**Domande “teoriche” di CS**

**Describe the meaning of assets and threats, with their relation**

Assets are the resources to protect from threats, using countermeasures in order to reduce the vulnerabilities of the system.

**Tell the 5 security properties**

Accountability: bind action to people

Authentication: identify accurately data origin & principles

Availability: service accessible when desiderd

Confidentiality: data not learned by unauthorized principles

Integrity: data not maliciously altered

**What does cryptography do?**

Technology that enables us to turn unsecure channels into secure ones.

need to provide confidentiality, integrity and authentication (non-repudiation desirable)

**What is steganography?**

Science of hiding messages in other messages (f.e. inside an image)

**Explain in general the two kinds of encryption.**

Symmetric: enc/dec keys equals or derivable one from each other

Asymmetric: different keys publishing public key doesn’t compromise the private one

**Who is the cryptoanalyst?**

job of looking at cipher text and trying to reconstruct either plaintext or key

**Describe the 5 types of attacks to a cryptographic system.**

cipthertext only: C must deduce the alg from the cipher

known plaintext: C must inverse the key or deduce the alg

chosen plaintext: // but C can choose the plaintext

adaptive chosen plaintext: // but C can edit plaintext based on encr. results

chosen ciphertext: C can choose text to decrypt

**Describe the two kinds of security.**

Unconditional: secure even vs infinite computing power (information theory)

Conditional: broken in principle but not realistically (complexity theory)

**What is and why isn't used the "security of obscurity"?**

Not revealing how things work (the alg) → not applicable (cryptanalysts may find vulnerabilities before serious damage)

**Describe the Monoalphabetic substitution cipher and Affine cipher.**

It’s a Caesar generaliz. with 26! complexity bcs of mapping between letters (A→C, B→Z, ..)

→ Statistical problems: freq. analysis (ex: natural lang) with letters or digrams (ex: “the”) → f.e Affine cipher: e(m) = (a\*m+b)%|A| with a & |A| relatively prime

**Describe the Homophonic cipher.**

Random string in a chosen set (f.e: “a” can be “01” or “10”). It mitigates freq. analysis

**Describe the Polyalphabetic substitution cipher and Vigenère cipher.**

Different transformation for every block (key is a sequence of substitutions). Vigenere is an example where the fun is the sum (first x letters shifted by n, then other x shifted by m, …)

**Describe the One time pads (Vernam cipher).**

key as long as plaintext, change every time (Unconditional but utopia)  
**Which is the idea of Homomorphic encryption? How is it obtained?**

*sec and efficient (client update triggers transf. of the cipher in server: no file transfer)*

**Explain the idea of Transposition ciphers.**

Set of all transformations like for block of length t and key-space set of permutations {1,..t}

→ factorial in t (length of msg, instead of length of alphabet) ; contro: letters unchanged

**Explain the idea of Composite ciphers.**

subst. → tranpos. → subst. → … by machine (good idea!)  
**What is the difference between block and stream ciphers?**

Stream cipher: bit/Byte at time VS Block cipher: each block is en/de-crypted, like a subst. on a large alphabet (>=64bits). Ideal is: huge subs. but would need a table of 2n entries for a n-bit block → so key size of n \* 2n → 2n! possible transformations.

**What is the idea of Feistel cipher structure?**

approx the ideal block cipher with a product cipher (comb of other simpler) → dev. a BC with a key-len of k and block-len of n bits = 2k possible transf. (rather than 2n!)

in detail: based on invertible block product cipher (partitions input block into 2 halves → multiple rounds with f an S-P network or any cipher. The right part, after the application of f with subkey, is xored with the left part. This result will be the new right part, while the previous right part becomes the new left part. → cipher is a concat → decrypt. uses same subkeys but in reverse order

**What are the objectives of Claude Shannon S-P ciphers?**

confusion & diffusion

**Describe DES and why it is deprecated today. Which version have been developed?**

Given 64b input and 56 bit keys (derived by key schedule alg): 16 rounds of Feistel f (with f consisting of two permutations & S-box subs) → permutation and then inverse.  
It has Avalanche Effect and it’s vulnerable by analytic & timing attacks (we have 2DES but it’s vulnerable by MeetInTheMiddle (known plaintext attack), and we have 3DES (secure, only 2112 brute-forces attacks) and even 3-Keys 3DES (even more secure))

