Security (CS4028)

Lecture 7. Digital Signatures

Chunyan Mu

chunyan.mu@abdn.ac.uk

The School of Natural and Computing Sciences

Lots of slides of this lecture adapted from Matthew Collinson's

Schedule

	Week	Lecture 1	Lecture 2	Tutorial
	1	Intro to course & security	Intro to Crypto	-
	2	Symmetric Crypto	Hash	Math for crypto
	3	Asymmetric Crypto-1	Asymmetric Crypto-2	Symmetric Crypto
⇒	4	Signatures	Zero Knowledge Proof	Asymmetric Crypto
	5	Certificates	Authentication	Signature & certificates
	6	Access Control	AC models	Authentication
	7	Information flow	Management	Access control
	8	Protocols	Communications	Concepts & management
	9	Network security	Network security	Protocols and communications
	10	Advanced topics	Advanced topics	Network
	11	Revision		

Lecture 7: Digital Signatures

Last lecture

- Public-key cryptography

This lecture

After having attended this session, you will be able to:

- understand what is a digital signature and how it works

Outline

Outline

Introduction

John Hancock's signature from the U.S. Declaration of Independence (public domain)

Physical signature

- What guarantees might we want to go along with a physical signature on a document?
- How will somebody relying upon the document be assured of these guarantees?

Introduction

Properties of handwritten signature on doc

We may (want to) have some confidence that the following properties hold of a handwritten signature on a document:

- The signature is authentic
  - the signature convinces the document's recipient that the genuine signer (deliberately) signed the document.
- The signature-document pair has integrity
  - it cannot be altered later
- The signature cannot be repudiated
  - the signer cannot claim later that he/she did not sign the document (as long as the document remains).

Introduction

Properties of handwritten sig on doc

Relating to the above:

- The signature is unforgeable
  - the signature is proof that the signer, and no one else, deliberately signed the document.
- The signature is unalterable
  - after the document is signed, it cannot be altered.
- The signature is not reusable
  - the signature is part of the document; an unscrupulous person cannot move the signature to a different document.

Outline

Digital signature

What is a digital signature scheme

- A **digital signature scheme** is a cryptographic mechanism.
- It produces **digital signatures** for messages.
- These are pieces of data that often accompany messages.

Digital signature

What is a digital signature?

- A **digital signature** is a mathematical scheme for demonstrating the authenticity (source) of a digital message or documents.
- A **valid** digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit.
- Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.

Digital signature

What is a digital signature?

- Digital signatures employ a type of asymmetric cryptography.
- Digital signature is used as a hand-written signature:
  - authentication (of source);
  - integrity (message-integrity);
  - non-repudiation.
- Basic security requirements
  - Unalterable: the document can not be modified once it has been signed.
  - Not reusable: no one, except the sender can move the signature to another document.

Digital signature

Status of digital signatures

- Digital signatures are central to (current implementations of) e-commerce, and to trust and security on the internet more generally.
- Legal status of some schemes is changing to become more solid, like handwritten signatures.

Digital Signature

Digital Signature Scheme

Digital Signature Scheme consists of

- Key generation algorithm
- Signing algorithm
- Verification algorithm
- (Protocol for use)

Digital signature

Cryptographic Operation

- Some signatures are encryptions
  - The verifier applies a decryption algorithm
  - Sometimes hashing is involved, so the verifier only gets back the hash value, not the original message.
- Some signatures are not encryptions (verifier does not apply a decryption)
  - ElGamal
  - Digital Signature Algorithm (DSA)

Digital signature

Digital Signatures & Verification Keys

- A **digital signature** is a value that depends on
  - the value of the document/message, and
  - a secret known only to the signer:
    - a private signature key
    - NOT known to the receiver, in particular.
- The signature associates the document with a **verification key**.
  - this is know by the receiver and any 3^{rd} party required to do verification,
  - it will often be public.

Digital signature

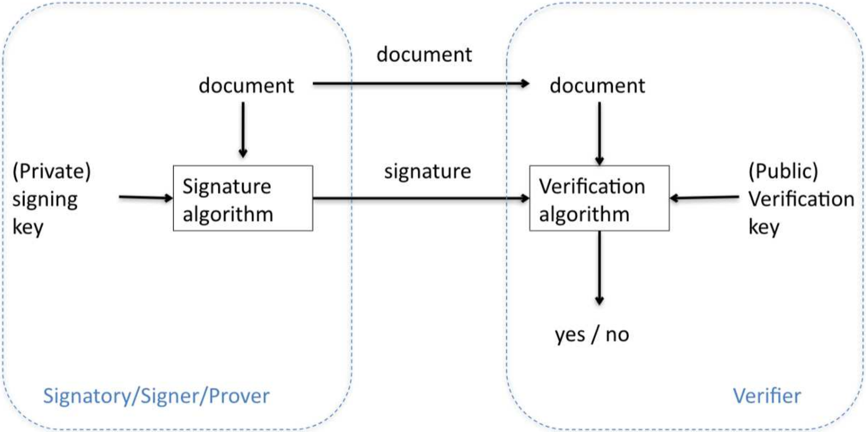
Algorithms

A digital signature scheme typically consists of three algorithms:

- A key generation algorithm:
  - input: a set of possible private keys
  - output: the private key and corresponding public key
- A signing algorithm:
  - input: a message and a private key,
  - output: a value (digital signature).
- A signature verifying algorithm:
  - input: a message, public key and a signature,
  - output: binary (yes/no) result

Digital signature

Digital signature basics



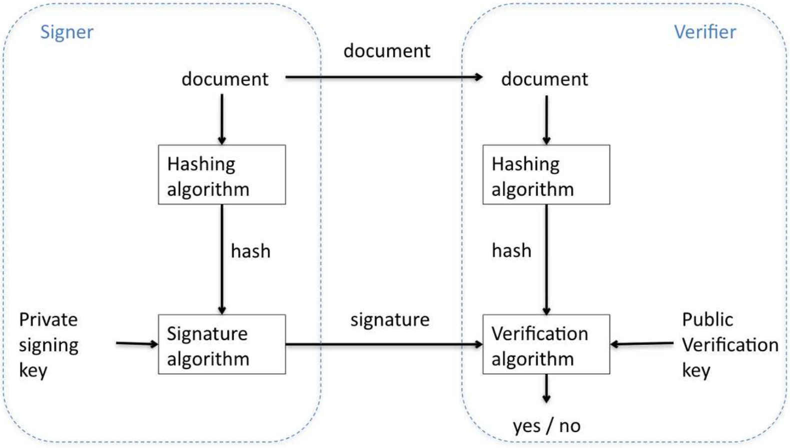
Hashed digital signature

Signing Hashes

- In practice, digital signatures are not very often not used quite as on the previous slide. In stead, we sign a hash of the original message (called 'message digest').
- There are two main types of reason:
  - Message might be big, and encrypting them might require many blocks and many expensive asymmetric cryptographic operations.
  - Security of the underlying scheme.

Hashed digital signature

Digital Signatures with Hashing



Hashed digital signature

Adding the Hashing stage

- The use of the hash does not disrupt the authenticity and integrity guarantees.
- **Exercise**: you should be able to explain why, write down arguments to shown this.

Outline

RSA Signatures

RSA signatures: the idea

1. The sender generates a public-private RSA key pair: (e, n), d, under the prime pair p, q with $pq = n$.
   - Message to be signed is (coded as) an integer m.
   - Message is hashed with suitable hash function h, with $1 < h(m) < n$.
2. The sender signs by '**encrypting**' with the private key d:
   - Signature value: $S = h^d(m) \bmod n$
3. The verifier can use the public verification key to compute:

$$S^e \bmod n$$

   this is '**decryption**' of S with the public key (e, n)
   - verifier can then checks whether the equation SIG below holds:

$$h(m) = S^e \bmod n \qquad (SIG)$$

RSA Signatures

RSA: verification of a true signature

- Verification of the true signature involves the calculation:

  $$S^e \bmod n = h^{de}(m) \bmod n = h^{ed}(m) \bmod n = h(m) \bmod n$$

- The equality holds because we showed that:

  $$M^{ed} = M \bmod n$$

  for an arbitrary M, when we showed that RSA decryption works.

Digital Signatures

Attack outcomes on signature schemes

The goal of an adversary in this context is to **forge** signatures; that is, produce signatures which will be accepted as those of some other entity.

- existential forgery. An adversary is able to forge a signature for <u>at least one</u> message.
  - The adversary may have little-or-no control over the message whose signature is obtained, and
  - the legitimate signer may be involved in the deception.
- selective forgery. An adversary is able to create a valid signature for a <u>particular</u> message (or class of messages) chosen *a priori (up front)*.
  - Creating the signature does not directly involve the legitimate singer.
- total break. An adversary is either able to compute the private key information of the signer, or finds an efficient signing algorithm functionally equivalent to the valid signing algorithm.

Digital Signatures

Assumptions: methods of attack

Two basic attacks against public-key digital signature schemes:

1. **Key-only attacks**: an adversary knows only the signer's public key.
2. **Message attacks**: an adversary is able to examine signatures corresponding either to known or chosen messages.
   a) Known-message attack: adversary, E, has signatures for messages not chosen by E.
   b) Chosen-message attack: E has signatures for messages chosen by E; all messages are chosen before any sigs are seen.
   c) Adaptive chosen message attack: E can use signer to produce sigs for messages chosen by E, using sigs already produced.

RSA Signatures

RSA: basic selective forgery defeated

- The attacker, Eve, has m and wishes to create a signature S for it.
- The attacker can calculate hash value $H = h(m)$
- S will be verified as correct if $S^e = H \mod n$, where e is the verification key.
- It suffices for Eve to calculate the e^{th} root of H in modulus n.
- This is an instance of the *RSA problem*.
- The (weak) **RSA assumption** is that this is not possible.

