Digital Signatures: A Panoramic View

Palash Sarkar

Applied Statistics Unit
Indian Statistical Institute, Kolkata
India
palash@isical.ac.in

International Conference on Electrical Engineering, Computing Science and Automatic Control, 2015
October 28, 2015

Structure of the Presentation

- Public key encryption and digital signatures.
- Examples of Digital Signatures.
- Public Key Infrastructure.
- Bitcoins: An Application of Digital Signatures.
- A Bit of Formalism.
- The Multitude of Digital Signatures.
- Signatures, E-Commerce and Law.
- Real-World Attacks on PKI.

Public Key Encryption and Digital Signatures

Cryptology: The Background Science

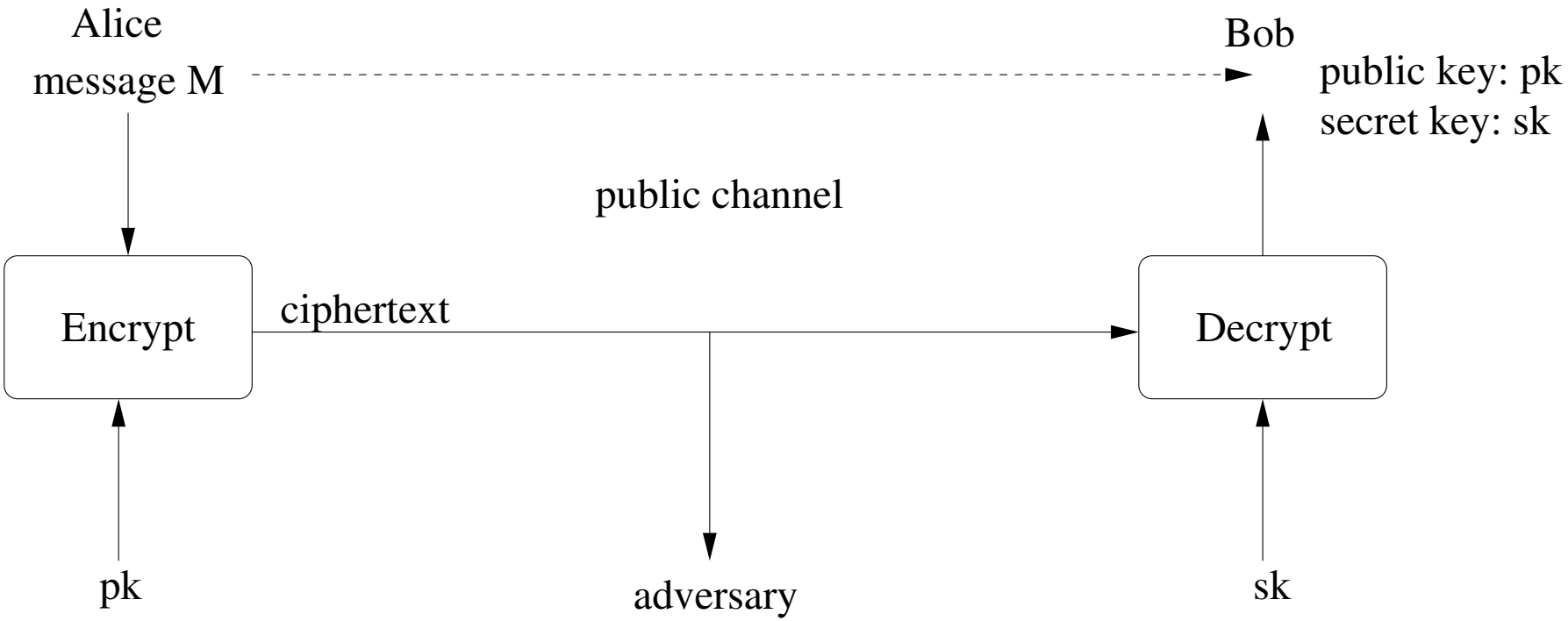
Two basic tasks.

- Encryption.
- Authentication.

