Hard Invalidation of Electronic Signatures

Lucjan Hanzlik[1], Mirosław Kutyłowski[1], Moti Yung[2]

Wrocław University of Technology, Poland [1]
Google Inc., USA [2]

ISPEC 2015, Beijing

Cryptographic protection of e-documents

Integrity:

the document has not been manipulated since its creation

Proof of origin:

the author of the document is the same as the declared author

Cryptographic protection of e-documents

Integrity:

the document has not been manipulated since its creation

Proof of origin:

the author of the document is the same as the declared author

Solution:

an electronic signature of the author is attached to the document

Properties of e-signatures

A high level of protection provided that:

1. **we are sure that the declared public key really corresponds to the signatory**
2. nobody but the signatory holds a secret key corresponding the the signatory
3. the signatory really intended to sign the document

ad 1: PKI infrastructure becomes necessary

Properties of e-signatures

A high level of protection provided that:

1. we are sure that the declared public key really corresponds to the signatory
2. **nobody but the signatory holds a secret key corresponding the the signatory**
3. the signatory really intended to sign the document

ad 2: the signing key is implemented on a secure device

Properties of e-signatures

A high level of protection provided that:

1. we are sure that the declared public key really corresponds to the signatory
2. nobody but the signatory holds a secret key corresponding the the signatory
3. **the signatory really intended to sign the document**

ad 3: the device protected by a PIN number, a document to be signed correctly presented to the signatory . . .

Problems

Despite the strength of digital signature schemes there are multiple problems:

1. a **certification authority** confirming the public key may **fail** to perform its duties correctly or even may turn out to be **malicious**
2. the “secure device” may turn out to **leak the signing key** or ill designed (e.g. bad implementation of randomness)
3. **weak access protection** for the signing device

Further problems
business model

Life cycle of paper documents:

1. create
2. use
3. destroy or archive

Most documents are intended to be used only a limited period of time.

Destroying documents
document shredders

The reasons behind destroying paper documents:

1. **storing is costly, so destroying reduces the costs**
2. even useless documents need to be protected. Strict European rules. Forthcoming rules even harder:

Destroying documents
document shredders

The reasons behind destroying paper documents:

1. storing is costly, so destroying reduces the costs
2. **even useless documents need to be protected. Strict European rules. Forthcoming rules even harder:**

   COM (2012) 11: Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)
   failing to protect personal data is generally a crime

Destroying documents
document shredders

The reasons behind destroying paper documents:

3. **destroying unnecessary information is an important security rule**

Destroying digital documents

Problems:

- **logical erasure is not enough: physical copies may remain and might be recovered with some effort**
- sometimes we are not sure where the document is physically stored
- physical erasure might be a problem in some archives

Destroying digital documents

Problems:

- logical erasure is not enough: physical copies may remain and might be recovered with some effort
- **sometimes we are not sure where the document is physically stored**
- physical erasure might be a problem in some archives

Destroying digital documents

Problems:

- logical erasure is not enough: physical copies may remain and might be recovered with some effort
- sometimes we are not sure where the document is physically stored
- **physical erasure might be a problem in some archives**

these problems might be particularly acute for cloud systems

Destroying electronic signature

1. digital signature might be a disadvantage: it provides data quality guarantees **even if we do not wish so**
2. **revoking an X.509 certificate does not solve the problem**:
   - A from **mathematical** point of view no changes: revocation is only a declaration, the cryptographic proof persist
   - B from **legal** point of view nothing changes: previously signed documents remain legally valid

Some application scenarios

- **business negotiations:** after reaching a final agreement all intermediate documents from the negotiations should be destroyed (as sensitive data that should not be available to the third parties)
- **bulk invalidation of signatures:** in case of a major failure (like discovering a trapdoor in a whole population of signing cards) a large number of signatures have to revoked at once
- **time limited documents:** like a binding business offer, which after the deadline should have no real proof value

Challenge

while removing all copies of a digital document might be infeasible, **we aim to make a signature worthless**

Such problems have been discussed at the very beginning of digital signatures:

Saltzer, J.H.: On digital signatures. Operating Systems Review 12(2), 12-14 (1978)

> *... sound cryptographic and authentication strategies developed in isolation may be defective when embedded in a complete system, involving people and legal considerations.*

Signatures with Hard-Revocation
Model

Main actors in the system:

- **Revocation Authorities**
- **Signers**
- **Verifiers**

Signatures with Hard-Revocation
Model

High level description (see paper for more details):

- **Setup:** on input of the security parameter and a number of revocation authorities, outputs system parameters and key pairs for revocation authorities,
- **UserSetup:** on input of system parameters, outputs key pair of an user,
- **CreateGroup:** on input of revocation condition, revocation authority private key, outputs revocation group parameters and trapdoor,

Signatures with Hard-Revocation
Model

- **Sign:** on input of revocation group parameters, user's private key, outputs signature,
- **RevokeGroup:** on input of the revocation group trapdoor, the revocation authorities private key and outputs a revokation token,
- **Verify:** on input of the signature, the user's public key, the revocation group parameters and revocation authorities public key, outputs *valid* or *invalid* according to the verification,
- **ForgeSign:** alternative signing procedure, on input of user's public key, the revocation group parameters and token, outputs a valid signature (optionally it may require a valid signature of the user).

