# CO 487 - Final prep

Bilal Khan (b54khan)
bilal.khan@student.uwaterloo.ca

December 6, 2024

## Contents

# 1 Sample Final 1

## 1.1 1

### 1.1.1 a

Give 1-sentence informal definitions of the following fundamental goals of cryptography: confidentiality, data integrity, data origin authentication, non-repudiation.

- Confidentiality: Only people who are authorized to see the data can see it.
- Integrity: Making sure the data hasn't been modified.
- Authentication: Making sure the data is coming from who it says it is.
- Non-repudiation: Making sure a sender cant deny sending the data.

### 1.1.2 b

Explain the difference between MAC algorithms and signature schemes.

MACs provide integrity and authentication guarantees, but not non-repudiation.

## 1.2 2

Recall that RC4 is a stream cipher which, on input consisting of a secret key $k$, outputs a keystream $RC4(k)$. The key stream is then used to encrypt a plaintext message by bitwise exclusive-or. What is the danger in using the same key $k$ to encrypt two different plaintext messages?

Using the same key for two messages means 1) if you encrypt the same message twice, the ciphertexts are the same, and 2) xor-ing the ciphertexts gives you the xor of the plaintexts (the $RC4(k)$'s cancel out from being xor-ed).

## 1.3 3

### 1.3.1 a

What type of interaction is required between the adversary and the legitimate user(s) in order to perform linear cryptanalysis? What goal can the adversary achieve under this interaction?

You need to see the plaintexts/ciphertexts for some messages (KPA). The goal is to recover the key

### 1.3.2 b

What type of interaction is required between the adversary and the legitimate user(s) in order to perform differential cryptanalysis? What goal can the adversary achieve under this interaction?

You need to see the plaintexts/ciphertexts for some *related* messages (CPA). The goal again is to recover the key.

### 1.4 4

Recall the following notation, used in protocols such as DSA and Diffie-Hellman: $p$ is an odd prime, $q$ is a prime divisor of $p-1$, and $g \in \mathcal{Z}_p^*$ is an element of order $q$.

Here is one method for generating $g$, given $p$ and $q$:

Repeat the following:

Select $h \in \mathbb{Z}$ randomly from within the range $2 \leq h \leq p-1$,

Compute $g = h^{(p-1)/q} \mod p$,

Until $g \neq 1$.

Output($g$).

Prove that this method works, i.e., prove that $g$ has order $q$.

Since $h$ is not 1 and is less than $p$, $\gcd(h,p) = 1$, and $h \in \mathbb{Z}_p^*$. $p-1$ is divisible by $q$, so $h^{(p-1)/q} \equiv 1 \pmod{p}$ will also be an element of $\mathbb{Z}_p^*$.

$$\begin{aligned} g^q &\equiv (h^{(p-1)/q})^q \pmod{p} \\ g^q &\equiv h^{p-1} \pmod{p} \\ g^q &\equiv 1 \pmod{p} \end{aligned}$$

The last step is true since we repeat until $g \neq 1$, so g is coprime to p, and by fermat's little theorem. $q$ is prime and $g \neq 1$, so $q$ is necessarily the smallest positive integer that makes the statement true (i.e. it has no factors that would make the statement true, except for 1).

### 1.5 5

Why are RSA public and private keys so much longer than secret keys in a symmetric-key encryption scheme such as AES (for the same level of security)?

Asymmetric encryption's hardness is based on the keys being hard to factor, symmetric encryption's hardness is based on the keys being hard to guess. We need comparatively longer keys for RSA because the keys themselves have mathematical structure that makes them easier to guess/brute force than AES keys (purely random keys).

### 1.6 6

Recall the specification of (basic) hybrid encryption. We have a public-key cryptosystem $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, and a symmetric-key cryptosystem $(E, D)$ with key length $\ell$. To send an encrypted message $m$, choose a random key $k \in \{0,1\}^\ell$, compute

$$(c_1, c_2) = (\mathcal{E}(pubkey, k), E(k, m))$$

and transmit this data to the recipient. To decrypt (c1, c2), compute

$$m = D(D(privkey, c_1), c_2).$$

Throughout this problem, we will use the one-time pad as our symmetric-key cryptosystem

### 1.6.1 a

Suppose that the public-key cryptosystem is insecure in such a way that an adversary intercepts a ciphertext and learns some partial (but not complete) information about $k$. What information can the adversary learn about $m$?

Since the key is bitwise xor-ed with the message, the adversary can learn information about individual bits of the message.

### 1.6.2 b

Suggest a practical modification of basic hybrid encryption which prevents the adversary from learning any information about $m$ even if the public-key cryptosystem is compromised in the manner described in part (a).