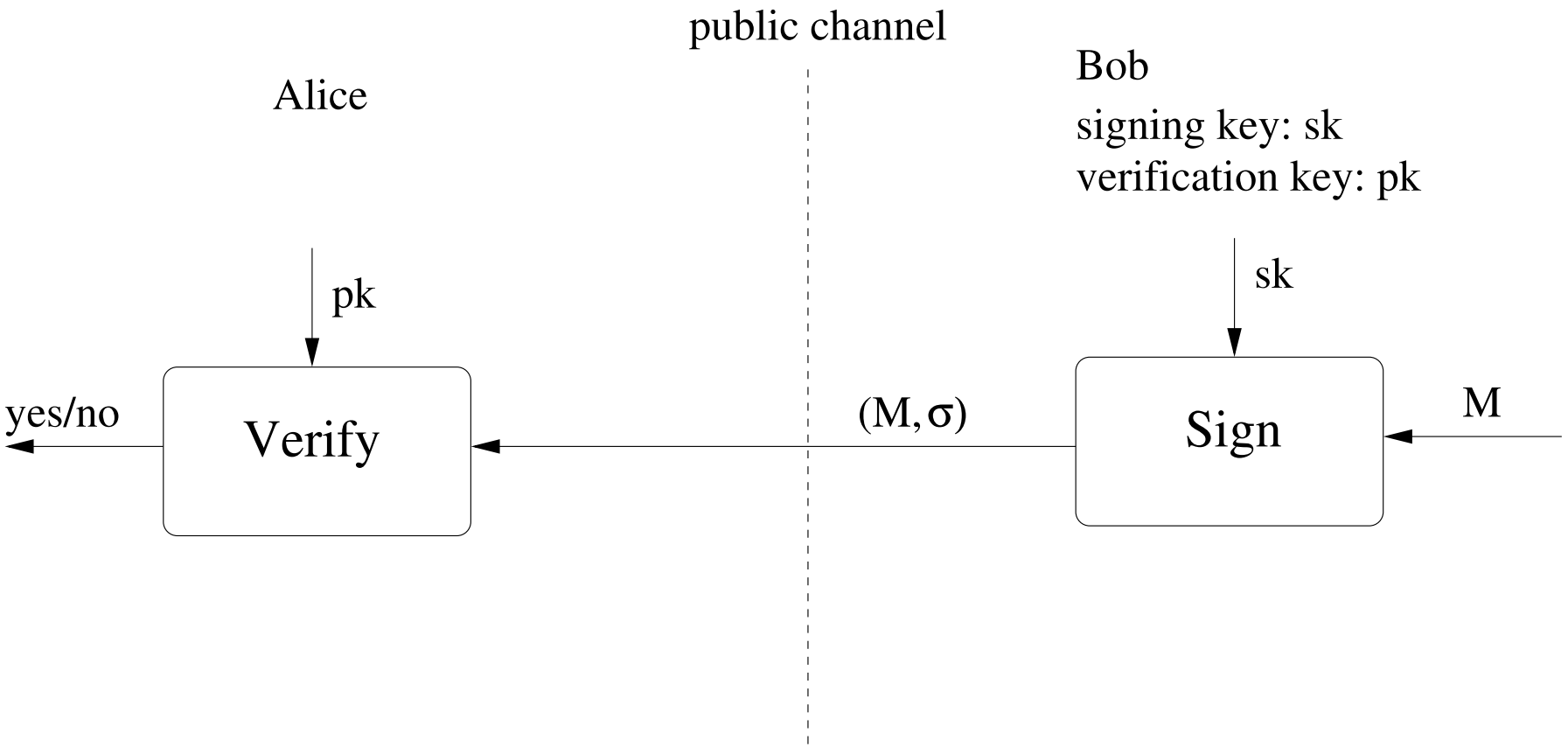
Two basic notions.

- Conventional or classical notion: secret or symmetric key cryptosystems.
- Paradigm shift: asymmetric key cryptosystem (Diffie-Hellman, 1976).
  - Public key agreement.
  - Public key encryption.
  - Digital signature.
- In practice, a combination of symmetric and asymmetric cryptosystems are used.

Overview of Public Key Encryption



Overview of Digital Signature Scheme



Digital Signature Schemes

Consists of three procedures: (Setup, Sign, Verify).

- **Setup:** generates (pk_B, sk_B) for Bob;
  pk_B is made public (placed in a public directory).
- **Sign:** Bob signs message m using sk_B to obtain signature σ.
- **Verify:** Alice can verify the validity of (m, σ) using pk_B;
  Alice does not need any secret information to verify a signature.

Hash-then-Sign

Signing a long message.

- Apply a hash function to the message to produce a short (160-bit, 256-bit) digest.
  - Collision resistant: It is computationally difficult to find two distinct messages which map to the same digest.
  - Pre-image resistant: Given a digest, it is computationally difficult to find a message which maps to the digest.
  - NIST standards: SHA-1, SHA-2 (256/512), SHA-3.
- Apply the signature scheme to the digest.

Examples of Digital Signatures

(Basic) RSA Signature Scheme

Set-up:

- Choose two "safe" primes p and q and set $n = pq$.
- Choose e and d such that $ed \equiv 1 \bmod \phi(n)$.
- Choose a hash function H which maps bit strings to $\{0, \ldots, n-1\}$.
- public key: (e, n, H); secret key: d.

Sign: message m is a bit string.

- Compute $y = H(m)$; $\sigma = y^d \bmod n$;
- the signature on m is σ.

Verify: message-signature pair is m and σ;

- Compute $y = H(m)$; accept if and only if $\sigma^e \bmod n = y$.

Correctness: $\sigma^e \bmod n = (y^d)^e \bmod n = y^{ed} \bmod n = y$.
Security: Factoring is computationally hard.

Elliptic Curves

Let $\mathbb{F}_q$ be a finite field of characteristic not equal to 2 or 3.

$$E/\mathbb{F}_q : \quad y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_q, \quad 4a^3 + 27b^2 \neq 0.$$

$$E(\mathbb{F}_q) = \{(\alpha, \beta) \in \mathbb{F}_q^2 : \beta^2 = \alpha^3 + a\alpha + b\}.$$

- $E(\mathbb{F}_q)$ forms an abelian group.
- Cryptography is done over a large prime order (cyclic) subgroup $\langle G \rangle$ of $E(\mathbb{F}_q)$.
- Characteristic 2 and 3: other forms of elliptic curves.

NIST Standard ECDSA: A Bare Description

Domain parameters (q, E, G, n, H):

- Elliptic curve E over $\mathbb{F}_q$;
- $\langle G \rangle$ a cyclic subgroup of $E(\mathbb{F}_q)$ of prime order n;
- hash function H.

Set-up:

- Choose d uniformly at random from $[1, \ldots, n-1]$.
- Set $Q = dG$.
- Signing key: d; Verification key: Q.

(Bare) ECDSA

Sign:

- Choose k uniformly at random from $[1, \ldots, n-1]$;
- $r = x_1 \bmod n$ where $kG = (x_1, y_1)$;
- $e = H(m)$; $s = k^{-1}(e + dr) \bmod n$.
- Signature on m is (r, s).

Verify:

- $e = H(m)$; $w = s^{-1} \bmod n$;
- $u_1 = ew \bmod n$; $u_2 = rw \bmod n$; $X = u_1 G + u_2 Q$;
- $v = x_1 \bmod n$ where $X = (x_1, y_1)$;
- if $v = r$ accept, else reject.

Correctness:

- $w = s^{-1} = k(e + dr)^{-1}$;
- $ewG + rwQ = ewG + rwdG = w(e + dr)G = kG$.

(Bare) ECDSA

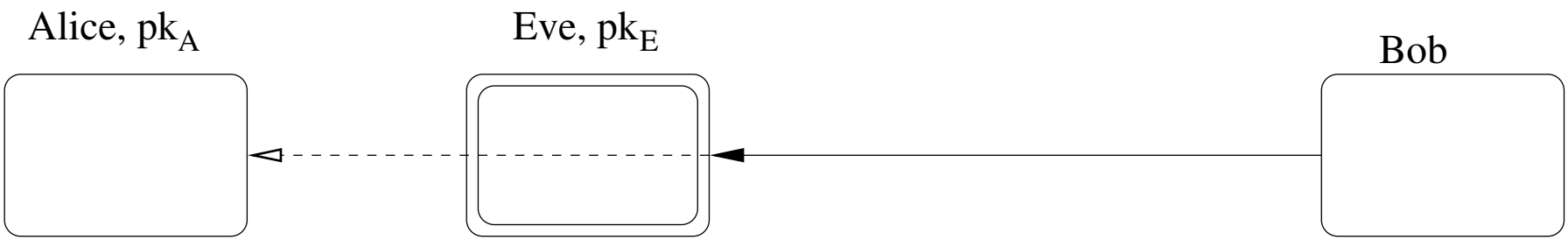
Security: The discrete log problem for certain elliptic curve groups is computationally difficult.

- Unlike factoring, there is no known sub-exponential algorithm for ECDLP.

Advantage: Possible to work with smaller size groups leading to shorter keys and signatures and faster signing and verification algorithms.

Public Key Infrastructure

How to Trust a Public Key?

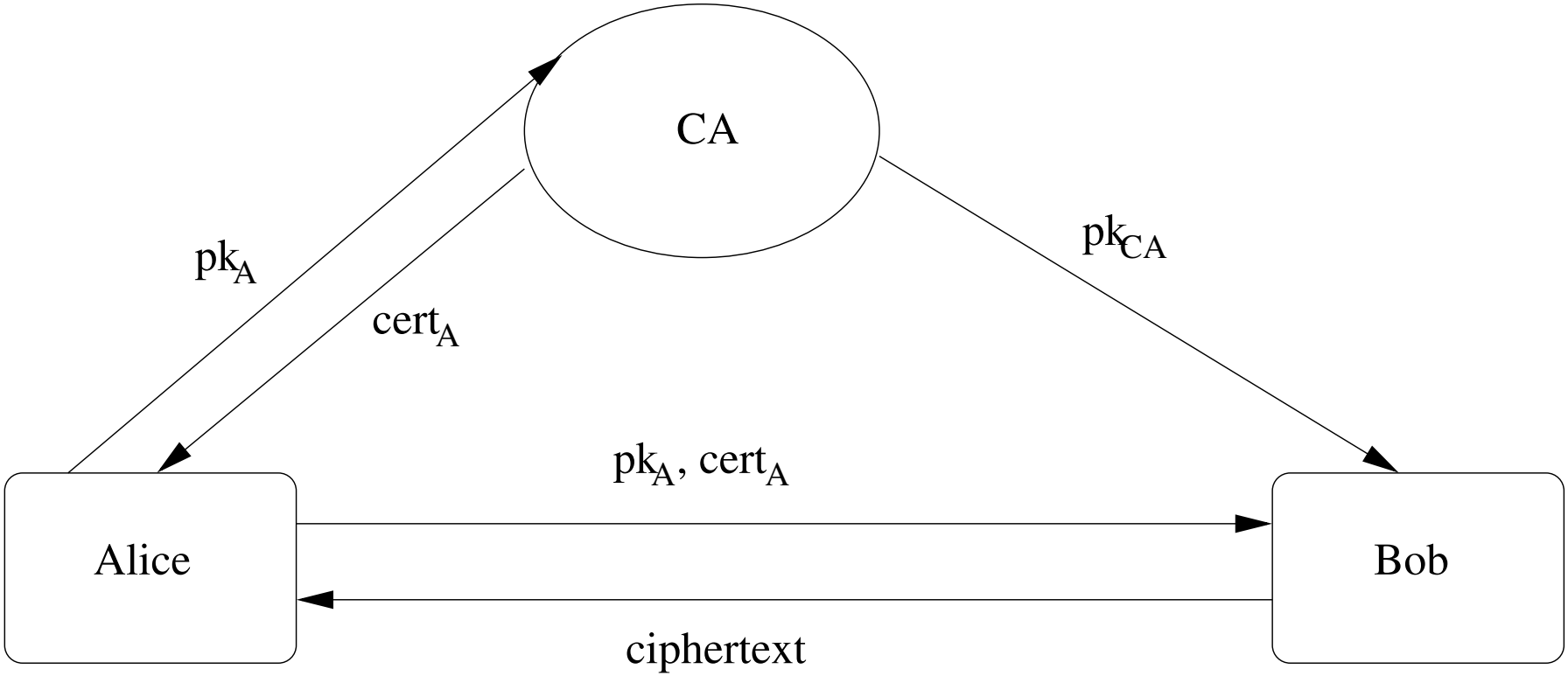


(Wo)man in the Middle

- Eve impersonates Alice.
- Puts a public key pk_E in the name of Alice.
- Eve signs a message M using sk_E.
- Bob verifies the signature using pk_E that he thinks is Alice's public key.

Question: when can Bob *trust* that the public key is indeed that of Alice?

