| Name: | Last name: | Id: |
|---|---|---|

**Cybersecurity**
**Computer and network security**
**Sicurezza nelle reti e nei sistemi informatici**
**Crittografia e sicurezza delle reti**

*Exam of 21 June 2023, a.y. 2022-23. Time: 2 hours*

*FOR NON-ENGLISH: 2 penalty points (only applicable to courses in English)*
*FOR UNREADABLE WRITING: arbitrary penalty points*

Q1: **Symmetric ciphers**

Q1.1 [1/30] Describe the scenario of symmetric ciphers and define the concepts of: synchronous/asynchronous stream ciphers, block ciphers and modes of operations.
[If definition is wrong subsequent questions cannot be correctly answered]

Q1.2 [3/30] Describe the RC4 cipher (both the key generation and the encryption process). What type of cipher is it?

Q1.3 [4/30] Describe and compare OFB and CTR. Can you suggest possible design criteria for their adoption?

Q2: **Man in the middle**

Q2.1 [2/30] Describe the attack *Man-In-The-Middle*, as well as a possible scenario where it could be run.

Q2.2 [2/30] Alice suspects she is currently being the target of a Man-In-The-Middle attack, and she decides to hire you as a personal adviser. Is it still possible to carry out safe actions? Discuss.

Q3: **Hashing**

Q3.1 [2/30] Define the properties that qualify a hashing function as cryptographic (the more formal, the better).

Q3.2 [2/30] Describe the Merkle–Damgård construction. If the underlying hash function maps 256b blocks into 128b blocks, how many rounds are required for hashing a 140KB file?

Q3.3 [2/30] Discuss and compare possible schemes for keying a hash function.

Q4: **RSA verification**

Q4.1 [2/30] Describe how to verify an RSA digital signature.

Q4.2 [2/30] Explicitly describe the possible causes of a verification failure.

Q4.3 [2/30] For having non-repudiation is the verification needed? [2/30].

Q5: **Authentication**

Q5.1 [2/30] Discuss the security of the following challenge-based scheme for mutual authentication, where Alice (A) and Bob (B) share a secret key K:
(information below is transmitted as clear text)
A -> B: (A, NA, B) { NA is a nonce chosen by A }
B -> A: (B, NB, K(NA), A) { NB is a nonce chosen by B }
A -> B: (A, K(NB), B)

Q5.2 [2/30] How would you improve the above schema?

| Name: | Last name: | Id: |
| --- | --- | --- |

Q6: **Miscellaneous**

Provide *short* answers to the following questions.

Q6.1 [2/30] Compute 5^12241 mod 13.

Q6.2 [2/30] Describe as best as you can the meaning of the following command iptables -A INPUT -p tcp -s 0/0 -d 195.55.55.78 --sport 513:65535 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

Q6.3 [2/30] Describe as best as you can the meaning of the following command ssh -L 44044:192.168.1.221:22 user@host.example.com

1.1 IN SYMMETRIC CRYPTOGRAPHY, THE SAME SECRET KEY IS SHARED BY SENDER AND RECEIVER, AND IS USED FOR BOTH ENCRYPTION AND DECRYPTION.

STREAM CIPHERS ENCRYPT DATA BIT BY BIT OR BYTE BY BYTE BY COMBINING THE PLAINTEXT WITH A KEYSTREAM

IN SYNCHRONOUS STREAM CIPHERS, THE KEYSTREAM DEPENDS ONLY ON THE SECRET KEY (AND AN IV) AND NOT ON THE PLAINTEXT OR CIPHERTEXT. ERRORS AFFECT ONLY THE CORRESPONDING BITS.

IN ASYNCHRONOUS STREAM CIPHERS, THE KEYSTREAM DEPENDS ALSO ON SOME PREVIOUS CIPHERTEXT BITS. A SINGLE ERROR PROPAGATES.

BLOCK CIPHERS ENCRYPT FIXED SIZE BLOCKS OF PLAINTEXT INTO CIPHERTEXT BLOCKS OF THE SAME SIZE, USING THE SAME KEY (DES, AES). TO PROCESS LONG MESSAGES, BLOCK CIPHERS USE MODES OF OPERATION.

