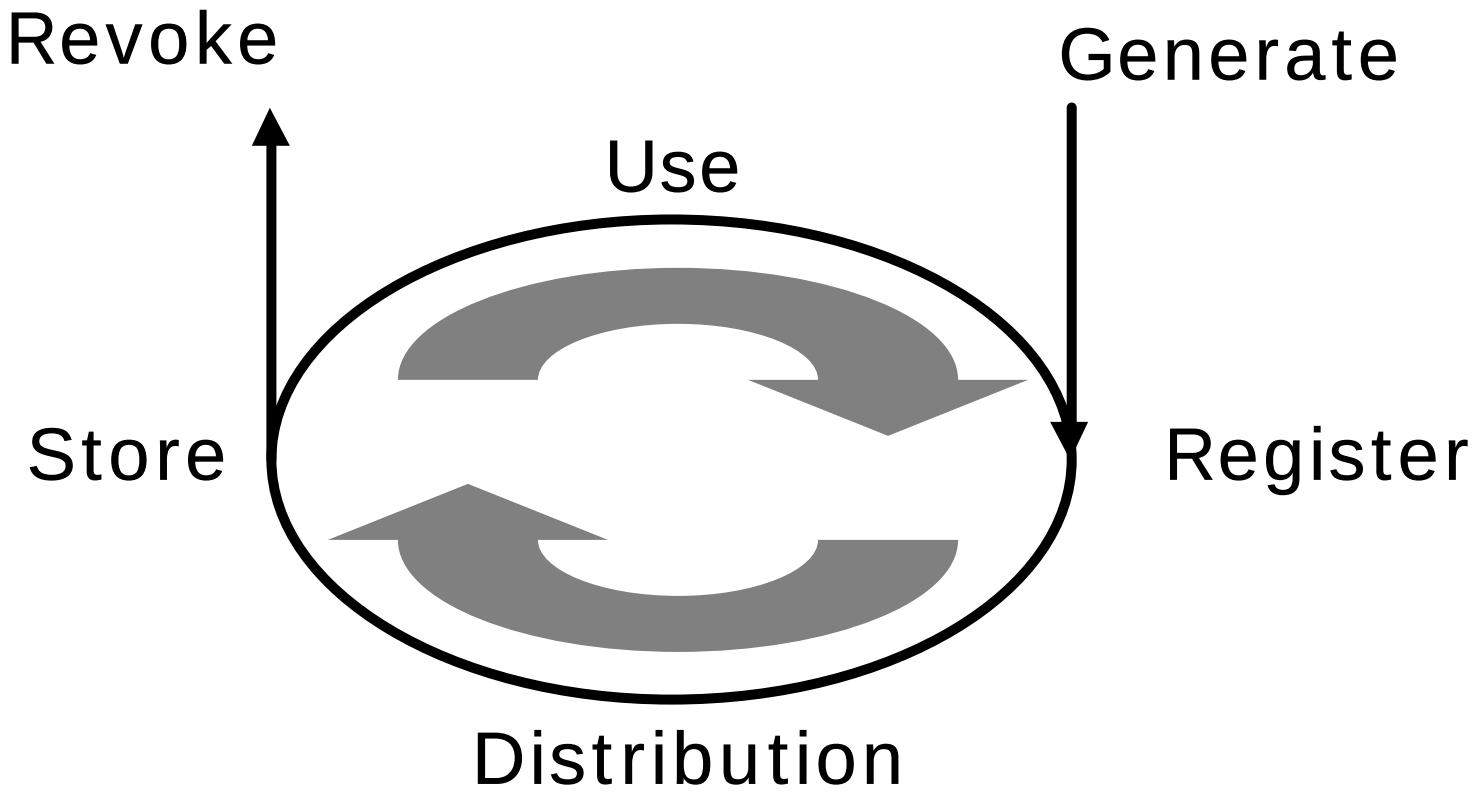
Key Management

Outline

- Definition of the key management problem
- Solutions based on SK (*Kerberos*)
- Solutions based on PK (*PKI and PGP*)
- SK vs PK