Signatures with Hard-Revocation
Model

Required properties:

- **Correctness:** should be obvious from context,
- **Unforgeability:** the goal of the adversary is to output a new signature, where the signature was not returned by the signing oracle, the corresponding user's and revocation authorities private key were not leaked and the corresponding revocation group was not revoked,
- **Hard Revocation:** the goal of the adversary is to distinguish signatures created by the user and third parties knowing the revocation token.

Solutions

We propose three solutions:

- two are based on a two tier signature method (require at least one valid signature of the user in a revocation group to forge signatures),
- the last one allows to use the main key pair and multiple revocation groups.

Verifiable Encryption
Core idea

- revocation group parameters contain the verifiable encryption public key,
- users use standard key pair as their long-term key,
- to sign a message, the user chooses an ephemeral key pair, certifies it using the long-term key and uses the verifiable encryption to encrypt the ephemeral private key (private key must be contained in the message space),

Verifiable Encryption
Core idea

- on revocation of a group, the revocation authority publishes the decryption key,
- verification checks the verifiable encryption part and the actual signature created using the ephemeral key pair, additionally it verifies the user's certificate on the ephemeral public key,
- to forge a signature one decrypts the ephemeral private key and uses it to sign the new message.

Verifiable Encryption
Instantiation

- Camenish-Shoup verifiable encryption, uses the group $\mathbb{Z}^*_{n^2}$ and message space $\mathbb{Z}_n$, can be used to verifiable encrypt $\log_\gamma \delta$, where γ can be a generator of prime order q of a different group,
- the signature scheme can be instantiated with any DLP based scheme in groups of prime order q, e.g. El-Gamal, Schnorr signature scheme.

Verifiable Encryption
Security

Unforgeability based on unforgeability of the signature scheme and the security of Camenish-Shoup verifiable encryption,

Hard Revocation based on soundness of the verifiable encryption.

Trapdoor DLP
Core idea

- each revocation parameters specify a group where DLP is hard but knowing a trapdoor makes it easy to compute (e.g. groups from Paillier cryptosystem $\mathbb{Z}^*_{n^2}$),
- users long-term keys are standard signature keys,
- to sign a message, the user chooses a random ephemeral key pair in the trapdoor group, certifies the ephemeral public key using the long-term key and uses a signature scheme in the trapdoor group to sign the actual message,

Trapdoor DLP
Core idea

- on revocation of a group, the authority publishes the trapdoor for computing DLP,
- to verify the signature one uses the verification procedure of the signature scheme in the trapdoor group, additionally one verifies the user's certificate on the ephemeral public key,
- to forge a signature one decrypts the ephemeral private key and uses it to sign the new message.

Trapdoor DLP
Instantiation

- trapdoor group from Paillier cryptosystem (multiplicative group $\mathbb{Z}^*_{n^2}$, where $n = pq$) or Okamoto-Uchiyama cryptosystem (multiplicative group $\mathbb{Z}_n$, where $n = p^2q$),
- Schnorr signature scheme with short exponents (order of group not always known to the signer).

Note: the GPS signature scheme (Girault-Poupard-Stern) introduced the idea of Schnorr signature scheme with groups of unknown order (we use this idea but additionally require that the group is a trapdoor DLP group).

Trapdoor DLP
Security

Unforgeability unforgeability of the GPS signature scheme in the trapdoor group,

Hard Revocation uncoditionally, since the ephemeral private key is computed from the public key.

Ring Signatures
Core idea

- common system parameters for all revocation groups,
- revocation group parameters specify a public key from the key space of the ring signature scheme,
- user's key pair is a ring signature key pair,
- to sign a message, the user creates a ring consisting of its public key and the revocation group public key (feature: can use multiple revocation groups at once),
- standard ring signature verification,

Ring Signatures
Core idea

- on revocation, the authority publishes the private key for the ring signature,
- to forge a signature, one uses the published private key and creates a ring consisting of the revocation group public key and the user's public key.

Note: a valid signature of the user is not needed in order to forge a message. However, we can define this contruction using the same two tier method as before and unify the definition.

Ring Signatures
Instantiation

- Schnorr based ring signatures (efficient solution),
- Shacham-Waters ring signatures (without random oracles),
- any other ring signature scheme.

Ring Signatures
Security

Unforgeability based on unforgeability of ring signature scheme,

Hard Revocation based on anonymity of ring signature scheme.

Conclusions

1. it is possible to build a system in which an electronic signature looses its proof value at a given time or event
2. such signature schemes can be created with well-known and well-studied techniques
3. the solutions are feasible for practical implementations

Thanks for your attention!