

CO 487 - Final prep

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

December 5, 2024

Contents

1	Sample Final 1	1
1.1	1	1
1.1.1	a	1
1.1.2	b	2
1.2	2	2
1.3	3	2
1.3.1	a	2
1.3.2	b	2
1.4	4	2
1.5	5	3
1.6	6	3
1.6.1	a	3
1.6.2	b	3
1.7	7	4
1.8	8	4
1.9	9	4
1.9.1	a	4
1.9.2	b	4
1.9.3	c	4
1.10	10	5
1.10.1	a	5
1.10.2	b	5
1.10.3	c	5

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 can't 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 $g \in \mathbb{Z}_p^*$ 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 $g = h^{(p-1)/q} \pmod{p}$,

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 ℓ . To send an encrypted message m , choose a random key $k \in \{0, 1\}^\ell$, compute

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

and transmit this data to the recipient. To decrypt (c_1, c_2) , compute

$$m = D(D(\text{privkey}, c_1), c_2).$$

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 $c' = (c \cdot r^e) \bmod n$, and decrypt c' instead of c .

1.9.1 a

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

We decrypt:

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

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 ℓ denote the bitlength of n .

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

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

Verification: Given a signature s of a message m , compute $s^e \bmod n$ 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 collision: $H(m) = H(m')$. 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 .