Private Key Encryption - One-time Pad

Q1: What does it mean to be secure against an unbounded adversary? Alternatively, what do you understand when one says perfect secrecy? Does what you understand match what the one time pad provides? Why or why not?

A1: Being secure against an unbounded adversary means that the algorithm is secure against an adversary that has infinite computational power and unlimited time. This security is also known as "unconditional security". Perfect secrecy is when the chances of guessing the message apriori is **perfectly** equal to the aposteriori probability. The emphasis on perfect here is due to the mathematical equivalence being without a margin of error. Even with the unlimited powers of the adversary, the fact that plaintext remains secret makes it sensible to use the word perfect too. OTP (One Time Pad) provides this, albeit it's drawbacks. In fact, in a private key scheme with perfect requirements, OTP is the optimal perfectly-secret scheme, also proven by Shannon's theory: $|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}|$.

Q2: What are some advantages and disadvantages of the one time pad? Why don't we use one time pad everywhere? Why do we need more research on cryptography?

A2: One Time Pad has several drawbacks. The immediately obvious one is the requirement of key length being at least equal to the message length. This is especially cumbersome if the message size is unbounded (not known apriori), and thus if parties decide to use a key they immediately bound their message length, which is not practical. Another case is the fact that this key is used only once -hence One Time-, as using a key more than once breaks perfect secrecy. Another problem is when the adversary is an "active attacker", i.e. the adversary can change the content of ciphertext. The receiver will be unable to obtain exactly the encrypted message in that case. Note that One Time Pad is a perfect cipher. Almost never in real life can we have a representation of perfect. A scheme that can be broken with probability 1 if the adversary works to break it in 500 years, is actually pretty secure! The further research in cryptography therefore approach this problem in terms of "asymptotic" or "concrete" security. Shannon already proved what is there to be proven regarding perfect schemes, and it was bad news for everyone.

Q3: Given any message m and a ciphertext c, how many one time pad keys provide $E_k(m) = c$? Explain your answer.

A3: In One Time Pad, we have $|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}|$. When we fix for some message m and ciphertext c, we can say $\exists k \in \mathcal{K}$ s.t. $k = Dec_k(c) = m$. Since OTP is perfectly secret, thus correct:

$$Dec_k(Enc_k(m)) = m$$

We see that there is only 1 key that can provide this. If more than one key could provide this correctness for a fixed message, the general correctness would be broken.

Q4: For the following encryption scheme, state whether the scheme is perfectly secret or not. Justify your answer.

- The message space is $\mathcal{M} = \{0, \ldots, 4\}$. Algorithm *Gen* chooses a uniformly random key from the key space $\{0, \ldots, 5\}$. $Enc_k(m)$ returns $[(k + m) \bmod 5]$, and $Dec_k(c)$ returns $[(c - k) \bmod 5]$.

A4: This scheme is not perfectly secret.

Proof. Assume this scheme is perfectly secret. Fix a cyphertext c. Given $k_1 = 0$ and $k_2 = 5$, there exists a message m such that $c = Enc_{k_1}(m) = Enc_{k_2}(m)$. Let there be another message $m' = m + 1 \bmod |\mathcal{M}| = (m + 1) \bmod 5$. Let there be keys k_1' and k_2' such that $Enc_{k_1'}(m) = Enc_{k_2'}(m) = c$, which was our fixed ciphertext. To ensure this, we must compensate the plus 1 by decrementing the keys: $k_1' = k_1 - 1$ and $k_2' = k_2 - 1$. We get $k_1' = 1 - 1 \bmod 6 = 5$, and $k_2' = 5 - 1 \bmod 6 = 4$. Looking at the encryption function we see that:

$$Enc_{k_1'}(m') = [(5 + m') \bmod 5] \neq Enc_{k_2'}(m') = [(4 + m') \bmod 5]$$

As a result: $Pr[C = c|M = m] \neq Pr[C = c|M = m']$, which breaks perfect indistinguishability, and therefore the perfect secrecy. □

Q5: For the following encryption scheme, state whether the scheme is perfectly secret or not. Justify your answer.

- The message space is $M = \{m \in \{0,1\}^\lambda |$ the last bit of m is $0\}$. *Gen* chooses a uniform key from $\{0,1\}^{\lambda-1}$. $Enc_k(m)$ returns the ciphertext as $[m \oplus (k||0)]$, and $Dec_k(c)$ returns the plaintext as $[c \oplus (k||0)]$.

A5: (Here $s_1||s_2$ denotes a concatenation of strings, see "Index of Common Notation".)Our message space can be thought of $M' = \{m \in \{0,1\}^{\lambda-1}$ where every message $m = m'||0$ and $m' \in M'$. So informally, a message of length $\lambda - 1$ ending with the bit 0. Our keys are also of

$\lambda - 1$ randomly, and concatenated with 0. The *Enc* and *Dec* functions are in essence the same of One Time Pad, so actually this is just One Time Pad with the message and key ending in 0. As a result, this is perfectly secret.

Q6: Using OTP, Alice and Bob agreed on a perfectly random key, and that Alice will send Bob the answer to the question : "Are you taking Comp443/543" as either Y or N encoded using string-to-bits:

$$Y \rightarrow 1011001$$

$$N \rightarrow 1001110$$

Charlie knows nothing about the message or the key but intercepts the transmission: 1001110.

What should Charlie send to Bob so that Bob thinks the opposite of Alice's answer? (If Alice indeed sent Y, Bob receives N. If Alice indeed sent N, Bob receives Y.)

A6: The adversary is defined to know the scheme and that $|\mathcal{M}| = \{Y, N\}|$. The intercepted transmission is a ciphertext $c = k \oplus m_c$ where $k \in \mathcal{K}, m_c \in \mathcal{M}$. Let us represent the message space as $\mathcal{M} = \{m, m'\}$ without necessarily forcing any of these to equal Y or N. The adversary should do as follows:

$$tmp_1 = c \oplus m \text{ and } tmp_2 = c \oplus m'$$

$$tmp_1 = k \oplus m_c \oplus m \text{ and } tmp_2 = k \oplus m_c \oplus m'$$

Let us assume without loss of generality, that $m_c = m$.

$$tmp_1 = k \oplus m \oplus m \text{ and } tmp_2 = k \oplus m \oplus m'$$

$$tmp_1 = k \text{ and } tmp_2 = k \oplus m \oplus m'$$

At this point, the adversary knows that one of these hold the key, to find out which:

$$c_{tmp_1,m} = tmp_1 \oplus m$$

$$c_{tmp_1,m'} = tmp_1 \oplus m'$$

$$c_{tmp_2,m} = tmp_2 \oplus m$$

$$c_{tmp_2,m'} = tmp_2 \oplus m'$$

The adversary will then compare these 4 results to find the one that equals to c. Again, without loss of generality, let us say $c_{tmp_1,m} = c$. Then, the adversary can infer that the key is tmp_1 and Alice sent the message m. So, the adversary can create a new ciphertext by using the same key but the other message: $c' = Enc_k(m')$. This c' should hold the opposite information. The adversary will forward c' to Bob, and the mission will be complete!