Strengthened Encryption in the CBC Mode

Vlastimil Klíma [1] and Tomáš Rosa [1,2]

[1] ICZ, V Olšinách 75, 100 97 Prague 10, Czech Republic, http://www.i.cz
[2] Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo náměstí 13, 121 35 Prague 2, Czech Republic
{vlastimil.klima, tomas.rosa}@i.cz

May 24, 2002

Abstract
Vaudenay [1] has presented an attack on the CBC mode of block ciphers, which uses padding according to the PKCS#5 standard. One of the countermeasures, which he has assumed, consisted of the encryption of the message *M´= M || padding || hash(M || padding)* instead of the original *M*. This can increase the length of the message by several blocks compared with the present padding. Moreover, Wagner [1] showed a security weakness in this proposal. The next correction, which Vaudenay proposed ("A Fix Which May Work") has a general character and doesn't solve practical problems with the real cryptographic interfaces used in contemporary applications. In this article we propose three variants of the CBC mode. From the external point of view they behave the same as the present CBC mode with the PKCS#5 padding, but they prevent Vaudenay's attack.

Category / Keywords: secret-key cryptography / block ciphers, block-cipher modes, CBC, side-channel, modes of operation, PKCS#5 padding, implementation, cryptoAPI

Introduction

Vaudenay's attack [1] on the CBC mode of block ciphers with the PKCS#5 padding [2] uses the information, which states, whether the deciphered text had the correct padding. One of Vaudenay's proposals consisted of the encryption of the message *M´= M || padding || hash(M || padding)* instead of *M*, but he rejected it because of the theoretical weakness. From the practical point of view, the disadvantage of this proposal is, in particular, that during encryption of the last plaintext block several cipher text blocks arise, instead of 1 or 2 blocks as at present. This noticeably disrupts the semantics of contemporary cryptographic interfaces, for instance CryptoAPI [3].

The present CBC mode (according to the common semantics of programming interfaces) works as follows: If a part of the plaintext is encrypted, a cryptographic device will always return one ciphertext block for each plaintext block. There is the only one exception and this is the encryption of the last block of the plaintext. In this case one or two ciphertext blocks can be returned (depending on the length of the last block). Decryption works in the reverse order: A cryptographic device returns one plaintext block for each ciphertext block, except the decryption of the last block. After decryption of the last plaintext block, its padding is determined, cut off and the valid plaintext is returned. The characteristic of the PKCS#5 padding is that the information, which part of the plaintext has to be cut off, is determined from and only from the last ciphertext block (this would be disrupted by Vaudenay's

proposal). Based on this principal the cryptographic interfaces were built and therefore they will not work if it is violated.*)

The goal of this contribution is to design an encryption for the last plaintext block, which respects the semantics of the widely used PKCS#5 padding (thereby preserving compatibility with the usual cryptographic interfaces) and at the same time prevents Vaudenay's attack. It is still possible that some systems do not enable the implementation of some proposed variants of encryption because of new requirements for working with key material. However we have paid the attention to minimizing the number of such systems.

This article does not deal in any way with the alternative definition of padding. We state that our goal was to design countermeasures, which are **practically usable**. That means: *They obviously eliminate Vaudenay's attack, they do not introduce other evidently practically exploitable weaknesses and they clearly do not violate the semantics of contemporary cryptographic interfaces.* The analysis of the theoretical characteristics of the proposed variants is an open question.

Example

According to Vaudenay's proposal, if a 7 bytes long message *M* is encrypted using 3DES and SHA-1, we have to encrypt 7 bytes of *M*, 5 bytes of the *padding* and 20 bytes of the hash value, which creates 4 blocks (32 bytes). The cryptographic interface would then obtain a request for an encryption of 7 bytes of *M* (marked as the last block) and it would return four ciphertext blocks to the calling application. However contemporary interfaces designed according to PKCS#5 expect to receive only one ciphertext block in such a situation. Similar problems arise during decryption. It has been practically verified, that the introduction of this type of padding into the subsystem CryptoAPI makes common applications crash.

New proposals for strengthened encryption of the last block in the CBC mode

We propose three variants for strengthened encryption of the last plaintext block in the CBC mode, which are compatible with contemporary cryptographic interfaces, including Crypto API [3]. We assume variant A as the ”middle”. Its "stronger" version is variant B and the "weaker" one is variant C. The security of these variants is estimated only heuristically in a short time after the presentation of the attack. An in depth theoretical analysis of their security remains an open problem. The variants reflect the capabilities of a designer to use different cryptographic tools.

Classical CBC encryption of plaintext by a block cipher with key *K1* is described by the following equations:

Let us denote plaintext: $x_1, x_2, \ldots x_N$, ciphertext: $y_1, y_2, \ldots y_N$, initialisation value: IV, encryption key: $K1$.