Key life cycle

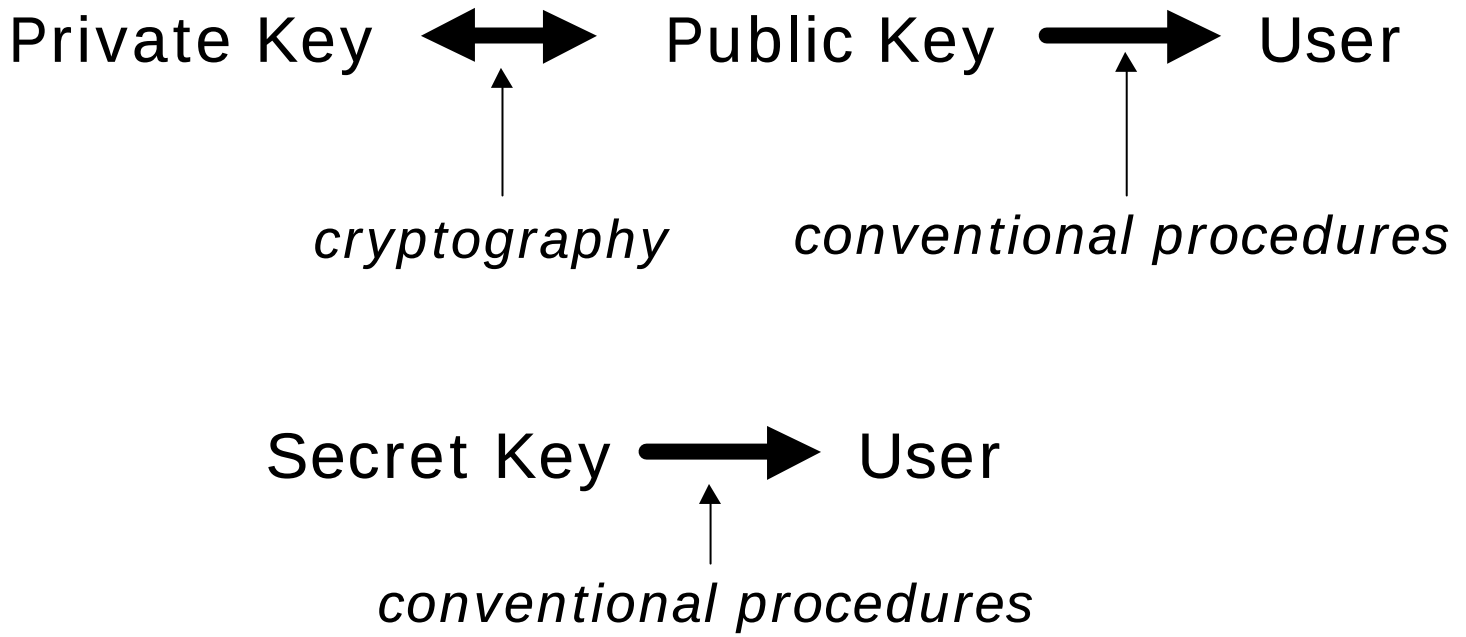


Key management problems

- Key generation
- Key registration
- Key storage
- Initial key distribution
- Electronic key distribution
- Key revocation

Key Registration

Binding keys to users (people)



