School of Computer Science and Engineering, UNSW

WK03-01: Authentication, PKI

Securing Fixed and Wireless Networks, COMP4337/9337

Never Stand Still

Sanjay Jha, Nadeem Ahmed

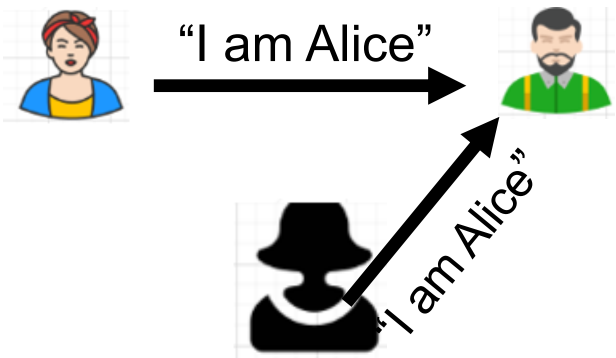
Authentication

- Goal: Bob wants Alice to "prove" her identity to him
- Protocol ap1.0: Alice says "I am Alice"

Failure scenario??

Authentication

- Goal: Bob wants Alice to "prove" her identity to him
- Protocol ap1.0: Alice says "I am Alice"



In a network,
Bob can not "see" Alice, so
Eve simply declares
herself to be Alice

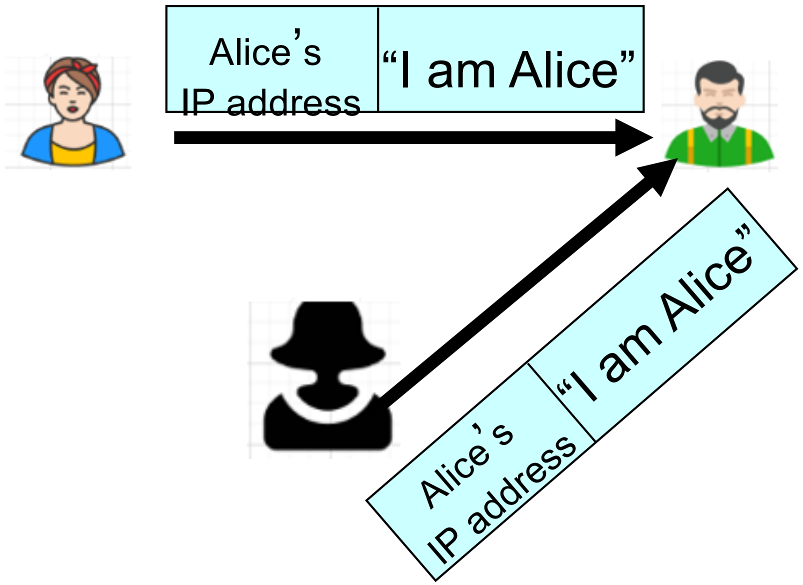
Authentication: Another try

- Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address

Failure scenario??

Authentication: Another try

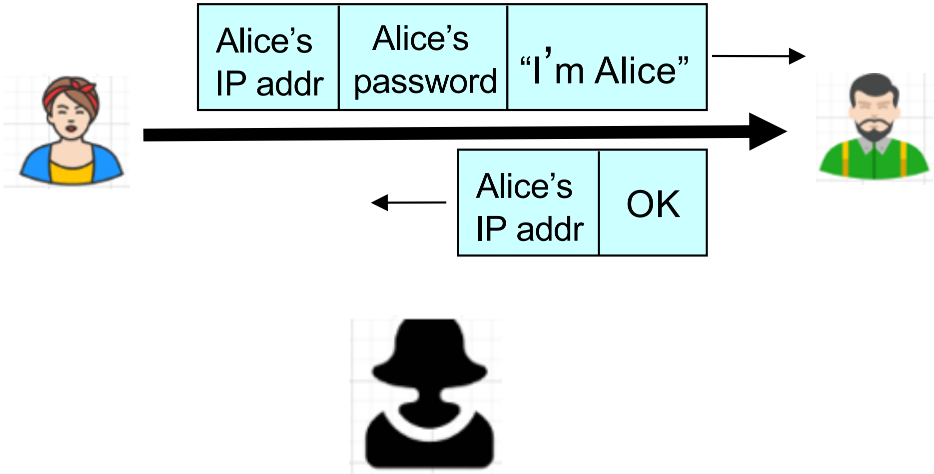
- Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address



Eve can create a packet "spoofing" Alice's address

Authentication: Another try

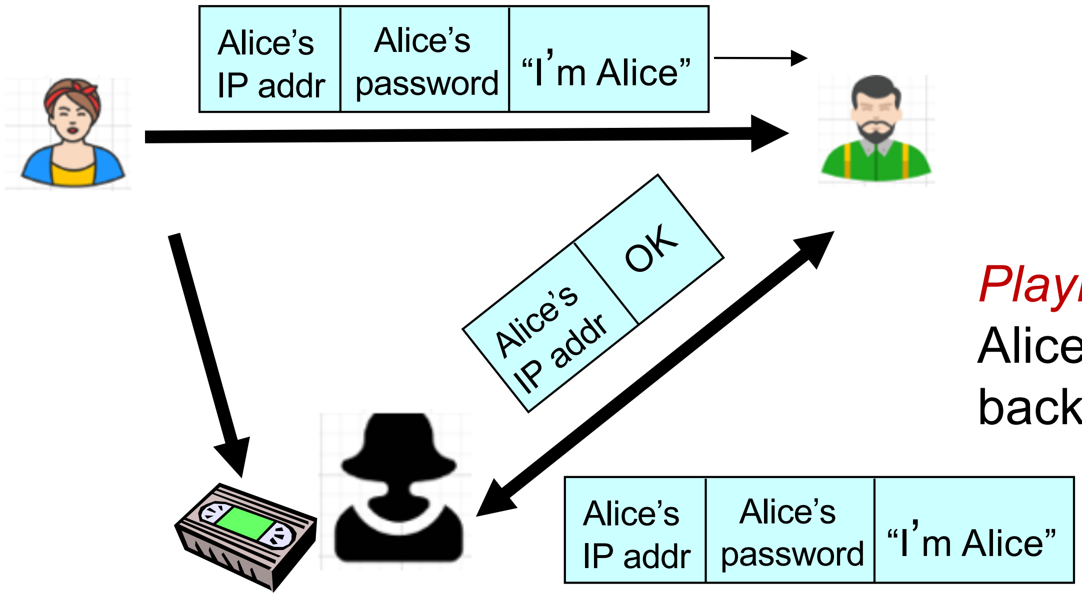
Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.



Failure scenario??

Authentication: Another try

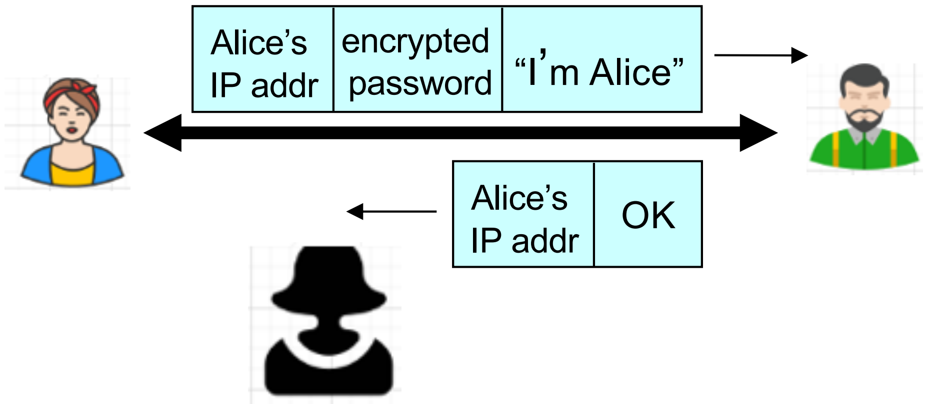
Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.