**Describe AES.**

blocks of 128b, 3 possible key sizes that specifies #reps of transformations rounds (128, 192, 256), fast on SW&HW, S-P

**Describe the difference between Electronic Codebook and Cipher-Block Chaining**

In ECB mode the msg is split in m blocks, each encrypted with the same key at different times (same problem of substitution) → Limits: Info leak (= ciphertxt block sse = msg block) & limited integrity (decrypting doesn’t state if integrity is broken).

In CBC mode, at the beginning, InitVector xored with plaintxt, then encrypted with key → C1, also used in xor with P2 for the next phase → Prop: Chaining dependency (Cj depends on all precedents) & Self-Syncro (if error in Cj but not Cj+1 then Cj+2 is correctly decrypted)

**What are Stream ciphers?**

Used to encrypt data flowing. Same idea of OTP but with pseudorandom gen (seed as key not long as msg: even knowing previous bits the next one is unknown). Usually faster & ez to implement, and with proper design is as secure as a block cipher with the same key length.  
**Where to place encryptions? Which layers?**

Link encryption (1,2 OSI layers): each link has to be protected, but nodes can read plaintext

E-to-E encr: as we move up: more secure but more complex. Protect the entire path + auth

→ header is in plain (network has to route), traffic flow not protected (but there’s link encr.)

→ You don’t have to trust nodes (es:Routers) and that's good, but you have to trust the final host (a specific process, not the whole system!) → higher level of encr (ports, socket, …).

**Tell the possible countermeasures to traffic analysis.**

1) link encryption protects header details 2) use spourious/padding traffic to confuse

3) use anonymous traffic (TOR)

**Tell the difference between master and session key.**

Session key (data between users for one logical session), ephimeral  
Master key (to encrypt the Session key), f.e shared by user and the KeyDistrib.Center

**What are the properties to meet about key distribution?**

- secrecy of Ks - freshness of Ks - Key auth.

**Describe the Needham-Schroder shared key protocol.**

Fst Step) A send in clear his ID, B’s ID and a nounce N (to avoid Replay Attack)

Snd) KDC sends back E(Ka,[Ks|IDa|IDb|N1]|E(Kb,[Ks,IDa])) (A can open but not 2nd part)

Trd) A send the sh-key Ks and his ID to B (encrypted with Kb as he received it from KDC)

Fth) B sends a nounce encrypted with Ks to check the auth of A → A will use f to transform

Ffh) A sends f(N2) (f can be every function except identity)

**Why is message authentication important?**

Important to know exactly the origin of the msg. Prevent repudiability

**What are MAC?**

The MAC, or crypto checksum, is the output of the Msg Authentication Code method (the message is given in input to the crypto function (“not an encryption”), together with crypto key, in order to produce C(M,K) = MAC).

**Describe the concept of collision and why it can or not be problematic.**

the fun is N-to-1 (potentially >1 msgs with same MAC) but it’s really difficult: not a problem

**Describe the hash function**

hash code = msg digest, no key in input) in detail:

sending M, h(M) the intruder can trick B to believe that A sent M’, so I send E(Kab,h(M)) aka the fingerprint of M, in order to avoid it. However this isn’t 100% a sign of A, because Bob can generate itself Kab and so A can repudiate his own sign to trick Bob (limit of symm crypt). Note that, in order to check if A “is” the sender, B has to compute h(M), decrypt the crypted part and then check if they’re equal. Note also that i can send h(M||S) instead.

**What is and why is the digital signature important over auth? In what differs?**

It’s important to guarantee non repudiability (in terms of “internal” attacks) & integrity. Authentication states the absence of external msg manipulation so provides only security against external attacks (f.e see Alice repudiation of her msg to Bob).

