CS 4770: Cryptography

CS 6750: Cryptography and Communication Security

Alina Oprea
Associate Professor, CCIS
Northeastern University

February 15 2018

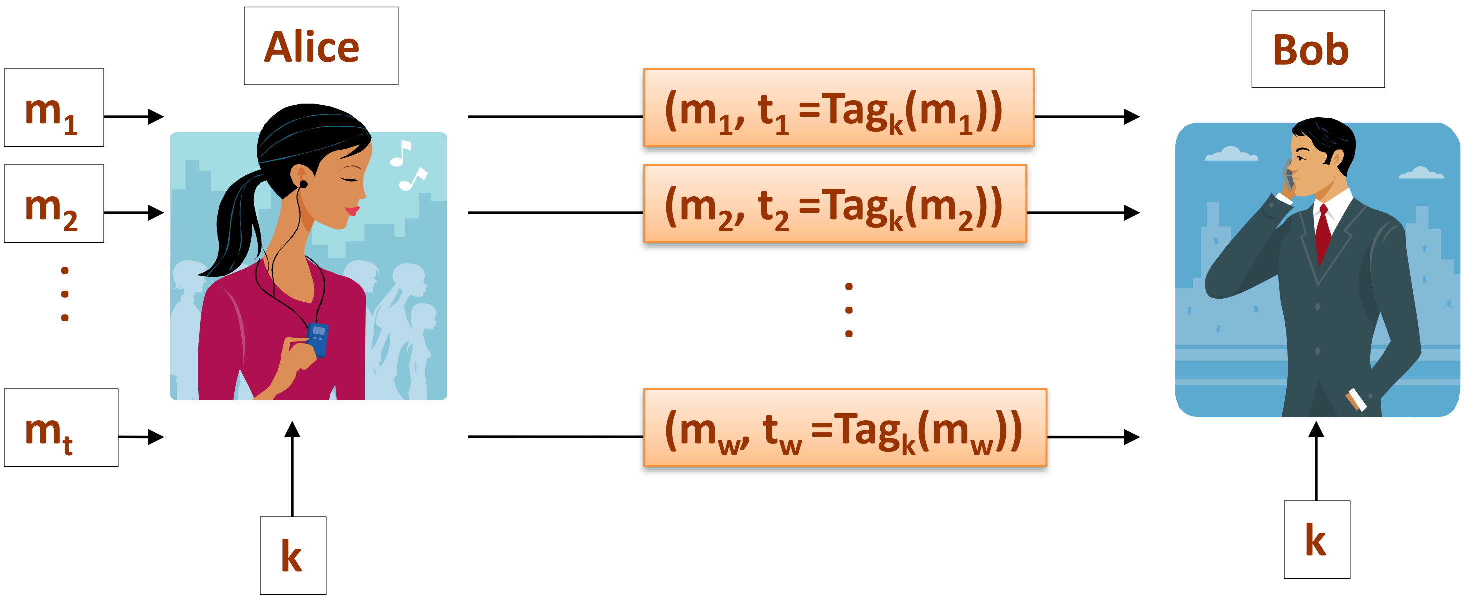
Announcements

- Schedule
  - Next week vacation on Monday (President's Day)
  - Class canceled on Thursday 02/22
  - Normal schedule on Monday 02/26
- Assignments
  - Programming project Thu 02/15 – Mon 02/26
- Midterm exam
  - Thursday 03/01
  - Topics
    - Notions of security for encryption (PS, EAV, CPA, CCA)
    - Modes of operation for encryption (CBC, CTR)
    - PRG, PRF, PRP
    - MAC for integrity
    - Authenticated encryption

Recap

- Integrity vs confidentiality
  - Complementary properties
  - Both are needed in practice
- Message Authentication Codes (MAC)
  - Secret key needed for integrity
  - Security definition
  - Encryption not sufficient for integrity
- Constructions
  - MACs on single block (e.g., 128-bit) can be built from PRFs
  - CBC-MAC for integrity on longer messages

Message Authentication Codes



Eve should not be able to compute a valid tag **t'** on any other message **m'**.

Security experiment for MAC

- Experiment $\mathrm{Exp}^{\mathrm{MAC}}_{\Pi,A}(n)$:
  1. Choose $k \leftarrow Gen(n)$
  2. *m*,*t* $\leftarrow A^{Tag()}(n)$
  3. Output 1 if Ver(*m*,*t*) = 1 and *m* was not queried to the Tag() oracle
  4. Output 0 otherwise

