Certificateless Public-Key Cryptography

Mohsen Toorani
Department of Informatics
University of Bergen

Norsk Kryptoseminar
Selmer Center
November 2011

Public-Key Cryptography (PKC)

- Also known as **asymmetric** cryptography.
- Each user has two keys: public & private.
- Alice's public key is typically used for:
  - encryption to Alice by Bob
  - verification of Alice's signatures by Bob
- Alice's private key is typically used for:
  - decryption by Alice
  - signing by Alice
- No need for Alice and Bob to share a common key before beginning secure communications!
  - Compared with **symmetric** key cryptography.

Public-Key Cryptography (PKC)

A significant problem in PKC is **verification of the authenticity of public keys**: Users must be assured that they cannot be fooled into using a false public key!

Solutions for authenticity of public keys:

1. **Public-Key Infrastructure (PKI)**
2. **Identity-based Cryptography**
3. **Self-Certified Public-Key Cryptography**
4. **Certificate-based Public-Key Cryptography (CB-PKC)**
5. **Certificateless Public-Key Cryptography (CL-PKC)**

1. Public-Key Infrastructure (PKI)

- PKI is a system for supporting deployment of PKC
- By the term “traditional PKI” we mean:
  - a combination of hardware, software and policies
  - needed to deploy and manage **certificates**
  - to produce trust in public keys
  - used in a particular application or set of applications.

Digital Certificates

A **certificate** binds an entity with its public key.

The certificate is issued and signed by a **trusted** Certificate Authority (CA).

Digital Certificate

entity's description (name, etc.)
+
entity's **public key**
+
expiration date, serial number, etc.
+
CA's name
+
a **signature** issued by a CA

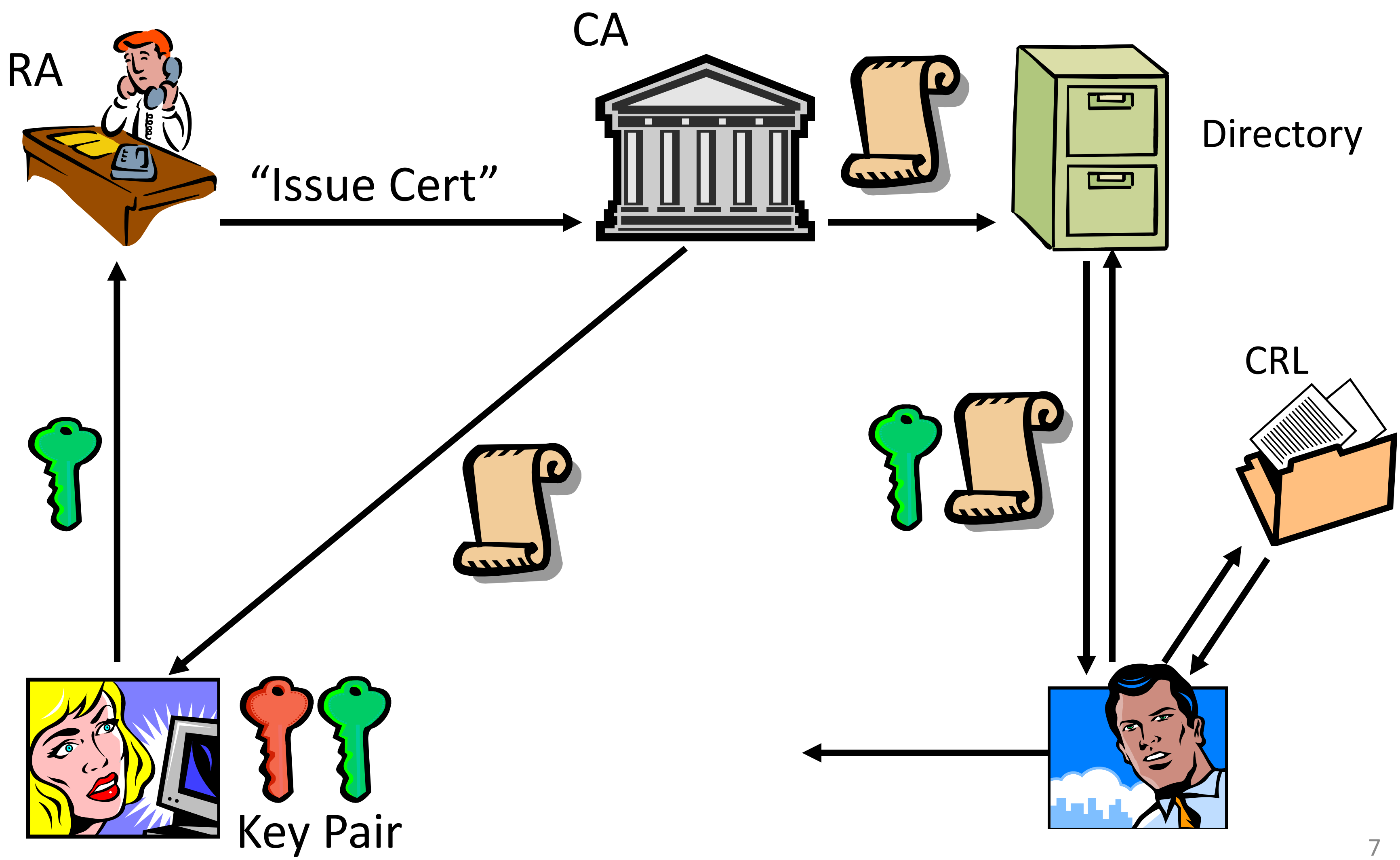
CA's signature: certificate's hash,

encrypted with CA's private key

PKI Components

- Registration Authority (RA)
  - Authenticates individuals/entities
  - Passes off result to Certification Authority.
- Certification Authority (CA)
  - Issues certificates
- Directory Service
  - Directory of public keys/certificates.
- Revocation Service
  - May involve distribution of Certificate Revocation List (CRL) or On-line certificate status checking (OCSP).