Playback attack: Eve records Alice's packet and later plays it back to Bob

Authentication: Another try

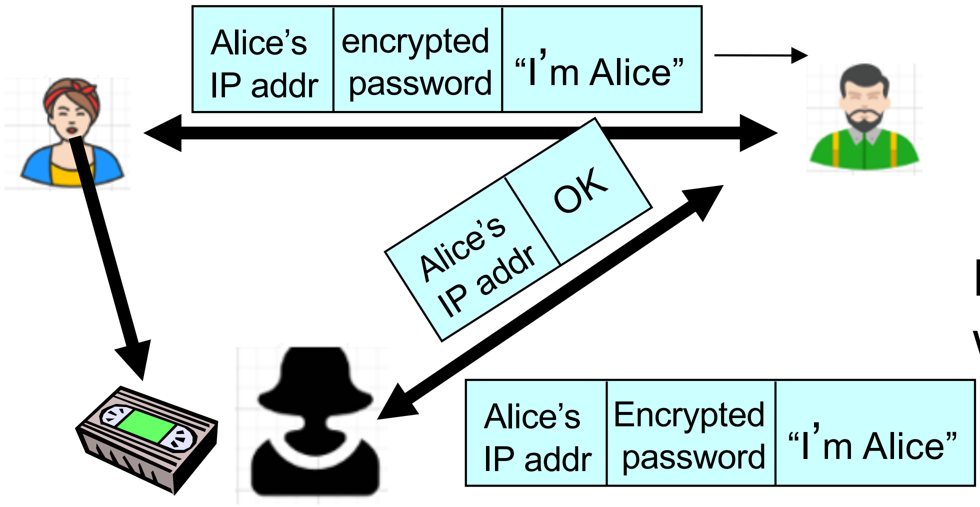
Protocol ap3.1: Alice says "I am Alice" and sends her encrypted secret password to "prove" it.



Failure scenario??

Authentication: Another try

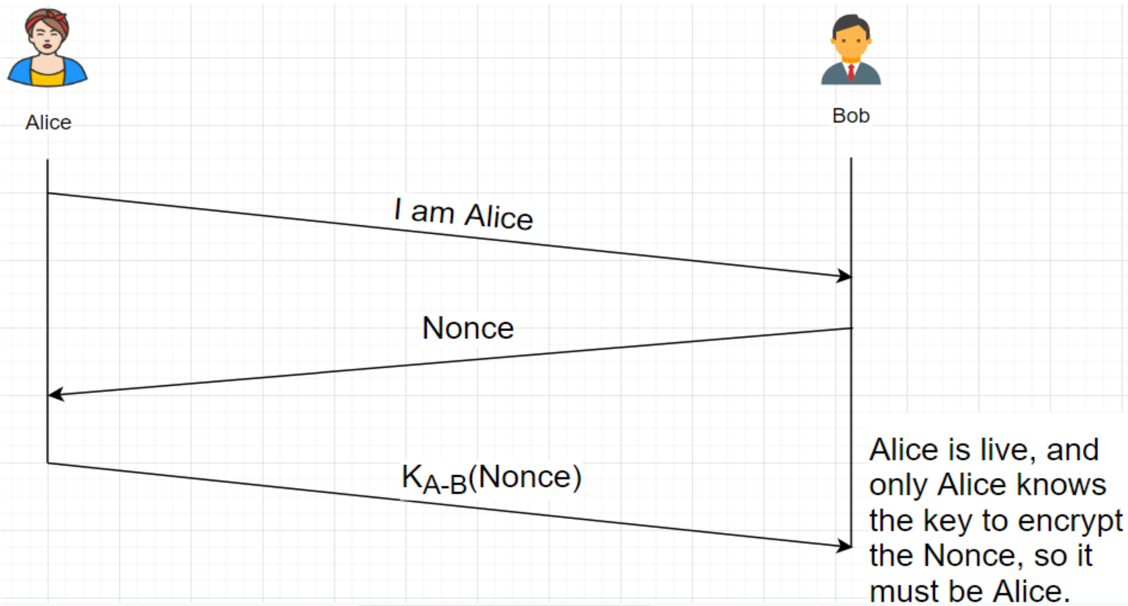
Protocol ap3.1: Alice says "I am Alice" and sends her encrypted secret password to "prove" it.



Record and playback *still* works

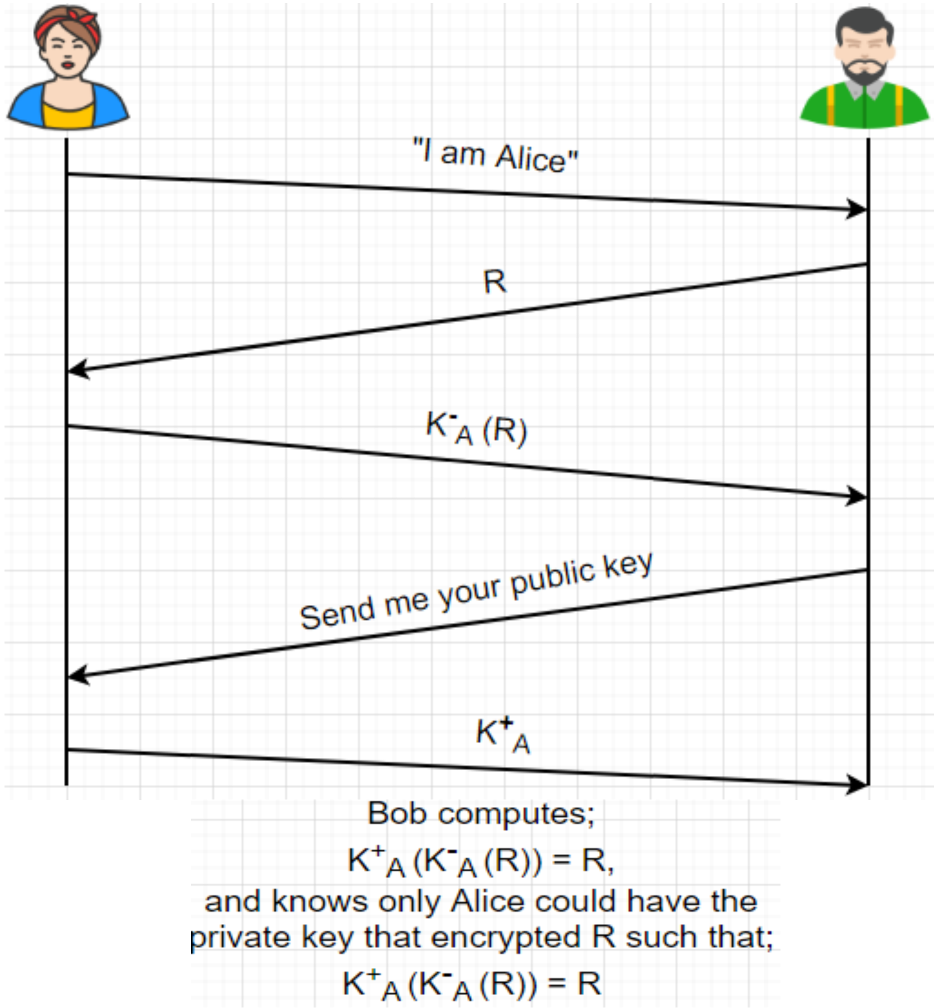
Authentication: Yet another try

- Goal: avoid playback attack
- nonce: number (R) used only once-in-a-lifetime
- ap4.0: to prove Alice "live", Bob sends Alice nonce, R. Alice must return R, encrypted with shared secret key



