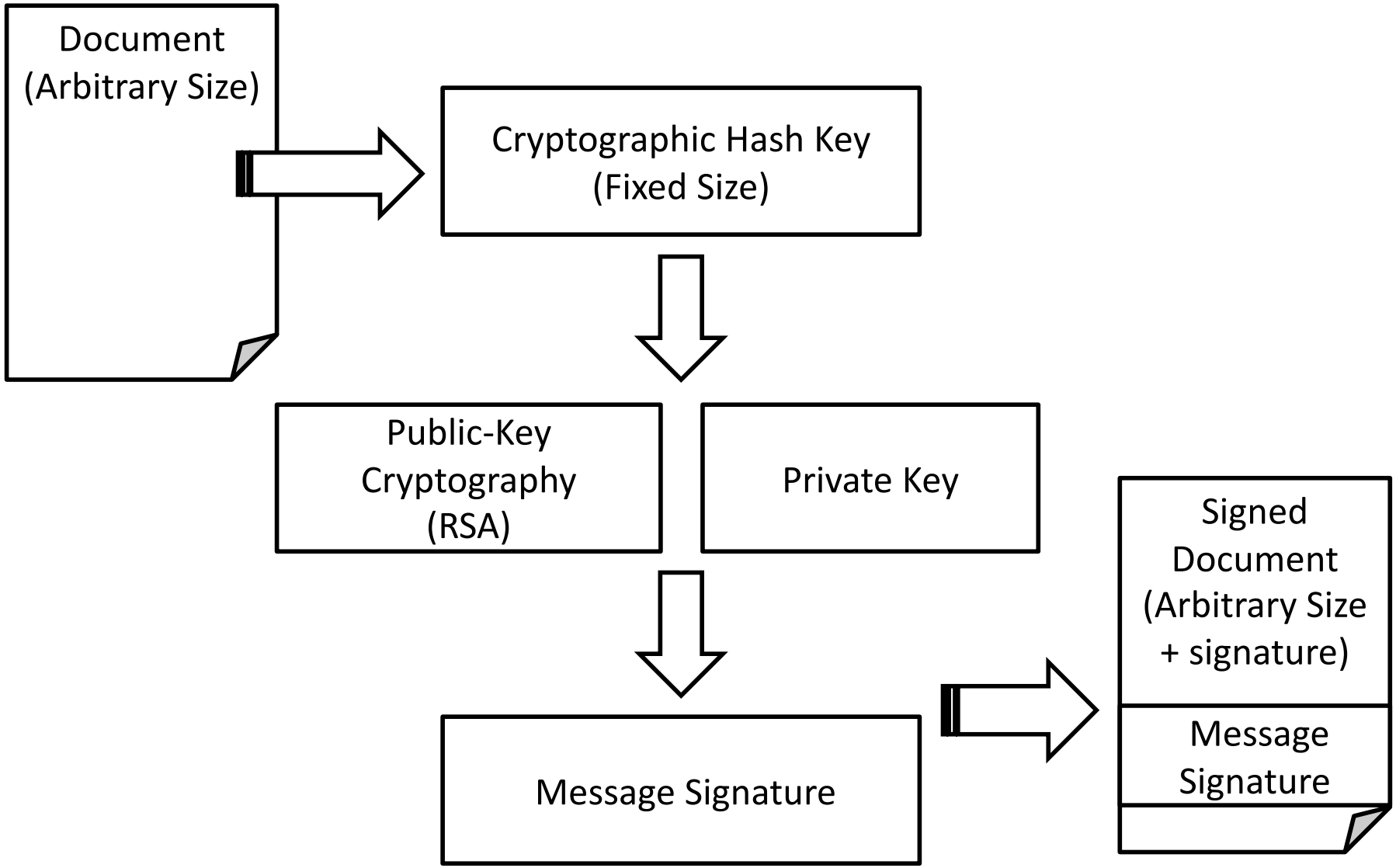
CS326 – Systems Security

Lecture 20

Message Authentication Codes (MAC)

Elias Athanasopoulos
athanasopoulos.elias@ucy.ac.cy

Recall: Digital Signing



Message Authentication Code (MAC)

- Digital Signing
  - Public-key cryptography (e.g., RSA, ElGamal)
  - Cryptographic Hash Function (e.g., SHA2)
- Perform an equivalent procedure to digital signing without using public-key cryptography
  - One shared key
  - Cryptographic Hash Function (e.g., SHA2)
  - Somehow mix the key with the hash function (*keyed hash function)*

MACs vs Digests

- Digest
  - For a given input x, a digest is m = H(x), where H() is a cryptographic hash function
- MAC
  - For a given input x, a MAC is m = H(x, k), where H() is a cryptographic hash function

How to create H(x, k)?