PKI Components



Some PKI Problems

- Acute where end-user are humans.
- Legal and regulatory issues
- Interoperability and standards
- Costs and business models
- Some technical issues:
    - How should the revocation be handled?
    - How the CA should be designed?
    - How should keys and algorithms be managed?

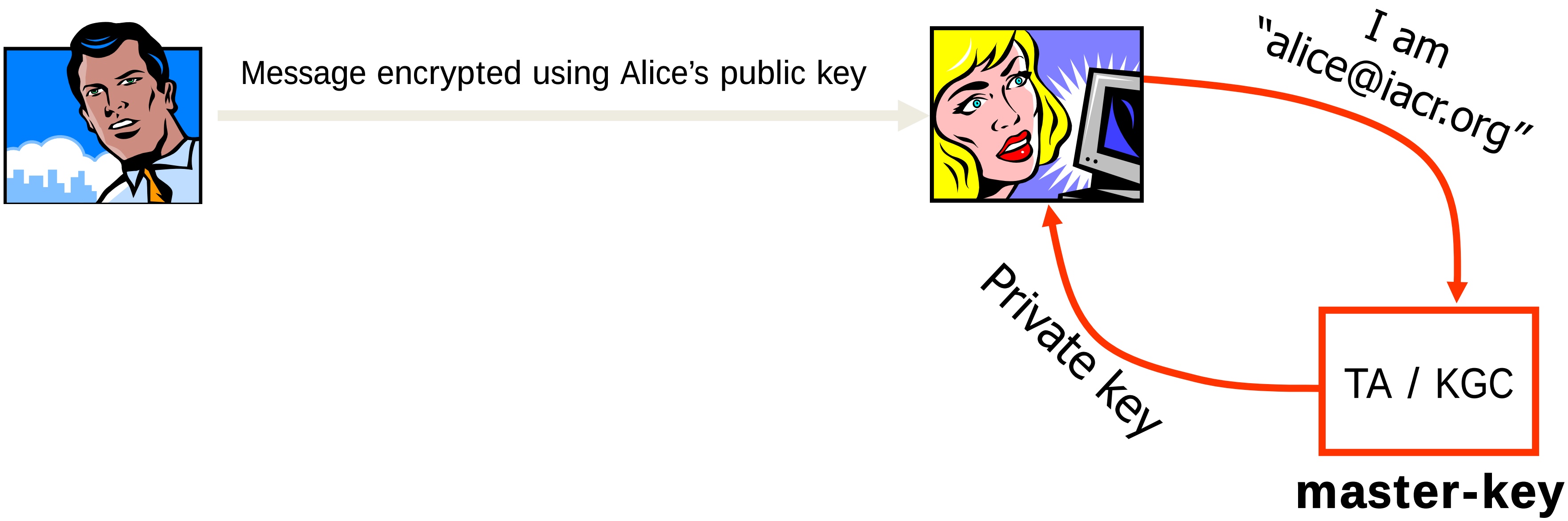
Certificates and their management are the source of some problems.

