CISC 468: CRYPTOGRAPHY

LESSON 13: THE RSA SIGNATURE SCHEME

Furkan Alaca

READINGS

- Section 10.1: Introduction (Digital Signatures), Paar & Pelzl
- Section 10.2: The RSA Signature Scheme, Paar & Pelzl

INTRODUCTION

- Digital signatures are one of the most important and widely-used cryptographic tools
  - Digital signatures use public-key cryptography
- Their objective is similar to that of handwritten signatures: to authenticate the originator of a message
- The many applications include:
  - Digital certificates for verifying the authenticity of public keys
  - Secure software updates
  - Secure boot

TODAY WE WILL LEARN...

- The principle of digital signatures
- Security objectives that can be achieved by digital signatures
- The RSA signature scheme

SECURITY SERVICES

The cryptographic schemes we have encountered so far have provided one of two *security services*:

1. *Confidentiality*, e.g., via symmetric-key algorithms (e.g., stream ciphers and block ciphers) or public-key algorithms (e.g., RSA, Elgamal)
2. Key establishment (Diffie-Hellman Key Exchange)

But there are security needs beyond these two!

REPUDIATION

- Suppose Alice and Bob share a secret key
- Bob creates an AES-encrypted contract to purchase a car from Alice
- When the car is delivered, Bob changes his mind
  - He claims that Alice (not he) created the contract
  - i.e., Bob *repudiated* the contract
- Problem: Alice and Bob share the same key, so a neutral third-party cannot verify which of the two created the contract
  - We can solve this with asymmetric-key cryptography, since each party has their own unique private key

MORE SECURITY SERVICES

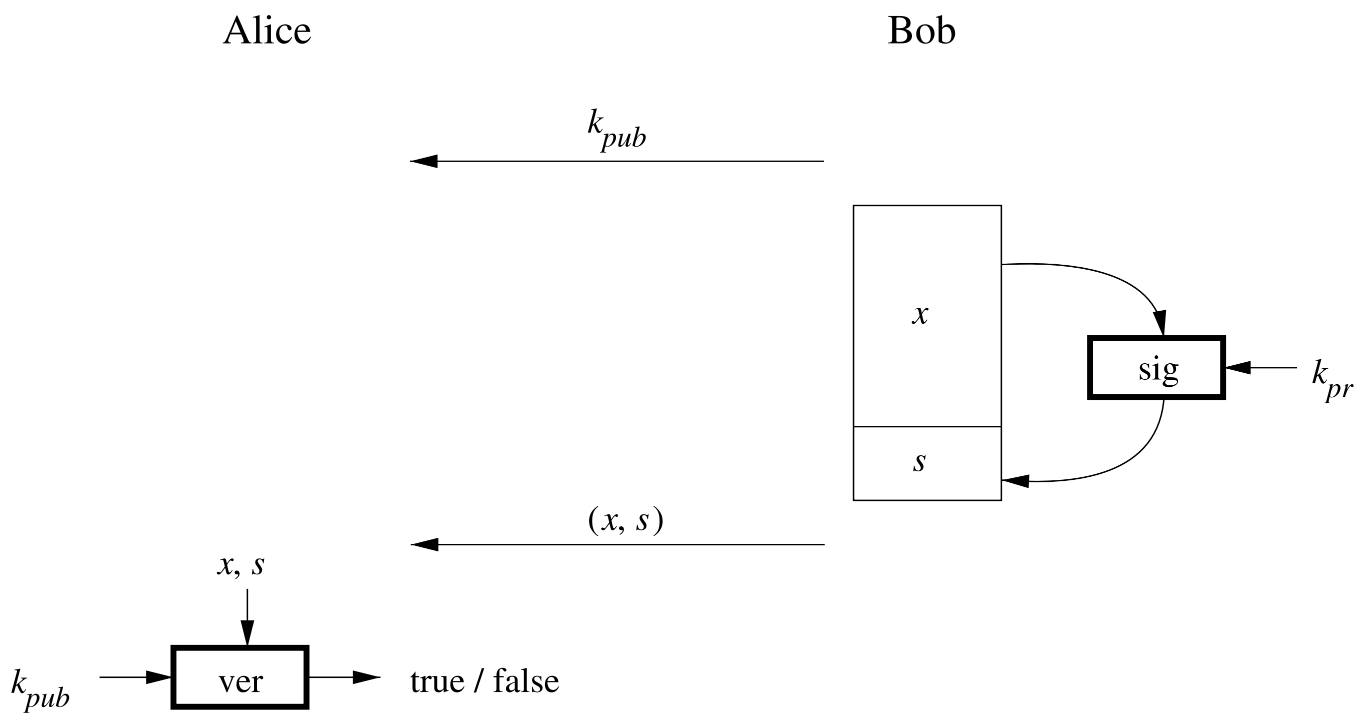
3. Message Integrity: Assure that messages have not been modified in transit.
4. Message Authentication: Assure that the sender of a message is authentic.
5. Nonrepudiation: Assure that the sender of a message cannot credibly deny the creation of the message.

