Constance Xu **Worksheet 2** CS 306 November 27, 2019

Problem 1: Can you crack a password?

1. The attack that Eve can use to figure out Alice's password is a dictionary attack. A dictionary attack is when you use commonly used words or phrases that are likely to be passwords (since people typically create passwords with common English words). This goes hand-in-hand with using a rainbow table (all possible plaintext permutations of encrypted passwords specific to a given hash algorithm). SHA-1 and SHA-256 are famous hashing schemes, so there is likely to find tables of common passwords for both. Since the password must explicitly be 10, capitalized letters, it makes it far easier to find since this is a huge limitation as to what it can be. Hence, because of this limitation, a brute force attack is also quite feasible as there are only so many passwords that can be created using ten, capitalized letters.

2. The use of user-specific salts appends characters to the password before it is hashed. Hence, any type of dictionary attack is not as strong against these user-specific salts. This would make figuring out the password more difficult. Furthermore, with these salts, if a user has the same password as another user, if one becomes compromised, the other remains the protected. However, Eve has compromised the security of the Stevens Authentication Server and hence, she has access to the salts' plaintexts and hence, her attack would likely not be slowed down.

3. A birthday attack is first and foremost a cryptographic attack that exploits the mathematics behind the birthday paradox. This means that you are more likely to find a collision by randomly choosing (also, the pigeonhole principle has a lot to do with this as well). Eve can utilize this tactic to find some distinct number of randomly chosen passwords and try out the random password which could potentially speed up the amount of time it takes for her to find the right password.

Problem 2: A password, but which password?

1. The honeywords model helps because it creates fake passwords that may seem like very real passwords. So say Eve did compromise the hashes and she figures out some of the passwords, and if she attempts to use one of the `honeywords`, then the system can be alerted that there might be some sort of compromise and that someone may be attempting to hack in.

2. password, pAsSWorD, password5, p4ssw0rd5, paSsw0rd555, Pa$$w0rd5, pa$sw0rd5, Pa$$word5, pa$$w0rd55, p4ssword5

3. I would specifically have chosen Blink-182 because it is similar to what any person could remember easily and it is a famous band. Similarly, the other honeywords are similar enough to this one that it makes it likely that the other passwords are simply trying

to mask it. The last two passwords are just highly unlikely that someone would spend the time and dedication it takes to memorize a password that long and convoluted.

Problem 3: Sir, is this your public key?

1. Eve can easily just store Bob's old public key. Mallory can store Bob's old public key so that whenever anyone asks for Bob's new public key, Mallory can provide Bob's old public key. Hence, Eve will still be able to eavesdrop on Bob's conversations. Bob will continue to believe that he is using his new public key and Eve can eavesdrop successfully and not be detected. Since Eve will still have the secret key, she will be able to decrypt each and every message. Eve will then be able to successfully make a man-in-the-middle attack through this.
2. Time-stamped signatures can provide the CA with information that is useful to prevent the man-in-the-middle attacks. The CA can authorize the timestamp for the new directory by simply creating a timestamp on the merkle-tree hash digest on the directory. Hence, the user could request Bob's public key and the key should then redirect them to the new timestamp on the hash for the new keys. If Mallory provides an old public key, they will notice that immediately with the timestamped signatures.
3. The main issue with doing this using a DNS server is that the one minute signing period would create too many requests and essentially be making a DDoS attack on the server. If clients wishing to make DNS requests too frequently, could make the servers crash as it would create a sort of DDoS attack. In theory, I believe that it is a great idea but it would just not work out in real life.

Problem 4: Can Cryptography Fail?

1. Eve has access to the previous messages so she can make changes by using this MAC scheme which will not notice if a semicolon is misplaced. If the semicolon was moved from the second no to the first so that it reads `no; no more homework assignments will come` then Eve can create havoc as this changes the meaning of the text entirely. She can do this through a replay attack.
2. RSA is deterministic, Eve can intercept the messages to encrypt the numbers 1 through 10 with their public keys so that she can see what they look like post encryption. From there, she can compare those messages to find what the changed numbers look like and compare those with their particular hashes. She can then find out the topics on the exam.
3. Bob should not because even though Charles is verified with the RSA, it is impossible to verify that Charles is actually Charles. He could be using a fake email or fake name or just everything could be fake so that he can obtain private information.

Problem 5: Are cloud based solutions cloudy?

1. Eve could rehash her assignment until it matches the hash of someone else's assignment and if that assignment is perfect, than Eve will get a perfect grade as well. Even if she does not, since she did not study for it, she will likely get a higher grade than a zero.

2. The given proposal will help make this cloud storage more secure because of encryption. The plain RSA is deterministic and hence, the files will have the same encrypted values and yield the same hash digests after encryption. This has no effect on deduplication. This will not reduce or increase costs in any way.

3. ElGamal is nondeterministic and hence, the file can be encrypted to a new, random ciphertext encoding. But because of this, deduplication benefits are not available because hashing the encrypted files will reduce collisions.