2. Identity-based Cryptography

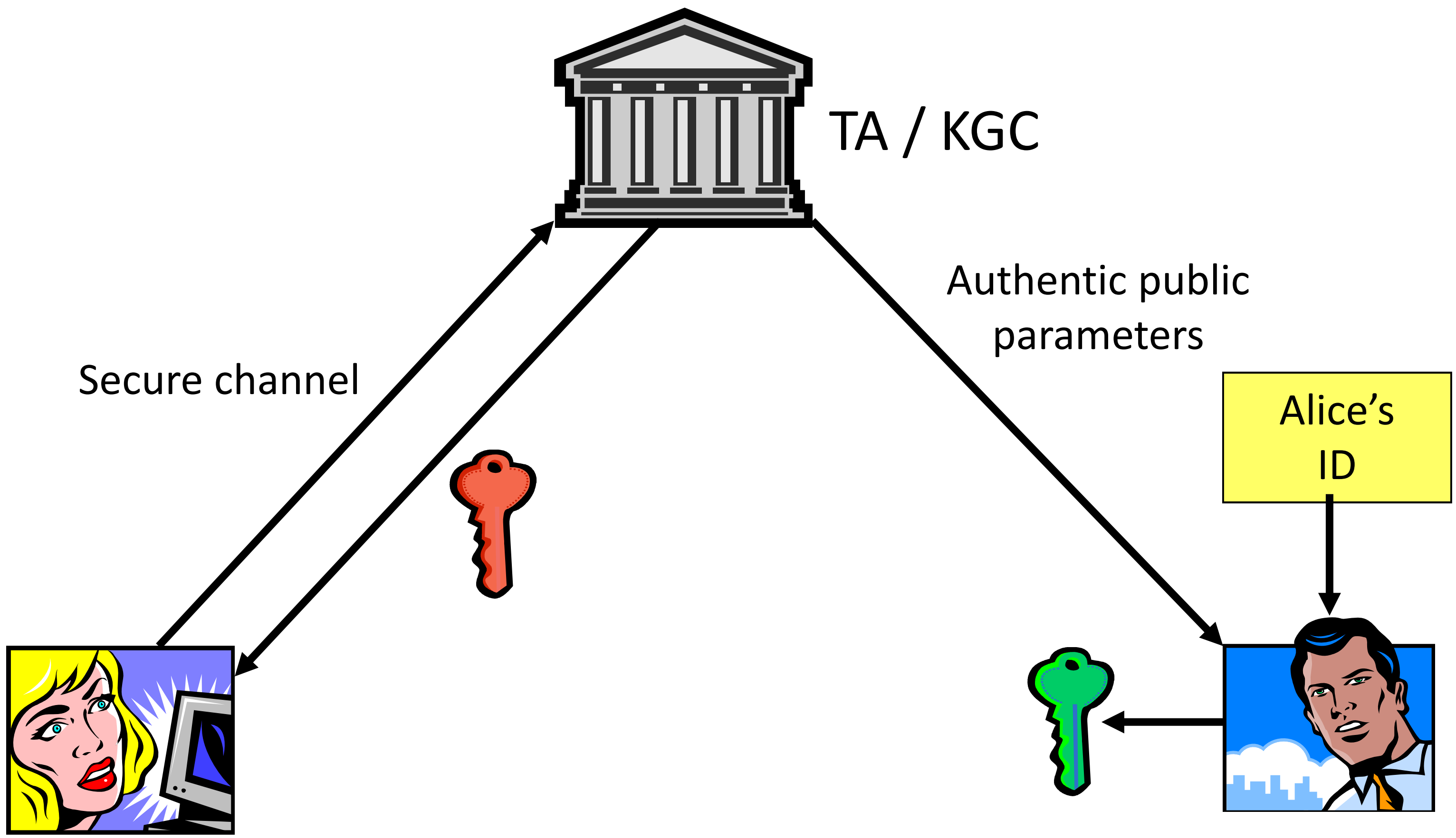
- In ID-based Cryptography, Public keys are directly derived from system identities (e-mail address, mobile number, IP address, etc).
- The first idea due to Shamir (1984) but it was just an ID-based signature scheme.
- Construction of *practical* and *secure* ID-based encryption scheme was an open problem until 2001 when Boneh and Franklin proposed a pairing-based IBE scheme in Crypto'01.

2. Identity-based Cryptography

Public key of Alice: “alice@iacr.org”



2. Identity-based Cryptography
(in Reality)



2. Advantages of ID-PKC

- **Certificate-free**
  - No production, checking, management or distribution of certificates.
- **Directory-less**
  - Bob can encrypt his message without looking-up Alice's public key.
  - Alice can get her private key after receiving Bob's encryption.
- **Automatic revocation**
  - No need for CRL or OCSP server.
  - It is possible to have the ability so that Alice have to obtain a new private key for each period for decrypting messages encrypted in that period and the private key becomes useless at end of each period. The identifier may also include a validity period.
- **Support for key recovery (can violate user's privacy)**
  - TA can calculate private key for any user.
  - It may be required when a user leaves the organization.
  - Enables applications like content-scanning of e-mails.

2. Disadvantages of ID-PKC

- **Catastrophic compromise**: What is the cost of compromise of the master secret?
  - All past encrypted messages are exposed & all old signatures become worthless.
  - Potentially has higher cost than compromise of CA's signing key in PKI: CA in PKI can re-issue all certificates under new signing key without compromising clients' private keys.
- **Key Escrow**
  - TA can calculate all the private keys.
  - We need to trust TA not to abuse this privilege.
  - PKI is more flexible in this respect.
- **Inability to Provide Non-repudiation**
  - Another consequence of key escrow.
  - TA can forge signatures if it uses an ID-based signature: Need to trust TA not to do that.
  - EU electronic signature legislation requires private key to be under "sole control" of signer. It is incompatible with some legislative regimes.

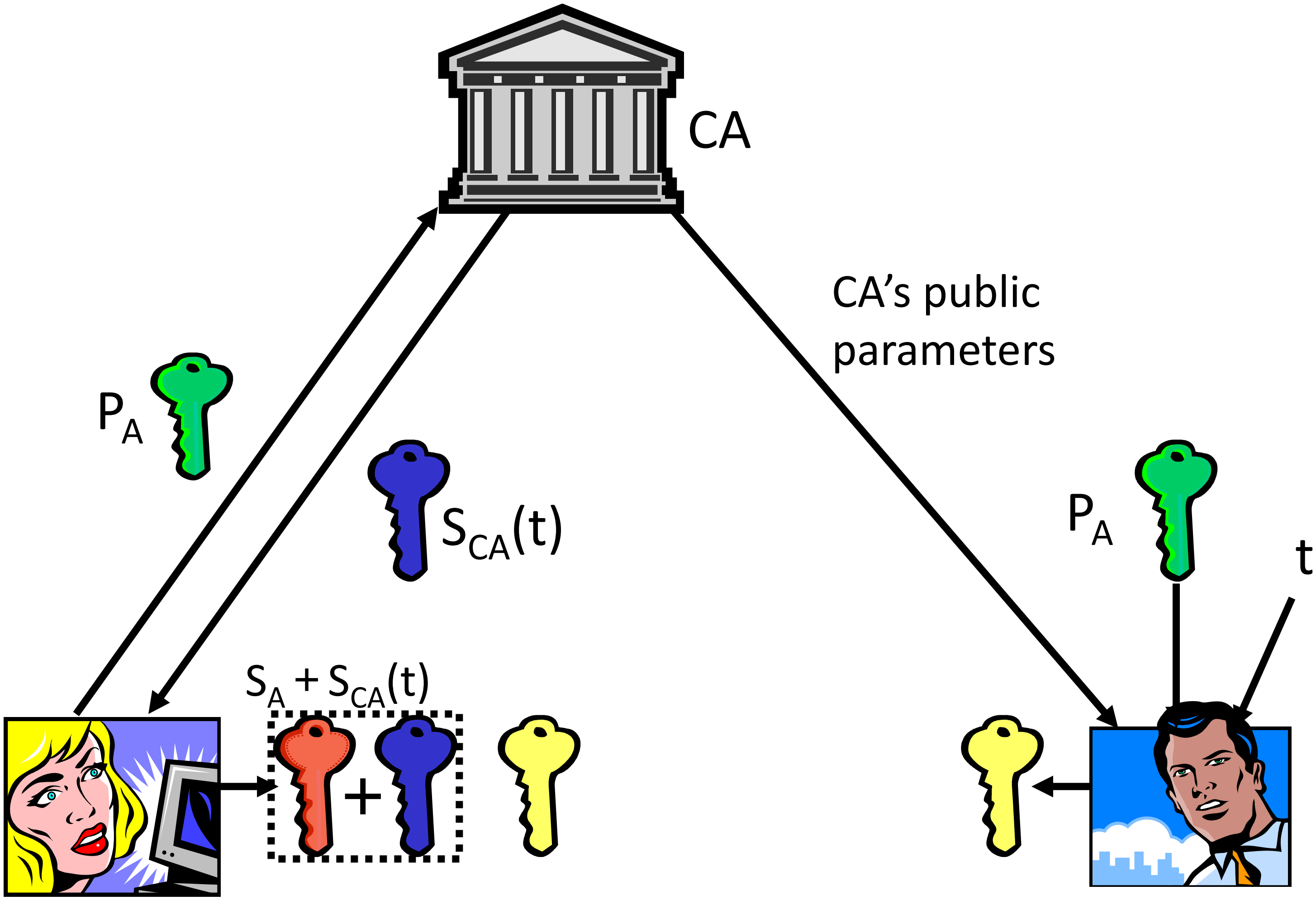
3. Self-Certified PKC

- Introduced by Girault (Crypto’91) to reduce storage and computation costs:
  - No key escrow
  - No need for hash functions in computing public keys
  - No need for a secure channel between CA and user.
- Users are associated with a 3-tuple (ID, s, P): (User's identity, User-chosen private key, the public key that doubles as a certificate).
- CA issues a certificate on ID that is used as the public key (different from traditional PKI where users have separate certificates validating their public keys.
- P cannot be immediately derived from ID (varies from ID-based schemes)

4. Certificate-based Public-Key Cryptography (CB-PKC)

- Introduced by Gentry (Eurocrypt 2003).
- Simplifies revocation in traditional PKIs.
- Alice's private key consists of two components:
  - A private component S_A that is part of a traditional key-pair (S_A,P_A).
  - A time-dependent component $S_{CA}(t)$ issued by CA as long as Alice is not revoked.
- Bob can compute a matching public key using the CA's public parameters, time *t* and Alice's public component P_A.
- Bob is assured that Alice can decrypt only if the CA has issued certificate $S_{CA}(t)$ for the current time interval *t*.

