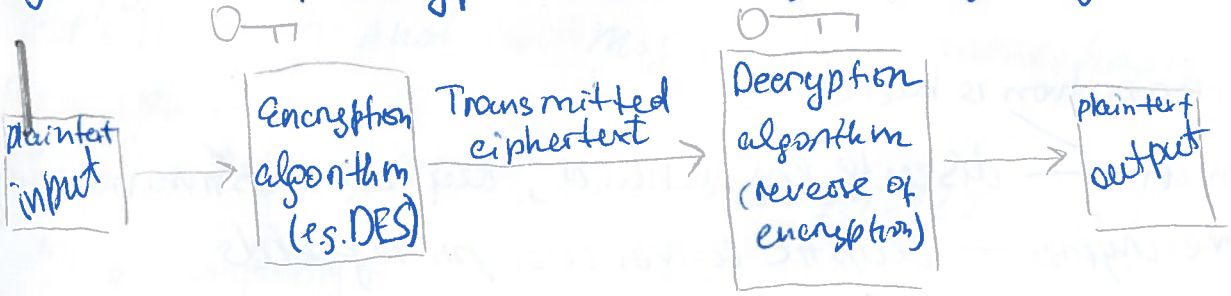
lecture 14

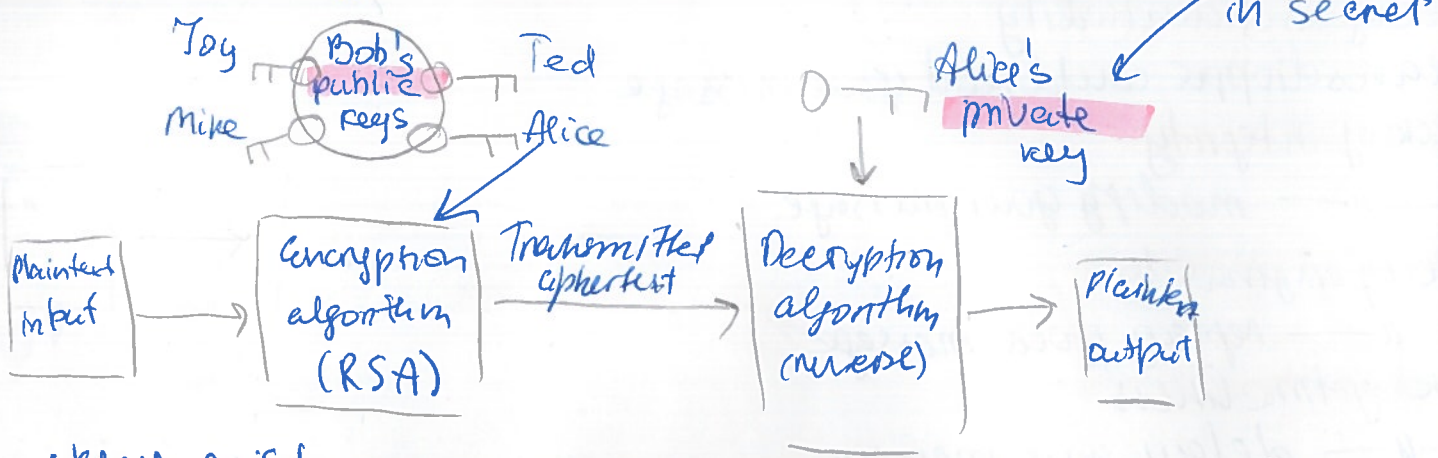
- Case study: online shopping
  - Lack of confidentiality
    - eavesdropper could read your message
  - Lack of integrity
    - —"— modify your message
  - Lack of originality
    - —"— replay your message
  - Lack of timeliness
    - —"— delay your message
  - Lack of auth
    - attacker might redirect user unknowingly to another website (by DNS)
  - Lack of access control
    - —"— gain access to the web server and change its contents ("defacing") or download stored credit card data
  - Lack of availability
    - ~~eg~~ DDoS, simultaneous shopping
- Symmetric en/decryption (secret-key cryptography)



- secret key shared by sender & recipient
  - Without knowing key, it is not possible to derive from ciphertext to plaintext
  - same key for both encryption & decryption
  - keys must remain secret
- Examples:
  - DES – length: 56; old
  - Triple DES – length: 112/168 (DES 3 times in a row)
  - AES/Rijndael – new standard America