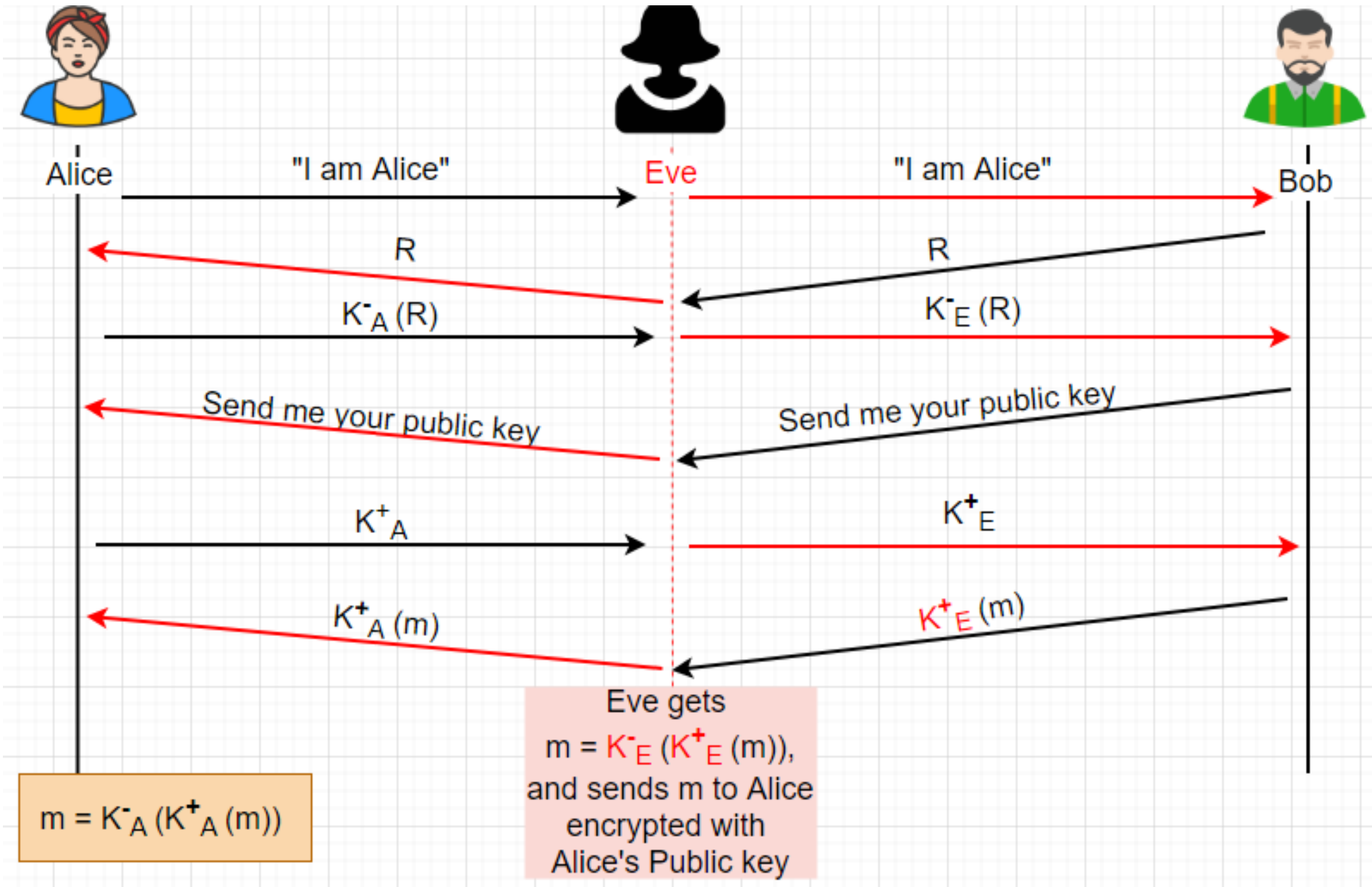
Authentication: ap 5.0

- ap4.0 requires shared symmetric key
- Can we authenticate using public key techniques?
- ap5.0: use nonce, public key cryptography



ap 5.0: Security hole

- Person in the middle attack: Eve poses as Alice (to Bob) and as Bob (to Alice)



ap 5.0: Security hole contd.

- PiTM attack: Eve poses as Alice (to Bob) and as Bob (to Alice)

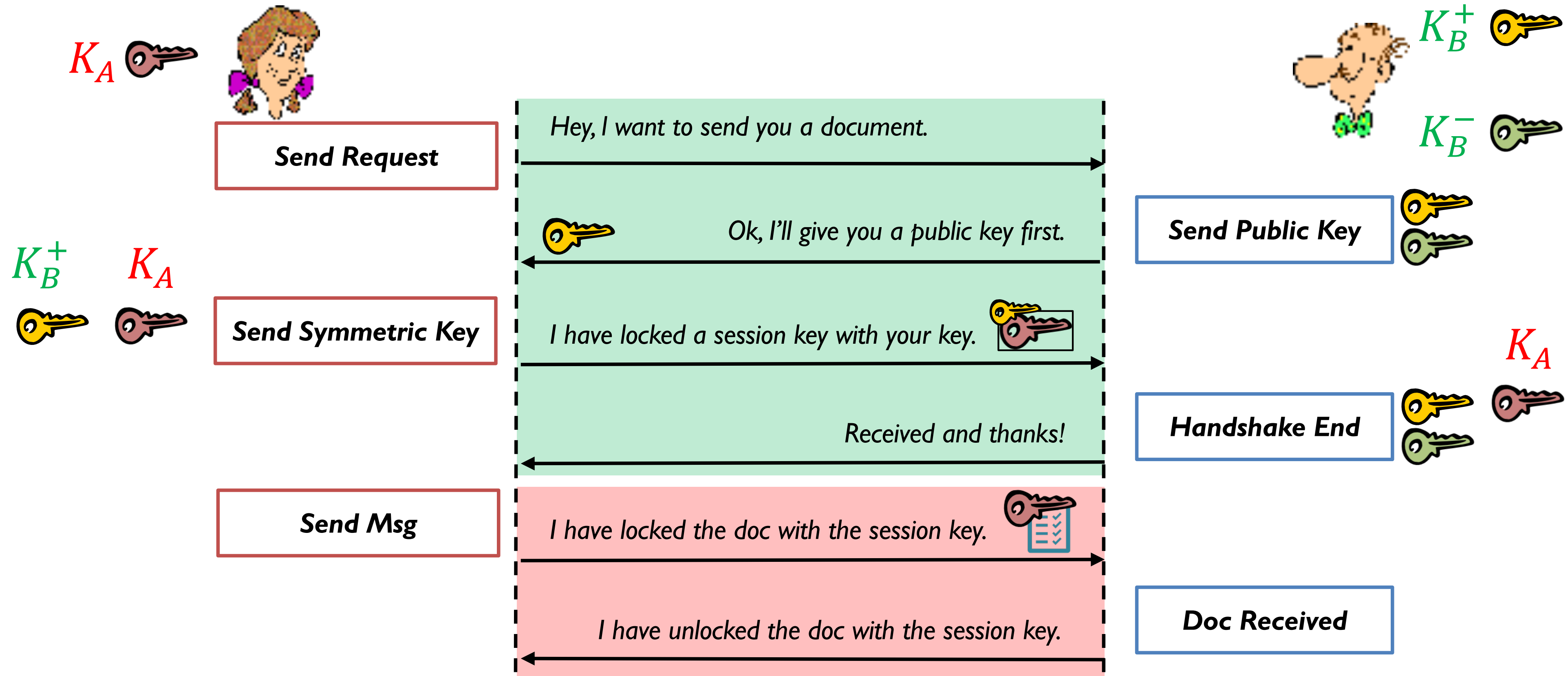
- Difficult to detect:
  - Bob receives everything that Alice sends, and vice versa. (e.g. Bob, Alice can meet one week later and recall conversation!)
  - Problem is that Eve receives all messages as well!
- What made this PiTM attack possible?