Notice

(1) || stands for concatenation

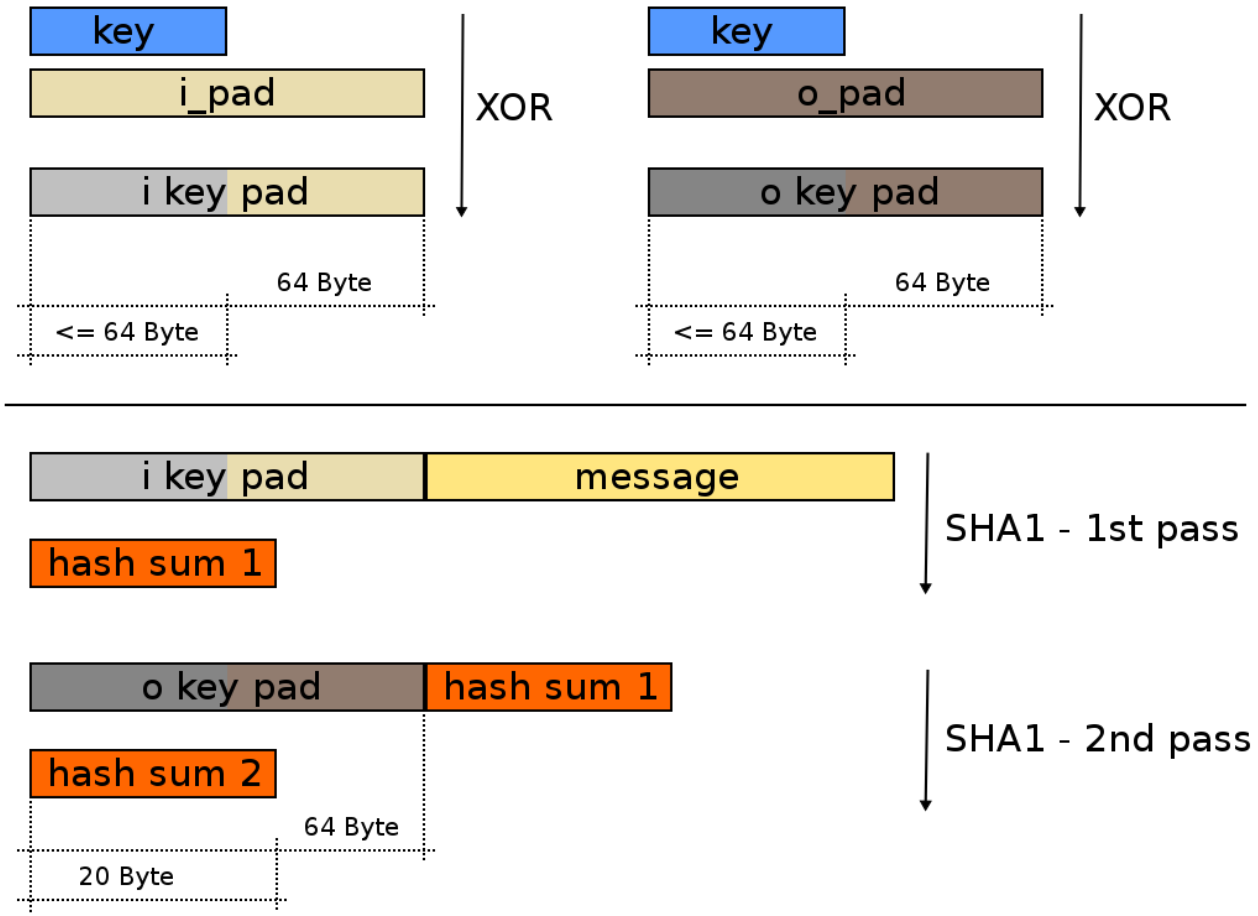
(2) attacks in this slide are *not* covered in the course

- m = H(k || x)
  - Length-extension attack
- m = H(x || k)
  - Collisions in the *unkeyed* hash function, introduces collision in the MAC
- m = H(k || x || k)
  - Better, but questionable

HMAC

- HMAC(K, m) = H((K'⊕opad) || H((K'⊕ipad) || m))
- Inputs
  - Key **K**, message **m**, and a cryptographic hash function **H()**
- Internals
  - **K'** is another secret key, derived from the original key *K* (by padding *K* to the right with extra zeroes to the input block size of the hash function, or by hashing *K* if it is longer than that block size)
  - **opad** is the outer padding (0x5c5c5c…5c5c, one-block-long hexademical constant)
  - **ipad** is the inner padding (0x363636…3636, one-block-long hexademical constant)
- Output
  - Fixed-length MAC, called: HMAC(K, m)

HMAC


key
i_pad
XOR
i key pad
64 Byte
<= 64 Byte
key
o_pad
XOR
o key pad
64 Byte
<= 64 Byte
i key pad
message
hash sum 1
SHA1 - 1st pass
o key pad
hash sum 1
hash sum 2
64 Byte
20 Byte
SHA1 - 2nd pass

MAC Properties

- Arbitrary input length
- Fixed output length
- Message Authentication
- Message Integrity
- Non-repudiation is not given

Hash, MAC, Digital Signature

	Cryptographic Hash	MAC	Digital Signature
Integrity	Yes	Yes	Yes
Authentication	No	Yes	Yes
Non-repudiation	No	No	Yes
Key	No keys	Symmetric	Asymetric

TLS Record Protocol

Byte	+0	+1	+2	+3
0	Content type			
1..4	Version		Length	
5..n	*Payload*			
n..m	MAC			
m..p	Padding (block ciphers only)			

The right record size

- Small records have larger CPU overhead due to frequent MAC verification
- Large records will have to be reassembled by the TCP layer before they can be processed by the TLS layer
- Not always possible to tune the record size