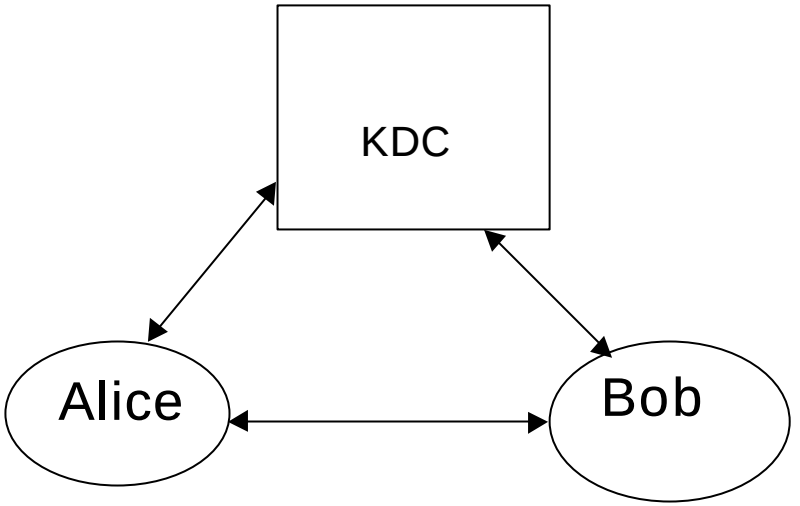
Chains

Agent(public key) $\rightarrow$Terminal(public key) $\rightarrow$
user's public key $\rightarrow$ user

Key Registration

- Initial registration *must* always involve a physical meeting to verify user's *conventional* identity (via passport, etc.).
- Initial registration *must* always produce conventional evidence (i.e. letter, contract, etc.) for both the user and the registration authority
- Evidence must be stored securely to prove the legitimacy of actions

Key Registration



- With PK users need to register their public key and to get server's public key. *Integrity* is needed
- With SK users and server need to exchange a shared key. *Secrecy* and *Integrity* are needed

Key Storage

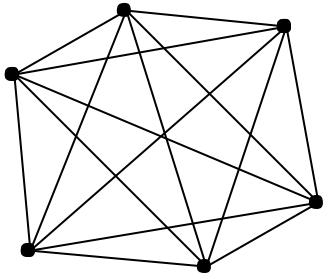
- To guarantee high security, keys (private or shared) need to be *generated*, *stored* and *used* in trusted environments.
- An environment is trusted if can be accessed only by the legitimate user.
- Ideally a trusted environment belong to a single user for all its lifetime so it has only a single legitimate user.
- It guarantees secrecy and integrity

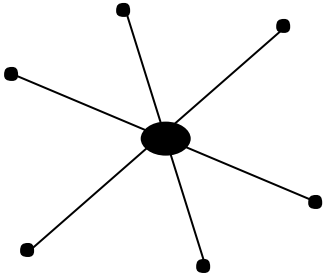
Trusted Environment

- Software cannot be considered a trusted environment
- Two types:
  - Tamper resistant.
    - Prevent unathorised access to keying material. Only theoretical
  - Tamper evident.
    - Detect unathorised access to keying material. Smart-card, Token usb, crypto processor

Initial Key Distribution

- **N** users
- Without any server **N(N-1)/2** channels are required for full connectivity (from the system point of view)
- With a server only **N** ® scalability





Electronic Key Distribution

- Initial ditribution is expensive cause requires *off-line* procedures
- Once bootstrapped the system can proceed cheaply with electronic ditribution and only *on-line* procedures
- Also the *electronic key distribution* phase makes use of the central server

Trusted Third Party

- The central server
  - stores keys
  - registers the binding between key and owner
  - distributes correct keys to requester
  - sometime generates also good session keys

- Thus the central server is not only a third party but need also to be *Trusted* for some tasks.

Public-key cryptosystem (PK)

- When AK crypto algorithms are used under the assumption that the encryption key (e) is public and the decryption key (d) (use and knowledge) is kept private, we refer the system as public-key based crypto systems.