Encryption in Practice : A hybrid approach

Alice wants to send Bob a document

– Exchange session (symmetric) key with *asymmetric* encryption

– Send msg with *symmetric* encryption



Encryption in Practice #1

Scenario A

1. Mallroy intercepts K_A encrypted with K_B^+ from Alice.
2. Mallroy sends Bob K_{M} encrypted with K_B^+.

3. Mallroy intercepts the document encrypted with K_A.
4. Mallroy sends Bob a changed document encrypted with K_{M}.

Question

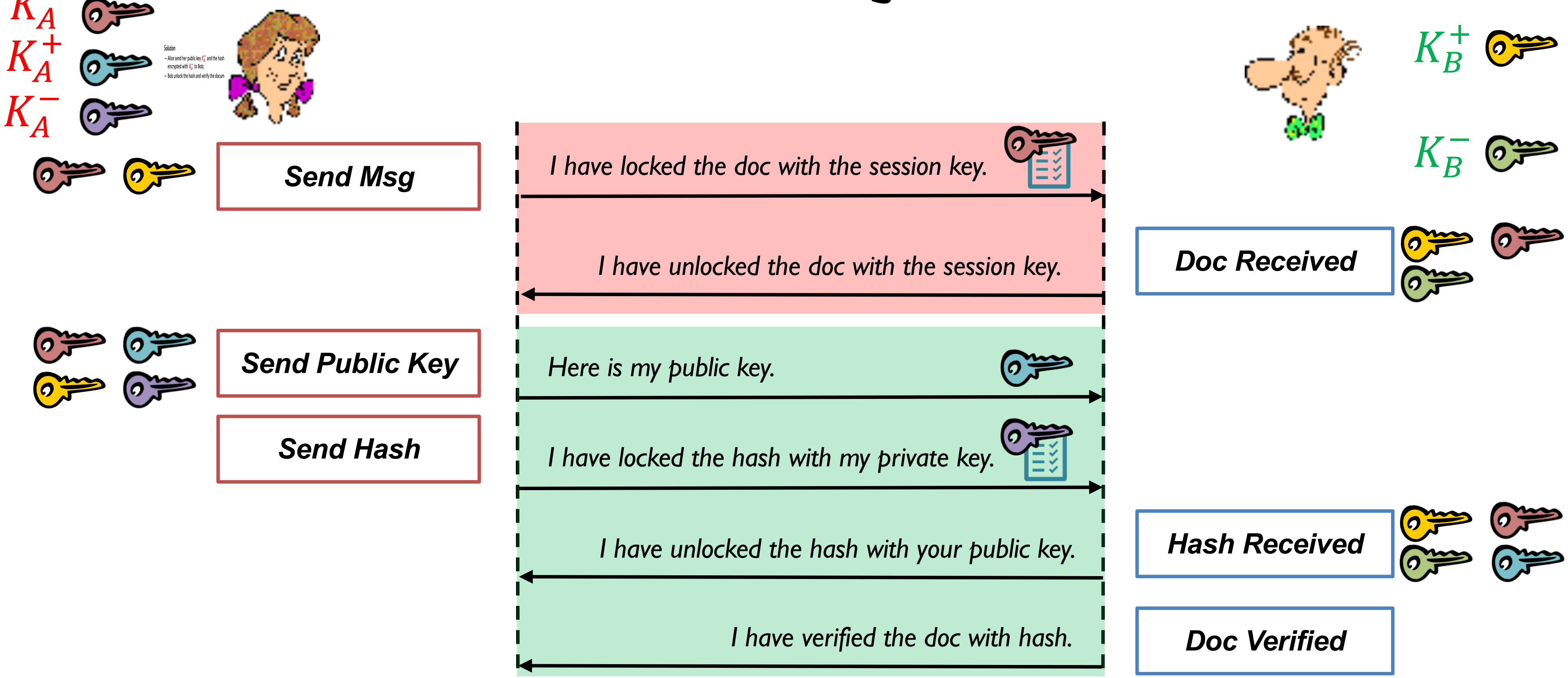
- Can Bob figure out if the document sent to him has changed?

Encryption in Practice #1 (cont'd)

Possible Solution

– Alice send her public key K_A^+ and the hash of file encrypted with K_A^- (digital signature) to Bob;

– Bob unlock the hash and verify the document.



Encryption in Practice #2

Scenario B

1. Mallroy intercepts the encrypted hash and K_A^+ sent from Alice;
2. Mallroy sends Bob his own public key K_M^+ and the hash of fake document encrypted with K_M^-.

Question

- Can Bob figure out the keys, doc and hash sent to him are all from Mallroy?

Challenge

- Identify the owner of K_A^+ and K_M^+.

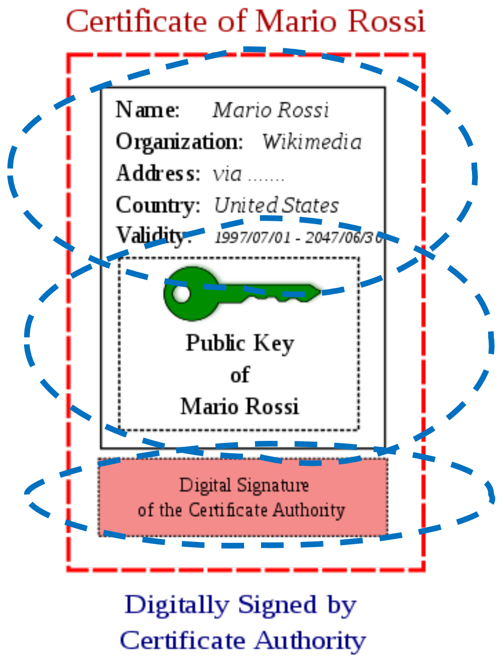
Public Key Certificate

Definition:

- An electronic document used to prove the validity of a public key.
- Also known as a digital certificate or identity certificate.

Key ingredients:

- – Information about the key;
- – Information about the identity of its owner (subject);
- – The digital signature of entity that verified the certificate's contents (issuer).



Public Key Infrastructure

Definition:

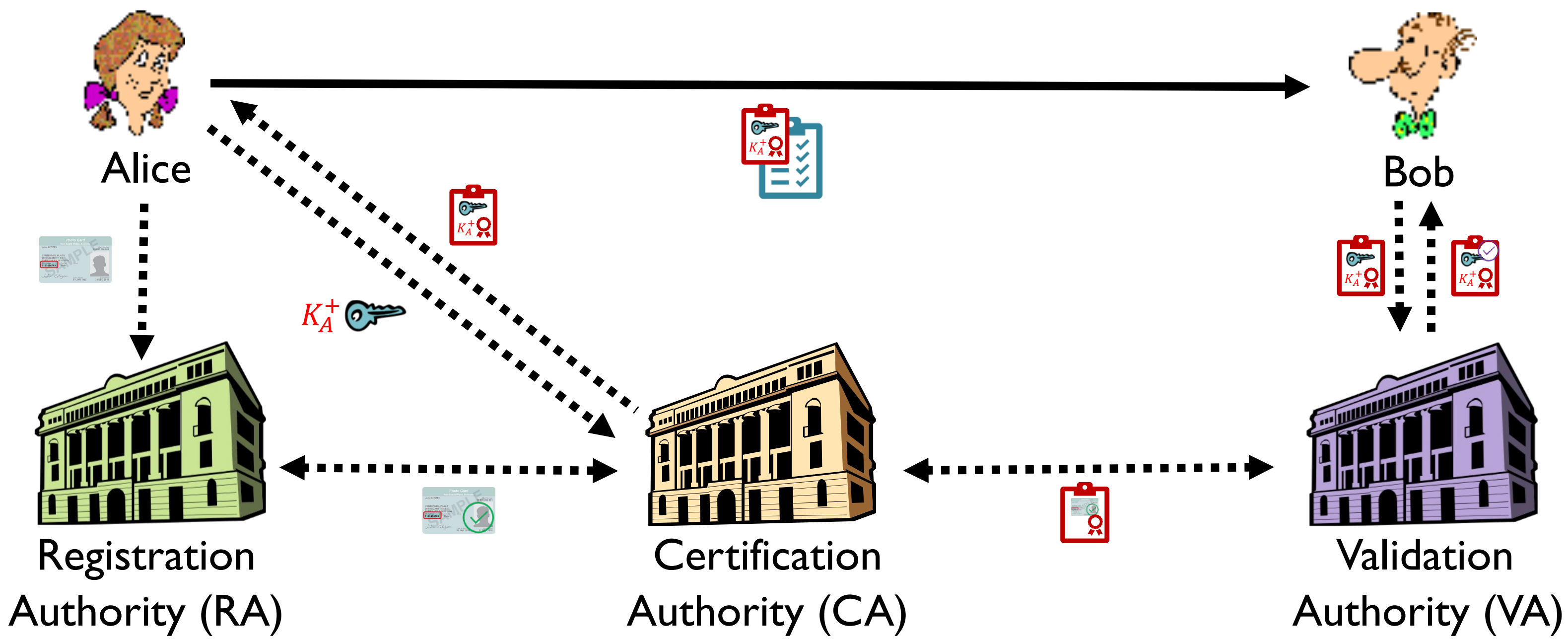
- A set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

Key ingredients:

- *Keys/certificate;*
- *Registration authority (RA):* verify the identity of entities;
- *Certificate authority (CA):* issue and sign the digital certificates;
  - Root certificate authority;
  - Intermediate certificate authority;
- *Validation authority (VA):* verify the digital certificates.
- …

Public Key Infrastructure

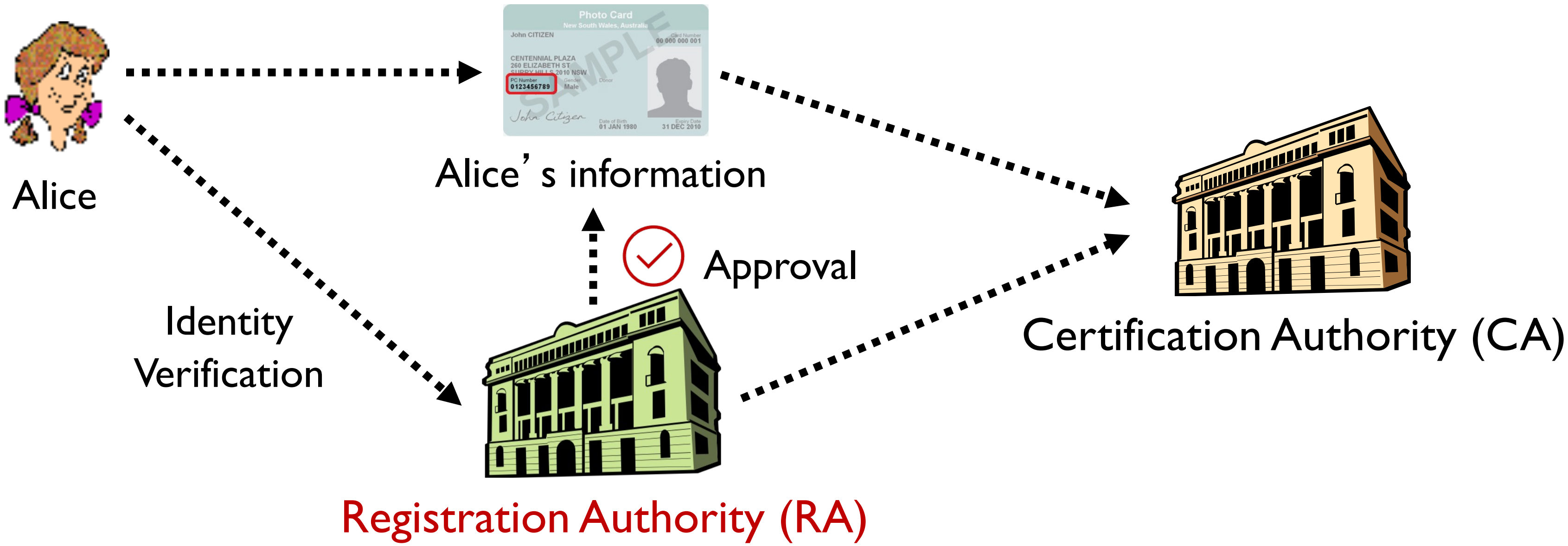
- Alice/Bob;
- Identity/Key/Certificate;
- RA/CA/VA.



Registration Authority

Registration Authority (RA): verify the identity of entity.

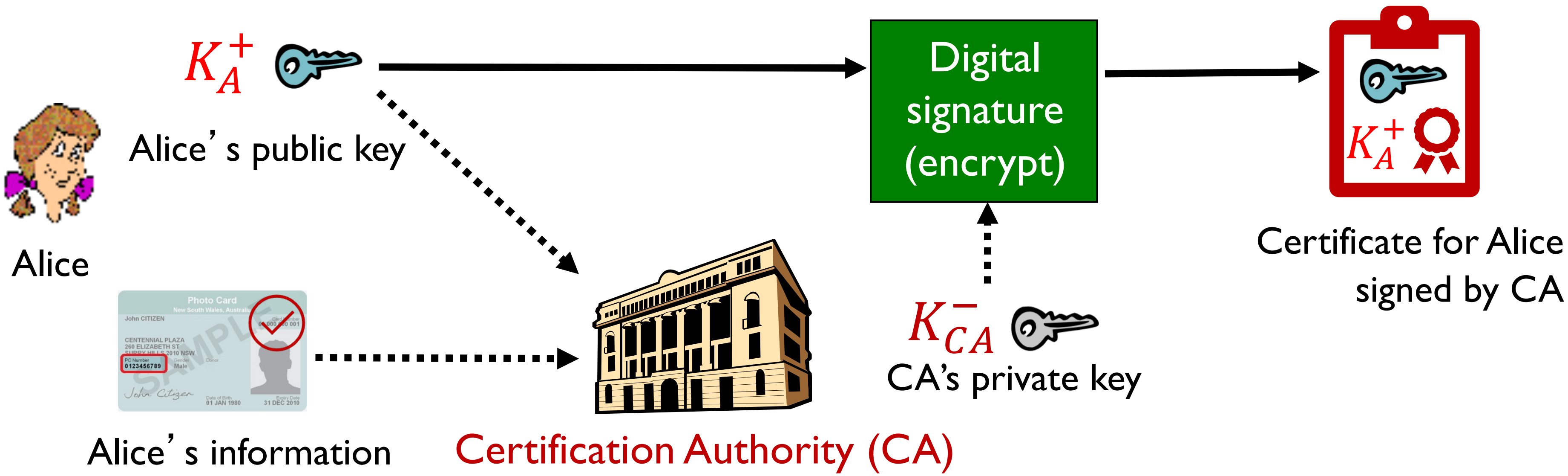
- Alice requests for a certificate.
  - Alice provides "proof of identity" to RA;
  - RA verifies the identity and gives approval to CA.