Certifying Authority



Certifying Authority

- A CA has a key pair (pk_C, sk_C).
- Alice obtains certificate.
  - Alice generates (pk_A, sk_A); sends pk_A to CA.
  - CA signs (Alice, pk_A) using sk_C to obtain σ_A; Alice's certificate: (Alice, pk_A, σ_A).
- Bob verifies (M, σ) signed by Alice.
  - Verifies (Alice, pk_A, σ_A) using pk_C.
  - Verifies (M, σ) using pk_A.
- **Trust:**
  - Bob trusts pk_C;
  - hence, Bob trusts pk_A.

Management of Certificates

- A CA may revoke Alice's certificate.
  - Alice has lost her private key.
  - The validity of the certificate has expired.
  - Other reasons?
- Bob needs to know whether Alice's certificate is "fresh".
  - Certificate revocation list (CRL).
  - Online certificate status protocol (OCSP).
  - One-way hash chains.

X.509 Certificate Format

- version number
- serial number
- signature algorithm ID
- issuer name
- validity period
- subject name (i.e., certificate owner)
- certificate owner's public key
- optional fields
- the CA's signature on all previous fields

TLS and HTTPS

Transport Layer Security: (successor of SSL)

- Creates a secure channel between a client and a server.
- Authenticates server's certificate using public key of the relevant CA.

HTTPS

- Uses TLS to create secure HTTP connection.
- Browsers store certificates and public keys of CAs.

GPG and Tor: Free Tools

Gnu Privacy Guard (GPG)

- Allows generation of public/private key pairs.
- Allows performing message signing, encryption and decryption.
- Compliant with OpenPGP.

Tor

- Functionality:
  - Provides anonymous communication; directs traffic through a volunteer network of several thousand relays.
  - Makes it difficult for internet activity to be traced back to the user.
- Role of signatures:
  - Download: **https**://www.torproject.org/.
  - Each download file has a GPG signature.
  - Signing keys are publicly known and can be imported using GnuPG.

Bitcoins: An Application of Digital Signatures.

Bitcoins Basics

Bitcoins: owned by public keys (addresses).

- Created by a mining procedure.
- At most (about) 21 million can be created.
- Each bitcoin is divisible into 10^8 parts; if required, divisibility can be increased.

Transaction: ownership of a coin is transferred when the current owner signs a hash of the previous transaction and the public key of the new owner.

- Use of ECDSA for signing and SHA-256 for hashing.
- Possible to have multiple INs and multiple OUTs.
- Transation fee: difference between input and output bitcoins.

How to prevent double spending?

Bitcoins: Proof of Work

Block: a group of transactions.

Block chain: a chain of blocks where the next block includes the hash of the previous block.

Proof of work: Generating a hash of a block requires some effort but, verifying the hash to be correct is fast.

- Repeatedly increment a nonce and hash the block until a required number of leading zeros is obtained.
- Incentive for block mining: (first) miner gets the transaction fees.
- Mining bitcoins is similar process and has a diminishing incentive.

Bitcoin Network: Doing Away with TA

A peer-to-peer network; nodes keep complete details of all transactions.

Transactions are broadcast to the network; nodes group them into blocks and chain it into the block chain and broadcast the new block.

Transactions are considered valid when they are embedded into the block chain to a certain depth.

Ownership of a coin can be verified by querying the network.

Double spending avoided by maintaining a public history of all transactions and with the majority of nodes agreeing to which of two transactions is the first one.

A Bit of Formalism

Digital Signature: Formal Security Definition

Adaptive Chosen Message Attack:
A game between an adversary and a simulator.

Simulator:
Runs the set-up algorithm to generate signing and verification keys; gives the verification key to the adversary; keeps the signing key.

Adversary:
Queries the simulator on messages of its choice and receives in return a proper signature generated using the secret signing key.

Finally outputs a forgery, i.e., a msg-sig pair; wins if the forged pair 'verifies' with the verification key.

Unforgeable: If the probability of the adversary winning is small.

Digital Signature: Formal Security Definition

Existentially forgery: If the msg-sig pair is ‘new’.

Selectively forgery: The adversary chooses the forgery message before receiving the verification key.

Universally forgery: The adversary is able to forge signatures on any message.

Complete break: The adversary is able to recover the signing key.

Reductionist Security Assurance

If

(smaller protocols are secure
and)
some problem Π is computationally hard

then

the main protocol is secure.

May require additional assumptions.

- Random oracle: assume one (or more) of the functions to be a uniform random function.

