

With CBC done on the inside, any change to ciphertext block n completely and unpredictably garbles all plaintext blocks from n to the end of the message. This makes CBC done on the inside more secure, and perhaps would therefore have been a better choice. However, sometimes people would prefer if garbling of a ciphertext block did not garble the entire rest of the message. They'd prefer that the encryption scheme be **self-synchronizing**, which means that after some small number of garbled blocks, the plaintext will start decrypting properly again. There are also subtle security flaws with CBC on the inside if the attacker can supply chosen plaintext and IV and examine the output.

Another advantage of CBC on the inside is performance. With CBC on the inside it is possible to use three times as much hardware and pipeline the encryptions so that it is as fast as single encryption. With CBC on the outside, this is not possible.

One reason that people choose CBC on the outside despite its disadvantages is that EDE encryption can be considered a new secret key block encryption scheme that uses a 112-bit key. This can then be used with any of the chaining methods (OFB, ECB, CFB, CTR, as well as CBC).

4.5 HOMEWORK

1. What pseudo-random block stream is generated by 64-bit OFB with a weak DES key?
2. The pseudo-random stream of blocks generated by 64-bit OFB must eventually repeat (since at most 2^{64} different blocks can be generated). Will $K\{IV\}$ necessarily be the first block to be repeated?
3. Let's assume you do DES double encryption by encrypting with K_1 and doing DES in decrypt mode with K_2 . Does the same attack work as with double encryption with K_1 and K_2 ? If not, how could it be made to work?
4. What is a practical method for finding a triple of keys that maps a given plaintext to a given ciphertext using EDE? Hint: It is like the meet-in-the-middle attack of §4.4.1.2 *Encrypting Twice with Two Keys*.
5. Let's assume that someone does triple encryption by using EEE with CBC on the inside. Suppose an attacker modifies bit x of ciphertext block n . How does this affect the decrypted plaintext?
6. Consider the following alternative method of encrypting a message. To encrypt a message, use the algorithm for doing a CBC decrypt. To decrypt a message, use the algorithm for doing

a CBC encrypt. Would this work? What are the security implications of this, if any, as contrasted with the "normal" CBC?

ably
side
ople
ey'd
um-
ecu-
ine

ssi-
gle

DE
ey.
).



u're working

Algorithm was
ly reasonable
message—has
ng security of
Then the best

to be secure if
MD4 Message

, x) for data x ,
nputs y , with y

roperties) pro-
Although there
fered the same
ndard.

the key to the
s the extension
detail, HMAC
ted-size output
h 0 bits to 512
128 bits or 160
to 512 bits. It

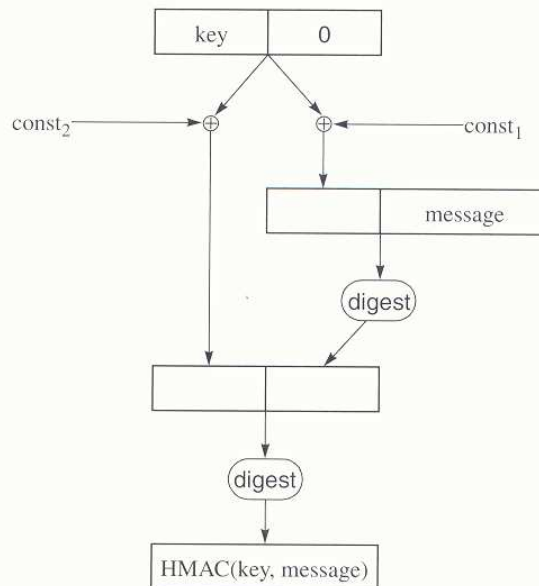


Figure 5-10. HMAC

then \oplus s the padded key with a constant string of octets of value 36_{16} , concatenates it with the message to be protected and computes a message digest. It \oplus s the padded key with a different constant string of octets of value $5c_{16}$, concatenates that with the result of the first digest, and computes a second digest on the result.

5.8 HOMEWORK

1. Doing a signature with RSA alone on a long message would be too slow (presumably using cipher block chaining). Suppose we could do division quickly. Would it be reasonable to compute an RSA signature on a long message by first finding what the message equals, mod n , and signing that?
2. Message digests are reasonably fast, but here's a much faster function to compute. Take your message, divide it into 128-bit chunks, and \oplus all the chunks together to get a 128-bit result. Do the standard message digest on the result. Is this a good message digest function?

12. Assume a good 128-bit message digest function. Assume there is a particular value, d , for the message digest and you'd like to find a message that has a message digest of d . Given that there are many more 2000-bit messages that map to a particular 128-bit message digest than 1000-bit messages, would you theoretically have to test fewer 2000-bit messages to find one that has a message digest of d than if you were to test 1000-bit messages?
13. Why do we expect that a randomly chosen 100-bit number will have about the same number of 1 bits and 0 bits? (For you statistics fans, calculate the mean and standard deviation of the number of 1 bits.)
14. For purposes of this exercise, we will define **random** as having all elements equally likely to be chosen. So a function that selects a 100-bit number will be random if every 100-bit number is equally likely to be chosen. Using this definition, if we look at the function "+" and we have two inputs, x and y , then the output will be random if at least one of x and y are random. For instance, y can always be 51, and yet the output will be random if x is random. For the following functions, find sufficient conditions for x , y , and z under which the output will be random:
- $\sim x$
 - $x \oplus y$
 - $x \vee y$
 - $x \wedge y$
 - $(x \wedge y) \vee (\sim x \wedge z)$ [the selection function]
 - $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ [the majority function]
 - $x \oplus y \oplus z$
 - $y \oplus (x \vee \sim z)$
15. Prove that the function $(x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$ and the function $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ are equivalent. (Sorry—this isn't too relevant to cryptography, but we'd stumbled on two different versions of this function in different documentation and we had to think about it for a bit to realize they were the same. We figured you should have the same fun.)
16. We mentioned in §5.2.2 *Computing a MAC with a Hash* that using $\text{MD4}(K_{AB}|m)$ as a MAC is not secure. This is not a problem if MD2 is used instead of MD4. Why is that the case?
17. In §5.2.3.1 *Generating a One-Time Pad*, we generate a pseudo-random stream of MD-sized blocks. This stream must eventually repeat (since only $2^{\text{MD-size}}$ different blocks can be generated). Will the first block necessarily be the first to be repeated? How does this compare to OFB (see Chapter 4 *Modes of Operation* Homework Problem 2)?
18. How do you decrypt the encryption specified in §5.2.3.2 *Mixing In the Plaintext*?

19. Can you modify the encryption specified in §5.2.3.2 *Mixing In the Plaintext* so that instead of $b_i = \text{MD}(K_{AB} \| c_{i-1})$ we use $b_i = \text{MD}(K_{AB} \| p_{i-1})$? How do you decrypt it? Why wouldn't the modified scheme be as secure? (Hint: what would happen if the plaintext consisted of all zeroes?)

6

6.1 INTRODUCTION

This chapter describes algorithms work number theory in detail for an introduction things. And in the algorithm exists ing a more complex Public key take a message the same thing— ods to convert them from each other We'll describe: