CNT4406 – Network Security **Spring 2024**

Homework 4: Deadline Thursday 4/25

Instructor: Viet Tung Hoang

Typically a homework has only 200 points, but this one has 300 points. That is, you have 100 bonus points.

1. (**Password hashing**) (100 points) Suppose that Alice shares a password pw and a salt s with a server. Let H be a good cryptographic hash function, say HMAC-SHA-256. Let H^k be the hash function obtained by iterating H for $k = 10,000$ times. Alice derives a key K via $K \leftarrow H^k(s\|\text{pw})$. If she wants to send a message M to the server, Alice encrypts M with a good AE scheme and sends the ciphertext C to the server. This kind of password-based encryption is widely used in practice.

a) (30 points) Suppose that an adversary captures a ciphertext C and knows that this is an encryption of a known message M. It knows that Alice's password is in a dictionary Dict of size D, and knows her salt s too. How would the adversary recover Alice's password? How many calls to H does it need to make?

b) (30 points) Suppose that an adversary is off-path and can't capture Alice's ciphertexts, but still knows Alice's salt s. To recover Alice's password, it impersonates Alice and uses the server as a decryption oracle. In particular, it can send ciphertexts to the server to learn if the ciphertexts are valid or not. (Note that the adversary doesn't get to see the decrypted message if somehow a ciphertext is deemed valid.)

How should the adversary attack? Compare the cost of this attack with the one in part (a).

Hint: To assess the cost of an attack, you should look at both computational and communication cost.

c) (40 points) Surprisingly, in the setting of part (b), the adversary can do a much better job. (This is a real attack on the Shadowsocks VPN and many other protocols.) To perform this attack, the adversary partitions the dictionary to subsets $P_1, \ldots, P_q$. (For simplicity, assume that each partition P_i has size N, meaning $q = D/N$.) For each P_i, the adversary crafts a small ciphertext C_i that can be properly decrypted under the corresponding key of *any* password $\text{pw}^* \in P_i$. For standard AE scheme like GCM, it's rather easy to do that, with $N \approx 10,000$.

How would the adversary use this to speed up the attack in part (b)? What's the cost of this attack? (For simplicity, let's ignore the cost of generating the ciphertexts C_i.) You can assume that each C_i has constant size.

2. (**Breaking IPSec encryption**) (100 points) Recall that IPSec works as follows. If Alice wants to send an IP packet P to Bob, she would send P to her gateway G_1. The latter will first pad P and append an additional Next-Header (NH) byte to obtain a string M, and then encrypt M to obtain a ciphertext C. It then creates an IPSec header H and sends a new IP packet P^* to G_2 in which the payload is $H\|C$. When G_2 receives P^*, it decrypts C, checks if the NH byte in the decrypted message M is `0x04`, and unpads M to recover P. If the checking of the NH byte fails or the padding is incorrect then G_2 drops the packet and sends an encrypted ICMP error message to G_1; otherwise it sends P to Bob.

IPSec allows several choices of the encryption scheme. One such choice is CBC with the following padding mechanism. Assume that the underlying blockcipher has 16B block length (like AES). You need to pad just enough bytes to the next multiple of 16 bytes *minus* one byte: recall that we need to add the NH byte to the padded string, and then run CBC on top of that. Moreover, in the padding,

append either `0x00` or `0x0101` or `0x020202` and so on. For example, if you have a 3B message then you must pad 12 additional bytes to it, each byte is `0x0b`.

In this exercise, you'll break IPSec encryption with the CBC choice above.

a) (30 points) Write a careful fragment of pseudocode for an algorithm Decrypt that decrypts an IPSec ciphertext C. (It should checks the NH byte and the padding format.) Let $\text{Decrypt}_K(C)$ return the distinguished symbol $\perp$ if it is provided an invalid ciphertext; otherwise, it returns a byte string P.

b) (40 points) Suppose that the adversary is given an oracle Valid that, given an IPSec ciphertext C, returns a single bit: the bit "1" if C is valid–meaning $\text{Decrypt}_K(C) \in \{0,1\}^*$—and the bit "0" if it is not—meaning $\text{Decrypt}_K(C) = \perp$. (This oracle can be realized by exploiting Bob's gateway.) Show how to use the oracle to decrypt a block $Y = E_K(X)$ for an arbitrary 16-byte X. Note that Y is **not** even a CBC ciphertext and thus you can't query it directly to Valid.

Hint: all your queries to the Valid oracle will be 32 bytes (namely 2 blocks). I don't mind if you make several thousand of them.

c) (30 points) Show how to decrypt any IPSec ciphertext C given a Valid oracle.

Note: If you can't solve part (b), you can assume that there is an adversary A that does the job for part (b), and show how to use it for part (c).

3. (**Challenge-response with public-key crypto**) (100 points) Recall that in the challenge-response setting, the client and the server share a secret key. If somehow the key is stolen from the server then all is lost. It's desirable that the client and server have *different* keys K_c and K_s, and the server key K_s is *public*.

Consider the following protocol.

- Let $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Run $(pk, sk) \leftarrow\!\!\$\ \mathcal{K}$. The server key K_s is pk, and the client key K_c is sk.
- If the client wants to identify himself, the server picks a random nonce $r \leftarrow\!\!\$\ \{0,1\}^n$ and sends $C \leftarrow\!\!\$\ \mathcal{E}(K_s, r)$ to the client.
- On receiving C, the client then computes $r' \leftarrow \mathcal{D}(K_c, C)$ and sends r' to the server.
- The server accepts only if $r' = r$.

This scheme is an attractive option for logging into a remote website from a laptop using a mobile phone as a second factor. The website displays C as a QR code on the laptop screen and the user scans the code using the phones camera. The phone decrypts C and displays the six least significant digits of r on the screen. The user then manually types the six digits into her web browser, and this value is sent to the remote web site to be verified.

a) (50 points) For the scheme above to be secure, the encryption scheme needs to be CCA-secure. Explain informally why this is the case.

b) (50 points) Find a public-key encryption scheme that is CPA-secure, but if we use it for the protocol above, a man-in-the-middle adversary can impersonate the client.