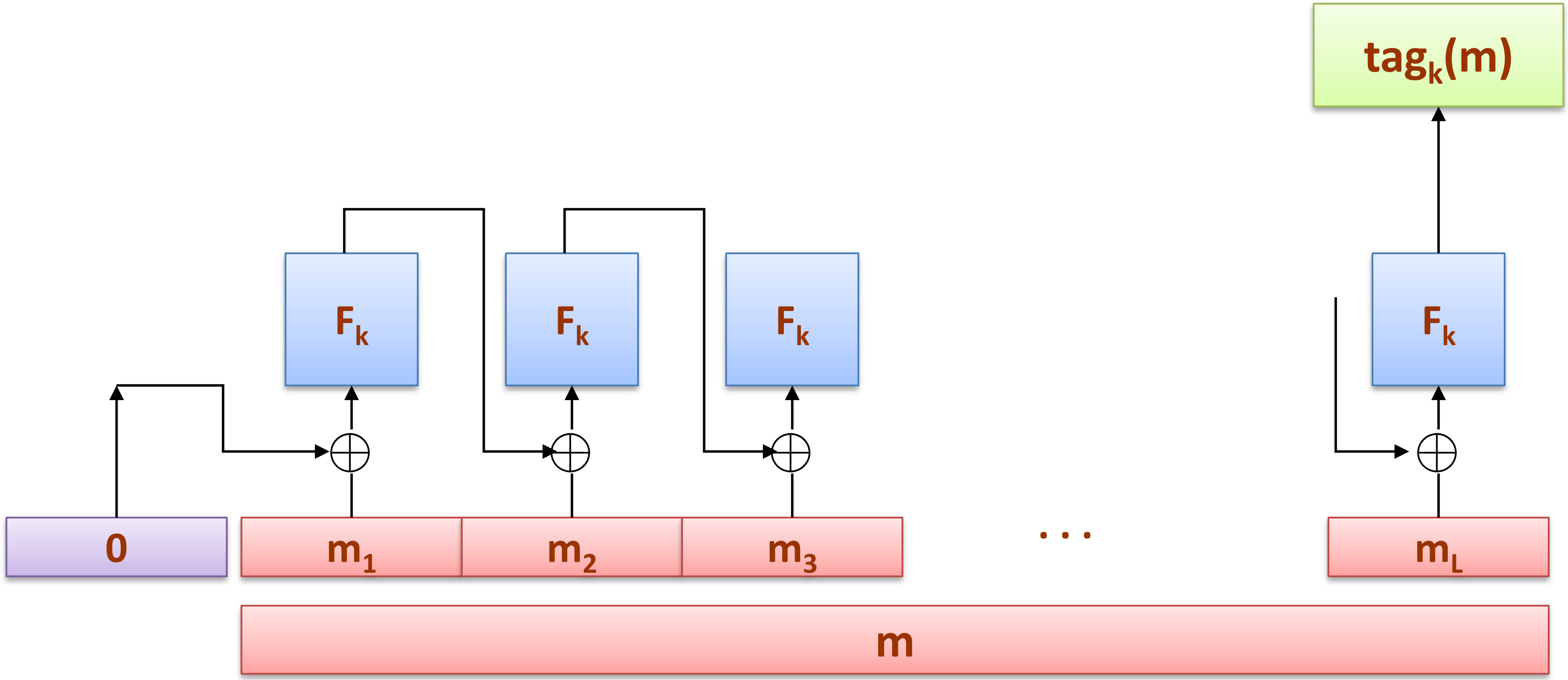
(Gen,Tag,Ver) is **a secure (existential unforgeable)** MAC if:

For every **PPT** adversary A:

Pr[$\mathrm{Exp}^{\mathrm{MAC}}_{\Pi,A}(n)$ **= 1] is negligible in n**

CBC-MAC

$F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ - a PRF



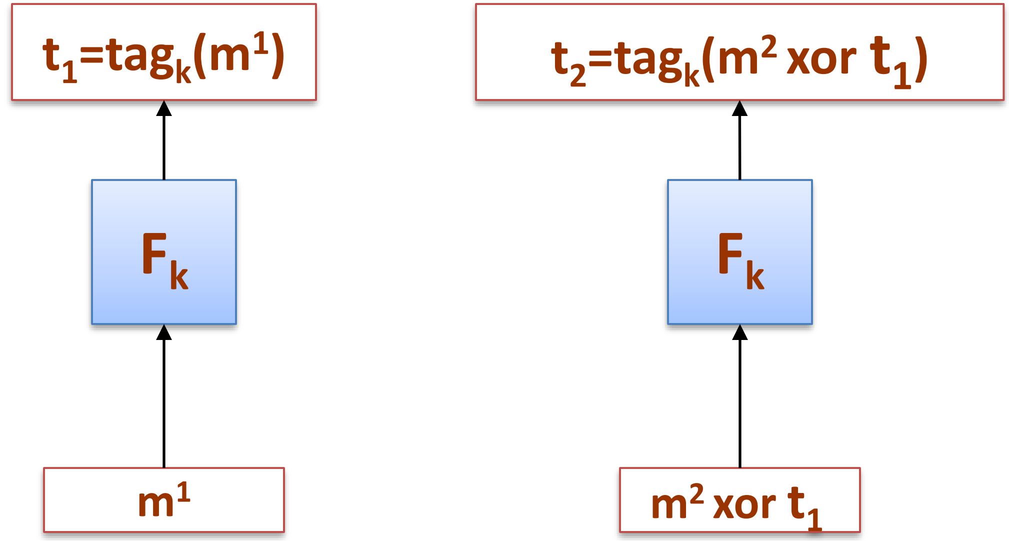
Theorem

Assuming that $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ is a **pseudorandom function** and messages of fixed length are tagged, then CBC-MAC construction is secure.

