Authenticated Encryption

Constructions from ciphers and MACs

… but first, 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)

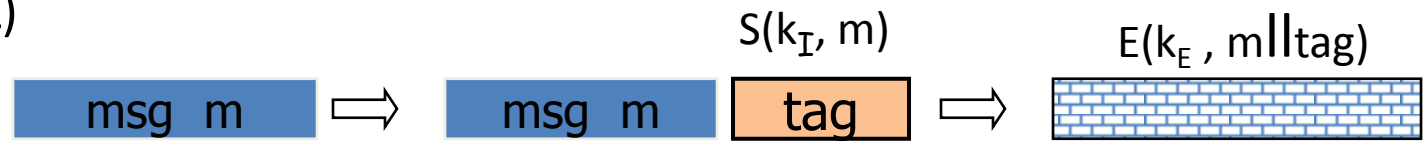
Every project had to combine the two itself without a well defined goal

- Not all combinations provide AE …

Combining MAC and ENC (CCA)

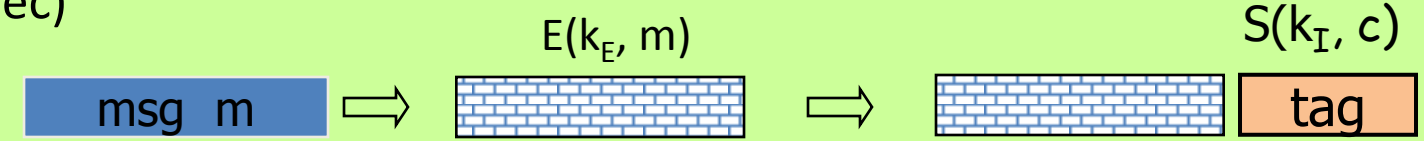
Encryption key k_E. MAC key = k_I

Option 1: (SSL)

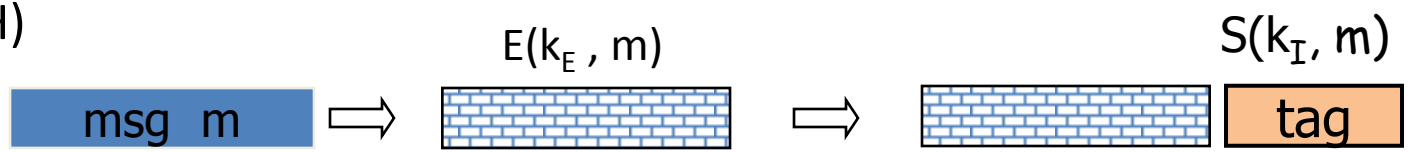


Option 2: (IPsec)

always correct



Option 3: (SSH)



A.E. Theorems

Let (E,D) be CPA secure cipher and (S,V) secure MAC. Then:

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

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

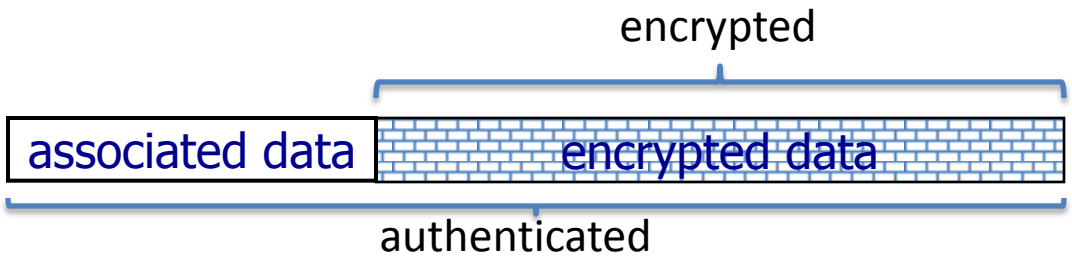
   however: when (E,D) is rand-CTR mode or rand-CBC
   M-then-E provides A.E.

   for rand-CTR mode, one-time MAC is sufficient

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: (auth. enc. 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,