ECB: EACH BLOCK ENCRYPTED INDEPENDENTLY, INSECURE DUE TO PATTERNS
CBC: EACH BLOCK CHAINED TO THE PREVIOUS ONE, STRONGER
CFB/OFB: TURN A BLOCK CIPHER INTO A STREAM CIPHER
CTR: COUNTER MODE, PARALLELIZABLE AND EFFICIENT

1.2 RC4 IS A SYNCHRONOUS STREAM CIPHER THAT GENERATES A PSEUDO RANDOM KEYSTREAM COMBINED WITH THE PLAINTEXT VIA XOR.

KEY SCHEDULING ALG (KSA): THE 256-BYTE STATE ARRAY S IS INITIALIZED WITH VALUES 0-255. USING THE SECRET KEY, THE ARRAY IS PERMUTED BY ITERATIVELY UPDATING AN INDEX $j$ AND SWAPPING ENTRIES $S[i]$, $S[j]$.

PSEUDO RANDOM GENERATION ALG (PRGA): TWO INDICES $i, j$ ARE UPDATED FOR EACH OUTPUT BYTE. AFTER SWAPPING $S[i]$ AND $S[j]$, THE KEYSTREAM BYTE IS SELECTED AS $S[(S[i]+S[j]) \bmod 256]$.

ENC/DEC: EACH PLAINTEXT BYTE IS XORED WITH THE CORRESPONDING KEYSTREAM BYTE. SINCE XOR IS SYMMETRIC, THE SAME PROCESS DECRYPTS THE CIPHERTEXT.

1.3 OFB: GENERATES KEYSTREAM BY REPEATEDLY ENCRYPTING AN IV. XOR WITH PLAINTEXT. SYNCHRONOUS, NOT PARALLELIZABLE. ONE BIT ERROR AFFECTS ONLY ONE BIT. REUSING IV IS INSECURE

CTR: GENERATES KEYSTREAM BY ENCRYPTING A COUNTER + NONCE. XOR WITH PLAINTEXT. FULLY PARALLELIZABLE, EFFICIENT. REUSING NONCE IS INSECURE.

BOTH TURN A BLOCK CIPHER INTO A STREAM CIPHER. OFB IS SEQUENTIAL AND SLOWER, IT CAN BE USED FOR SIMPLE CASES. CTR IS FASTER AND PREFERRED FOR PERFORMANCE AND PARALLELISM.

2.1 IT OCCURS WHEN AN ADVERSARY INTERCEPTS AND POSSIBLY ALTERS THE COMMUNICATION BETWEEN TWO PARTIES WHO BELIEVE THEY ARE TALKING TO EACH OTHER. THE ATTACKER CAN READ, MODIFY OR INJECT MESSAGES

ALICE CONNECTS TO BOB'S SERVER OVER AN UNSECURED NETWORK. TRUDY PLACES HERSELF BETWEEN THEM, ALTERING DATA.

2.2 THE ONLY WAY TO ACT SAFELY IS TO SWITCH TO A SECURE CHANNEL WITH A TLS CONNECTION, AND VERIFY THE AUTHENTICITY OF THE COUNTERPART.

3.1 IT MUST BE PREIMAGE RESISTANT, MEANING THAT GIVEN AN OUTPUT $h$ IT'S INFEASIBLE TO FIND $x$ SUCH THAT $h = H(x)$. IT MUST BE SECOND PREIMAGE RESISTANT, MEANING THAT GIVEN AN INPUT $x$ IT'S INFEASIBLE TO FIND $x' \neq x$ SUCH THAT $H(x) = H(x')$. IT MUST BE COLLISION RESISTANT, SO THAT FINDING TWO DIFFERENT INPUTS THAT PRODUCES THE SAME DIGEST REQUIRE $2^{n/2}$ WORK WITH AN $n$-BIT HASH. IT SHOULD HAVE THE AVALANCHE EFFECT, WHERE A ONE BIT CHANGE IN THE INPUT CHANGE THE ENTIRE OUTPUT (INTEGRITY).

3.2 IT BUILDS A HASH FOR ARBITRARY LENGTH INPUTS FROM A FIXED SIZE COMPRESSION FUNCTION. THE MESSAGE IS PADDED TO A MULTIPLE OF THE BLOCK SIZE, THEN PROCESSED BLOCK BY BLOCK: STARTING FROM AN IV, EACH BLOCK IS COMBINED WITH THE PREVIOUS CHAINING VALUE THROUGH THE COMPRESSION FUNCTION, AND THE FINAL CHAINWG VALUE IS THE HASH OUTPUT.