PRINCIPLES OF DIGITAL SIGNATURES

- Only the person who creates a message should capable of generating a valid signature
  - So, the sender's private key is used for signing
- Any person who receives a message must be capable of verifying the validity of the signature
  - So, the sender's public key is used for verifying
- Note that the role of the keys is swapped compared to public-key encryption and decryption

DIGITAL SIGNATURES: SIGNING AND VERIFYING

- Bob creates a message x and then uses his private key k_{pr} to generate a digital signature s
- Alice verifies the signature s using Bob's public key k_{pub}



DIGITAL SIGNATURES: SIGNING AND VERIFYING (2)

- After signing a message x, it must be sent together with the signature s to Alice
  - Thus, message confidentiality is **not** provided
  - A signature s without an accompanying message is useless
- If an active attacker modifies the message x in transit, the signature s will be invalid for the modified message x'
  - Thus, integrity is provided
- Assuming Bob keeps his private key secret, only he can sign a message x on his behalf
  - Thus, nonrepudiation is provided

SCHOOLBOOK RSA DIGITAL SIGNATURE

- The RSA signature scheme is based on RSA encryption
  - Key generation step is identical
- The sender signs a message x by calling the RSA encrypt function using their own private exponent d
- The receiver verifies the signature s by calling the RSA decrypt function using the sender's public exponent e and modulus n
- To prove that RSA decryption works, we already showed that $(x^e)^d \equiv x \bmod n$
  - To prove that RSA signature verification works we show that $(x^d)^e \equiv x \bmod n$, i.e., the proof is essentially the same

SCHOOLBOOK RSA DIGITAL SIGNATURE: EXAMPLE

Bob sends a signed message $x = 4$ to Alice:

Alice		**Bob**
		1. choose $p = 3$ and $q = 11$ 2. $n = p \cdot q = 33$ 3. $\Phi(n) = (3-1)(11-1) = 20$ 4. choose $e = 3$ 5. $d \equiv e^{-1} \equiv 7 \bmod 20$
	$\xleftarrow{(n,e)=(33,3)}$	
		compute signature for message $x = 4$: $s = x^d \equiv 4^7 \equiv 16 \bmod 33$
	$\xleftarrow{(x,s)=(4,16)}$	
verify: $x' = s^e \equiv 16^3 \equiv 4 \bmod 33$ $x' \equiv x \bmod 33 \Longrightarrow$ valid signature		

Alice concludes that Bob generated the message $x = 4$ and that it was not altered in transit

RSA DIGITAL SIGNATURES: COMPUTATIONAL ASPECTS

- The signature s is as long as the modulus n
  - i.e., at least 2048 bits
- Section 7.5 discusses speed-up techniques for RSA encryption that are also applicable for digital signatures
  - Of particular interest is the ability to use short public exponents, which makes signature verification much faster than signature generation

RSA DIGITAL SIGNATURES: SECURITY

- The verifying party must be be assured that it is using the sender's authentic public key for verification
  - We will need digital certificates for this
- Just as with RSA encryption, modulus should be large enough to be secure against factoring (i.e., at least 2048-bit)
- *Existential forgery* is an attack against Schoolbook RSA Signatures that allows an attacker to generate a valid signature for a random message x

RSA DIGITAL SIGNATURES: EXISTENTIAL FORGERY ATTACK

- Oscar chooses a signature $s \in \mathbb{Z}_n$, then computes a matching message $x \equiv s^e \bmod n$ (which will likely be gibberish)

Alice		**Oscar**		**Bob**
				$k_{pr} = d$ $k_{pub} = (n, e)$
	$\xleftarrow{(n,e)}$		$\xleftarrow{(n,e)}$	
		1. choose signature: $s \in \mathbb{Z}_n$ 2. compute message: $x \equiv s^e \bmod n$		
	$\xleftarrow{(x,s)}$			
verification: $s^e \equiv x' \bmod n$ since $x' = x$ $\Longrightarrow$ valid signature!				

RSA DIGITAL SIGNATURES: PADDING

- Existential forgery attacks can be prevented by imposing rules on the message format
- A simple rule could require that all messages x have 100 trailing 0 bits
  - Then, if Oscar chooses a signature s and computes a matching message $x \equiv s^e \bmod n$, the probability that it will match the required message format is 2^{-100} (nearly zero)

HASH FUNCTIONS

- A *hash function* takes an aribtrary-length input and generates a fixed-size output, e.g., 256 bits
  - Output is called a *message digest*, i.e., a compact representation of the message
- Cryptographic hash functions have some special properties that non-cryptographic hash functions do not have
  - Our next topic in the course
  - Play an important role in digital signatures and other security applications

RSA PROBABILISTIC SIGNATURE STANDARD (PSS)

- RSA-PSS is a standardized RSA signature scheme that incorporates:
  - Padding, to defend against existential forgery attacks
  - A random *salt* value, to generate a different signature if the same message is signed more than once
  - A hash function, so that the message digest is signed instead of the actual message

RSA-PSS: MESSAGE ENCODING