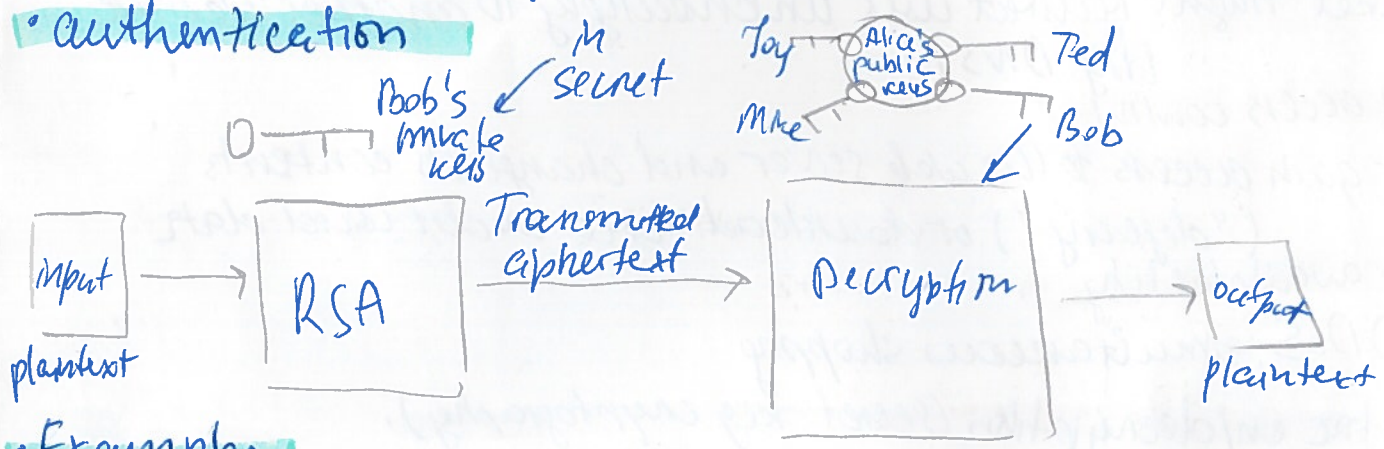
Public key cryptography (assymetric cryptography)

Confidentiality



- keys exist in pairs; one for encryption and other for decryption
- can't derive one key from another

Authentication



Examples

- RSA — factorization is heard
- Diffie-Hellman — discrete log. is hard; key-establishment
- Elliptic-curve crypto — elliptic curves over prime fields

- Symmetric algo are much faster than public key
- Usually used in combination:
  - public key for auth and set up of temporary public key
  - symmetric algo for confidential transfer of the actual user, using temporary key

Cryptographic hash function

computes a fixed-length "hash" value from an (arbitrary long) input
same input gives same output
given hash it is hard to construct corresponding input

Applications:

- signatures in e-mail (encrypted hash of message)
- test of integrity of files
- password storage

instead of verifying integrity/auth of a long message, verify the integrity of the hash of the message

Security requirements:

preimage resistance / "one-way res"
second preimage resistance / weak collision resistance
- given a msg, it's hard to find another message giving the same hash

strong collision resistance
- it's heard to find two msg giving the same hash

Examples

MD2; MD4; MD5; — hash length: 128; basic unit: 512
- 128 complexity 3

SHA-1, SHA-2, SHA-5 — hash
- complexity 2^{63}

RIPEMD

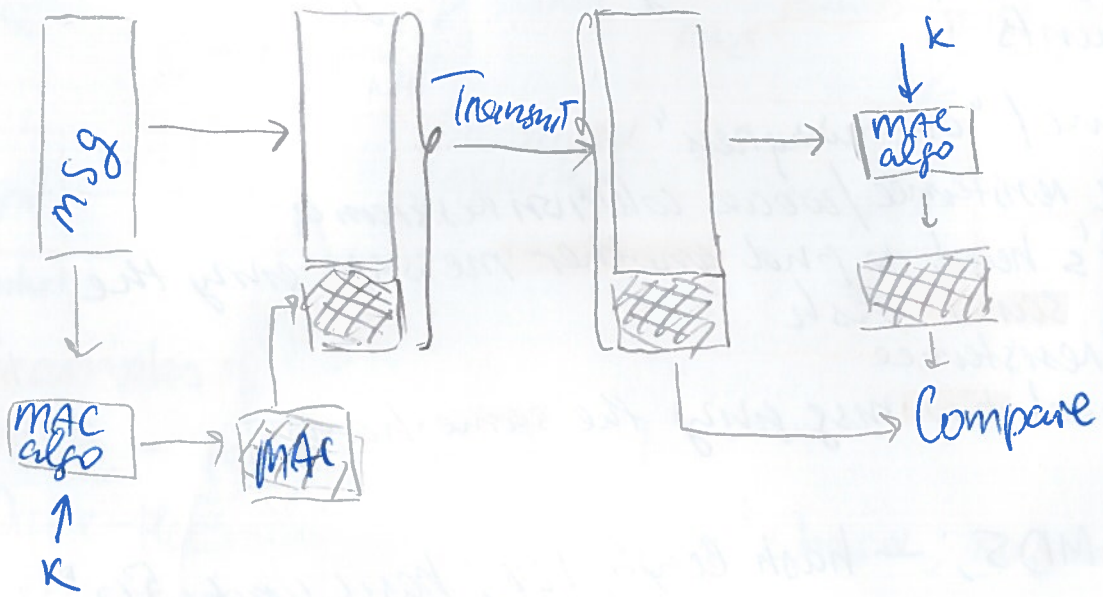
no attacks

• Message authentication code (MAC)

- fixed length value calculated from
  - var-len msg
  - fixed-len key

such that
- without knowing the key, it's hard to calculate the MAC, or update the MAC after making a change to the msg.