CBC-MAC vs CBC-Enc

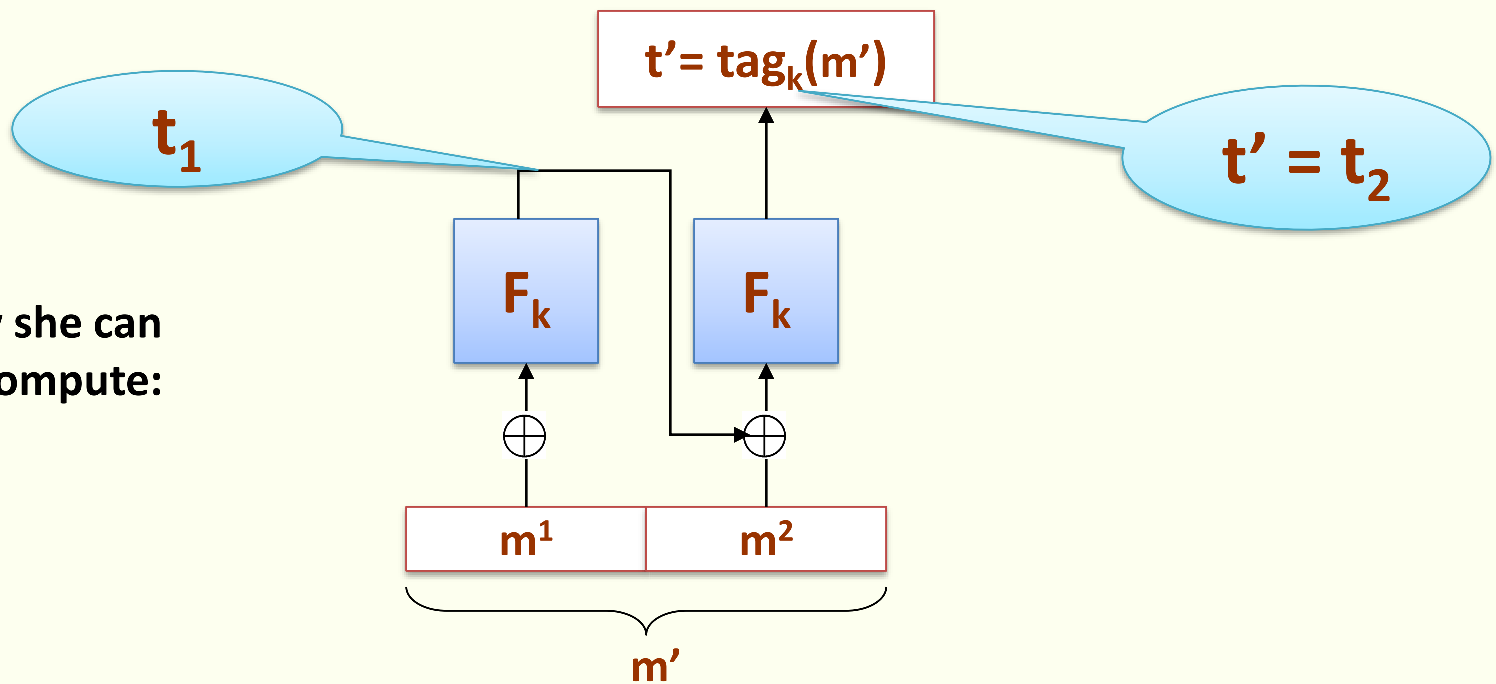
- Different security properties
  - CBC-Enc is CPA secure encryption
  - CBC-MAC is secure MAC
- Initialization
  - CBC-Enc uses random IV
  - CBC-MAC uses first block fixed at 0
  - CBC-MAC with random IV is insecure!
- Output
  - CBC-Enc outputs all intermediate blocks (to decrypt)
  - CBC-MAC outputs only last block

the adversary chooses:



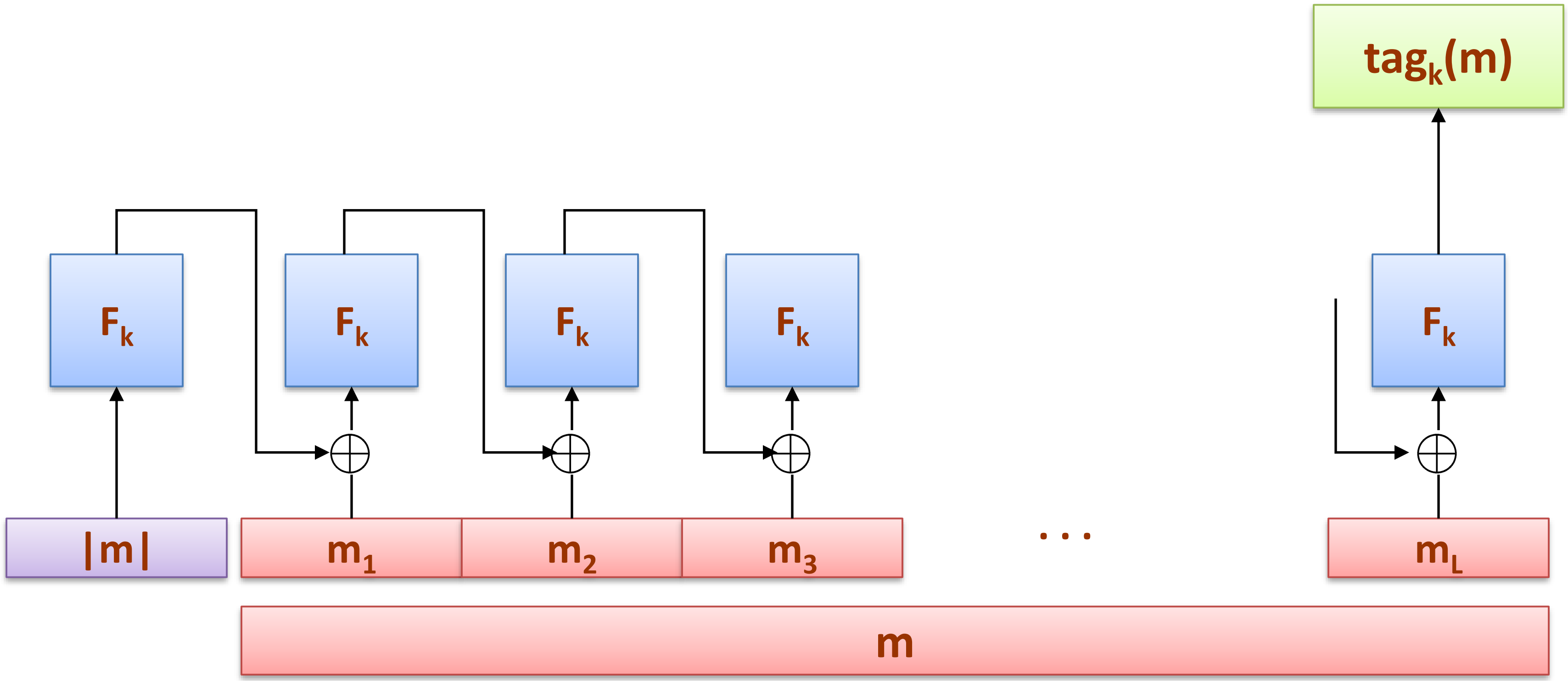
Messages of different length

now she can compute:



CBC-MAC for variable length messages

$F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ - a PRF



CBC-MAC analysis

Theorem: For any L>0,

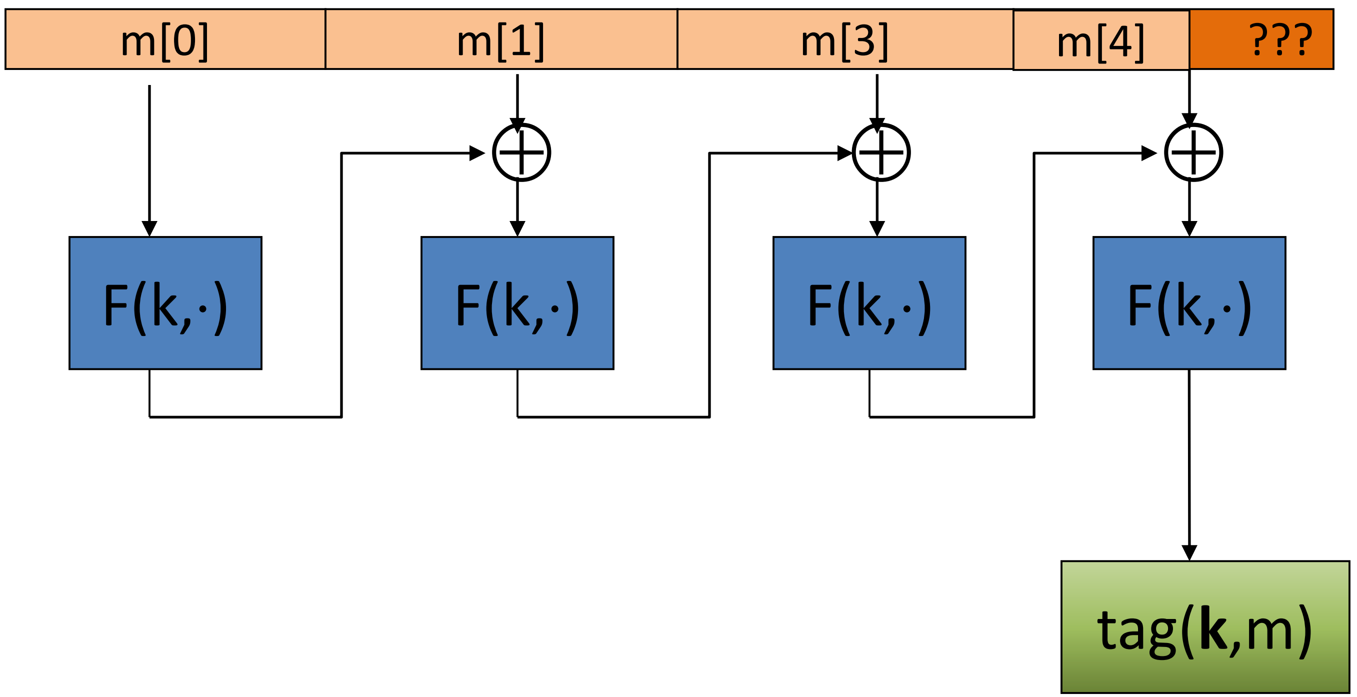
For every PPT q-query PRF adversary A attacking the CBC-MAC