Structure of Proofs

$$\underline{\text{A Game Sequence}}$$
$$G_0,$$
$$G_1,$$
$$\vdots$$
$$G_k$$

- Let X_i be the event that the adversary is succesful in Game G_i. We consider

$$\begin{aligned}
&\Pr[X_0],\\
&\Pr[X_0] - \Pr[X_1],\\
&\vdots\\
&\Pr[X_{k-1}] - \Pr[X_k]\\
&\Pr[X_k].
\end{aligned}$$

Structure of Proofs (contd.)

- G_0 is the game which defines the security of the protocol and so

$$\mathsf{Adv}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins in } G_0].$$

- G_k is designed such that $\Pr[\mathcal{A}$ wins in $G_k]$ is small.
- Games G_{i-1} and G_i differ:
  - the difference is not too much;
  - the adversary should not be able to notice whether it is playing Game G_{i-1} or Game G_i.

(Modern) Goals of Designing Signature Schemes

Efficiency:

- 'Short' signatures: reduce the size of signatures so that transmission bandwidth is reduced.
- Sizes of the verification and signing keys.
- Fast signing algorithm.
- Fast verification algorithm.

Reductionist Security:

- Based on some well studied (standard) computationally hard problem.
- Tightness of the reduction.
- Use of random oracles.

Bilinear Maps

$$e : G_1 \times G_2 \rightarrow G_T$$

- G_1, G_2 and G_T are groups of the same prime order p; practical examples arise from elliptic curve groups.
- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$.
- Non-degenerate.
- Efficiently computable.
- Symmetric (Type-I): $G_1 = G_2$;
  Asymmetric (Type-II or Type-III): $G_1 \neq G_2$.
- Currently Type-III pairings offer the best performance.

Provable Signature Scheme: An Example

Waters' 2005 signature:

- Type-I pairings; signatures consist of two group elements.
- For signing 256-bit messages, the public key consists of 260 group elements.
- Security based on the Computational Diffie-Hellman assumption.
- Basis for many later proposals of signature schemes.

Improvements (Chatterjee-Sarkar):

- Type-3 pairings; signatures consist of 2 elements of G_1.
- For signing 256-bit messages, the public key has less than 40 group elements; trade-off is some loss of tightness.
- Security based on the co-CDH problem.
- Extensions to hierarchical identity-based setting.

The Multitude of Digital Signatures

The Digital Signature Zoo

Signcryption: combined encryption and signing with the goal of being faster than individual encryption and signing.

Proxy signatures: allows the partial delegation of signing capability.

Blind signatures: the message is blinded before it is signed; later the signature can be verified with the unblinded message.

Group signatures: allows a member of a group to anonymously sign a message on behalf of the group.

Ring signatures: allows an entity to form a group and sign a message on behalf of the group without the consent of the group.

The Digital Signature Zoo

Designated verifier signatures: verification can be done only by a single entity which is chosen by the signer during the signing process.

Aggregate signature scheme: allows the aggregation of signatures of different users on different messages into a single signature; verification on the single signature verifies all the signatures.

Structure preserving signatures: Messages, signatures and verification keys are elements of groups involved in a bilinear pairing; Verification is done by evaluating pairing product equations.

Other notions: unique signatures, multi-signatures, identity-based, short signatures, ...

Post-Quantum Signatures

- Hash function based: One-time signatures, Merkle signatures.
- Coding theory based.
- Lattice based.
- Multivariate-quadratic-equations.

`http://pqcrypto.org/`

Signatures, E-Commerce and Law

Digital World

A new way of interaction and communication.

- **e-commerce:** “trading in products or services using computer networks, such as the Internet.” (Wikipedia)
- **e-government:** “The employment of the Internet and the world-wide-web for delivering government information and services to the citizens.” (Wikipedia)

Digital signatures are the basic building blocks.

The Legal Aspect

- For digital signatures to be accepted, the law has to recognise these as legal.
- United Nations Commission on International Trade Law (UNCITRAL).
  - Formulated a model law on e-commerce in 1996.
  - Adopted by the General Assembly resolution 51/162 of 16 December 1996.