**What are the properties of hash functions? Describe some applications of hash function. What is a birthday attack?**

1) H can be applied to a block of data of any sz 2) produces a fixed-len output

3) ez to compute (ideally linear) for a given x: not good in pwd context (for attacker is better)

4) ONE WAY: given y it’s computationally infeasible (c.i) to find a x such that h(x) = y

→ important in msg auth tech with a secret values (A→B:M|H(M|S)) (attacker deconstruct S)

5) WEAK COLLISION (pre-img) resistance: given x, it’s c.i to find y!=x such that h(y)=h(x)

→ important to prevent forgery when encr. hash code is used (A→B:M|E(K,h(M))), means that C can’t “sign” a document of not his own.

& protect pwd file storing h(p), often combined with salt ((s,h(s|p)) to protect from dictionary attacks (attacker could download pwd file & then offline try to match pre-computed hashes)

6) STRONG C. (2nd pre-img) res.: it’s c.i to find any pair (x,y) such that h(y)=h(x)

→ against birthday attack (known-plaintxt attack): h sz must be doubles if collision detection is important (with 2m search-space h must be applied 2m/2 so that p\_of\_collision > 0.5)

The attacker can be A itself (A prepares M, good for B, and M’ → A generates good msgs and bad ones, and she stops when h(xi) = h(yj) → B signs h(xi) but later on A uses sign for Yj (B has his bad msg, he wants to change words in the good one so that the changed good one (but with same semantics) has same hash of the bad one) → MD5 and SHA

**List some of the services provided by PGP.**

digital sig, msg encr., compression, email compat. (radix), segmentat.