there exists a PPT adversary B for F s.t.:

$$\Pr[\mathrm{Exp}^{\mathrm{MAC}}_{\mathrm{CBC\_MAC},A}(n) = 1] \leq \mathrm{Adv}^{\mathrm{PRF}}_{\mathrm{F},B}(n) + 2\,q^2 / 2^n$$

CBC-MAC is secure as long as $q \ll 1/ 2^{n/2}$

What if msg. len. is not multiple of block-size?



CBC MAC padding

Bad idea: pad m with 0's

Is the resulting MAC secure?

Yes, the MAC is secure

It depends on the underlying MAC

No, given tag on msg **m** attacker obtains tag on **m||0**

Problem: pad(m) = pad(m||0)

Collision in padding function

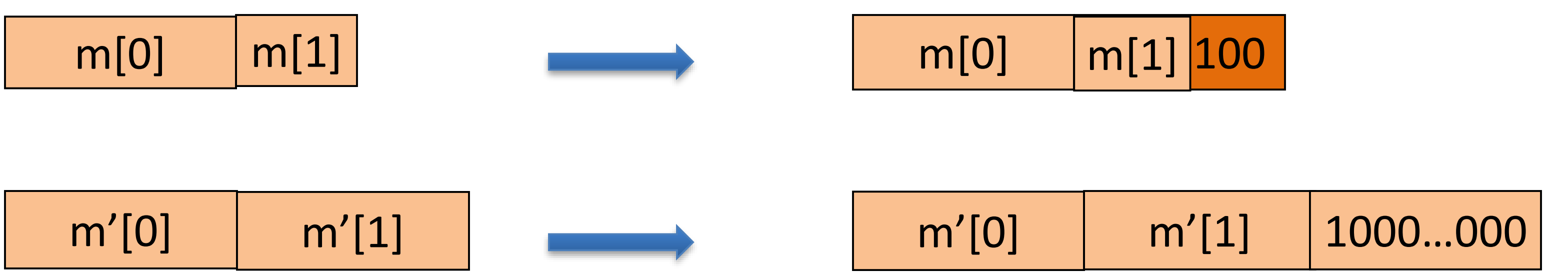
CBC MAC padding

For security, padding must be invertible !

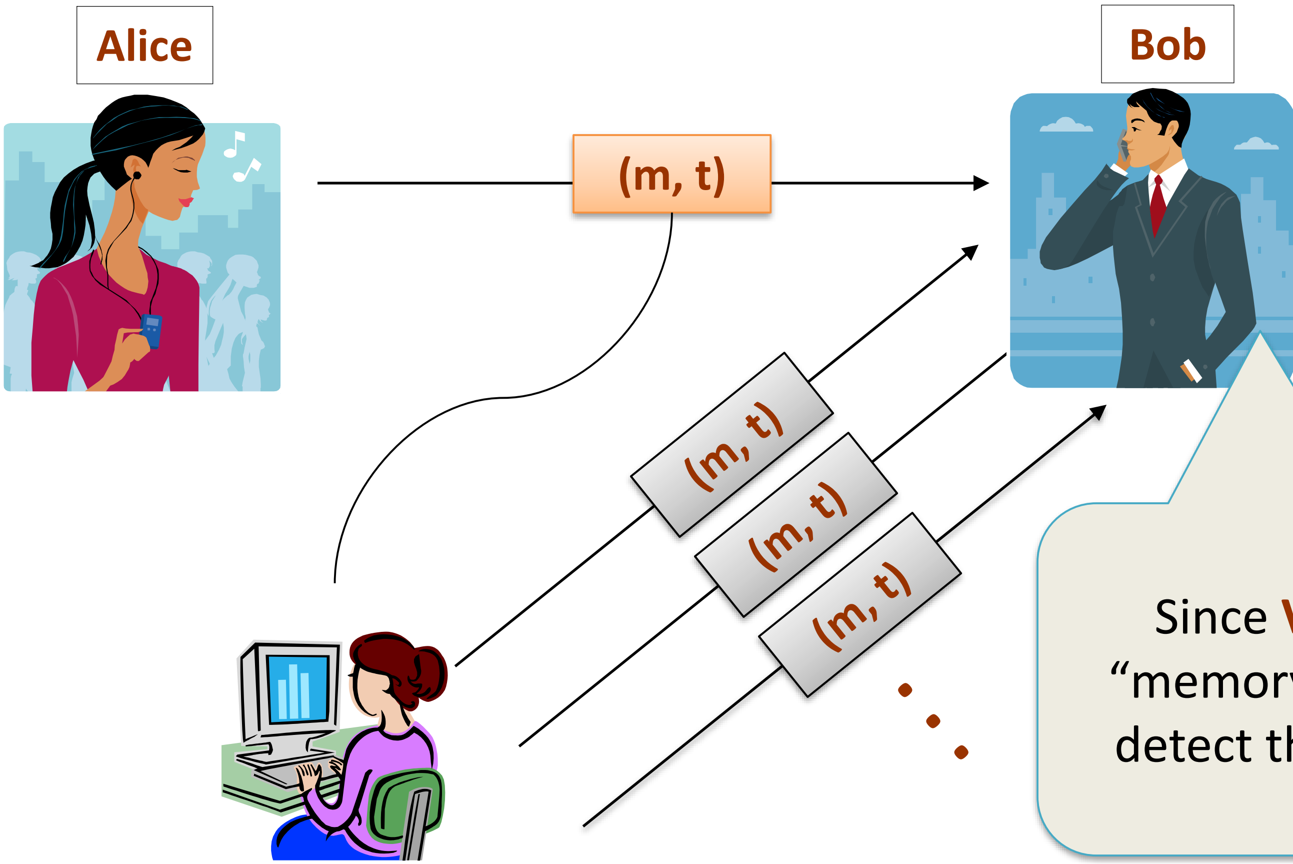
$$m_0 \neq m_1 \quad \Rightarrow \quad pad(m_0) \neq pad(m_1)$$