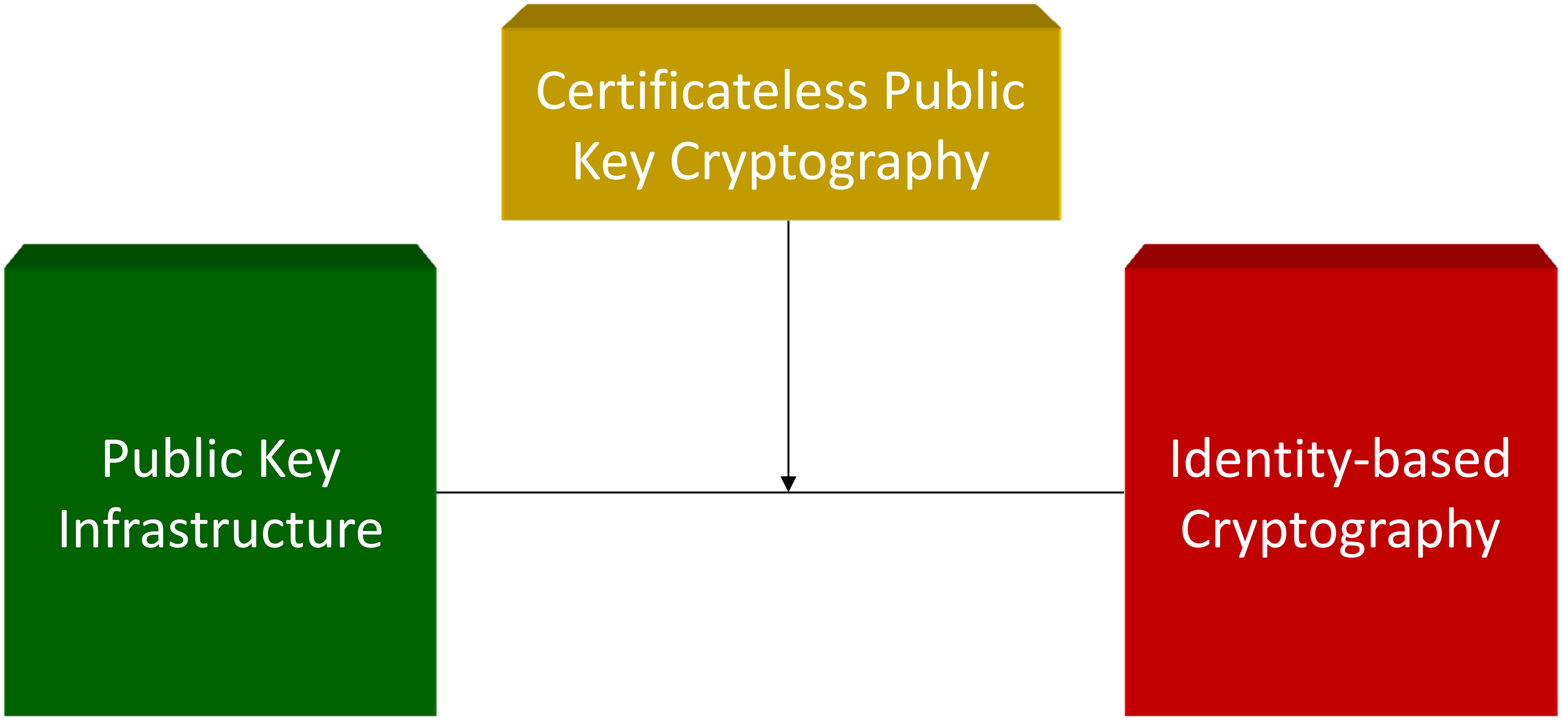
4. Certificate-based PKC (CB-PKC)



5. Certificateless Public-Key Cryptography (CL-PKC)

- Introduced by Al-Riyami and Paterson (Asiacrypt 2003).
  - A thriving sub-area of ID-PKC.
- Design objective:
  - To remove the key escrow problem of ID-PKC without introducing certificates.

CL-PKC



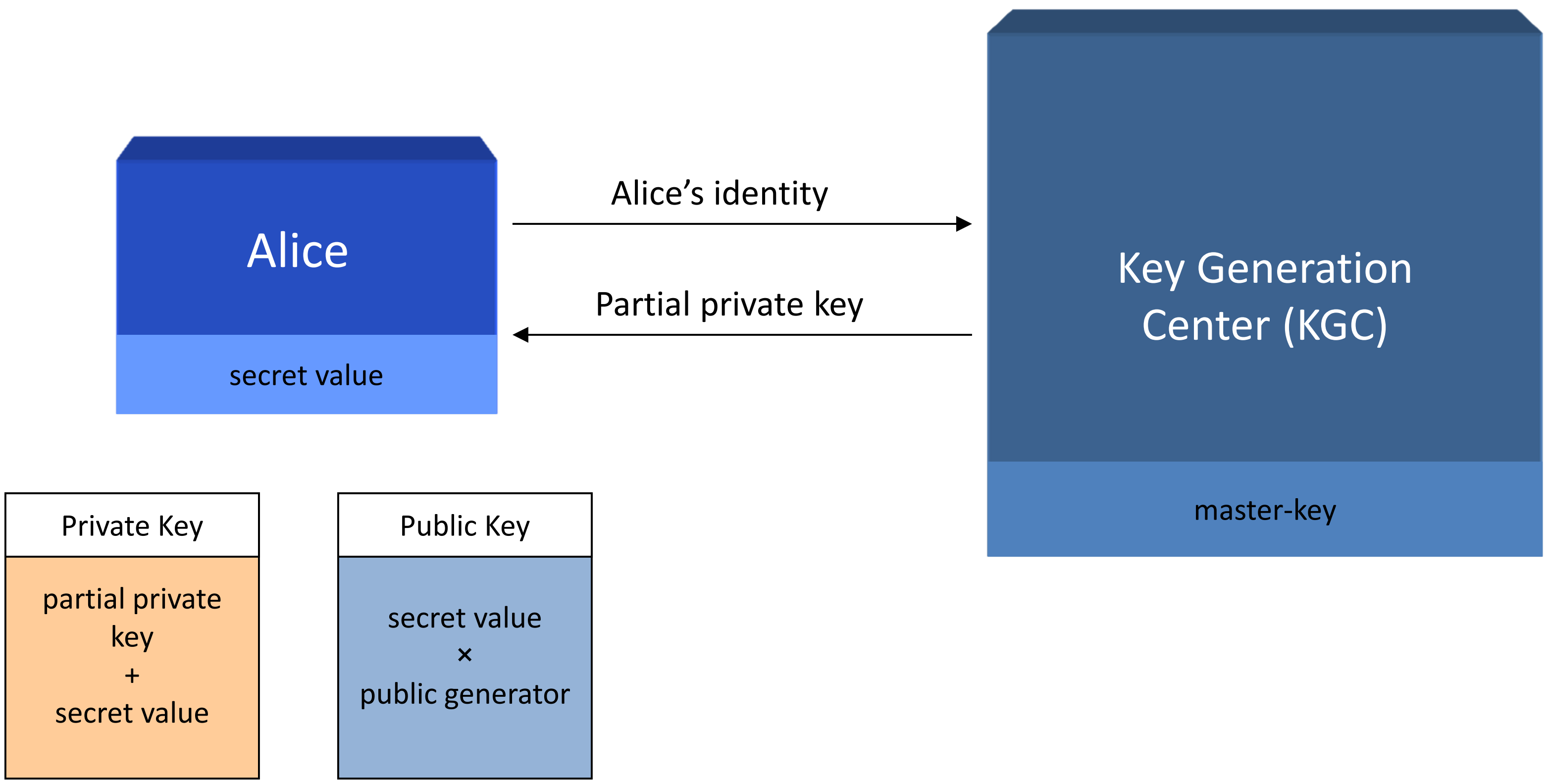
CL-PKC:

- A paradigm for generating trust in public keys.
- Lies midway between traditional PKI and ID-PKC in terms of trust model and functionality

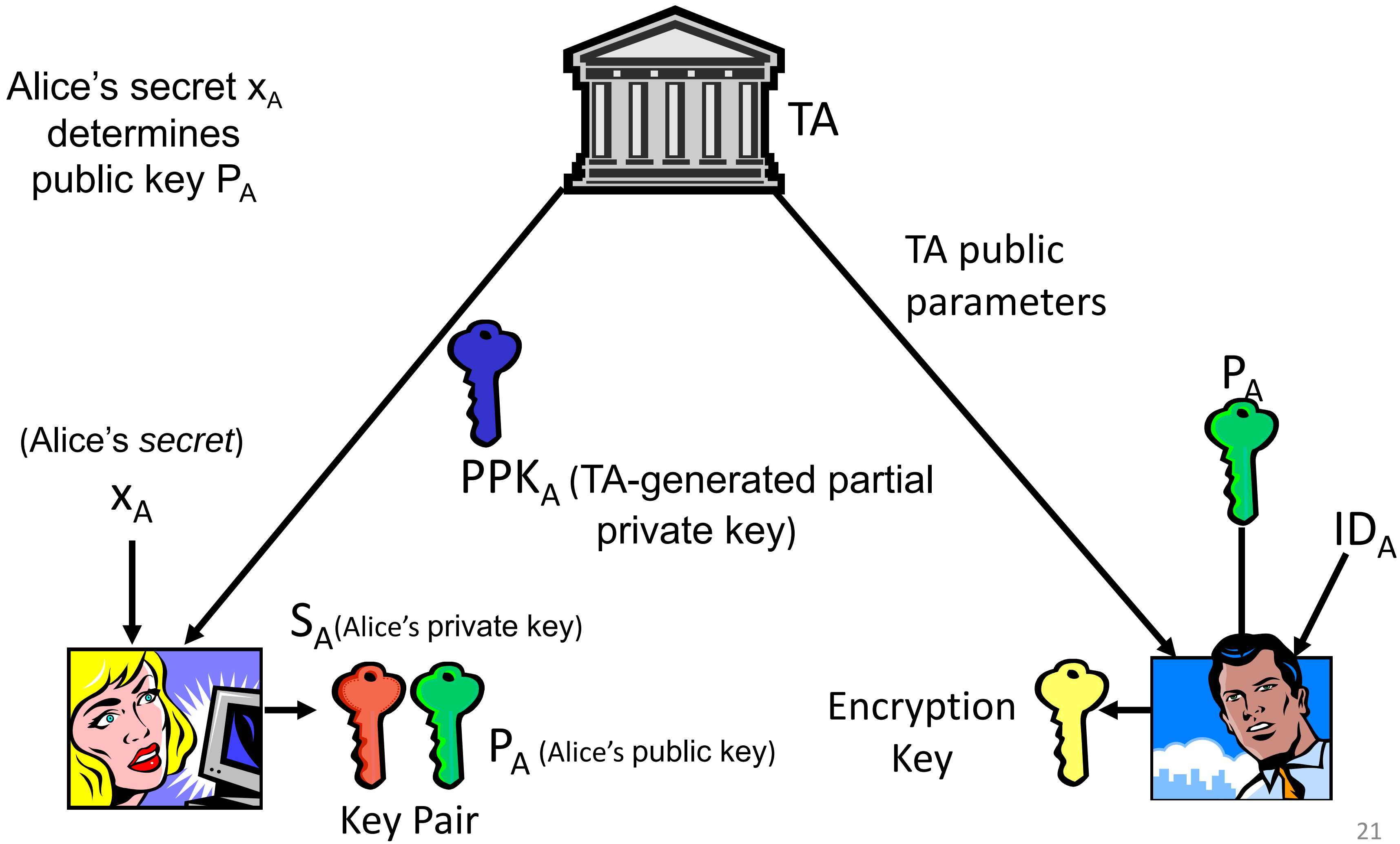
Why CL-PKC?

- No need to certificates (as PKI)
  - Low storage and communication bandwidth
  - No need to verify certificates and certificate chains
  - Higher degree of privacy
- Public keys are always valid
  - No need for CRLs
- No key escrow (as ID-PKC)
  - TA cannot recover session keys
  - TA cannot forge signatures

CL-PKC


Alice
secret value
Alice’s identity
Partial private key
Key Generation Center (KGC)
master-key
Private Key
partial private key
+
secret value
Public Key
secret value
×
public generator

CL-PKE


Alice's secret x_A determines public key P_A
TA
TA public parameters
PPK_A (TA-generated partial private key)
P_A
ID_A
(Alice's *secret*)
x_A
S_A(Alice's private key)
P_A (Alice's public key)
Key Pair
Encryption Key

CL-PKE

- Each user generates its own public key from a randomly generated “secret value”.
- KGC provides a partial private key for user’s identity.
- Encryption requires the user’s public key and user’s identity.
- Decryption requires a private key based on user’s secret value and partial private key.

