CSIT 495/595 - Introduction to Cryptography
Digital Signatures

Bharath K. Samanthula
Department of Computer Science
Montclair State University

Course Performance Statistics

- Average and Highest scores so far (out of 38%)

	Quiz 1 (4%)	Quiz 2 (4%)	Assign 1 (15%)	Midterm (15%)
Average	2.64	2.08	10.36	10.80
Highest	4	3.75	15	14.6

- What is remaining? (62%)
  - Active Class Participation Credit (2%)
  - Assignment 2 (15%) $\rightarrow$ *Due on December 1*
  - Project Demo (20%) $\rightarrow$ *TBD*
  - Final Exam (25%) $\rightarrow$ *Scheduled on December 8*

Public-Key Encryption: Recap

Def: a public-key encryption system is a triple of algs. (G, E, D)

- G(): randomized alg. outputs a key pair (pk, sk)
- E(pk, m): randomized alg. that takes $m \in M$ and outputs $c \in C$
- D(sk,c): det. alg. that takes $c \in C$ and outputs $m \in M$ or $\perp$

Consistency: $\forall$ (pk, sk) output by G :

$$\forall m \in M: \quad D(sk, E(pk, m)) = m$$

Private-Key vs. Public-Key: Recap

	Private-Key Setting	Public-Key Setting
Secrecy	Private-key encryption	Public-key encryption
Integrity	MACs	Digital Signatures

- In public-key setting, Integrity/authenticity is achieved using digital signatures
- Digital Signatures can be considered as the public-key analogue to MACs (with some differences)

Digital Signatures: Basic Idea

- Consider a user Alice (often called *signer* in this context) owns a public-private key pair
- Suppose Alice wants to send a message m to Bob
- Assume that Bob knows the public key of Alice (this can be achieved using techniques discussed earlier)
- Key Goal of Signature Schemes:
  1. Alice signs the message and sends it to Bob
  2. Bob should verify that m was originated from Alice and was not modified in transit
- In general, anyone with Alice's public key should be able to verify m
- Observation: In the context of digital signatures, the owner of the public-key is the sender (note that this is not the case in a typical public-key encryption)

Digital Signature: Real-World Example

- Consider a company who wants to release software update/patches to its clients in an authenticated manner (what does this mean?)
- The company can generate a public-private key pair pk, sk and distribute pk to all its clients (Example: pk can be incorporated with the original software purchased)
- When releasing a software update m:
  1. The company can compute a digital signature S on m using sk
  2. Company sends (m, S) to all its clients
  3. Each client can verify whether S is a valid signature on m using pk

MACs vs. Digital Signatures (DS in short)

- MAC: sender need to compute a separate MAC for each receiver
- DS: compute only a single signature for each message sent to multiple receivers

- MAC: only the intended receiver with the corresponding secret key can validate the message
- DS: Publicly Verifiable (implies that signatures are transferable)

MACs vs. Digital Signatures (DS)

- MAC: cannot provide non-repudiation
- DS: provide non-repudiation

- Key Observation 1: MACs are shorter and roughly 2-3 orders of magnitude faster than digital signatures
- Key Observation 2: If the sender is sending a message only to a single party and if public verifiability, transferability, and/or non-repudiation is required, **MAC should be used**

Digital Signatures: Definition

Consists of three algorithms

- Key Generation: takes a security parameter (usually key length) and outputs public key *pk* and private key *sk*
- Signing: takes a message *m* and private key *sk* and generates a signature *S*
- Verification: takes public key *pk*, message *m* and a signature *S* as inputs and outputs 1 if valid, and 0 otherwise

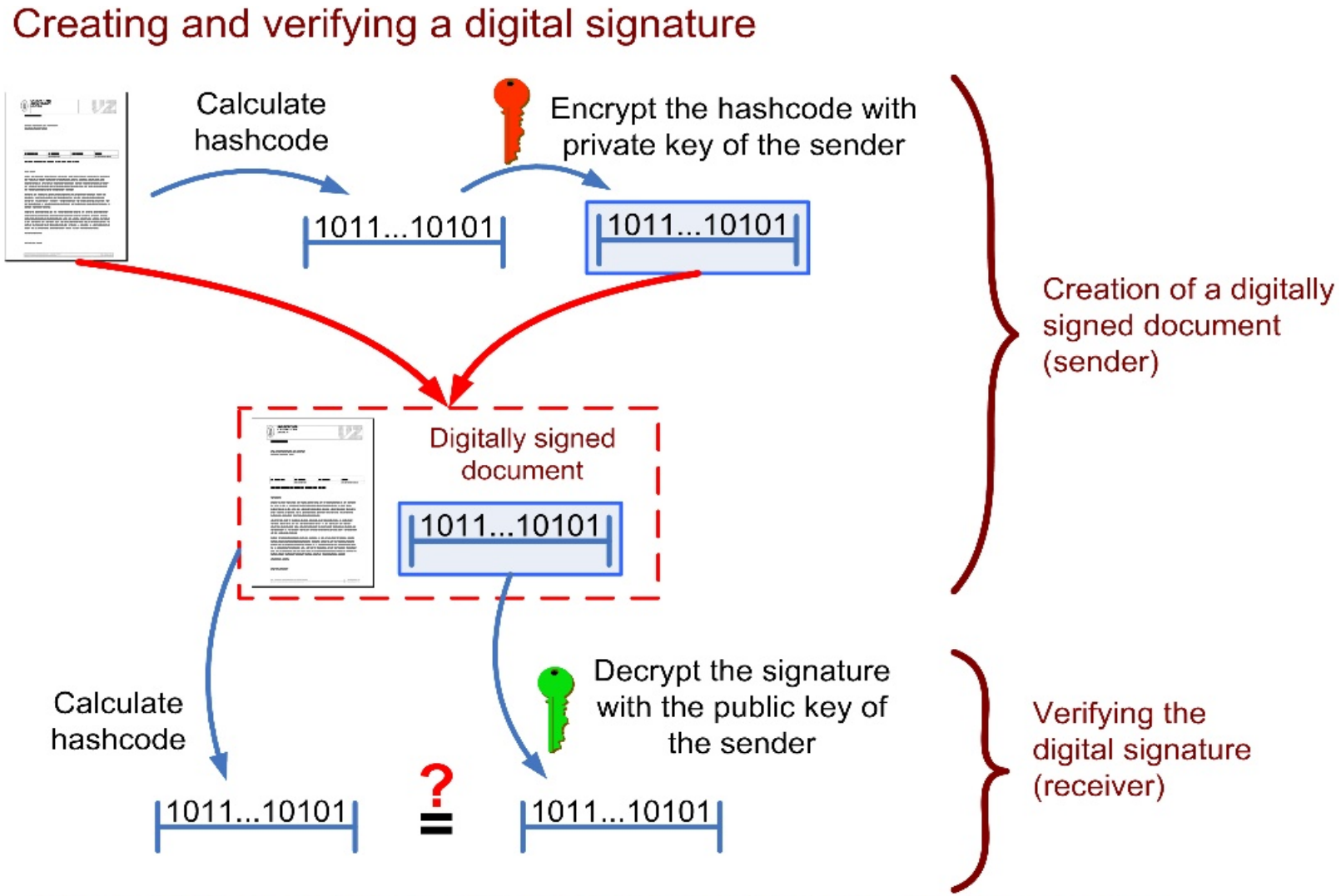
Security: the adversary should not be able to generate a forgery with a probability better than a negligible function

Hash-and-Sign Approach (1)

Message is first hashed and then signed due to the following reasons:

- It is not every efficient to sign longer messages (say a 1 GB file)
- To save bandwidth

Hash-and-Sign Approach (2)



RSA-Based Digital Signatures (1)

Key generation (as in RSA encryption):

- Select 2 large prime numbers of about the same size, p and q
- Compute $n = pq$, and $\Phi = (q - 1)(p - 1)$
- Select a random integer e, $1 < e < \Phi$, s.t. $\gcd(e, \Phi) = 1$
- Compute d, $1 < d < \Phi$ s.t. $ed \equiv 1 \bmod \Phi$

Public key: (e, n)

Secret key: d, p and q must also remain secret

RSA-Based Digital Signatures (2)

Signing message M

- M must verify $0 < M < n$
- Use private key (d)
- compute $S = M^d \bmod n$

Verifying signature S

- Use public key (e, n)
- Compute $S^e \bmod n = (M^d \bmod n)^e \bmod n = M$

Note: in practice, a hash of the message is signed and not the message itself.

RSA-Based Digital Signatures: Security

Example of forging

- Attack based on the multiplicative property of property of RSA.

  $y_1 = sig_K(x_1)$

  $y_2 = sig_K(x_2)$, then

  $ver_K(x_1x_2 \bmod n, y_1y_2 \bmod n) = true$

- So adversary can create the valid signature

  $y_1y_2 \bmod n$ on the message $x_1x_2 \bmod n$
- This is an existential forgery using a known message attack.

Methods based on DLP: ElGamal and Schnorr signature schemes

Digital Certificates

- Important application of digital signatures
- Use public-key cryptography to securely distribute public keys
- A single public-key of a trusted-party, called *certificate authority*, is first distributed in a secure fashion. Then this key can be used to distribute many other public keys
- Digital certificate: signature binding an entity to some public key

Digital Certificates: Basic Idea

- Consider two users Charlie and Alice with (pk_C, sk_C) and (pk_A, sk_A)
- Assume that Charlie knows Alice's public key pk_A
- Charlie Computes:

$$Cert_{C \to A} = Sign_{sk_C}(Alice's\ public\ key\ is\ pk_A)$$

We call $Cert_{C \to A}$ a certificate for Alice's pubic key pk_A issued by Charlie (trusted or certificate authority)

- In general, other details are used in the certificate generation, such as Alice's
  - full name
  - email address
  - URL of personal website

Digital Certificates: Basic Idea

- Suppose Bob wants to communicate with Alice
- Assume Bob knows pk_C
- Alice sends $(pk_A, Cert_{C \to A})$ to Bob
- Bob verifies whether $Cert_{C \to A}$ is a valid signature on the message *Alice's public key is* pk_A with respect to pk_C
- If the verification succeeds, Alice and Bob can communicate over an insecure and authenticated channel

Digital Certificates: Key Factors

- To what extent Bob can trust Charlie? (usually only well-established companies play the role of CA)
- How Bob learns the public key of CA?
  - many operating systems/web browsers typically come pre-configured with many CA's public keys
- Single or Multiple CAs
- Expiration of certificates (include expiration date as part of the message before generating the signature)

Useful References

- Chapter 12, Introduction to Modern Cryptography by Jonathan Katz and Yehuda Lindell, 2nd Edition, CRC Press, 2015.
- https://msdn.microsoft.com/en-us/library/hk8wx38z(v=vs.110).aspx
- https://www.entrust.com/wp-content/uploads/2013/05/cryptointro.pdf
- http://www.cgi.com/files/white-papers/cgi_whpr_35_pki_e.pdf