*UBI/UTI 523 Cryptosystems and Cryptographic Protocols*

*Home Work Problems*

**5.2.** We consider known-plaintext attacks on block ciphers by means of an exhaustive key search where the key is k bits long. The block length counts n bits with n > k.

1. How many plaintexts and ciphertexts are needed to successfully break a block cipher running in ECB mode? How many steps are done in the worst case?
2. Assume that the initialization vector IV for running the considered block cipher in CBC mode is known. How many plaintexts and ciphertexts are now needed to break the cipher by performing an exhaustive key search? How many steps need now maximally be done? Briefly describe the attack.
3. How many plaintexts and ciphertexts are necessary, if you do not know the IV?
4. Is breaking a block cipher in CBC mode by means of an exhaustive key search considerably more difficult than breaking an ECB mode block cipher??

**5.10.** Sometimes error propagation is an issue when choosing a mode of operation in practice. In order to analyze the propagation of errors, let us assume a bit error (i.e., a substitution of a “0” bit by a “1” bit or vice versa) in a ciphertext block *yi*.

1. Assume an error occurs during the transmission in one block of ciphertext, let’s say *yi*. Which cleartext blocks are affected on Bob’s side when using the ECB mode?
2. Again, assume block *yi* contains an error introduced during transmission. Which cleartext blocks are affected on Bob’s side when using the CBC mode?
3. Suppose there is an error in the cleartext *xi* on Alice’s side. Which cleartext blocks are affected on Bob’s side when using the CBC mode?
4. Assume a single bit error occurs in the transmission of a ciphertext character in 8-bit CFB mode. How far does the error propagate? Describe exactly *how* each block is affected.
5. Prepare an overview of the effect of bit errors in a ciphertext block for the modes ECB, CBC, CFB, OFB and CTR. Differentiate between random bit errors and specific bit errors when decrypting *yi*.

**5.12.** We now analyze the security of DES double encryption (2DES) by doing a cost-estimate:

 *2DES(x) = DESK2 (DESK1 (x))*

1. First, let us assume a pure key search without any memory usage. For this purpose, the whole key space spanned by K1 and K2 has to be searched. How much does a key-search machine for breaking 2DES (worst case) in 1 week cost?

In this case, assume ASICs which can perform 107 keys per second at a cost of $5 per IC. Furthermore, assume an overhead of 50% for building the key search machine.

1. Let us now consider the meet-in-the-middle (or time-memory tradeoff) attack, in which we can use memory. Answer the following questions:
* How many entries have to be stored?
* How many bytes (not bits!) have to be stored for each entry?
* How costly is a key search in one week? Please note that the key space has to be searched before filling up the memory completely. Then we can begin to search the key space of the second key. Assume the same hardware for both key spaces.

For a rough cost estimate, assume the following costs for hard disk space: $8/10 GByte, where 1 GByte = 109 Byte.

1. Assuming Moore’s Law, when do the costs move below $1 million?