"Recommends that all States give favourable consideration to the Model Law when they enact or revise their laws, in view of the need for uniformity of the law applicable to alternatives to paper-based methods of communication and storage of information;"

Adoption of Digital Signature Laws

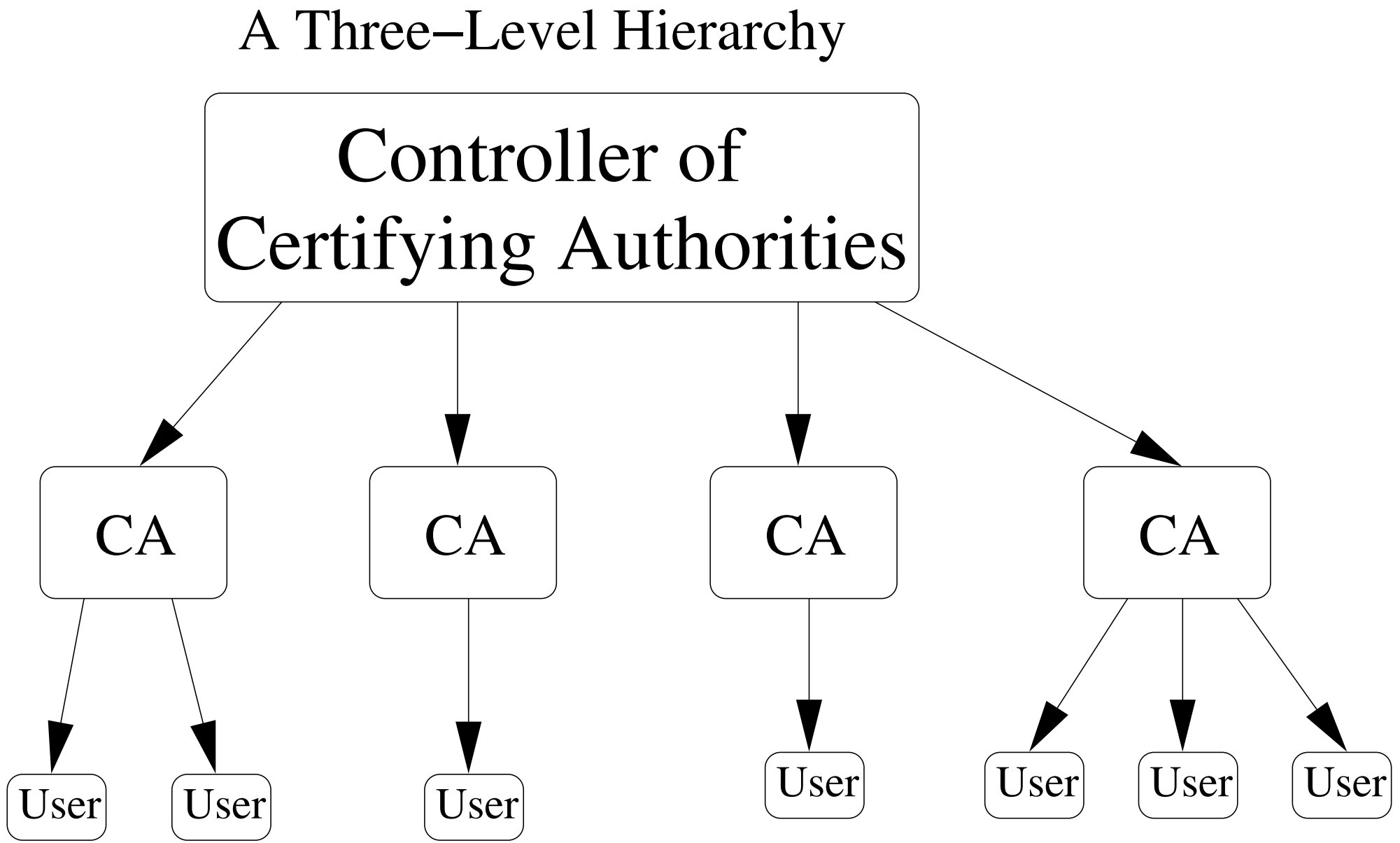
More than 60 countries (as per a 2011 report) have enacted laws providing legal force to digital signatures.

- Mexico.
  - In 2000, modifications to the Civil Code, the Civil Procedures Code, the Commerce Code and the Consumer Protection Law were made.
  - In 2011, an advanced digital signature law was adopted by the senate.
- India: Information technology act, 2000 and 2006.