Trusted Third Party

- In SK-based systems this is usually referred as Authentication Server

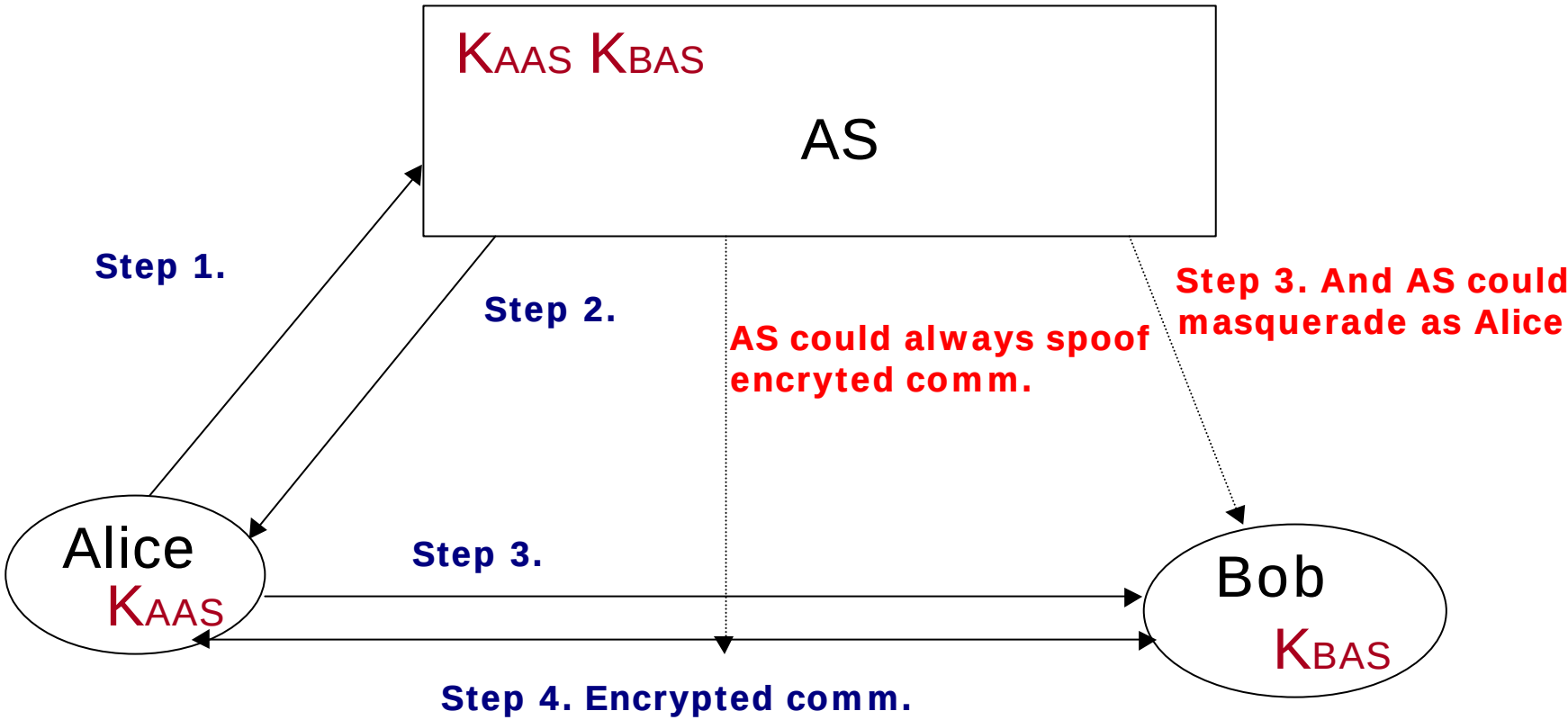
- In PK-based systems this is usually referred as Certification Authority