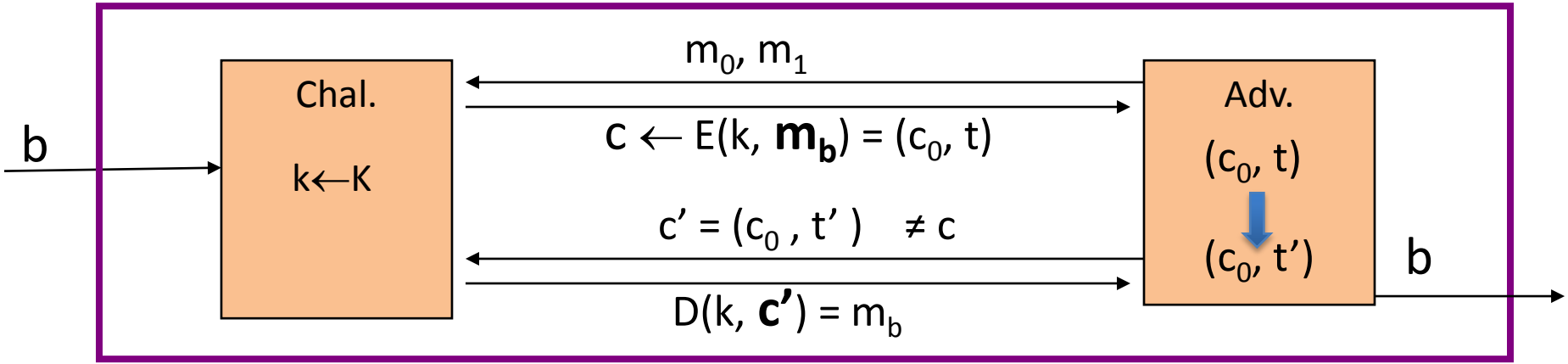
unsigned char ***out**, unsigned long *outlen)

MAC Security -- an explanation

Recall: MAC security implies $(m, t) \not\Rightarrow (m, t')$

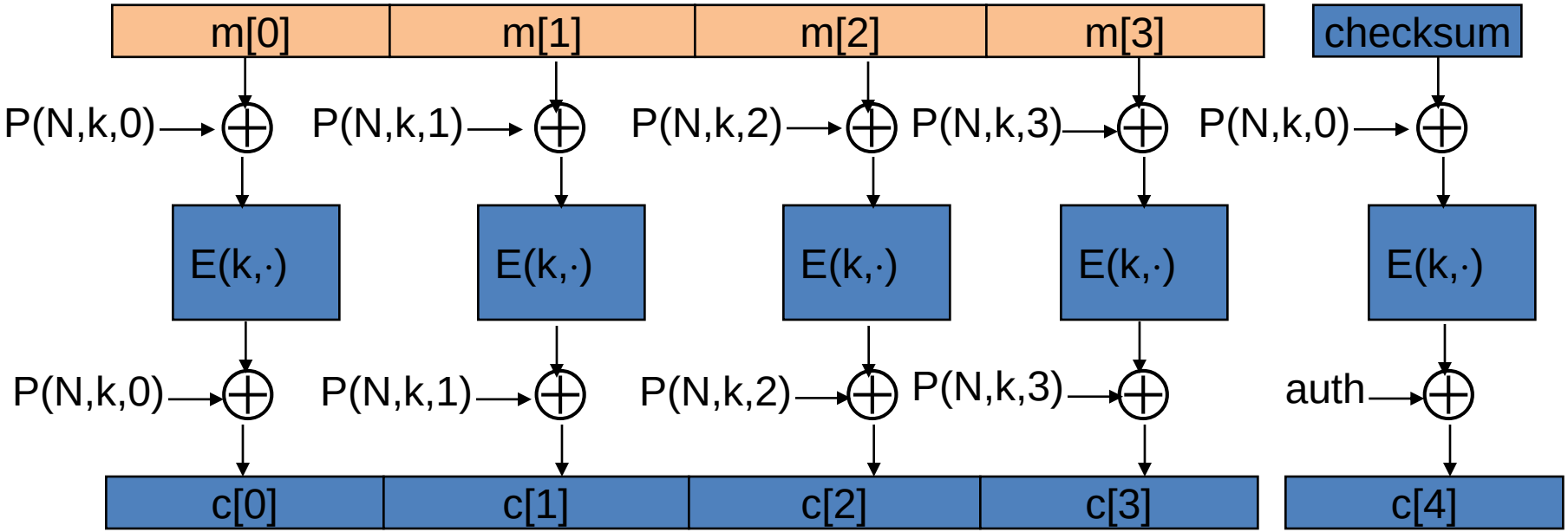
Why? Suppose not: $(m, t) \longrightarrow (m, t')$

Then Encrypt-then-MAC would not have Ciphertext Integrity !!



OCB: a direct construction from a PRP

More efficient authenticated encryption: one E() op. per block.



Performance:

Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

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

* extrapolated from Ted Kravitz's results ** non-Intel machines

End of Segment