '97* IEEE Computer Society, 1997, pp. 394–403.
- [4] E. Biham, “Cryptanalysis of triple-modes of operation,” *Technion Technical Report CS0885*, 1996.
- [5] E. Biham, “Cryptanalysis of triple modes of operation,” *Journal of Cryptology*, Vol. 12, No. 3, 1999, pp. 161–184.
- [6] E. Biham, L.R. Knudsen, “Cryptanalysis of the ANSI X9.52 CBCM mode,” *Journal of Cryptology*, Vol. 15, No. 1, 2002, pp. 47–59.

- [7] J. Black, P. Rogaway, "Ciphers with arbitrary finite domains," *Topics in Cryptology – CT-RSA 2002, LNCS 2271*, B. Preneel, Ed., Springer-Verlag, 2002, pp. 114–130.
- [8] J. Black, H. Urtubia, "Sidechannel attacks on symmetric encryption schemes: the case for authenticated encryption," *Proceedings of the 11th USENIX Security Symposium*, 2002, pp. 327–338.
- [9] B. Canvel, A. P. Hiltgen, S. Vaudenay, M. Vuagnoux, "Password interception in a SSL/TLS channel," *Advances in Cryptology, Proceedings Crypto'03, LNCS 2729*, D. Boneh, Ed., Springer-Verlag, 2003, pp. 583–599.
- [10] D. Coppersmith, D.B. Johnson, S.M. Matyas, "A proposed mode for triple-DES encryption," *IBM Journal of Research and Development*, Vol. 40, No. 2, 1996, pp. 253–262.
- [11] D.W. Davies, W.L. Price, "*Security for Computer Networks. An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer*," (Second Edition), Wiley, 1989.
- [12] FIPS 81, "*DES Modes of Operation*," Federal Information Processing Standard (FIPS), Publication 81, National Bureau of Standards, US Department of Commerce, Washington D.C., December 1980.
- [13] FIPS 197, "*Advanced Encryption Standard (AES)*," Federal Information Processing Standard (FIPS), Publication 197, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., November 26, 2001.
- [14] P. Flajolet, A.M. Odlyzko, "Random mapping statistics," *Advances in Cryptology, Proceedings Eurocrypt'99, LNCS 1592*, J. Stern, Ed., Springer-Verlag, 1999, pp. 329–354.
- [15] H. Handschuh, B. Preneel, "On the security of double and 2-key triple modes of operation," *Fast Software Encryption, LNCS 1636*, L.R. Knudsen, Ed., Springer-Verlag, 1999, pp. 215–230.
- [16] ISO/IEC 10116, "*Information technology – Security techniques – Modes of operation of an  $n$ -bit block cipher algorithm*," IS 10116, 1991.
- [17] B. S. Kaliski, M.J.B. Robshaw, "Multiple encryption: Weighing security and performance," *Dr. Dobb's Journal*, January 1996, pp. 123–127.
- [18] L. Knudsen, "*Block Ciphers – Analysis, Design and Applications*," PhD thesis, Aarhus University, Denmark, 1994.
- [19] U.M. Maurer, "New approaches to the design of self-synchronizing stream ciphers," *Advances in Cryptology, Proceedings Eurocrypt'91, LNCS 547*, D.W. Davies, Ed., Springer-Verlag, 1991, pp. 458–471.
- [20] A. Menezes, P.C. van Oorschot, S. Vanstone, "*Handbook of Applied Cryptography*," CRC Press, 1998.
- [21] C.H. Meyer, S.M. Matyas, "*Cryptography: A New Dimension in Data Security*," Wiley & Sons, 1982.
- [22] NIST, "*SP 800-38A Recommendation for Block Cipher Modes of Operation – Methods and Techniques*," December 2001.
- [23] K.G. Paterson, A. Yau, "Padding oracle attacks on the ISO CBC mode encryption standard," *Topics in Cryptology – The Cryptographers' Track at the RSA Conference, LNCS 2964*, T. Okamoto, Ed., Springer-Verlag, 2004, pp. 305–323.

- [24] S. Vaudenay, “Security flaws induced by CBC padding – Applications to SSL, IPSEC, WTLS ...,” *Advances in Cryptology, Proceedings Eurocrypt’02, LNCS 2332*, L. Knudsen, Ed., Springer-Verlag, 2002, pp. 534–546.
- [25] D. Wagner, “Cryptanalysis of some recently-proposed multiple modes of operation,” *Fast Software Encryption, LNCS 1372*, S. Vaudenay, Ed., Springer-Verlag, 1998, pp. 254–269.