- Purpose:
  - prevent adversary from forging or changing msgs;
  - provides auth and integrity
- Similar to digital signature, but based on symmetric cryptography



ey (pre) distribution

articipants need to know what keys to use
ymmetric cryptography: how to securely share a key?
ublic-key cryptography: how to be sure about which key belongs to whom?
distinguish between short-lived session keys and longer-live pre-distributed keys

ssion keys

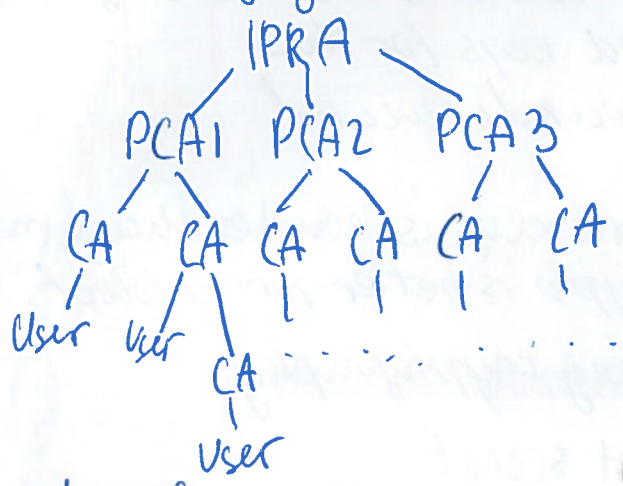
sed only during single, relatively short episode of communicatn ("session")
always symmetric cryptography, for speed
ach session gets new key
determined using a protocol
that protocol needs to be secure: uses the longer-lived pre-distributed keys for this
limited damage if key is found/revealed

Pre-distribution of symmetric keys is harder than public keys, but symmetric crypto is better for en/decr. speed.

y-distribution for public-key cryptography

ublic keys need not be kept secret
ut they do need to be authenticated!
therwise, man-in-the-middle attack is possible
uth: different ways
- only trust keys that you have personally verified in real life
- Also trust keys that have been personally verified by someone whose key you have personally verified
- etc.

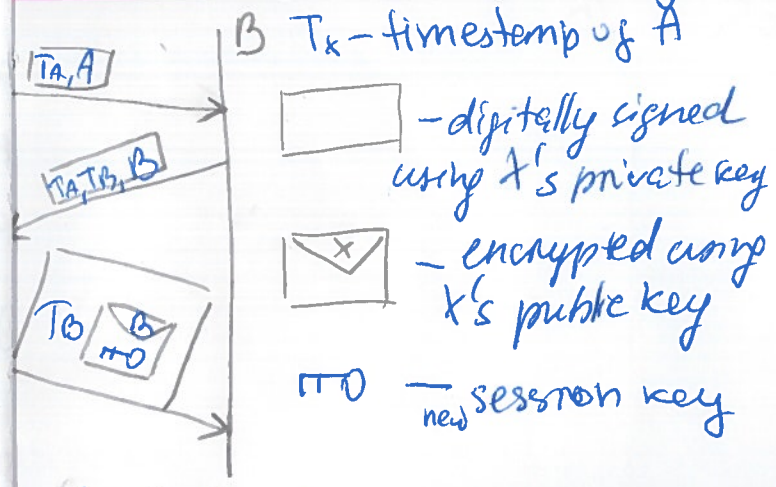
- X.509 Certificate
  - Contains fields:
    - subject name: whose certificate is this?
    - subject's public key
    - period of validity
    - issuer's identity: who created this certificate?
    - signature: hash of the other fields, encrypted with the issuer's private key
  - If you have the issuer's public key, you can verify the certificate. Then if trust the issuer, you can also trust the certificate subject's public key
- Public Key Infrastructure (PKI)
  - Two ways:
    - Very Hierarchy of Certificate Authorities



Used for example https web auth

    - Web of trust
      - Users exchange certificates at real-life meeting
      - Users decide for themselves how much they trust each certificate (or actually its owner)
      - Used in PGP (Pretty Good Privacy; software for email security)

th and key exchange using PKI



y distribution for symmetric cryptography

$\frac{n(n-1)}{2} \approx n^2/2$ keys are needed and must be exchanged securely between n parties