RSA Signatures

RSA: basic existential forgery defeated

- Attacker, Eve, has a signature value s, and wishes to find a document m, s.t. s is its signature.
- The attacker can calculate s^e
- Eve wants m s.t. the equation (SIG) holds, i.e.,

$$h(m) = s^e \bmod n.$$

- The use of the hash function makes it hard to find such an m.
- Important implementation details needed to avoid several vulnerabilities have been omitted.

RSA Signatures

Check RSA sig. has required properties

1. **authentic**.
   - the signature value must have come from the genuine signer.
2. **unforgeable**.
   - only the signer has the private signature key.
3. **not reusable**.
   - the signature is highly unlikely to match another given document.
4. **unalterable**.
   - if the document is altered, the signature is highly unlikely to match.
5. **cannot be repudiated**.
   - again, only the signer has the private key.

Outline

ElGamal: overview

Overview

1. asymmetric key encryption algorithm for public-key cryptography, which is based on Diffie-Hellman key exchange
2. described by Taher ElGamal in 1984

The algorithm

1. key generation
2. Signature and verification
3. encryption
4. decryption

ElGamal: key generation

Initial generation of keys

Alice wants to send a message to Bob.

1. Bob choose a prime number p
2. Bob choose two random numbers:
   - generator g and an integer x, both less than p
3. Bob compute $y = g^x \bmod p$

p, g and y are PUBLIC as verification key, x is PRIVATE as signature key.

ElGamal: signature

Signature

1. Consider a given message coded by integer m $(0 \leq m < p)$ Bob wants to sign, he:
   1.1 chooses a SECRET random number k $(1 < k < p - 1)$ relatively prime to $p - 1$
   1.2 computes $s_1 = g^k \bmod p$
   1.3 computes s_2, such that $m = (xs_1 + ks_2) \bmod (p - 1)$
2. Signature $S = (s_1, s_2)$
3. Note that:
   - s_2 depends on x which is PRIVATE
   - s_1 depends on k which is SECRET
4. Signature verification: Alice verifies
   $y^{s_1} {s_1}^{s_2} \bmod p = g^m \bmod p$

ElGamal: signature

Why it works?

$$
\begin{aligned}
& y^{s_1} \cdot s_1^{s_2} \mod p \\
= \;& g^{xs_1} \cdot g^{ks_2} \mod p \quad (\text{since: } y = g^x \bmod p, s_1 = g^k \bmod p) \\
= \;& g^{xs_1 + ks_2} \mod p \\
= \;& g^{(p-1)j+m} \mod p \quad (\text{for some } j, \text{ since:} xs_1 + ks_2 = m \bmod p-1) \\
= \;& g^m \cdot (g^{(p-1)})^j \mod p \\
= \;& g^m \cdot 1^j \bmod p \qquad (\text{since: } g^{p-1} = 1 \bmod p) \\
= \;& g^m \mod p
\end{aligned}
$$

ElGamal

Example: Consider the message $m = 5$ Bob wants to sign

1. consider $p = 11$ (PUBLIC), $g = 2$ (PUBLIC)
2. choose private key $x = 8$ (PRIVATE)
3. calculate $y = g^x \bmod p = 2^8 \bmod 11 = 3$ (PUBLIC)
4. choose a random $k = 9$ (SECRET) such that $gcd(k, p-1) = gcd(9, 10) = 1$
5. compute $s_1 = g^k \bmod p = 2^9 \bmod 11 = 6$

ElGamal

Example: Consider the message $m = 5$ Bob wants to sign

6. compute s_2 such that $m = (xs_1 + ks_2) \bmod p - 1$ that is $m = (8 \times 6 + 9 \times s_2) \bmod 10 \to s_2 = 3$
7. generate signature $(s_1, s_2) = (6, 3)$
8. Alice verify the signature:

$$\begin{aligned} y^{s_1} \times {s_1}^{s_2} \bmod p &= 3^6 \times 6^3 \bmod 11 \\ &= 729 \times 216 \bmod 11 = 10 \\ g^M \bmod p &= 2^5 \bmod 11 = 10 \end{aligned}$$

ElGamal Signature

Forging Elgamal signatures

- Forgery: given (p, g, y), find any message m and signature (s_1, s_2) with $y^{s_1} s_1^{s_2} = g^m \mod p$
- Try direct attack to find the private key, we have y and must solve for x in:

$$y = g^x \mod p$$

- Technicalities - there are many:
  - There are attacks when ephemeral keys, k, are re-used or non-random
  - In the real Elgamal scheme, we sign the hash and not the message. This is because existential forgery is possible otherwise.
  - A further technicality is that in verification one should also only accept (s_1, s_2) with $1 < s_1 < p - 1$.

Outline

Final Thoughts

Going around the crypto: what if?

- I trick you into signing a document.
- I break into your machine and steal/use your signing keys.
  - SSH often uses signatures for re-authentication to remote machines.
- Your server uses signature for challenge-response, but your certificate for the verification key doesn't specify that your signing key can only be used for that purpose (e.g. not for signing documents, executables. ...)
  - `https://en.wikipedia.org/wiki/Stuxnet`

Outline

Summary

This lecture

- Digital signature scheme: RSA, ElGamal

Next lecture

- Zero knowledge proof