Use $k$ in a KDF to generate a key to encrypt $m$.

## 1.7 7

What is certificate revocation? Why is it important to have certificate revocation in a public-key infrastructure (PKI)?

Certificate revocation lets you revoke certificates in the case that the private key used to generate them has been compromised.

## 1.8 8

Discuss the security implications for Bitcoin if: (a) SHA-2 is broken, (b) ECDSA is broken.

Bitcoin uses SHA as the proof of work system, so if it is broken, it comes easier to generate new blocks and it becomes easier to fake transactions for longer e.g in a 51% attack.

ECDSA is used to sign and guarantee the authenticity of transactions, so if it is broken, you can forge transactions in ways that would seem legitimate to the network.

## 1.9 9

In an implementation of RSA, we may choose to blind the ciphertext $c$ as follows: after receiving the value of $c$, choose a random integer $r$, compute $c' = (c \cdot r^e) \mod n$, and decrypt $c'$ instead of $c$.

### 1.9.1 a

After decrypting $c'$, how does the implementation recover the originally intended plaintext?

We decrypt:

$$\begin{aligned} m' &= (c')^d \mod n \\ &= (c \cdot r^e)^d \mod n \\ &= c^d \cdot r^{ed} \mod n \\ &= c^d \cdot r^1 \mod n \\ m' &= m \cdot r \mod n \\ m &= (m' \cdot r^{-1}) \mod n \\ m &= ((c')^d \cdot r^{-1}) \mod n \end{aligned}$$

### 1.9.2 b

Does blinding protect against simple side-channel attacks, such as Simple Power Analysis (SPA)? Why or why not?

??

### 1.9.3 c

Does blinding protect against second-order side-channel attacks, such as Differential Power Analysis (DPA)? Why or why not?

??

## 1.10 10

Recall the specification of Full Domain Hash RSA (RSA-FDH):

**Key generation**: Same as in basic RSA. Let $\ell$ denote the bitlength of $n$.

**Public parameters**: A hash function $H : \{0,1\}^* \to \{0,1\}^\ell$.

**Signing**: The message space is $\{0,1\}^*$. For any message $m \in \{0,1\}^*$, the signature of $m$ is $s = H(m)^d \mod n$.

**Verification**: Given a signature $s$ of a message $m$, compute $s^e \mod n$ and check whether this value equals $H(m)$.

### 1.10.1 a

Describe how an adversary capable of mounting a known-message attack can forge Alice's signatures if $H$ is not 2nd-preimage resistant.

Given a known (message, signature) pair $(m, s)$, we can find a $m' \neq m$ such that $H(m') = H(m) = s$. We can now forge a signature for $m'$.

### 1.10.2 b

Describe how an adversary capable of mounting a chosen-message attack can cheat if $H$ is not collision resistant.

An adversary can find a pair of messages $m \neq m'$ that yields a collisiaon: $H(m) = H(m')$. It can then choose to get a signature $s$ for $m$ from the oracle and then pass off $s$ as a signature for $m'$.

### 1.10.3 c

Describe how Alice can repudiate signatures if $H$ is not collision resistant.

If you want to repudiate a signature for a message $m$, you can find a $m' \neq m$ such that $H(m') = H(m)$. You can then get a signature for $m'$ from the oracle and claim that you never signed $m$.

# 2 Sample Final 2

## 2.1 1

Define what it means for a MAC scheme to be secure.

If it is passes EUF-CMA (existential unforgeability under chosen message attacks) aka an attacker can't generate a tag for a message without the key.

## 2.2 2

Explain why SHA-256 and AES-Small are said to have the same security level. (Recall that SHA-256 is a hash function with 256-bit hash values, and AES-Small is a block cipher with secret keys of bitlength 128

To break SHA-256, a hash function, you need to find a collision, by the birthday paradox, you need to try $\approx 2^{128}$ values;

To break AES-Small, a symmetric block cipher you need to brute force all $2^{128}$ keys

## 2.3 3

Alice is given two hash functions $F : \{0,1\}^* \to \{0,1\}^{160}$ and $G : \{0,1\}^* \to \{0,1\}^{160}$. She is told that one of these functions is collision resistant (and the other one is not collision resistant), but she does not know which is which. Alice wishes to use $F$ and $G$ to create a new hash function $H$ which is definitely collision resistant. She defines the hash function $H : \{0,1\}^* \to \{0,1\}^{320}$ as follows: $H(x) = F(x)||G(x)$ (where | denotes concatenation). Either prove that $H$ is collision resistant, or provide a counterexample which shows that $H$ is not necessarily collision resistant.