ISO: pad with "1000…00". Add new dummy block if needed.

- The "1" indicates beginning of pad.



Warning: MACs do not offer protection against the "replay attacks".



Since **Ver** has no state (or "memory") there is no way to detect that **(m,t)** is not fresh!

This problem has to be solved by the higher-level application (methods: **time-stamping**, **sequence numbers**...).

Authenticated encryption

- Combine confidentiality and integrity
- Security properties
  - *Confidentiality*: CCA security
  - *Integrity*: attacker cannot create new ciphertexts that decrypt properly
- Decryption returns either
  - Valid messages
  - Or invalid symbol (when ciphertext is not valid)

Some history

Authenticated Encryption (AE): introduced in 2000 [KY'00, BN'00]

Crypto APIs before then: (e.g. MS-CAPI)

- Provide API for CPA-secure encryption (e.g. CBC with rand. IV)
- Provide API for MAC (e.g. HMAC)

Every project had to combine the two itself without a well defined goal

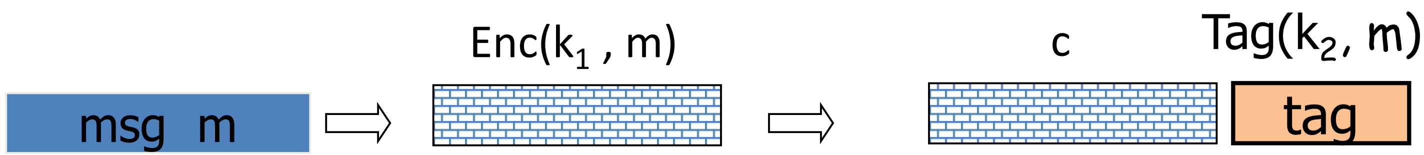
- Not all combinations provide AE …

Combining MAC and ENC (CCA)

Encryption key k_1. MAC key = k_2

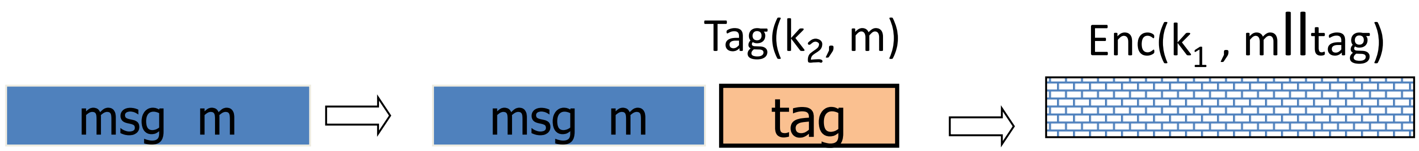
Enc-and-MAC

Option 1: (SSH)



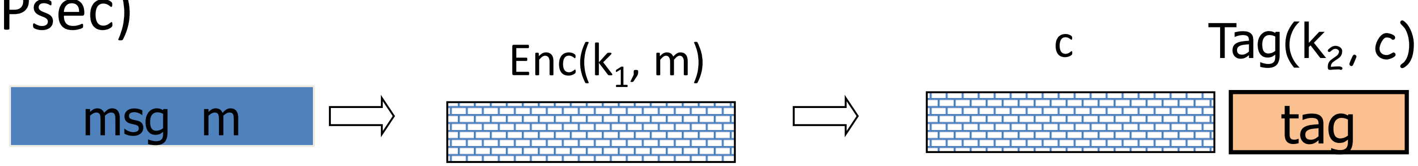
Option 2: (SSL)