Encryption: $y_1 = E_{K1}(IV \oplus x_1)$, $y_i = E_{K1}(y_{i-1} \oplus x_i)$, $i = 2, \ldots, N$

Decryption: $x_1 = IV \oplus D_{K1}(y_1)$, $x_i = y_{i-1} \oplus D_{K1}(y_i)$, $i = 2, \ldots, N$

Strengthened encryption of the last block uses the present definition of the PKCS#5 padding. It means that x_N is the block padded according to PKCS#5. All three variants encrypt all

*) There can be situations where the system receives the ciphertext blocks consequently and the fact, that a block is the last block of the whole message, will be recognized after receiving this block, not before. In Vaudenay's original proposal after cutting the padding the system would have to take back several plaintext blocks, which in some cases would not be possible.

plaintext blocks $x_1, x_2, ... x_{N-1}$ (excluding the last block x_N) in the CBC mode with the key *K1* in the usual way, i.e.
Encryption: $y_1 = E_{K1}(IV \oplus x_1)$, $y_i = E_{K1}(y_{i-1} \oplus x_i)$, $i = 2, ..., N-1$ and
Decryption: $x_1 = IV \oplus D_{K1}(y_1)$, $x_i = y_{i-1} \oplus D_{K1}(y_i)$, $i = 2, ..., N-1$.
The only difference is in the equation for the encryption and decryption of the last block, which we will describe now.

Using the function *PBKDF2* from the standard PKCS#5 [2] and three different values of the salt (SALT) and three counters (COUNT) we derive three different values of keys *K2*, *K3* and *K4*:
K2 = PBKDF2(K1, SALT1, COUNT1, dklen),
K3 = PBKDF2(K1, SALT2, COUNT2, dklen),
K4 = PBKDF2(K1, SALT3, COUNT3, dklen),
where dklen is the length of keys *K2*, *K3* and *K4*, values *SALT1*, *SALT2* and *SALT3* are different constants and *COUNT1*, *COUNT2* and *COUNT3* are other constants, chosen according to the standard PKCS#5.

Strengthened encryption - variant A
We define the equation for the last block in the following way:
Encryption: $y_N = E_{K4}(D_{K2}(x_N) \oplus E_{K3}(y_{N-1}))$
Decryption: $x_N = E_{K2}(E_{K3}(y_{N-1}) \oplus D_{K4}(y_N))$
The goal of this type of encryption is that the influence of the variables (x_N, y_{N-1}, y_N) affecting encryption and decryption of the last block, is indirect, non-linear and "masked" by transformations, unknown to the attacker. The keys *K2*, *K3* and *K4* must have these properties:

- They are derived from key *K1* by a one-way function.
- It is impossible to determine the individual keys *K2*, *K3* or *K4* from the remaining two.

These additional keys are introduced in order to prevent an attacker from deducing any information about E_{Ki} and D_{Ki}, for $i = 2, 3, 4$, from eventual knowledge of the behaviour of the transformation E_{K1} on many pairs of plaintext-ciphertext blocks. Note that it is possible to define other kinds of derivations of the keys *K2*, *K3* and *K4*, which are different from those defined in PKCS#5. It is only necessary to preserve the properties mentioned above.

Strengthened encryption - variants B1 and B2
According to the designer's capability to use the hash function we propose variants B1 and B2. Both are designed in such a way, that the feedback from the penultimate ciphertext block is a one-way function of the variable y_{N-1}. This property is guaranteed by the value $E_{K3}(y_{N-1}) \oplus y_{N-1}$ in the B1 variant and by the value $h(E_{K3}(y_{N-1}))$ in the B2 variant. Derivation of the keys *K2*, *K3* and *K4* is the same as in the variant A. Note that we will use only the *n* most significant bits from the hash function output, where *n* is the length of the block of the block cipher.
The equation for the last block is defined in the following way:
The variant B1:
Encryption: $y_N = E_{K4}(D_{K2}(x_N) \oplus E_{K3}(y_{N-1}) \oplus y_{N-1})$
Decryption: $x_N = E_{K2}(E_{K3}(y_{N-1}) \oplus y_{N-1} \oplus D_{K4}(y_N))$
The variant B2:
Encryption: $y_N = E_{K4}(D_{K2}(x_N) \oplus h(E_{K3}(y_{N-1}))$
Decryption: $x_N = E_{K2}(h(E_{K3}(y_{N-1})) \oplus D_{K4}(y_N))$

Strengthened encryption - the variant C

This variant is proposed as the "minimal" variant for the case where the designer does not have the possibility of using other transformations than E_{K1} and D_{K1}, i.e. he/she has no possibility to derive new keys from key *K1* and he/she has no possibility to use a hash function.