Certification Authority

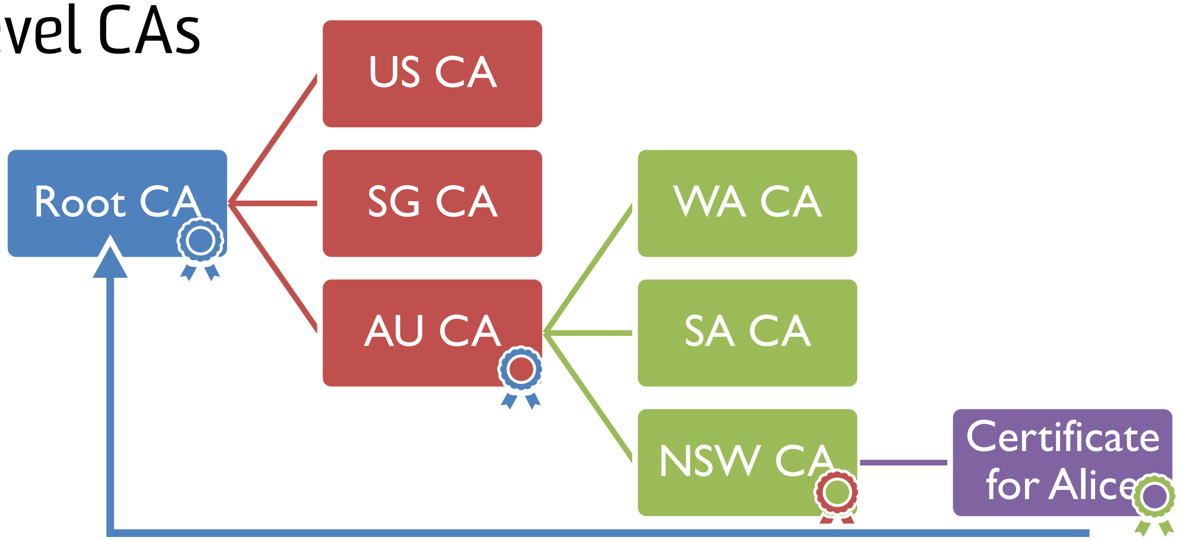
Certification Authority (CA): binds public key to particular entity.

- Alice registers her public key with CA.
  - Alice provides proofed identity and public key to CA;
  - CA creates certificate binding Alice to its public key.



Certification Authority (cont'd)

- Hierarchical structure of CAs
  - Root CA (level 1)
  - Subordinate CAs (level 2+)
- Chain of trust
  - Start from Root CA
  - Top down: Upper-level CAs issue certificates for lower-level CAs

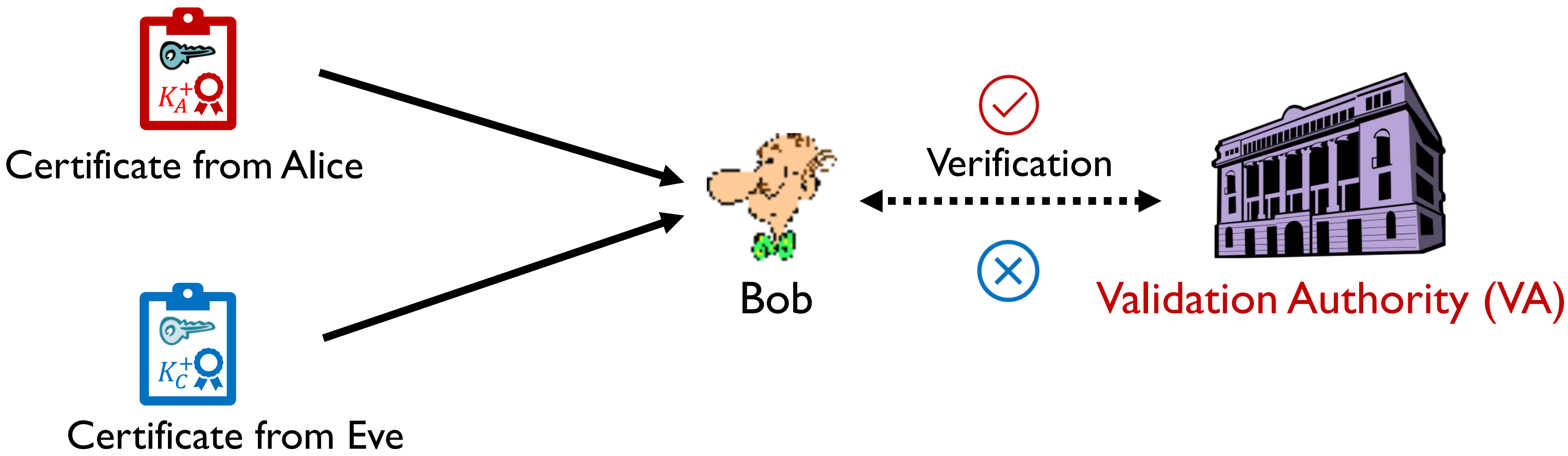


Verification: backward trace to Root CA

Validation Authority

Validation Authority (VA): verify the digital certificates.

- Bob validates the certificate from "Alice".
  - Bob sends the certificates to VA;
  - VA verifies the signature of certificates and sends the result back to Bob.



Lifecycle of PK Certificates

Enrollment

- Request a certificate;

Issuance

- Validate the identity of entity and issue the certificate;

Validation

- Confirm the certificate is valid and hasn't expired or been revoked;

Revocation

- An expiration date specified when first issued;
- When that date is reached, the certificate will automatically be considered invalid;

Renewal

- Renew certificates upon expiration date, though typically re-verifying identity.