MAC-then-enc



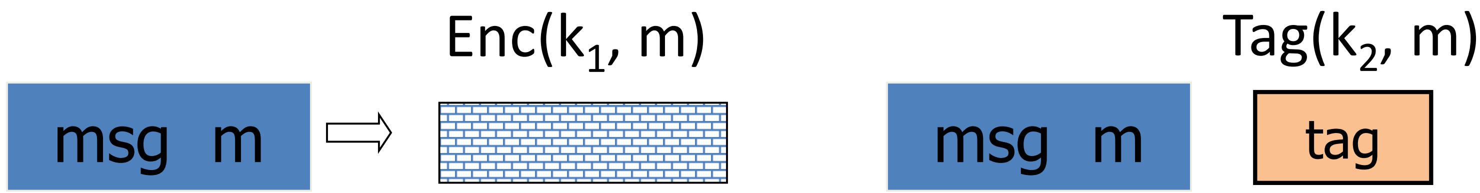
Enc-then-MAC

Option 3: (IPsec)



Encrypt-and-MAC (SSH)

Encryption key k_1. MAC key = k_2



- Tag does not protect confidentiality of message
  - Could output first message bit, for example
- If adversary gets Enc(k_1, m) and Tag(k_2, m), he can distinguish encryption of two messages in challenge phase

Insecurity of Encrypt-and-MAC

- Assume that (Tag, Ver) is a secure MAC
  - Define $Tag'_k(m) = (m[1] || Tag_k(m))$ and m[1] first bit of m. Ver' runs Ver and checks first bit of m.
  - Then (Tag',Ver') is a secure MAC
- Consider Encrypt-and-MAC scheme
  - $c = Enc_{k_1}(m)$, $t = Tag'_k(m) = m[1] || Tag_k(m)$
  - Attacker can break security of encryption
  - How?
  - Not even EAV secure!

MAC-then-Enc

Let (Enc,Dec) be CPA secure encryption and (Tag,Ver) secure MAC. Then:

MAC-then-Encrypt (SSL): is not always secure

$$t = Tag_{k_2}(m),\ c = Enc_{k_1}(m||t),$$

Properties:

- Vulnerable to padding oracle attack if CBC encryption is used
- If no padding oracle, Mac-then-Encrypt provides A.E. when when (Enc,Dec) is rand-CTR mode or rand-CBC

Encrypt-then-MAC

Let (Enc,Dec) be CPA secure encryption and (Tag,Ver) secure MAC. Then:

Encrypt-then-MAC (IPSec): always provides A.E.

$$c = Enc_{k_1}(m),\ t = Tag_{k_2}(c)$$

Intuition:

- Adv. can not modify valid ciphertext and still get a valid Tag (by unforgeability of MAC)
- All queries to Dec oracle will return valid for c returned from Enc oracle; or invalid otherwise
- Dec oracle is not useful, CCA security reduces to CPA security

A.E. Theorems

Let (Enc,Dec) be CPA secure encryption and (Tag,Ver) secure MAC. Then:

1. **Encrypt-then-MAC** (IPSec): always provides A.E.

2. **MAC-then-encrypt** (SSL): may be insecure against CCA attacks

However: when (Enc,Dec) is rand-CTR mode or rand-CBC and no padding oracle available, Mac-then-Encrypt provides A.E.

Important: Encryption and MAC keys need to be independent

Counter-example for same key

- F a secure PRP
- $Enc_k(m) = F_k(m \mid\mid r)$ for r a random number
  - CPA secure
- $MAC_k(c) = F_k^{-1}(c)$
  - F_k^{-1} is also a PRP
  - MAC is secure
  - But $MAC_k(c) = m||r$ (because same key is used)
- $Enc_k(m)$, $MAC_k(c)$ is not secure A.E.!

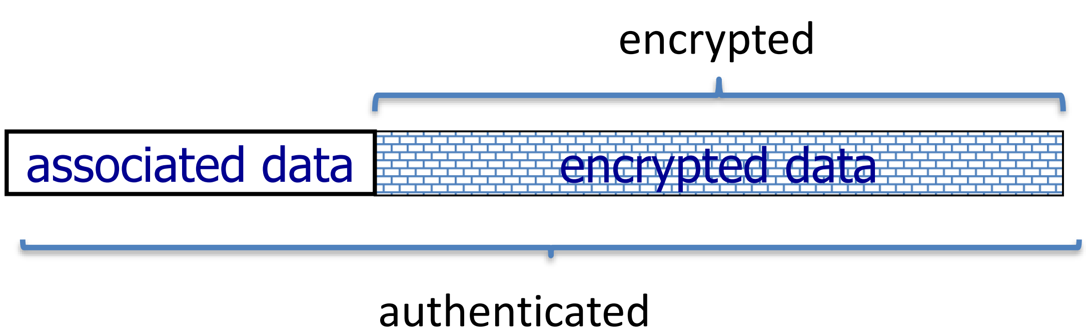
Standards (at a high level)

- **GCM**: CTR mode encryption then CW-MAC
  (accelerated via Intel's PCLMULQDQ instruction)

