Graded Homework 3 – Due Tuesday, April 24

1. As you saw in your last graded homework, certain meta-data fields used in Nextcloud e2e encryption were saved in a special format called Base64. Do some searching on the Internet, and find out how Base64 works, and then answer the following questions.

   (a) What is Base64 used for (in other words, what problem does it solve)?

   (b) Describe how data is encoded using Base64 — give the algorithm, but do not copy from anywhere. Learn the algorithm first so that you understand it (it's not that complex!), and then describe it in your own words.

   (c) Consider binary data that is 600 bytes long. How many bytes long is the Base64 encoding of that data? (as always, show your work!)

   (d) Writing binary data in hexadecimal can solve the same problem. How long would the hexadecimal encoding of 600 bytes be?

2. One weakness of the Nextcloud end-to-end encryption design is that it uses GCM mode on the file as a whole, so it is impossible to update *part* of an encrypted file on the server — the entire file must be re-uploaded. Viewing a file as a sequence of 8192-byte chunks, design and describe a scheme that would allow chunks to be individually updated. For example, if a single chunk is changed in a 10 Gigabyte file, the amount of data uploaded should be close to 8192 bytes (a little more is OK!). After you describe your design, explain how your design protects both confidentiality and integrity of data. Are there any weaknesses to your scheme (or the idea of partial updates in general) that reveal information to an attacker that wouldn't be revealed in the current Nextcloud scheme?

3. The SSH protocol can use digital signatures to perform authentication without having to enter a password. Let's say you want to be able to log in from your home system to two remote systems, RemoteA and RemoteB. You create an identity on your home system, which is a keypair: the signing (private) key is kept on the home system only, while the verification (public) key is copied to both RemoteA and RemoteB. When you connect to RemoteA to log in, it sends a random challenge nonce to your home system, which is signed and sent back to RemoteA. Since RemoteA has the verification key, it can verify the signature to authenticate you.

   (a) If an attacker gains full access to system RemoteB, what information is stored on that system that is potentially compromised?

(b) With this information, can the attacker log in to RemoteA? Why or why not? Be precise, referring to the security properties of digital signatures (Section 13.1).

(c) With this information, can the attacker log in to your home system? Why or why not?

(d) What if the attacker's access to RemoteB includes the ability for the attacker to monitor the plaintext of all communication with RemoteB — does that change the answer to any of the preceding questions?

(e) Now consider a scenario in which basic passwords are used for authentication. As before, the same authentication information (in this case a plaintext password) is shared between the two remote systems. If the attacker has fully compromised RemoteB in this scenario, is access to any other system possible?

4. Consider including a special-purpose chip in a computer (or phone) that internally stores a random 256-bit secret k in such a way that the secret can only be used as the key for executing HMAC (using SHA-256) in the chip. Specifically, all the user can do is send a bitstring x to the hardware so that it computes and returns HMAC(k, x). Consider a system that uses this hardware to create a 256-bit key for AES for data storage: The user supplies a password or PIN as x, and then the output of the hardware HMAC is an AES key that is used to encrypt and decrypt all data stored on the hard drive or other connected storage device. We can quickly test whether the hardware gives the correct key by attempting a decryption using it. For this problem, assume that HMAC with SHA-256 satisfies the desired MAC security properties from Section 12.4 of the textbook.

   (a) Consider a case where the storage device is removed from the system and then stolen or sold to an adversary. In addition to having the data, the attacker knows the algorithm that is used and knows that x is a 4-digit PIN (but doesn't know x). How secure is this? Reason about the "best possible attack" given the security of HMAC, and give some indication of how much time this would take. I'm not looking for a mathematical proof — informal reasoning is OK, but make sure the logic of your reasoning is clear!

   (b) If the entire device (the security chip and the storage device) is captured, so that the attacker can access the HMAC-computing hardware device, what is the best attack in this case? How much time would this take (state any assumptions you need to make about the speed of the hardware device, etc.).

   (c) If the hardware deletes the secret k, then the attacker is basically back at the situation in part (a), even if they have the entire device. How could you improve the system so that it can detect and delete the secret k after a certain number of failed attempts (your goal is to stop brute-force attacks on the PIN). Note that the security chip needs some way of knowing when a use has been successful or not, and an attacker should not be able to trick this detection.