CL-PKE Advantages

- No key escrow
  - User-generated secret-value prevents TA to recover keys.
- No explicit certification of public keys is required
  - Adversary does not know partial private key so cannot calculate the full private key.
  - However, we should assume that TA is not engaged in active adversarial behavior.
- A complete suite of certificateless cryptographic primitives and protocols is available:
  - Digital Signatures
  - Key Exchange (KE) and Authenticated-Key Exchange (AKE) protocols
  - Hierarchical schemes
  - Signcryption

CL-PKC Drawbacks

- Is not purely identity-based: Identifier and public key are required for encryption.
- As in ID-PKC, a secure channel is required for delivery of partial private keys.
- Revocation is a potential problem
- Does not attain full security of a traditional PKI, since TA may cheat.
  - TA should mount an active attack for replacing public keys. It is better than the ID-PKC where it can be done by a passive attack.

Al-Riyami & Paterson's Certificateless AKE (2003)

A		B
$a \in_R \mathbb{Z}_q^*;\ T_A = aP$		$b \in_R \mathbb{Z}_q^*;\ T_B = bP$
	$\xrightarrow{T_A,\, P_A}$	
	$\xleftarrow{T_B,\, P_B}$	
$e(X_B, P_{\text{KGC}}) \stackrel{?}{=} e(Y_B, P)$		$e(X_A, P_{\text{KGC}}) \stackrel{?}{=} e(Y_A, P)$
$K_A = e(Q_B, Y_B)^a e(S_A, T_B)$		$K_B = e(S_B, T_A) e(Q_A, Y_A)^b$
$h'(K_A \| aT_B)$		$h'(K_B \| bT_A)$.

KGC's master private key: s
KGC's master public key $P_{KGC} = sP$
Public parameters: $(G_1, G_T$, e, q, P, P_{KGC}, h, h')
Alice's secret value: x_A
$Q_A = h(ID_A)$
Alice's partial private key (issued by KGC): $D_A = sQ_A$
Alice's Public key: $(X_A, Y_A) = (x_i P,\ x_i P_{KGC})$

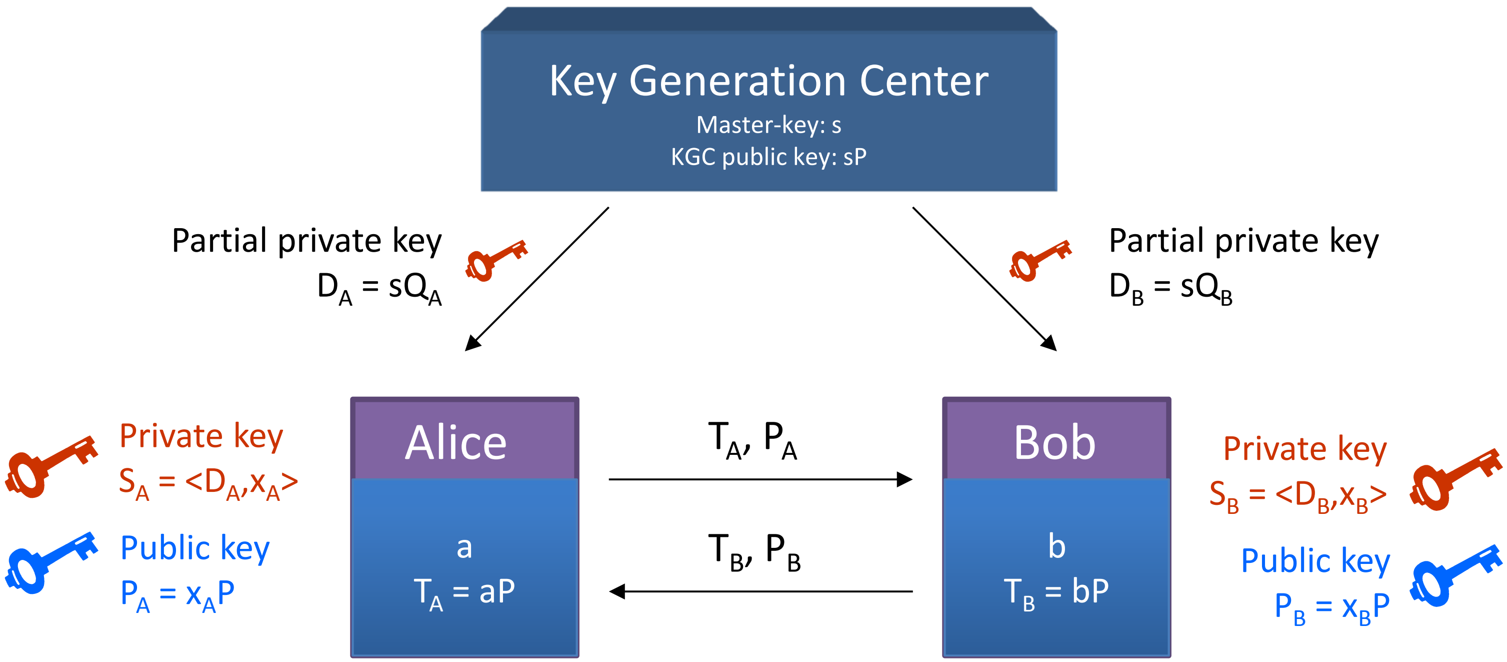
K_A = e(QB, YB)^a e(SA, TB) =

e(QB, xBsP)^a e(xAsQA, bP) =

e($x_B sQ_B$, aP) e(QA, x_AsP)^b =

e(SB, TA) e(QA, YA)^b = K_B

Another example of a Certificateless AKE Protocol (Mandt, 2006)



