Midterm - Cryptology

Name:

November 7, 2016

Question 1, 5 points. An encryption scheme `(Gen, Enc,Dec)` over a message space $\mathcal{M}$ is perfectly secret if for every probabilistic distribution over $\mathcal{M}$, every $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ for which $Pr[C = c] > 0$ we have $Pr[M = m|C = c] = Pr[M = m]$. Prove that this definition is equivalent to requiring that for every probabilistic distribution over $\mathcal{M}$, every $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ for which $Pr[C = c] > 0$ we have that $Pr[C = c|M = m] = Pr[C = c]$.

Question 2, 5 points. Give a formal definition of the one-time pad, that is, formally specify it in terms of the algorithms `(Gen, Enc,Dec)` and its message and cipher text spaces. Prove that the one-time pad is perfectly secure.

Question 3, 5 points. Why is it difficult to use the one-time pad in practice?

Question 4, 5 points. Define CPA security. Show that no deterministic encryption scheme can be CPA secure.

Question 5, 5 points. Define the ECB and CTR modes of operation. Are they CPA secure? Why? Are they CCA secure? Why?

Question 6, 5 points. Prove or Refute: If the encryption scheme is secure against chosen-plaintext attacks, an eavesdropper cannot learn anything about the plaintext by looking at the ciphertext, but can still tell if two ciphertexts encrypt the same message by comparing them for equality.

Question 7, 5 points. Suppose you have a computer than can efficiently perform parallel computations. Consider CBC mode of operation of a block cipher. Which operation is faster encryption or decryption? Why?

Question 8, 5 points. Let F be a pseudorandom function. Explain why the following message authentication code is secure or insecure. The shared key is a random $k \in \{0,1\}^n$

- To authenticate a message $m = m_1||\cdots||m_l$, where $m_i \in \{0,1\}^n$, compute $t = F_k(m_1) \oplus \cdots \oplus F_k(m_l)$, where $\oplus$ is the bit-wise XOR function.

Question 9, 5 points. A famous cryptologist from Tacoma once proposed a cryptographic hash function. The hash function is based on the Merkle-Damgard paradigm. First define a collision-resistant compression function. The compression function takes a fixed length input message $m \in \{0,1\}^n$ and breaks it into two parts m_1 and m_2, with $|m_1| = |m_2|$ so that we have $m = m_1||m_2$. The output of the compression function $Comp(m)$ is defined as $Comp(m) = m_1 \oplus m_2$. The function $Comp(\cdot)$ is then used within the Merkle-Damgard construction to obtain a general hash function. Is this hash function secure?

Question 10, 5 points. Our famous cryptologist from Tacoma proposed a way to add authenticity to any CPA secure encryption scheme $E_k(\cdot)$. To obtain confidentiality and authenticity do as follows:

- Given a message m, first encrypt it with a pre-shared key k obtaining a cipher text $c = E_k(m)$
- Compute the hash of the ciphertext $t = hash(c)$ and send $< c, t >$ to the receiver.
- Upon receiving $< c, t >$ the receiver first check if $t = hash(c)$. If this is the case, then the receiver decrypts c with the key k and outputs the resulting message.

Does this scheme provide authenticity? Why?

Question 11, 5 points points. Show that reusing a key is a fatal mistake when encrypting with the one-time pad.

Question 12, 5 points. Explain how a key can be securely reused with a stream cipher

Question 13, 5 points. Does storing Hash(username || password) (hash of the username concatenated with the user's password) on the server better defend against an attacker who breaks into the server and tries to crack passwords than just storing Hash(password)?

Question 14, 30 points. Being a bit paranoid, you decided that Google is spying into your emails and decide to encrypt your messages before sending them as attachments to gmail messages. Propose and implement a symmetric key based cryptosystem for implementing a secure email system. Your solution should accept a text file as an input and output an encrypted text file. The encrypted text file can be copied and pasted into any email system. Your system should be based on well-known security primitives (AES,3DES, SHA1, etc.) and it should provide CCA secure authenticated communication between two parties that share one secret key. Implement your cryptosystem and test it with the King James Bible as input. How much time does it take to encrypt this file? How larger is the output file when compared to the input? Explain your design principles and why your proposed system is secure. How do you deal with replay attacks in your solution? Would you use the system you just proposed and implemented in question 1 to encrypted packets in a computer network where the packet length is about 500 bits long? Explain.

Question 15, 5 points. Suppose that you have to encrypt a large message and that this message is split in several small packets and the algorithm proposed in question 1 is used to encrypt each one of the packets before transmission. Suppose the packets have different lengths. Do you consider this system secure? Explain.