Authentication Server

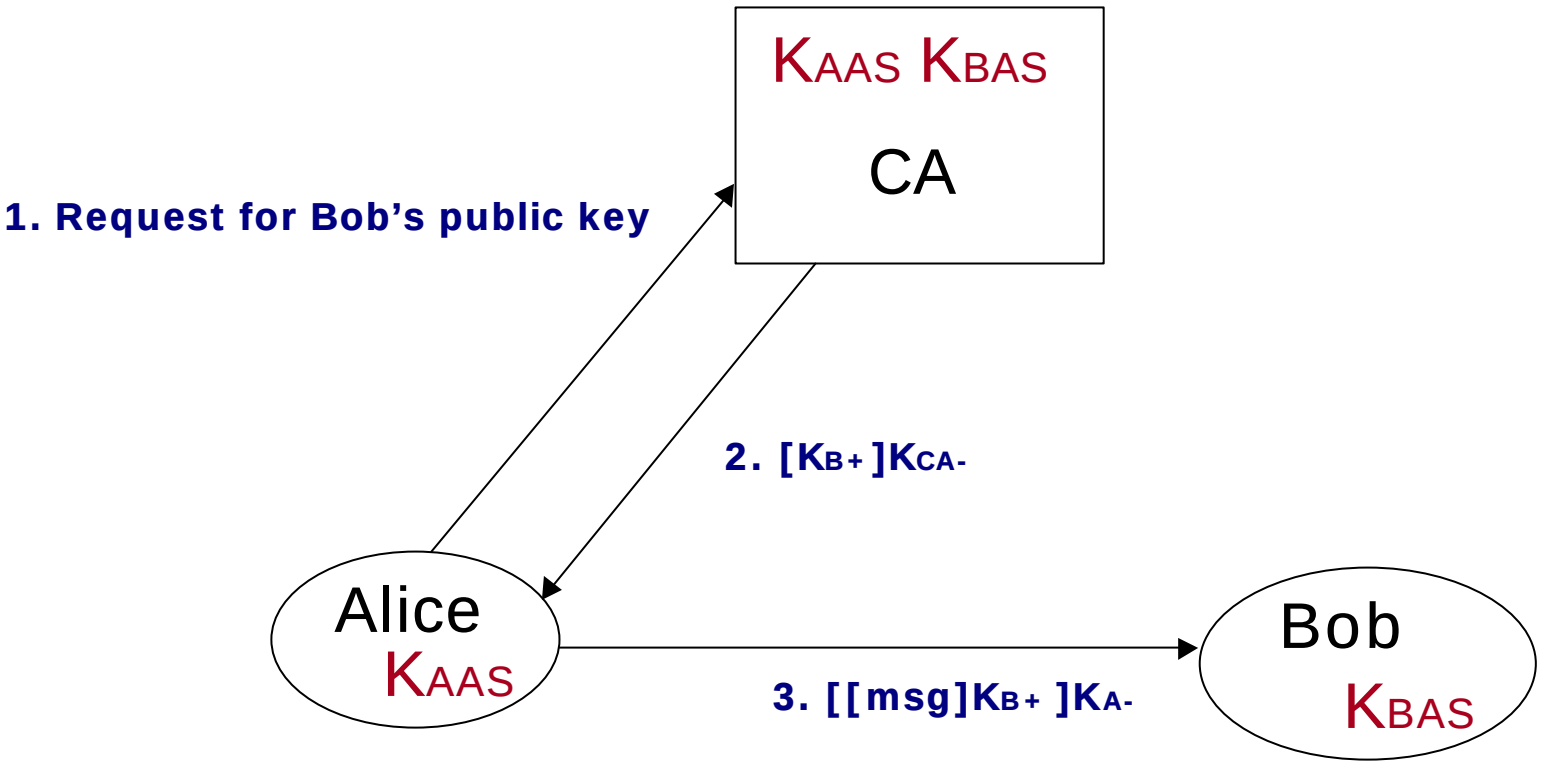
trusted to

- Initial registration of users (revocation of keys)
- Store shared key
- Confidentiality of shared key
- Generate good session keys
- Forget session keys once distributed
- Not masquerade as a user
- Execute correctly the protocol