Indian IT Act, 2000, 2006

- Provides legal sanctity to digital signatures based upon the principle of equivalence to handwritten signatures.
- Provides for the creation and management of PKI in India.
- Cascaded amendments to several other acts.
  - Indian Evidence Act, 1872.
  - Banker's Book Evidence Act, 1891.
  - Reserve Bank of India Act, 1934.
  - Indian Penal Code.
- Covers aspects other than digital signatures.
  - Issues related to digital distribution of obscenity.
  - Issues related to wire-tapping by governmental agencies.
  - Issues related to 'offensive' digital messages; was recently struck down by the Supreme Court.

PKI-India Framework



Three-Level Hierarchy

- The CCA (or root CA) only issues certificates to CAs.
- The CAs issue certificates to individual users.
  - Certain CAs issue certificates to certain category of users.
- There are no lower level CAs, i.e., a CA cannot issue a certificate to another CA.
- Trust in a certificate is ultimately derived from the root CA.
- Cross-certification with a foreign CA.
  - An individual CA can arrange for cross-certification after due approval by the CCA, India.

Functions of the CCA

- Creation and maintenance of the Root CA of India (RCAI).
  - Root CA certificate is a self-signed certificate. It is based on the ITU-T X.509 standard.
  - Protection of private key of CCA (using tamper proof hardware and 3-out-of-3 access control).
- Issue certificates to individual CAs.
- Maintain the national repository of digital certificates (NRDC) (mandated under Section 20 of the IT Act): copies of all certificates and certificate revocation lists.
- Empanel auditors for auditing infrastructure of CAs.
- Generally act as the controlling authority of all PKI-related issues in India.

From the Indian IT Act

"If, by application of a security procedure agreed to by the parties concerned, it can be verified that a digital signature, at the time it was affixed, was –

(a) unique to the subscriber affixing it;

(b) capable of identifying such subscriber;

(c) created in a manner or using a means under the exclusive control of the subscriber and is linked to the electronic record to which it relates in such a manner that if the electronic record was altered then digital signature would be invalidated,

then such digital signature shall be deemed to be a secure digital signature."

Question. What is the relationship of the above to the scientific definition of secure digital signature?

From the Indian IT Act

"A has a letter of credit upon B for Rupees 10,000, written by Z. A, in order to defraud B, adds a cipher to the 10,000, and makes the sum 1,00,000 intending that it may be believed by B that Z so wrote the letter. A has committed forgery."

"A signs his own name to a bill of exchange, intending that it may be believed that the bill was drawn by another person of the same name. A has committed forgery."

There are 16 such illustrations.

Question: Can one come up with a good explanation of how and why the scientific definition of secure digital signature rules out these and similar cases?

Digital versus Physical Signatures

Physical signatures:

- Directly linked to the “cognitive bio-mechanics” of a person.
- Forgery can be detected only by experts.
- Can be archived for a long time.

Digital signatures:

- The signing key is a bit string which is divorced from the biological entity.
- Long term archival is a serious problem.
- Offers a wider range of functionalities.

It is unlikely that digital signatures will *replace* physical signatures.

Real-World Attacks on PKI

Forgery

Malware 'Flame': Reported in 2012.

- The attackers identified a Microsoft certificate that was used for signing code updates.
- The signing algorithm used the weak MD5 hash algorithm.
- The attackers created a (new) chosen prefix collision for MD5.
- This was used to fraudulently sign some components of the malware to make them appear to have originated from Microsoft.

> *"the design of Flame is partly based on world-class cryptanalysis."*
>
> *– Marc Stevens*

Fake Certificates

Comodo: Reported in 2011.

- A certificate selling authority.
- A user account of one of its registered authorities was breached.
- Some fake certificates were issued for several popular domain names such as Yahoo, Google and Skype.
- With additional control over DNS servers this would have proved very dangerous.

Diginotar: Reported in 2011.

- Was hacked and fake certificates were issued; about 500 have been reported.
- The company went bankrupt.

Snowden Revelations (2013)

Alleged Man-in-the-Middle attack by NSA:

- An internet router was hacked and targeted traffic was redirected.
- A fake certificate was used for authentication.
  - Matthew Green suggests that NSA could have obtained their own signing key from a less trustworthy CA and then used it to create and sign fake certificates.
  - Bruce Schneier suggests that the attack could be linked to the fake certificate issue from Diginotar hack.
- Could intercept information in unencrypted format.

Gracias por su amable atención!