- **CCM**: CBC-MAC then CTR mode encryption
  (802.11i)

- **EAX**: CTR mode encryption then CMAC

All support AEAD: (authenticated encryption with associated data)
All are nonce-based



An example API (OpenSSL)

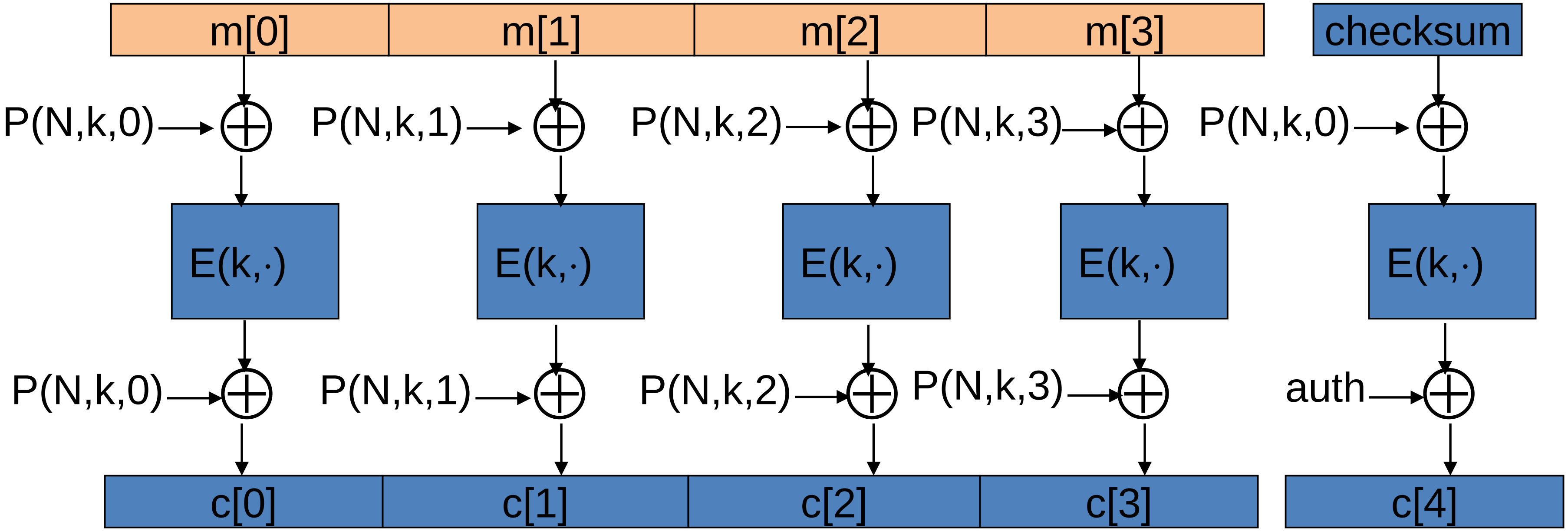
int **AES_GCM_Init**(AES_GCM_CTX *ain,
unsigned char ***nonce**, unsigned long noncelen,
unsigned char ***key**, unsigned int klen)

int **AES_GCM_EncryptUpdate**(AES_GCM_CTX *a,
unsigned char ***aad**, unsigned long aadlen,
unsigned char ***data**, unsigned long datalen,
unsigned char ***out**, unsigned long *outlen)

OCB: a direct construction from a PRP

More efficient authenticated encryption

- one Enc() operation per block
- Parallelizable



Performance: Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

NIST standards: AES/GCM, AES/CCM

Cipher	code size	Speed (MB/sec)		
AES/GCM	large	108	AES/CTR	139
AES/CCM	smaller	61	AES/CBC	109
AES/EAX	smaller	61	AES/CMAC	109
AES/OCB		129	HMAC/SHA1	147

Further reading

- The Order of Encryption and Authentication for Protecting Communications. H. Krawczyk, Crypto 2001.
- Authenticated-Encryption with Associated-Data. P. Rogaway, Proc. of CCS 2002.
- Password Interception in a SSL/TLS Channel. B. Canvel, A. Hiltgen, S. Vaudenay, M. Vuagnoux, Crypto 2003.
- Plaintext Recovery Attacks Against SSH. M. Albrecht, K. Paterson and G. Watson, IEEE S&P 2009
- Problem areas for the IP security protocols. S. Bellovin, Usenix Security 1996.

Review secret-key cryptography

- Stream ciphers
  - PRG
- Block ciphers
  - PRF, PRP
  - Modes of operation to encrypt longer messages
- Integrity
  - Message Authentication Codes
- Authenticated encryption
  - Encrypt-then-MAC always secure
  - MAC-then-Encrypt secure only sometimes
- Practical attacks
  - Padding oracle has serious security implications

Acknowledgement

Some of the slides and slide contents are taken from
http://www.crypto.edu.pl/Dziembowski/teaching
and fall under the following:

We have also used slides from Prof. Dan Boneh online cryptography course at Stanford University:

http://crypto.stanford.edu/~dabo/courses/OnlineCrypto/