INPUT: 256 BIT = 32 BYTE
OUTPUT: 128 BIT

$140 \cdot 1024 = 143\,360$ BYTES
$143\,360 / 32 = 4480$ BLOCKS OF 256 BIT

WE NEED ANOTHER BLOCK FOR PADDING + LENGTH → 4481 BLOCKS

3.3 KEY PREPENDING $H(K \| M)$: VERY SIMPLE BUT INSECURE AGAINST LENGTH EXTENSION ATTACK.

KEY APPENDING $H(M \| K)$: SLIGHTLY BETTER, BUT STILL VULNERABLE.

ENVELOPE CONSTRUCTION $H(K \| M \| K)$: BETTER PROTECTION BUT NOT STANDARD

HMAC: SECURE. PREVENTS EXTENSION ATTACKS.

$$HMAC(K, M) = H((K \oplus OPAD) \| H((K \oplus IPAD) \| M))$$

4.1 BOB RECEIVES THE DIGITAL SIGNATURE AND THE MESSAGE $(M, s)$. HE THEN COMPUTES THE HASH OF $M$ AND USES THE SIGNER'S PUB KEY $(n, e)$ TO RECOVER THE VALUE $v = s^e \bmod n$. IF $h' = v$ THE SIGNATURE IS VALID.

4.2 MESSAGE ALTERATION: THE HASH DOESN'T MATCH THE SIGNATURE
WRONG PUBLIC KEY: THE VERIFIER IS USING AN UNTRUSTED PUB KEY
INVALID ENCODING: THE FORMAT OF THE SIGNATURE IS NOT STANDARDIZED.
TRANSMISSION OR STORAGE ERRORS: DATA CORRUPTION

4.3 YES. NON REPUDIATION MEANS THAT ONLY THE SIGNER, USING THEIR PRIV KEY, COULD HAVE PRODUCED THE SIGNATURE, AND ANY THIRD PARTY CAN CHECK THIS FACT WITH THE CORRESPONDING PUB KEY.

5.1 BOTH PARTIES PROVE KNOWLEDGE OF THE SHARED KEY K WITH FRESH NONCES, GIVING MUTUAL AUTHENTICATION AND REPLAY PROTECTION. HOWEVER THIS METHOD IS VULNERABLE TO REFLECTION AND MITM, SINCE AN ATTACKER COULD REFLECT A CHALLENGE BACK. MOREOVER ENCRYPTING NONCES IS WEAKER THAN USING A MAC.

5.2 REPLACING SIMPLE NONCE ENCRYPTION WITH HMAC TO PROVIDE INTEGRITY AND AVOID REFLECTION ATTACKS.
BINDING THE NONCES TO THE IDENTITIES INSIDE THE MAC TO PREVENT MANIPULATION.
USING SEPARATE KEYS FOR ENC AND AUTH TO AVOID KEY REUSE.

6.1 $5^{12241} \bmod 13$ $\quad p=13$ PRIME $\quad a=5$ NOT A p'S MULTIPLE

$a^{p-1} \bmod p \rightarrow 5^{12} = 1 \bmod 13 \rightarrow 12241 = 12 \cdot 1020 + 1$

$5^{12241} = 5^{1} \bmod 13 \rightarrow 5^{12241} \bmod 13 = \mathbf{5}$

6.2 THIS RULE ALLOWS INCOMING TCP PACKETS TO THE LOCAL HOST AT IP 195.55.55.78, DESTINATION PORT 22 (SSH), COMING FROM ANY SOURCE ADDRESS WITH A SOURCE PORT BETWEEN 513 AND 65535. IT ACCEPTS BOTH NEW SSH CONNECTION ATTEMPTS AND PACKETS BELONGING TO ALREADY ESTABLISHED SSH SESSIONS.

6.3 THIS RULE OPENS AN SSH CONNECTION TO HOST.EXAMPLE.COM AS USER AND SETS UP LOCAL PORT FORWARDING. CONNECTIONS MADE LOCALLY TO PORT 44044 WILL BE SECURELY TUNNELED THROUGH THE SSH SESSION AND FORWARDED TO 192.168.1.221:22 AS SEEN FROM THE REMOTE HOST. SO, IT ALLOWS ACCESSING THE SSH SERVICE OF 192.168.1.221 VIA THE TUNNEL.