**CS255, Winter 2020**
**Introduction to Cryptography**

# Project 1 Short Answer Questions

**Silvia Gong, Julian Cooper**

---

**Question 1:** In our implementation, Alice and Bob increment their Diffie-Hellman ratchets every time they exchange messages. Could the protocol be modified to have them increment the DH ratchets once every ten messages without compromising confidentiality against an eavesdropper?

Idea: ..

**Question 2:** What idf they never update their DH keys at all? Please explain the security consequences of this change with regards to Forward Secrecy and Break-in Recovery.

Idea: ..

**Question 3:** Consider the following conversation between Alice and Bob, protected via Double Ratchet Algorithm according to the spec:

```
A: Hey Bob, can you send me the locker combo?
A: I need to get my laptop
B: Sure, it's 1234!
A: Great, thanks! I used it and deleted the previous message.
B: Did it work?
```

What is the length of the longest sending chain used by Alice? By Bob? Please explain.

Longest sending chain used by Alice is length 2. This is occurs when Alice sends 2 messages without response from Bob. Since Bob did not respond in between, Alice completes a second symmetric ratchet to update the chain and message keys without needing to complete a DH ratchet to update the root key.

Longest sending chain used by Bob is length 1. This is because Bob never sends more than one message in sequence without Alice responding.

**Question 4:** Unfortunately, in the situation above, Mallory has been monitoring their communications and finally managed to compromise Alice's phone and steal all her keys just before she sent her third message. Mallory will be unable to determine the locker combination. State and describe the relevant security property and justify why double ratchet provides this property.

Forward Secrecy. This property states that compromising long term keys or current session key must not compromise past communications.

The double ratchet provides this property: output keys from the past appear random to an adversary who learns the KDF key at some point in time. This happens because each ratchet step uses a KDF which acts like a one-way PRF (deterministic and output looks random) so long as we include sufficient entropy.

**Question 5:** The method of government surveillance is deeply flawed. Why might it not be as effective as intended?

We interpreted this question to be about citizens' ability to evade government eavesdropping. Idea: what is the user (citizen) simply did not follow the protocol and used the wrong government public key. Without updates to the protocol, the government would likely not even know this had happened!

What are the major risks involved with this method?

The major risk we saw was around the government becoming a single source of failure. For example, if the government was hacked, and their secret key was stolen, the attacker could access and read all communciations that use the messanging service.

**Question 6:** The SubtleCrypto library is able to generate signatures in various ways, including both ECDSA and RSA keys. For both the ECDSA and RSA-based signature technqiues, please compare:

1. Which keys take longer to generate (timing SubtleCrypto.generateKey)? ...
2. Which signature takes longer to generate (timing SubtleCrypto.sign)? ...
3. Which signature is longer in length (length of output of SubtleCrypto.sign)? ...
4. Which signature takes longer to verify (timing SubtleCrypto.verify)? ...