CS114 - Homework 1, part 2*

Assigned February 2nd, 2023; Due 11:59pm on March 7th, 2023

<Duc Anh Nguyen>

Note: Use one late day token and submitted on March 8th, 2023

1 Written questions {40 points}

(a) {10 points} A cryptosystem that offers *perfect secrecy* prevents an eavesdropper who observes an encrypted transmission from learning anything about the plaintext, other than its size.

Show with a counterexample that the Substitution Cipher doesn't provide perfect secrecy.

Answer: Substitution Cipher is an encryption technique that involves replacing the plaintext with a fixed set of ciphertext characters. For example, the Caesar Cipher has a simple fixed rule of shifting the letters of the alphabet by a certain number of positions. If we shift each letter of the alphabet by three positions, "a" would be replaced by "d", "b" would be replaced by "e", and so on. With this knowledge, the attacker can easily revert the ciphertext by reversing the rule of the technique. This defeats the purpose of perfect secrecy, as the attacker can learn about the plaintext just by looking at the ciphertext.

(b) {10 points} Consider the following modification to one-time pad (OTP) encryption. Rather than share a single one-time pad, Alice and Bob have shared knowledge of two pads, P_1 and P_2.

Given a plaintext M, Alice creates the ciphertext $C = M \oplus P_1 \oplus P_2$, where $\oplus$ denotes xor and $|M| = |P_1| = |P_2|$ (i.e., the size of the message and the two pads are all equal). To decrypt, Bob takes the ciphertext and xors it with P_1 and P_2; i.e., $D(C) = C \oplus P_1 \oplus P_2$.

Argue that if a one-time pad offers perfect secrecy, then the above scheme must also be perfectly secure.

Answer: One-time pad provides perfect secrecy because of the ciphertext provides no information about the plaintext to an attacker , as the pad is completely random. In this case, Alice creates the ciphertext $C = M \oplus P_1 \oplus P_2$. To decrypt this, Bob needs to perform the XOR operation of $D(C) = C \oplus P_1 \oplus P_2$. Since P_1 and P_2 are one-time pads, they are completely random and provide no information about the plaintext M. Therefore, C is also completely

*Last revised on March 8, 2023.

random and provides no information about M. Therefore, this scheme must also be perfectly secure.

(c) {5 points} Prof. Pedantic, the esteemed Ineptitude Professor of Computer Science and Quackery at Wikipedia University, is developing a new terminal program (and associated service) to log into the servers in his lab. Although he is aware of `ssh`, he refuses to use it because he doesn't like being hushed.[1] Instead, he decides to construct his own novel protocol. Like `telnet` and `ssh`, his remote console/terminal program should allow a remote user to type commands and execute them on a remote machine. Since Prof. Pedantic doesn't trust anyone — particularly the students in his introduction to network security class — he decides that all communication should be encrypted.

Prof. Pedantic decides to use the AES encryption algorithm in ECB mode. Is this a good choice? Give **two** reasons why or why not.

Answer:

- **Bad reasons:** AES encryption algorithm in ECB mode is not a good algorithm in this case because it is vulnerable to pattern analysis as identical plaintext blocks produce identical ciphertext blocks. Therefore, an attacker could potentially intercept and modify the encrypted communication without detection. Moreover, it lacks integrity protection, as encrypted blocks can be shuffled without detection, meaning an attacker can modify the encrypted data without detection.
- **Good Reasons:** As Prof. Pedantic suggested, he refuses to use other algorithms because of the hassle, which means the AES encryption algorithm is easy to implement. This algorithm involves encrypting each block of plaintext separately, which can be implemented with minimal overhead. Another potential reason that Prof. Pedantic might think it is good is because of performance; with less overhead, there is less need for computing power. However, in most cases, the performance improvement is not significant enough to outweigh the security concerns.

(d) {15 points} Prof. Pedantic designed a "secure" communication protocol for two parties (Alice and Bob) that have preshared secrets k_1 (the confidentiality key) and k_2 (the authenticity key).

Prof. Pedantic doesn't believe in traditional MACs, so he constructs his protocol as follows: to send a message m, Alice (A) sends to Bob (B) the following:

$$\begin{aligned} A \to B : \langle \quad & r, \\ & \mathrm{iv}_1, \\ & \mathrm{iv}_2, \\ & \mathrm{S}(k_1, \mathrm{iv}_1) \oplus (m \| r) \\ & \mathrm{S}(k_2, \mathrm{iv}_2) \oplus (m \| r) \quad \rangle \end{aligned}$$

where $\|$ denotes concatenation, r is a nonce (to prevent replay attacks), iv_1 and iv_2 are fresh initialization vectors (IVs), and $\mathrm{S}(k, \mathrm{iv})$ denotes a cryptographically secure pseudorandom sequence based on key k and IV iv (i.e., a stream cipher).

The professor claims that the protocol achieves *confidentiality* and *authenticity*, **as defined as follows**:

[1] Extra credit {0.0000001 points}: Explain that joke.

- *confidentiality:* an eavesdropper that observes a run of the protocol cannot learn the message m unless it knows the confidentiality key k_1; and
- *authenticity:* if Bob receives $\langle r, \mathrm{iv}_1, \mathrm{iv}_2, \mathrm{S}(k_1, \mathrm{iv}_1) \oplus (m \| r), \mathrm{S}(k_2, \mathrm{iv}_2) \oplus (m \| r) \rangle$ and r is a fresh nonce and the decryption of $\mathrm{S}(k_1, \mathrm{iv}_1) \oplus (m \| r)$ equals the decryption of $\mathrm{S}(k_2, \mathrm{iv}_2) \oplus (m \| r)$ (using the corresponding IVs and keys), then message m must have been transmitted by a party that knows both the confidentiality and authenticity keys (i.e., k_1 and k_2).

The professor's intention is that Bob obtains m by decrypting $\mathrm{S}(k_1, \mathrm{iv}_1) \oplus (m \| r)$ using key k_1 and iv_1. Further, Bob performs an authenticity check by ensuring that the decrypted message matches the decryption of $\mathrm{S}(k_2, \mathrm{iv}_2) \oplus (m \| r)$ (via key k_2 and IV iv_2). He reasons that only a sender that knows *both* k_1 and k_2 can cause the decryptions to match.

Does Prof. Pedantic's scheme achieve confidentiality and/or authenticity, as defined above? Briefly argue why or why not, for both confidentiality and authenticity. Assume that k_1 and k_2 are random 128-bit keys that have been securely shared apriori between Alice and Bob, that $k_1 \neq k_2$, and that the two IVs are also fresh.

Answer: With the scheme described above, Prof.Pedantic does achieve confidentiality as an eavesdropper who observes a run of the protocol without knowing k1 cannot learn the message. The message is confidential because it is encrypted with a secure stream cipher and each initialization vector are refresh every runs. However, the scheme does not achieve authenticity because the authenticity check that Bob performs only ensures that the decrypted message matches the decryption of $\mathrm{S}(k_2, \mathrm{iv}_2) \oplus (m \| r)$. It does not ensure that the sender is the one that have both k_1 and k_2. An attacker who intercepts the message can modify it by XORing the encrypted message with a different value.

2 Eavesdropping on unencryptedim {15 points}

Show that the `unencryptedim.py` program from Part I of Homework 1 is susceptible to eavesdropping.

To do this, you will look at a packet capture I generated while running my version of the program and write down what the password was that I transmitted between the client and server. The pcap can be found at https://www.cs.tufts.edu/comp/114/hws/hw1p2.pcap

To open the captured pcap file you can use Wireshark, which will help you visualize the stream of packets sent between the server and client. Note that Wireshark is available (for free!) on Linux, Mac OSX, and Windows. Unless you already have it, you will need to install it. Submit the password string that I typed to receive points for this section of the homework.

Answer: the string being sent is ”password=allthep oints”