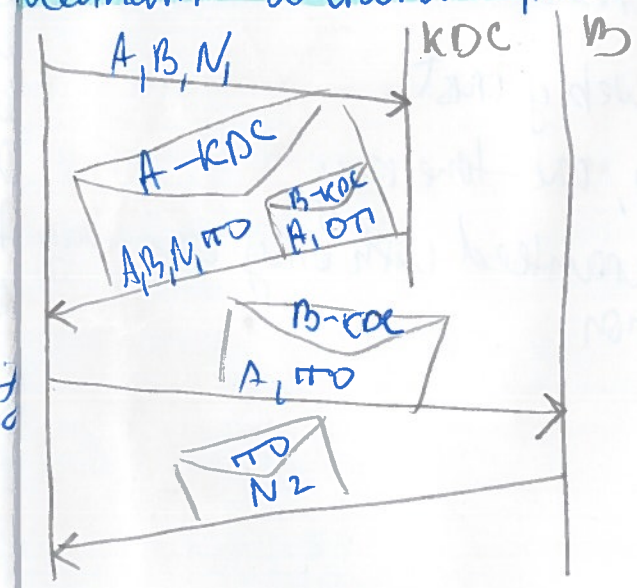
y distribution Center

Principle:

- each party gets one secret key known to him/her and the KDC
- KDC generates session keys, and encrypts them with the secret keys

careful design needed to prevent replay and man-in-the-middle attacks

Needham - Schroeder protocol

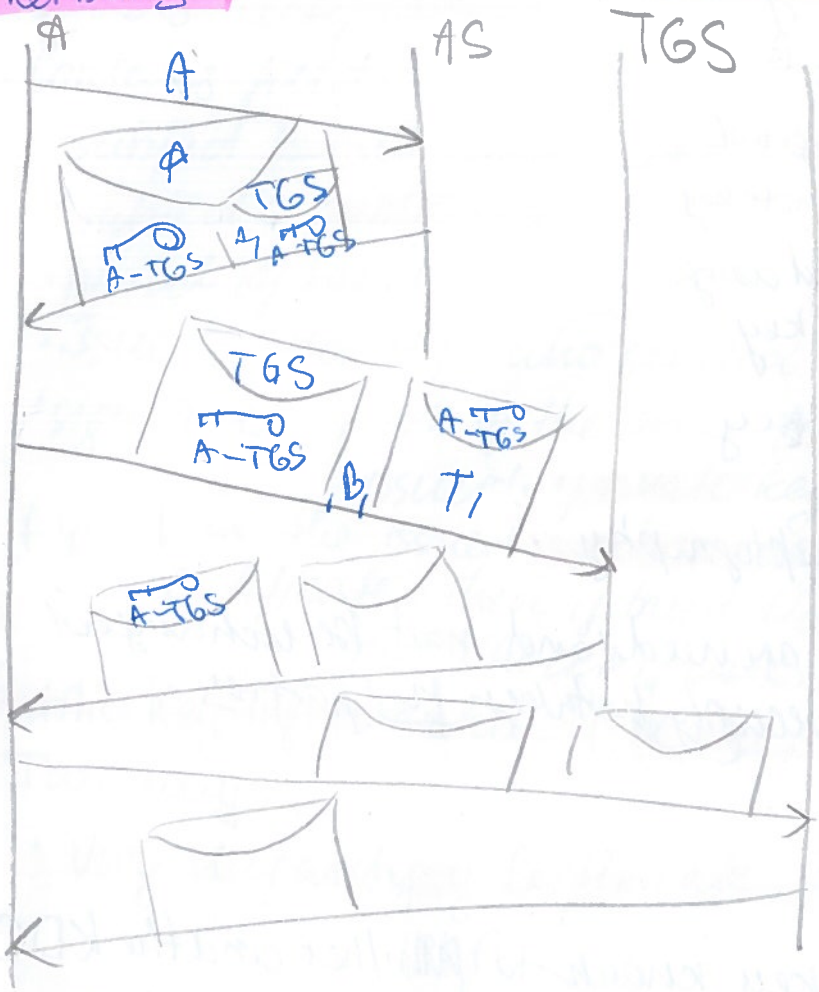


K-Y - encrypted using key shared by X and Y

OTT - new session key

- Advantages
  - only n keys for n parties
  - real secure channel needed only once for each party
- Disadvantages
  - KDC - single point of failure
  - If secret key is compromised, all earlier communications of that party is also compromised

- Kerberos



- PGP
  1. Digitally sign using A's private key
  2. Encrypt using a newly generated one-time session key
  3. Encrypt session key using Bob's public key and append that
  4. Use base 64 encoding to obtain ASCII

- Key distribution: public keys using web of trust
- Encryption: symmetric cryptography, one-time key
- Timeliness, replay: cannot be guaranteed with only one-way communication