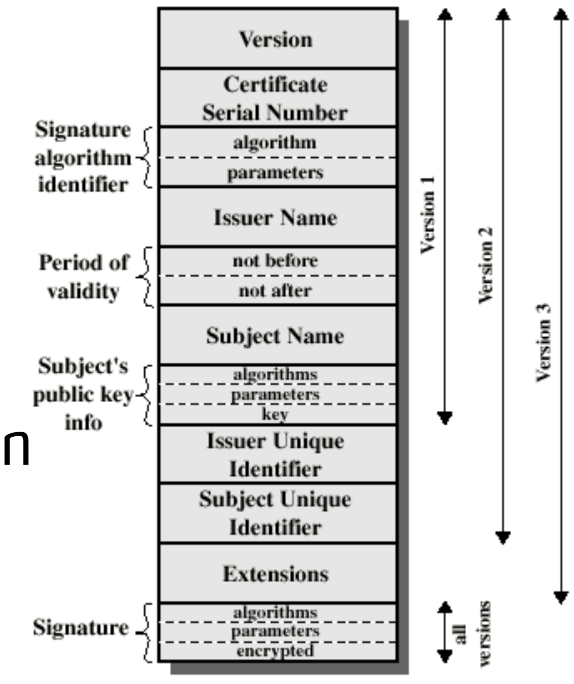
X.509 Formats

Definition:

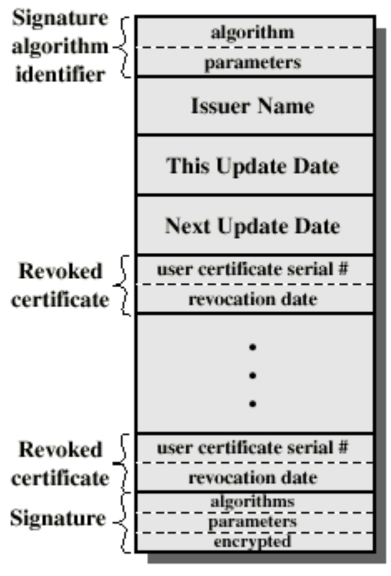
- The format standard for the public key certificate.

Ingredients:

- Public key;
- Identity information;
- Signature information;
- Certificate revocation list;
- Certificate validity verification algorithm.



(a) X.509 Certificate



(b) Certificate Revocation List

Certificate Revocation List (CRL)

Definition:

- A list of all revoked certificates;
- Base CRL
  - A large file contains all revoked certificates;
  - Update in a longer time span;
- Delta CRL
  - A small file contains the certificates that have been revoked since the last base CRL was published;
  - Update in a shorter time spam: 15 min - 1 day

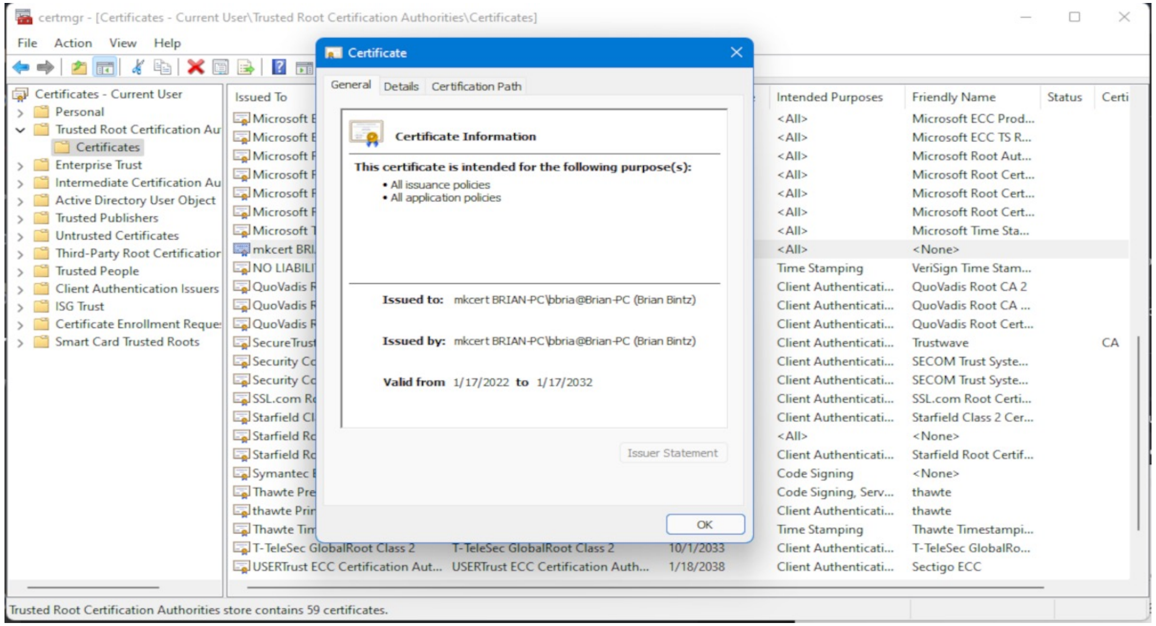
Certificate Store

Definition:

– Store a list of certificates that you trust;

– Users can manage the list by themselves;

  ○ Disable an insecure certificate before it's reported.

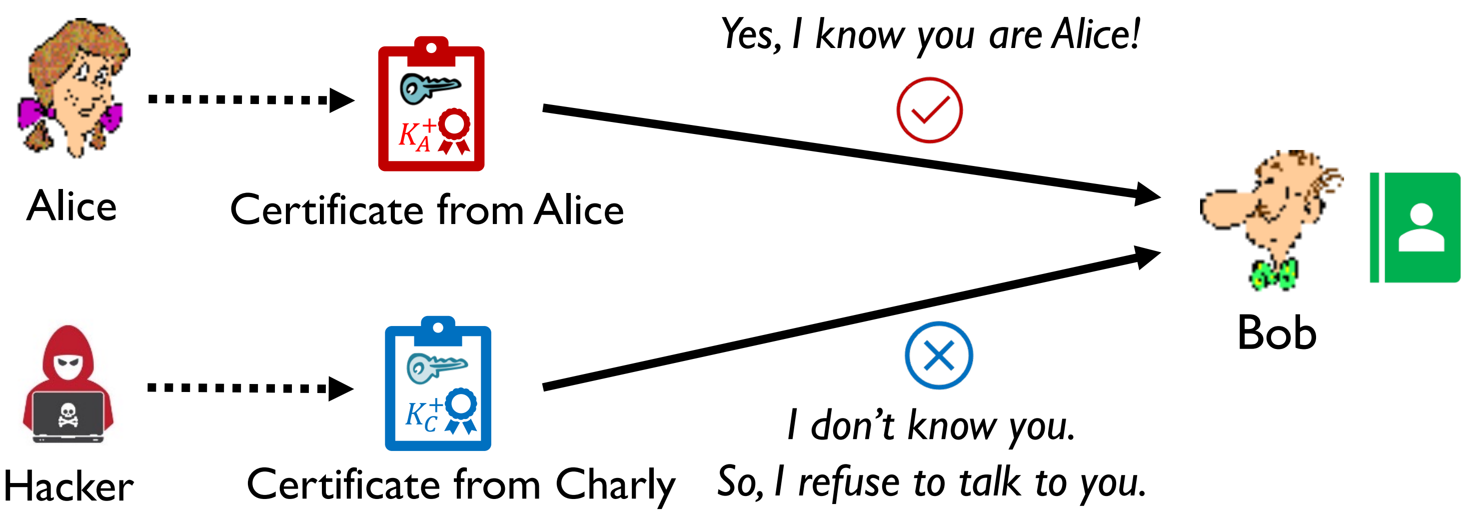


It's risky to add certificates from unknown sources.

Certificate Store (cont'd)

Certificate pinning: associate a host with their expected certificate or public key.

– Accepts only authorized ("pinned") certificates for authentication of connections



Threats in Practice

- Case #1
  - Attackers are using malware with valid certificates
    - Signed malware samples with certificates came from CAs such as DigiCert, Entrust, GlobalSign, Go Daddy, Symantec, Thawte, and VeriSign.
    - Be careful with unknown certificates because CAs can make mistakes.