**What is GPG?**

GPG is the GNU implementation of PGP  
**Explain the cryptographic functions of PGP (confidentiality and authentication)**1) **Auth only**: hash code of M encrypted by A with PRkey is concatenated with M itself  
→ then the dest. B compares hashed M with decrypted M (with PUkey (everyone can))  
2) **Confidentiality only**: M compressed → encrypted with a brand new fresh Ks → results concat with Ks itself (but encrypted with public key of B, so that only B can recover Ks decrypting with his private key), then B decrypt with Ks → key exchange problem is no more  
3) **both**: “(1) with (2) built in inside” **How is a message generated in PGP?**see slides **Data protection: which kinds exist? What’s the problem of key-rings integrity?**Data protection can be in transit, at rest or in-use (f.e in memory). The Integrity of the public key-ring (the place where public keys are stored) is crucial becouse C can substitute B’s PUk with his own. Also the integrity of the private key-ring is obv crucial, so we must protect it with a private key, usually consisting of an hashed passphrase keeped by the user.  
**What are the requirements in order to apply public key encryption?  
1)** comput. ez for B to generate pair (PUb, PRb) **2)** comput. ez for sender A, knowing PUb and M, to generate C=E(PUb,M) **3)** comput. ez for receiver B to decrypt C using PRb to recover M **4)** comput. infeasible, for attacker, knowing PUb to determine PRb **5)** comput. infeasible, for an attacker, knowing PUb and C=E(PUb,M) to recover M **(!)  
6)** 2 keys applicable in 2 orders: D(PUb,E(PRb,M))=M=D(PRb,E(PUb,M)) (not alw. necess.) **What are one-way functions (in this context)?**ez to compute ∀x but f-1 is hard  
**What trapdoor means?**Is a one-way function parametric in k such that x= f-1 is ez if both y & k are known **What kinds of attack can be done against this kind of encryption?  
1)** brute force (counterm. is using large keys, but we need a tradeoff → in practice pubk confined to key managm. & digital sign) **2)** computing PR from PU (no proof that is infeas.) **3)** probable msg attack: short M is sent encrypted with PUa. The attacker computes all Yi=E(PUa,Xi) ∀ possible plaintxt Xi as soon as Yi=C → means that M=Xi → solution? attach random bits to M **What modular airthmetic and euler totient function are applied to?**RSA (see slides if u don’t remember stuff) **Explain how RSA algorithm works.  
Alg**: **1)** keypair gen   
 **(1)** gen. 2 large p&q (primes) **(2)** n = pq & Φ=(p-1)(q-1) **(3)** select e ∈ (1;Φ) rel prime  
 to Φ **(4)** find d such that ed modΦ = 1 **(5)** publish (e,n), keep (d,n) priv., discard p&q  
**2)** encr. with (e,n)  
 (1) break M into M1,... with Mi<n (2) Ci=Mie modn  
**3)** decr. with (d,n): Mi=Cid modn **What does mean that RSA is malleable? Is it a problem?**  
An attacker can obtain another C by transforming the first C applying F (the other C is likely a result as if computed on a message G(M)), with a message chosen by him.  
So ciphertext can be altered and that can be a problem. As a countermeasure we can attach some random bits (before computing RSA) such that if an alteration occurs, it is detected. **What are the problems about key distribution with RSA? What is the PF property?**Problem1: how can B know S auth? We must ensure auth here → digital signatures  
Problem2: PRk of S has to be protected at rest, but we can’t ask everytime for a passphr. We can’t solve it! Is the problem only for future communications? No, it’s retroactive due to accessing logs with PRk C has discovered → Diffie-Hellman (perfect forwarding secrecy)  
→ solutions to Problem2: 1) not storing Ks in the server, 2) Diffie Hellman  
**Describe the Diffie-Hellman algorithm. What is man in the middle attack?**Security depends on difficulty on computing discrete logs.  
Alg: 1) sharing of prime q & primeRoot a of q (both may be public), 2) A&B randomly generate Xa & Xb (both <q), 3) A computes Ya=aXamod(q) and B computes Yb=aXbmod(q), 4) A&B exchange results 5) A computes Ka=YbXamod(q) and B viceversa → Ka=Kb=Ks  
Pro: better than RSA bcs Ks is created out of nothing & never transmitted: PFS (C can’t recover prev communications even “knowing Ks”) …  
Contro: no auth (suffers from MIDattack, with C substituting with his Yc) → solution is using RSA combined with D-H in order to sign the exponents. **What is the HW security model? Why is it useful with RSA?**Module with ez interface (hopefully) that decrypts E(Kb, Ks) with PRb **What are some possible approaches to protect private keys in digital signature**Smart Card, USB, Remote Signature **What is man-in-the-middle attack and why digital certificates are useful to counter that?**C can take the msg between A and B and substitute the real PUk with his own, so now A is “securely” communicating with C. Digital Certificates are used to serve key distribution & auth with no repudiaton, verifying auth of PUk by binding PUk to the owner name. We write (B, PUb, CA)[PRca] as DC of B issued by CA **Describe generally the PKI.**Starting from the server, it requests DC from RA → CA generates it. In order to assure CA’s auth we need an IssuingDC from a root CA. The root CA provides rootDC, that is the CA’s public key list, to software vendors (this rootDC will be provided to final users by the sw vendors). We also have CRL (certificate revocation list), bcs before validating certificates the Browser check if it isn’t in the CRL (or better, asks the OCSP responder that avoids overhead for browsers that should check the whole (often huge) list). Then there’ll be PUk exchange from Bob to Alice + digital certificate → then exchange of msg from A to b. **Describe the two types of Digital Certificates.  
