# EECE 412, Fall 2014

## Quiz #2 - Key

Your Family name: _____

Your Given name: _____

Your student ID: _____

Name of your left neighbor: _____

Name of your right neighbor: _____

| # | Points | Out of |
|---|---|---|
| 1 | | 2 |
| 2 | | 2 |
| 3 | | 2 |
| 4 | | 4 |
| 5 | | 10 |
| 6 | | 8 |
| TOTAL | | 28 |

---

**1. (2 points) State the Kerckhoff's principle (in your own words)?**

**The secrecy of a system should not rely on secrecy of its design. Its design should be open and the secrecy of the key should only be assumed.**

**2. (2 points) Explain why Kerckhoff's principle is paramount to the security of crypto-systems? Provide some examples.**

Through out the history we saw many examples where secrecy of the design lead to security compromise later. For example, GSM encryption or DRM. That is why it is important to make the design open, so that security professionals can review it.

**3. (2 points) Why is it important to reconsider design assumption every so often? Provide an example.**

**Adversarial capabilities change over time as well as the use of technologies. For example, in early days of SMTP, no one assumed that someone will use it for SPAM, hence no authentication was put in place, which, as we can see today, was false.**

**4. (4 points) Draw random oracle models for block and stream ciphers below. Describe how these ciphers work according to the models.**

In this answer a student is expected to draw something close to the following (for stream and block ciphers):

## Random Generator (Stream Cipher)

as Random Oracle

In:

short string (key)

length of the output

Queries

Responses

Out: long random stream of bits (keystream)

Applications:

Communications encryption

Storage encryption

Properties
- Should not reuse
  - Use *seed*

## Random Permutation (Block Cipher)

as Random Oracle

In

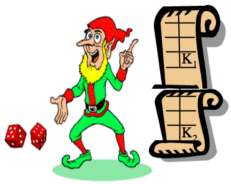
fixed size short string (plaintext) M,

DES -- 64 bits

Key K

Queries

Responses




Out

same fixed size short string (ciphertext) C

Notation
- $C = \{ M \}_K$
- $M = \{ C \}_K$

Properties
- Invertible

**Block Cipher:** In terms of block cipher, Random Oracle (RO) works as a random permutations, where, for each key K, it stores a table of plain text (P) to cipher text (C) mapping. For a given P a corresponding C can be read. If there is no such record yet, RO generates a random value C and adds P, C record to the table.
**Stream Cipher:** Is merely works as a random number generator dependent on the key.

**5) (10 points) Consider the following example: Alice encrypts a plaintext P with AES in CBC mode using key K to get a ciphertext C. Lets also limit size of P to exactly 3 blocks and, hence, C to 3 blocks too. Thus, E(P, K) = E({$P_0$, $P_1$, $P_2$}, K) = {IV, C} = {IV, $C_0$, $C_1$, $C_2$}, where $C_i$ = E($P_i$ XOR $C_{i-1}$, K). Then Alice sends C to Bob and Bob uses the same key K to decrypt the message.**

**5.1) (5 points) Demonstrate how Trudy can use this knowledge to change the C so that corresponding P changes as well.**

Considering that $P_i$ depends only on $C_{i-1}$, it is possible to change a single C block, so that, the corresponding $P_i$ changes. If Trudy knows enough about structure of the $P_i$, he can force desired changes by modifying $C_{i-1}$ in a such way that he can obtain $P_i^{desired} = D(C_i, K)\ XOR\ C_{i-1}^{modified}$

The root cause of such an attack lays in the fact chaining only combines two sequential blocks. Thus, modification of a single block leads to corruption of two blocks at most.

**5.2) (5 points) Propose and explain changes to the mode of operation of encryption that will mitigate the vulnerabilities:**

One of the approaches to defend against corruption of cipher-text is to use message authentication code (MAC) to protect integrity of the cipher text. For example, one can use an HMAC or MAC based on block cipher to generate the MAC. This will require sending it along with IV, C, i.e., Alice will have to send {IV, C, MAC}.

**6) (8 points). For each mode of operation (OFB, CTR, CBC and ECB) state it's pros and cons. For each of the modes, also, provide an example where would you use it.**

ECB is useful for encryption small pieces of information that fit into a single block, e.g., encryption another encryption keys.
CTR shines in streaming applications, e.g., audio over Bluetooth, where fast and continuous encryption is required.
CBC is good where parallel encryption is not required, e.g., storage.
OFB also useful for streaming applications. In contrast, to CTR, it cannot be computed in parallel, because $K_i$ needs all previous ($i$-1) keys to be calculated.