

## Public Key Cryptography in Practice

## How Cryptography is Used in Applications

The main drawback of public key cryptography is the inherent slow speed of the public key schemes.

There are only a few schemes which are relatively faster, but they require use of huge keys, and are thus impractical.

Therefore, public key schemes are not used directly for encryption.

Instead, public key schemes are used in conjunction with secret key schemes where encryption is performed by the secret key schemes (e.g., Triple-DES) and the agreement on the keys is performed by public key distribution schemes (e.g., using RSA or Diffie-Hellman).

This is similar to the case described in the public key signature schemes, where the signature scheme does not sign the original message, but rather signs the result of a fast hash function.

Moreover, in many application even the single public key signature on a message is too cumbersome. In such cases, MACs are used, and their key is distributed in advance by a public key distribution scheme.

## Recommended Key Sizes

In secret key schemes the trend changes from keys of 56–64 to keys of 128 bits. Keys of 128 bits are large enough to thwart any practical attack, as long as the cipher does not have weakness due to its design. Paranoids can use even longer keys, which are supported by various ciphers.

The situation is different in public key schemes, where considerably longer keys are required, as the keys are not uniformly selected from all the possible keys with the same length. Therefore, the number of keys is (slightly) smaller than the number of values of the same length as the keys.

However, the main reason that requires longer keys is the information inherited in the key due to the properties of the cipher.

## Recommended Key Sizes (cont.)

In RSA, the public key is a product of two primes. The best known factoring algorithms are the quadratic sieve and the number field sieve whose complexities are about

$$\text{Complexity(QS)} = e^{c\sqrt{\ln n \ln \ln n}} ; \quad \text{Complexity(NFS)} = e^{c(\ln n)^{1/3}(\ln \ln n)^{2/3}}$$

Due to the different constant factors (and other smaller terms) the quadratic sieve is faster when factoring up to about 129 decimal digits. The quadratic sieve algorithm was used to factor the number RSA-129, proposed by the designers of RSA in 1978 as an example of a number whose factoring will take about 40 quadrillion years. This factorization took a few months on several thousands computers over the Internet.

Over a similar computer network the NFS can factor numbers up to about 140 digits. It is expected that within a decade numbers of up to 154 digits (512 bits) will be factorable.

Therefore, all new applications should use public keys of 1024 bits. Long-term keys should have at least 2048 bits. Paranoids can use longer keys.

## Recommended Key Sizes (cont.)

Breaking DLOG-based schemes requires computation of discrete logarithms. Advances in designing algorithms for computing DLOG were performed in parallel to the design of algorithms for factoring, and actually the best known algorithms for computing DLOG are variants of those used for factoring.

Nowadays, 400-bit primes moduli are still secure for DLOG-based schemes. However, it would better be that new applications use longer keys (e.g., 512 bits).

## Public Key Infrastructure

Public key cryptography provide a tool for secure communication between parties by letting them trust messages encrypted or signed by the **already known** public keys of the other parties.

However, no algorithmic scheme can solve the original trust problem of accepting the identity of a party that you never met.

The usual face-to-face identification is by a trusted third party (a friend) who presents the two parties to each other.

Such a presentation protocol is also required for cryptographic protocols.

The presenting party in the cryptographic environment is called a **certification authority** (גורם חתימה), or briefly a **CA**. The management of the CA's requires a **public key infrastructure (PKI)**.

## Certificates

During face-to-face presentation, the presenter gives the relation between the name and the face of the presented party, together with some side information (e.g., he is a friend of the presenter).

For cryptographic use the certification authority should give the relation between the **public key** and the **identity** of his owner.

This information should be transmitted authenticated from the CA to the receiver, e.g., signed under the widely known public key of the certification authority.

Note that it is not necessary that the receiver communicate directly with the CA. Instead, the CA signs all the required information, and gives the key owner, who can then give it to anybody he wishes to communicate with, or publish widely. This, the receiver should only verify the signature of the CA, rather than to communicate with him for verifying every new key.

Such a signed information is called a **certificate** (תעודת אישור).

## Certificates (cont.)

A certificate includes

1. The CA name
2. Sequential number of the certificate
3. The public key of the user
4. The identity of the user
5. Date
6. Last validation date
7. signature of the CA on all the above

### Certificates (cont.)

It might happen that the secret key of some user become known to other, due to theft, factoring, or other reasons. Therefore, certification authorities maintain **Certificate Revocation Lists** (CRLs, blacklists) of canceled certificates which must not be trusted. Users can ask to add their old certificates to the blacklists if they suspect that their secret keys became known.

The last validation date field in the certificate ensures that the blacklists will not have to keep such certificates for more than a selected time, as after the last validation date, the certificates become invalid anyway, and the user should select another key instead.

### The Legal Status of Digital Signatures

Several countries, including Israel, made special law to approve digital signatures for legal purposes.

Under the Israeli law, there are three kinds of digital signatures:

1. **Digital signature** (חתימה אלקטרונית): Any kind of electronic data that is added to a document to show the identity of the signer (e.g., a scanned hand-written signature at the bottom of a document). This kind has no legal status.
2. **Protected digital signature** (חתימה אלקטרונית מאובטחת): A digital signature that allows verification of the identity of the signer, and ensures that the signed message is original and was not modified after the signature generation.
3. **Certified digital signature** (חתימה אלקטרונית מאושרת): A protected digital signature, whose key is certified by a certificate (signed by a CA).

### The Legal Status of Digital Signatures (cont.)

A CA should be approved by the CA registrar (רשם הנרמטים המאשרים), sign the certificates with a long key (at least 2048 bits in case of RSA), and satisfy many other security and financial requirements.

Certified digital signatures can be used whenever a signature is required by law (a few exceptions apply, e.g., wills), and courts accept certified digital signatures (and with the proper evidence also protected digital signatures) as valid signatures.

### The X.509 Public Key Hierarchy

The X.509 standard defines a tree hierarchy of CA's. Each CA has some "parent", who signs the certifies the CA's public key. The only required widely known public key is the key of the root CA. All other public keys of CA can be verifies using certificates.

Then, a receiver verifies a certificate of another user by verifying the certificate of the CA first, and then verifying the signature of the CA on the certificate of the user. In turn, verification of the certificate of the CA is performed by verifying the certificate of his parent CA and the signature of the parent CA, and so on. Only the public key of the root CA should not be verified, as it is widely known.

### The PGP Hierarchy

The drawback of the X.509 hierarchy is that every user must trust all the CA's. If a user does not trust even one CA, he cannot trust the system at all. That like "If you do not trust the government, you cannot trust your brother".

In the PGP hierarchy every user is also a CA, and users can select which CA's they trust, and which they do not trust.

As a CA a user signs certificates to his friends. His signature ensure that he recognizes the friend, and checked his identity. It does not mean that the friend is trustworthy.

Each user then asks for certificates from many other users, and collects as many as he wishes.

### The PGP Hierarchy (cont.)

When a user need to prove his identity, he publishes (or sends) the certificates he collected to the other user, and the other user verify them.

The receiver can select his own trust scheme. He can decide to trust certificates signed by some CA's unconditionally, and not trust certificates signed by some other CA's. He can also decide to trust some CA with some medium trust, i.e., a certificate is only trusted if he got one (or more) additional certificates for the same key from medium trust CA's.