D**omain **V**alidation: CA verifies if requester has control of the domain  
**O**rg/**E**xtended **V**alidation: CA also verifies real-world ID **What is a replay attack?**C re sending A’s msg after some time → freshness needed → security protocols **- What is man-in-the-middle attack and why are digital certificates useful to counter that?**In the MID attack an intruder puts itself in the middle of the communication between A & B, trying to do some malicious operations (so it’s an active attack). We can use DC to verify the communication, so that A can be sure that he’s talking with the expected agent (thanks to the binding of B, PUb, CA encrypted with the private key of the TRUSTED CA that obv can only be decrypted by the CA public key). More in the next question **- Describe generally the PKI.**CA is divided between an interface (“front-end”) for accepting requests (Registration Authority, that is seen by who request) & the inside part for generating (“back-end”).  
We also have CRL (certificate revocation list): before validating certificates, the Browser check if it isn’t in the CRL (or better, asks the OCSP responder that avoids overhead for browsers that should check the whole (often huge) list).  
**- Describe the two (three) types of Digital Certificates.**Domain Validation: CA verifies if requester has control of the domain  
Org/Extended Validation: CA also verifies real-world ID  
?? **- What is a protocol?**A set of rules describing precisely the communication between two+ agents **- What are the goals of security protocols?**Using crypto mech. to achieve security objectives **- Descibe the Dolev-Yao attacker model.**It’s the most powerful attacker model, in which the attacker has the total control of the channel (Strongest possible assumptions about the attacker)  
**- List kinds of attacks that exist in this context.**MiD: see above  
Replay Attack: replay a previous msg (es: Needham Schroeder)  
Type Flaw Attack: substitute type semantics to force a different interpretation of msg  
Reflection: your own msg can be used against yourself **- What is NSPK protocol? Which attack suffers and why?**It’s used to obtain mutual (entity) auth.   
Nominal run: **1.** A→B:{A,Na}Kb **2.** B→A:{Na,Nb}Ka (Bob’s auth) **3.** A→B:{Nb}Kb (Alice’s auth)  
→ recall principles can be involved in multiple runs. Goal should hold in all interleaved protocol runs → **MID attack on NSPK** is possible with 2 concurrent runs, with A trying to communicate with C, C decrypting A’s msgs and re-encrypting them with Kb to let B think he’s speaking to A. The solution is adding B’s ID in the second msg to prevent reply as it is. **- Describe the Otway-Rees protocol and its vulnerability.**It’s used for auth. key distribution + fresh, but not entity auth & key confirmation. It suffers Type Flaw + Reflection (your own msg defeats u). Here 2 kinds of attack:  
 1. TF+Refl: after step1 A is in wait, so C can replay s1 in s4 and, if |Kab|=|I,A,B|, A  
 will see I,A,B as the key because they have the same length  
 2. Reflection: NO key auth/secrecy: C play S’s role  
→ solution: preserve typing info, assigning codes to types & (ap|pre)pending the code **- Which is an example of a similar protocol that suffers from the same kind of attack?**Andrew Secure RPC protocol **- Explain how key-exchange with CA (Denning & Sacco) works and how to fix its vulnerability.**A→S: A,B S→A: Ca, Cb A→B: Ca, Cb, {{Ta, Kab}ka(-1)}kb  
→ MID attack (B believes that was sent by A, so Kac will be used instead of Kab → C can overhear) → countermeasure: be explicit about purpose: include A,B ID’s in the last step **- List some of the principles of Abadi-Needham.**see slides the day before  **- What is Kerberos? What are its aims?**auth for C/S (mutual), based on NSSK but without nested encr & T instead N  
Aims: user pwd never travel over the net & never stored on client’s machine (discard after usage) & never stored un-encr even on the server → Single Sign-On: user asked for pwd once x session (but not for every service) → auth info only in auth S, not in AppS  
Req: secure, reliable, transparent, scalable  
Architecture: KerberosAuthenticationServer & TicketGrantingServer (authorization)  
**- Describe the three phases of Kerberos.**see notes & slides **- What are the limitations for Kerberos IV?**Limit. of Kerb IV: DOS (C can flood S), nested encr, relies on clocks