Threats in Practice

- Case #2
  - NVIDIA's stolen Code-Signing Certs used to sign Malware
    - The stolen NVIDIA code-signing certificates are expired, but they're still recognized by Windows.
    - Be careful with Expired certificates.

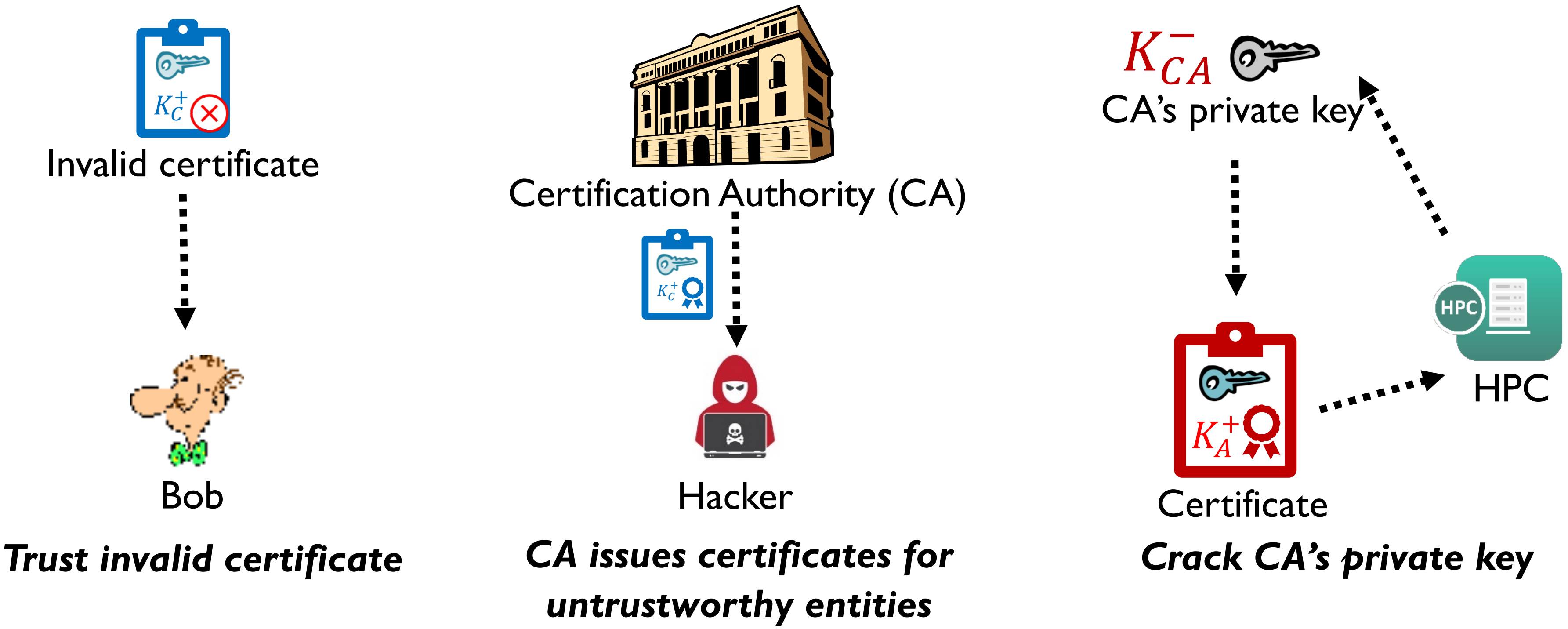
Threats in Practice

- Case #3
  - Fraudulent certificates issued by Comodo, is it time to rethink who we trust?
    - 8 certificates were fraudulently issued for 6 high-profile websites, including Google, Yahoo! and Skype.
      - mail.google.com
      - www.google.com
      - login.yahoo.com
    - Be careful with Revoked certificates.

Security issues in PKI

Key points for PKI Security:

- Humans follow security rules;
- RA/CA/VAs are responsible entities;
- Encryption algorithms are secure.



Web of Trust

- Decentralized PKI
  - The web of trust is an alternative to PKI where users can create a community of trusted parties by mutually signing certificates without needing a registrar
    - Alice and Bob can sign each other's certificates certifying their public keys;
    - Other entities if they trust Alice, can use Bob's certificate duly certified by Alice;
    - Similar to the certificate chain in centralized PKI.
  - E.g., PGP and GnuPG
  - Reading: *The Official PGP User's Guide*

Acknowledgements

- Network Security Essentials: Stallings, Chapter 4 provided by Henric Johnson, Blekinge Institute of Technology, Sweden (Please refer to Section 4.3 and 4.4 from Stallings)
- Computer Networking A Top-Down Approach: Jim Kurose and Keith Ross, Chapter 8
- Optional read (Public Key Infrastructure X.509 (PKIX)) WG, RFC 5280
  https://datatracker.ietf.org/doc/html/rfc5280

Public Key Cryptography

All remaining slides are self-read and examinable

Public Key Encryption Algorithms
(taken from COMP3331/9331)

- Requirements:

  1. need $K_B^+(\,)$ and $K_B^-(\,)$ such that

$$K_B^-(K_B^+(m)) = m$$

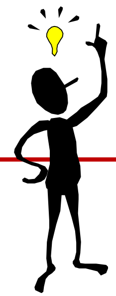
  2. given public key K_B^+, it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

Public Key Cryptography

Symmetric key crypto

- Requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?



Public key crypto

- Radically different approach
- Sender, receiver do not share secret key
- Public encryption key known to all
- Private decryption key known only to receiver

RSA: Getting Ready

- A message is a bit pattern.
- A bit pattern can be uniquely represented by an integer number.
- Thus, encrypting a message is equivalent to encrypting a number.
- Example
- m= 10010001. This message is uniquely represented by the decimal number 145.
- To encrypt m, we encrypt the corresponding number, which gives a new number (the ciphertext).

RSA: Creating Public/Private Key Pair

1. Choose two large prime numbers p, q. (e.g., 1024 bits each)
2. Compute $n = pq$, $z = (p-1)(q-1)$
3. Choose e (with $e<n$) that has no common factors with z. (e, z are "relatively prime"). E.g.: 4 and 9 are relatively prime. 6 and 9 are not.
4. Choose d such that $ed-1$ is exactly divisible by z. (in other words: $ed \bmod z = 1$).
5. Public key is $\underbrace{(n,e)}_{K_B^+}$. Private key is $\underbrace{(n,d)}_{K_B^-}$.

UNSW AUSTRALIA

RSA: Creating Public/Private Key Pair

0. Given (n,e) and (n,d) as computed
1. To encrypt bit pattern, m (m < n), compute:

   $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)

2. To decrypt received bit pattern, c, compute:

   $m = c^d \bmod n$ (i.e., remainder when c^d is divided by n)

Magic happens! $m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

RSA Simple Example

- Bob chooses p=5, q=7. Then n=35, z=24.

  e=5 (so *e, z* relatively prime).
  d=29 (so *ed-1* exactly divisible by z).

encrypt:

letter	m	m^e	$c = m^e \bmod n$
l	12	248832	17

decrypt:

c	c^d	$m = c^d \bmod n$	letter
17	481968572106750915091411825223071697	12	l

Note: Assume that letters a-z are numbered 1 to 26 and hence l=12

RSA Another Important Property

- The following property will be very useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

Result is the same!