Authentication Server



Certification Authority

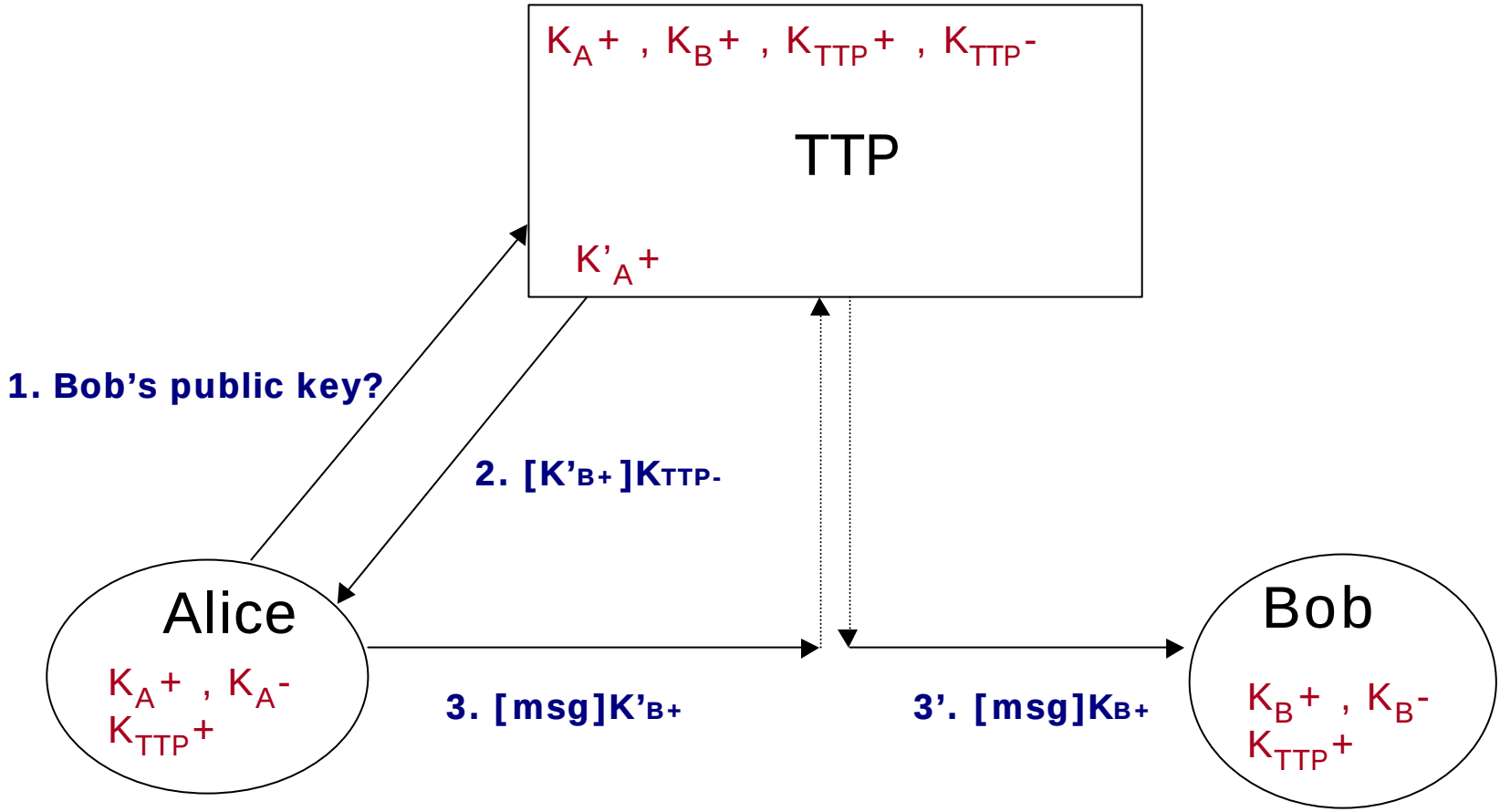


Certification Authority

trusted to

- Initial registration of public key to user
- Store public keys
- Distribute correct public key
- Integrity of public keys
- Revocation of public keys
- Execute correctly the protocol

Trusted Third Party



Key revocation

- Keys can be stolen, damaged, lost, forgotten thus the necessity to be revoked
  - With SK each key is shared only with one other party so revocation involve only two parties
  - With PK the public key is shared with N-1 parties so revocation involve N parties