**## Internet Security  
- How is the transport layer implemented in SSL and IPsec?**In SSL, after tcp/ip socket there’s the ssl interface (a superset of socket’s API)  
IPsec instead puts itself under the transport layer. **- Describe IPsec with AH, ESP, IKE.**IPsec is implemented with three options. AH is the auth header and it serves both auth and integrity, while ESP serves confidentiality and, optionally, integrity. AH & ESP can be either in transport or tunnel mode. The first one, put after the IP header, is usually used end-to-end by the single host implementing IPsec. The second one, protecting all the IP header, is used for VPN’s.  
IKE is used for exchanging both keys and SA (security assoc.) that contain all the useful info for the exchange of data between the principles (as type of crypto, keys, …)   
**- Diffie-Hellman for IKE. Why?**For exchanging keys with perfect-fowarding secrecy.

**## Buffer overflow  
- Describe the threat and the possible solutions.**threat: overflowing the memory zone allocated for your data  
solutions: canary (value for integrity check before the intended return zone), code design, stop using deprecated bs,…

**## Web Security  
- Describe the HTTP protocol.**[📦](https://emojiterra.com/it/pacco/) **- What are cookies, why are they useful?**[📦](https://emojiterra.com/it/pacco/) **- Describe the two types of HTTP authentication.**Basic authentication: Login/password based.Information is sent unencrypted. Credentials are sent on every request to the same realm. Supported by nearly all S/C. Widely used.  
Digest authentication: Server sends nonce. Client hashes nonce based on login/password. Client sends only cryptographic hash over the net. Seldom used  
**- What is an injection flaw by unvalidated input?**A threat caused by a malicious piece of executable, injected by an attacker via an user input that hasn’t been validated server-side by the programmer. **- What is SQL Injection?**a’ OR ‘1’ = ‘1 **- What is XSS?**A type of attack called cross site scripting, in which the attacker tricks the user to click a link that steals his legit cookies, f.e by js malicious code that alerts the content of document.cookie. This is done in order to send them to an attacker script that is listening for catching the cookies. **- What is CSRF?**A type of attack called cross site req forgery, in which the attacker tricks the user to do an unwanted action on a legit website (f.e buying lothar copies on Amazon)  
**- What is Clickjacking?**Tricking the user to interact with something visible that is different from what he thinks. **- What does Broken Access Control vulnerability mean?**Bad design leads to vulnerabilities → Broken Auth, Session Managment → see slides

**## Access Control  
- What is a security policy? Describe the three main categories.**A relationship between subjects & objects that defines what is allowed. A security model provides a formal representation of a security policy, while a security mechanism defines the low level sw/hw functions that implements the controls imposed by the policy and formally stated in the model. ACL can be: DAC (based on ID of requestor), MAC (based on rules by a central authority), RBAC (based on roles that users play). DAC and RBAC are usually coupled with an administrative policy that defines who can specify rules governing AC  
**- What is a reference monitor?**  
Every request passes through a trusted component called reference monitor (with these properties: tamper-proof (impossible altering it or at least with accountability), non-bypassable, security kernel (confined in a limited part of the sys), small). **- What does it mean that a system is 'secure'?**A secure system is a system that starts in an authorized state and can’t enter an unauthorized state. A state of a system is the collection of current values of all its components. Let P be the system state space and Q⊆P be the states in which system is authorized to reside in. A security mechanism prevents a system from entering P\Q. **- How does UNIX manage control access?**Simple ACLs (actually triples with scheme owner/group/other) → chown/grp/mod  
File Mode Bits: file permission bits & special mode bits (4 executable files & dirs)  
Special bits: When a non-owner executes the file, the process will run with user and/or group permissions set upon it by its owner. If used on a folder (bit t), files can be accessed as specified by the owner but they can’t be deleted. **- What vulnerability is DAC open to?**Trojan: Discretionary policies DP do not distinguish users from subjects. Once connected to the system, users originate processes (subjects) that execute on their behalf. So DP evaluates all requests submitted by a process running on behalf of some user. Once this information is acquired by a process it is possibleto leak info to users not allowed to read it. **- Describe MAC and tell some model examples.**AC controlled by comparing security labels (indicating criticality of objs) with formal authoriz of subjects. Mandatory bcs subjects may not grant (> rigid than DAC but > secure).  
2 principles to hold for confidentiality: Read down (a subject’s clearance must dominate (>=) the security level of the obj being read, while a subject’s clearance must be dominated (>=) by the security level of the obj being written **- What is RBAC?**User Assignment with User-Role & then Permission Assignment with Role-Permission  
→ hierarchies simplify / As DAC an admin is needed to manage static rules in times (adm. policies)  
**- Roles VS Groups**G defines sets of users while roles define sets of privileges. R can be “activated” and “deactivated” by users at their discretion. Gmembership always applies, that is, users cannot enable and disable group memberships (and corresponding authorizations) at their will  
 **- For which purpose is ARBAC used?**  
Adm. policies (modifying UA): UEmployee : {Student, !TA} => PTEmployee (a UEmp can become a PTEmp as long as he doesn’t have another job yet) → so that means that this can be done