The equation for the last block is defined in the following way:

Encryption: $y_N = E_{K1}(D_{K1}(x_N) \oplus E_{K1}(y_{N-1}) \oplus y_{N-1})$

Decryption: $x_N = E_{K1}(E_{K1}(y_{N-1}) \oplus y_{N-1} \oplus D_{K1}(y_N))$

The brief analysis

From the general attack point of view we can assume all presented variants as an application of the two modes. Blocks $x_1, x_2, \ldots x_{N-1}$ are encrypted using the first mode and block x_N is encrypted using the second mode. There is no change in the case of encryption of the blocks $x_1, x_2, \ldots x_{N-1}$ - it is the original CBC mode. Therefore the proposed variants do not impose new weaknesses here. The encryption of the last block can be assumed also as the CBC mode, the initialisation value of which is derived pseudorandomly from the penultimate ciphertext block y_{N-1}. In this way the dependency on the original *IV* is preserved.

From the point of view of defence against Vaudenay's attack, it is natural to use the notion of the confirmation oracle [4], which is a useful tool in the study of side channels. We have used it for the fault attacks on RSA-KEM in [4]. In the case of the CBC mode the confirmation oracle has the form of a decryption engine, which the attacker sends chosen ciphertexts to. The engine accepts or refuses the given ciphertext according to whether the last plaintext block has the correct padding or not. We assume that an attacker has the possibility of obtaining information about the acceptance or refusal of the last block. Therefore he/she has access to a confirmation oracle, which allows him/her to confirm whether the last block of the decrypted plaintext has correct padding or not.

Let us denote *PAD* the set of allowed paddings according to PKCS#5. In the case of the classical CBC mode with the PKCS#5 padding, it holds $x_N = D_{K1}(y_N) \oplus y_{N-1}, x_N \in PAD$. Using the confirmation oracle it is possible to confirm the validity of this relation for an arbitrarily chosen y_N and y_{N-1}. With respect to the definition of the set *PAD* and with respect to the way in which the value y_{N-1} enters the expression, the transformation E_{K1} can be easily inverted using the confirmation oracle. That is exactly what Vaudenay has shown in his article [1].

Now, let us look at the relations (note that the statement $x_N \in PAD$ is their crucial part), which can be confirmed in our variants of a strengthened encryption.

A) $x_N = E_{K2}(E_{K3}(y_{N-1}) \oplus D_{K4}(y_N)), x_N \in PAD$

B1) $x_N = E_{K2}(E_{K3}(y_{N-1}) \oplus y_{N-1} \oplus D_{K4}(y_N)), x_N \in PAD$

B2) $x_N = E_{K2}(h(E_{K3}(y_{N-1})) \oplus D_{K4}(y_N)), x_N \in PAD$

C) $x_N = E_{K1}(E_{K1}(y_{N-1}) \oplus y_{N-1} \oplus D_{K1}(y_N)), x_N \in PAD$

The influence of y_{N-1} on $x_N = D_{K1}(y_N) \oplus y_{N-1}$ is in the original CBC mode "direct and non-masked". In the proposed variants the variables y_{N-1} and y_N always act indirectly and via non-linear transformations, unknown to an attacker. Thus from the confirmation oracle an attacker could obtain only information about a relation among unknown images of input variables. Moreover, except for variant A, the input variable y_{N-1} goes through a one-way function. This prevents the attacker preparing special values for a test in the case, where he/she has partial

knowledge about the transformation $E_{K3}(y_{N-1})$ or $E_{K1}(y_{N-1})$. In this way defence against Vaudenay's attack is ensured.

Conclusion

Vaudenay has described a practical attack on the CBC mode based on a fault side channel. The correction, which Vaudenay proposed has a general character and doesn't solve practical problems with the real cryptographic interfaces used in contemporary applications. In this contribution we have presented practical countermeasures, which are semantically compatible with current cryptographic interfaces. On the basis of the above brief analysis we presume that the proposed variants are not vulnerable to attacks of the Vaudenay type. Their theoretical security is an open problem. We suggest considering and implementing them in the order B2, B1, A, C.

References

[1] Vaudenay, S.: Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS, … , Eurocrypt 2002, pp. 534 - 545

[2] PKCS#5 v. 2.0: Password-Based Cryptography Standard, RSA Laboratories, March 25, 1999, *http://www.rsasecurity.com/rsalabs/pkcs/pkcs-5/index.html*

[3] Microsoft: MSDN Library - July 2001, Platform SDK Documentation, Security, Cryptography, 2001

[4] Klíma, V., Rosa, T.: Further Results and Considerations on Side Channel Attacks on RSA, available on http://eprint.iacr.org/2002/071/, final version is to be publlished in proceedings of CHES 2002