We can prove this by contradiction. If $H$ is not collision resistant, then there exist two inputs $x \neq x'$ such that $H(x) = H(x')$. This means that $F(x)||G(x) = F(x')||G(x')$. Since exactly one of $F$ or $G$ is collision resistant, this would be a contradiction and therefore the hash function $H$ is collision resistant.

## 2.4 4

Why is encryption exponent $e = 3$ commonly used in deployments of the RSA encryption scheme and the RSA signature scheme? Why not use $e = 2$?

We didn't cover this? Claude says its an optimization to minimize the number of steps you need to take in the square-and-multiply algorithm

## 2.5 5

Recall that in the Diffie-Hellman key agreement protocol, Alice selects $x \in_{\mathbb{R}} [0, q-1]$ and sends $g^x \mod p$ to Bob. Similarly, Bob selects $y \in_{\mathbb{R}} [0, q-1]$ and sends $g^y \mod p$ to Alice. Their shared secret key is $k = H(K)$ where $K = g^{xy} \mod p$. (Here, $p$ is a prime, $q$ is a prime divisor of $p-1$, $g$ is an element of order $q$ in $\mathbb{Z}_p^*$, and $H$ is a hash function.) Show that the protocol is insecure if the communications channel between Alice and Bob is not authenticated.

A mitm attacker can intercept the key messages and instead select two new values $a, b \in_{\mathbb{R}} [0, q-1]$ and pass off $g^a \mod p$ and $g^b \mod p$ as the messages from Alice and Bob respectively. The new shared secret keys are then $g^{xa} \mod p$ and $g^{yb} \mod p$ respectively. The attacker can decrypt and reencrypt messages.

### 2.6 6

We recall the DSA signature scheme. The public domain parameters consist of a 1024-bit prime $p$, a 160-bit prime divisor $q$ of $p-1$, and an element $g \in \mathbb{Z}_p^*$ of order $q$. SHA-1 is a 160-bit hash function. Alice's private key is $a \in_{\mathbb{R}} [0, q-1]$, while her public key is $h = g^a \mod p$. To sign a message $M \in \{0,1\}^*$, Alice does the following:

1. Select $k \in_{\mathbb{R}} [1, q-1]$.
2. Compute $m = \text{SHA-1}(M)$.
3. Compute $r = (g^k \mod p) \mod q$, and check that $r \neq 0$.
4. Compute $s = k^{-1}(m + ar) \mod q$, and check that $s \neq 0$.
5. Alice's signature on $M$ is $(r, s)$

To verify Alice's signature $(r, s)$ on $M$, Bob does the following:

1. Obtain an authentic copy of Alice's public key $h$.
2. Check that $1 \leq r, s \leq q-1$.
3. Compute $m = \text{SHA-1}(M)$.
4. Compute $u_1 = ms^{-1} \mod q$ and $u_2 = rs^{-1} \mod q$.
5. Accept if and only if $r = (g^{u_1} h^{u_2} \mod p) \mod q$.

Show that DSA is existentially forgeable under a key-only attack if SHA-1 is not preimage resistant. (A key-only attack means that the attacker knows Alice's public key, but does not have access to a signing oracle.)

We can use the property that $H$ is not preimage resistant to sign a message without access to a signing oracle.

Choose a random $u_1, u_2 \in_{\mathbb{R}} [1, q-1]$. Compute $r = (g^{u_1} h^{u_2} \mod p) \mod q$ and check that $r \neq 0$. Set $s = u_2^{-1} r \mod q$. Set $m = u_1 s \mod q$. Invert the hash to find a $M$ such that $H(M) = m$. The signature is $(r, s)$.

To verify this is a valid forged signature, note that $r, s$ are in the correct range, $m = H(M)$, $u_1 = ms^{-1} \mod q = (u_1 s)s^{-1} \mod q = u_1$, $u_2 = rs^{-1} \mod q = r(u_2^{-1} r)^{-1} \mod q = u_2$, and that $r = (g^{u_1} h^{u_2} \mod p) \mod q$ by definition.

### 2.7 7

N/a

### 2.8 8

N/a

### 2.9 9

Explain why ECDSA (the elliptic curve analogue of DSA) may be advantageous over DSA.

ECC provides greater security at smaller key sizes, so its more efficient and more future-proof against post-quantum attacks.

### 2.10 10

What is certificate revocation? Why is it important to have certificate revocation in a public-key infrastructure (PKI)?

Certificate revocation lets you revoke certificates in the case that the private key used to generate them has been compromised.

### 2.11 11

N/a