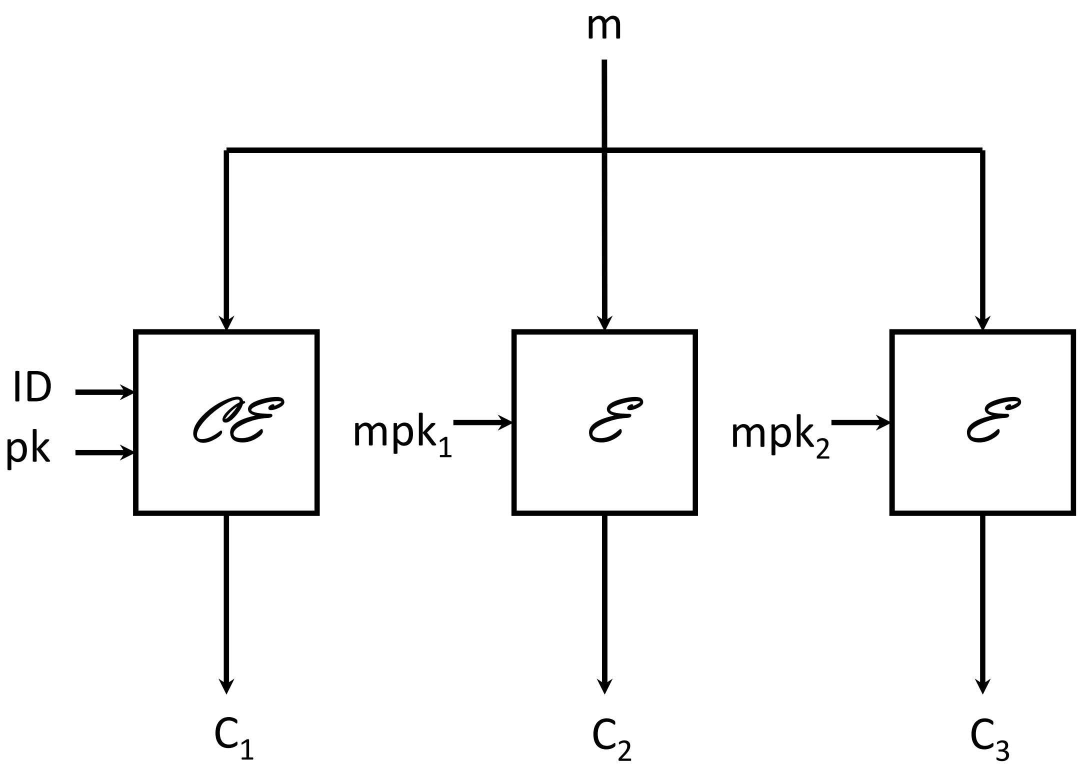
$$K_A = \hat{e}(Q_B, P_B + sP)^a \cdot \hat{e}(x_A Q_A + D_A, T_B) \quad \Big| \quad K_B = \hat{e}(Q_A, P_A + sP)^b \cdot \hat{e}(x_B Q_B + D_B, T_A)$$

$$K = \hat{e}(Q_B, P)^{a(s+x_B)} \cdot \hat{e}(Q_A, P)^{b(s+x_A)}$$

A Certificateless AKE Protocol without bilinear pairings (He et. al, 2011)

A		B
Generate a random number a;		Generate a random number b;
$T_A = a \cdot P$;	$M_1 = \{ID_A, T_A\} \longrightarrow$	$T_B = b \cdot P$;
	$\longleftarrow M_2 = \{ID_B, T_B\}$	
$K_{AB}^1 = (x_A + s_A)T_B + a \cdot (P_B + R_B + H_1(ID_B, R_B, P_B)P_{pub})$;		$K_{BA}^1 = (x_B + s_B)T_A + b \cdot (P_A + R_A + H_1(ID_A, R_A, P_A)P_{pub})$;
$K_{AB}^2 = a \cdot T_B$;		$K_{BA}^2 = b \cdot T_A$;
$sk = H_2(ID_A \parallel ID_B \parallel T_A \parallel T_B \parallel K_{AB}^1 \parallel K_{AB}^2)$		$sk = H_2(ID_A \parallel ID_B \parallel T_A \parallel T_B \parallel K_{BA}^1 \parallel K_{BA}^2)$

Strongly Secure Certificateless Encryption
(Dent et al., PKC'08)



- ID and pk are the user's identity and public key.
- mpk_1 and mpk_2 are part of system parameters
- Decryption process uses the certificateless encryption scheme

\+ NIZK proof that (C_1,C_2,C_3) are all encryptions of the same message.

One passively secure certificateless encryption scheme: $\mathcal{CE}$
Two instances of a passively secure public-key encryption schemes: $\mathcal{E}$

Questions?