symmetric *cryptography*

cs642

computer security

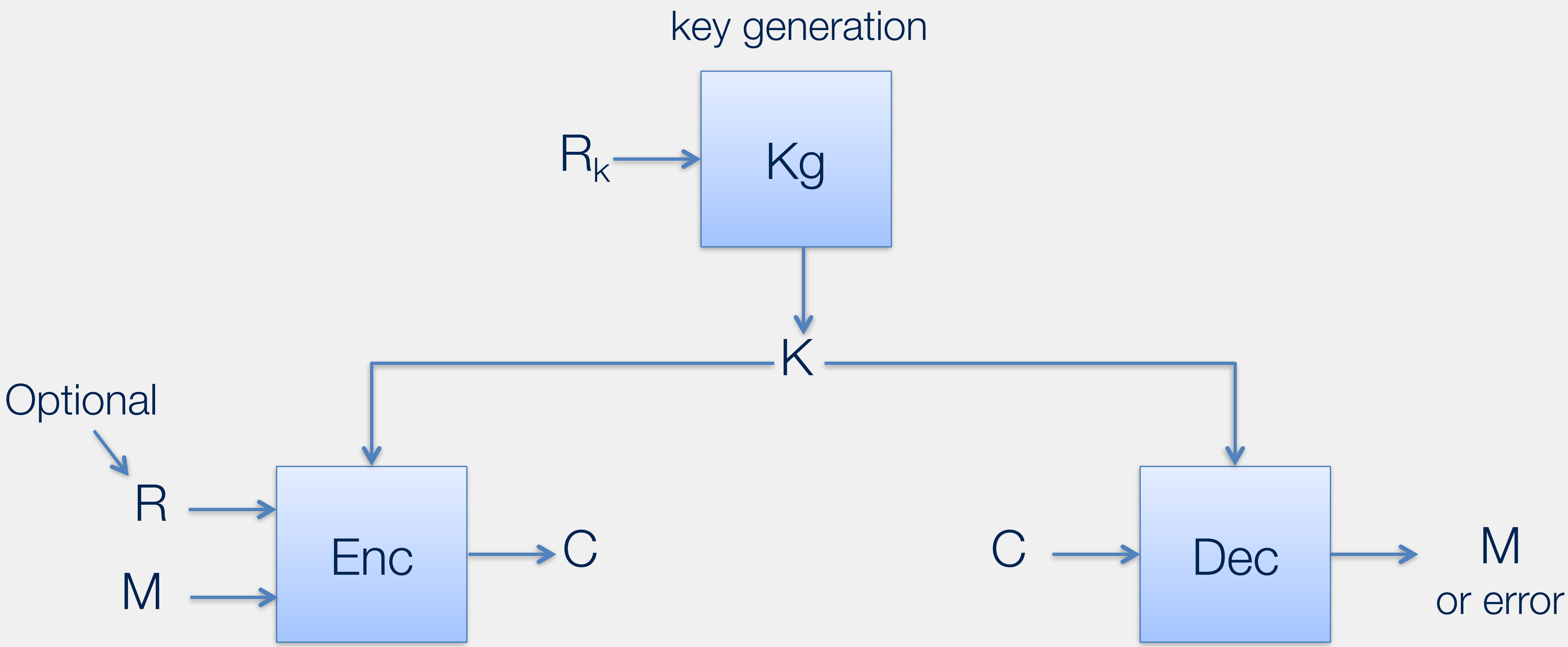
adam everspaugh
ace@cs.wisc.edu

Announcements

- Midterm next week: Monday, March 7 (in-class)
- Midterm Review session Friday: March 4 (here, normal class time)
- Talk today: Adversarial Machine Learning - Scott Alfed
  / CS 4310, 4-5p

today

* Block ciphers (AES)
* Block cipher modes of operation
* Hash functions
* Message authentication codes (MAC), HMAC
* Authenticated encryption

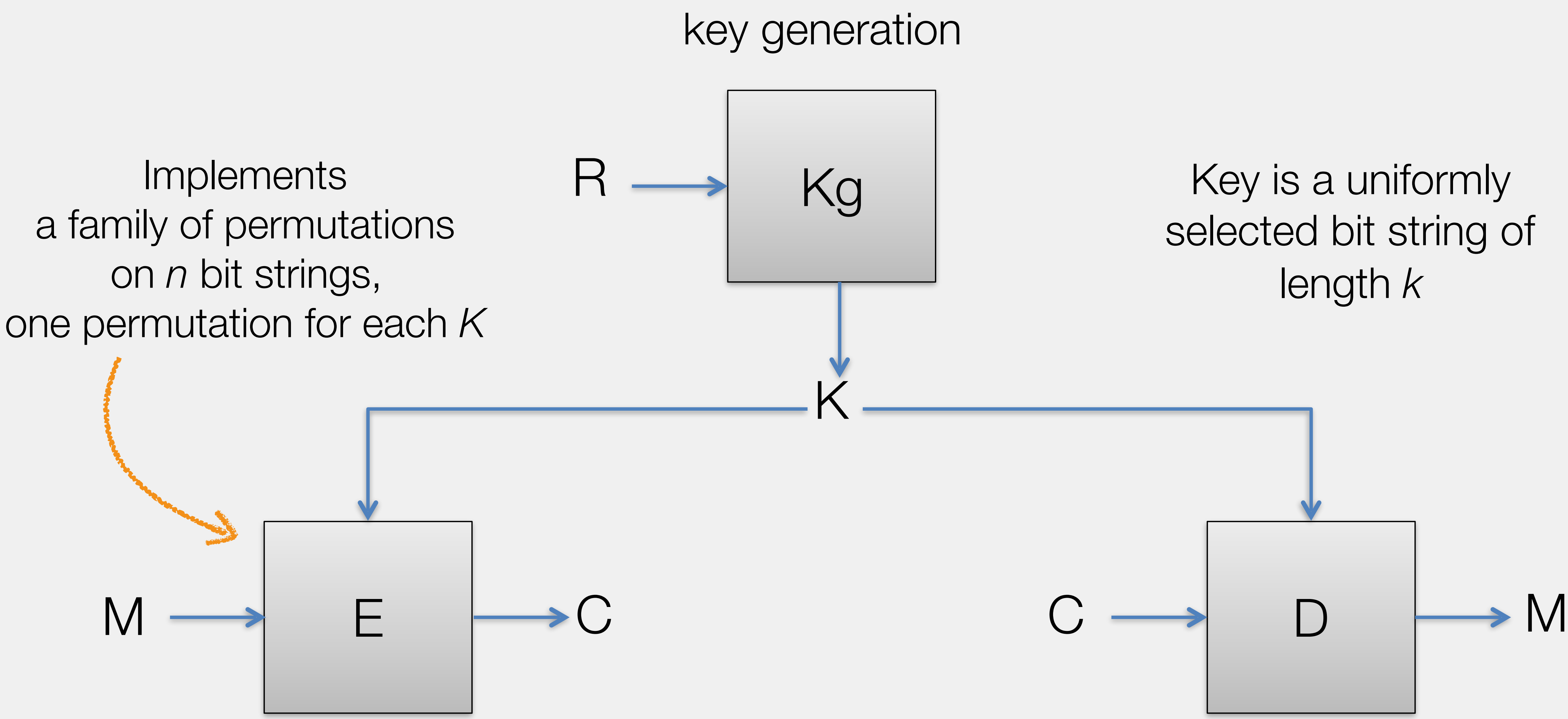


Correctness: Dec(K, E(K,M,R)) = M

with probability 1 over all randomness

symmetric encryption scheme

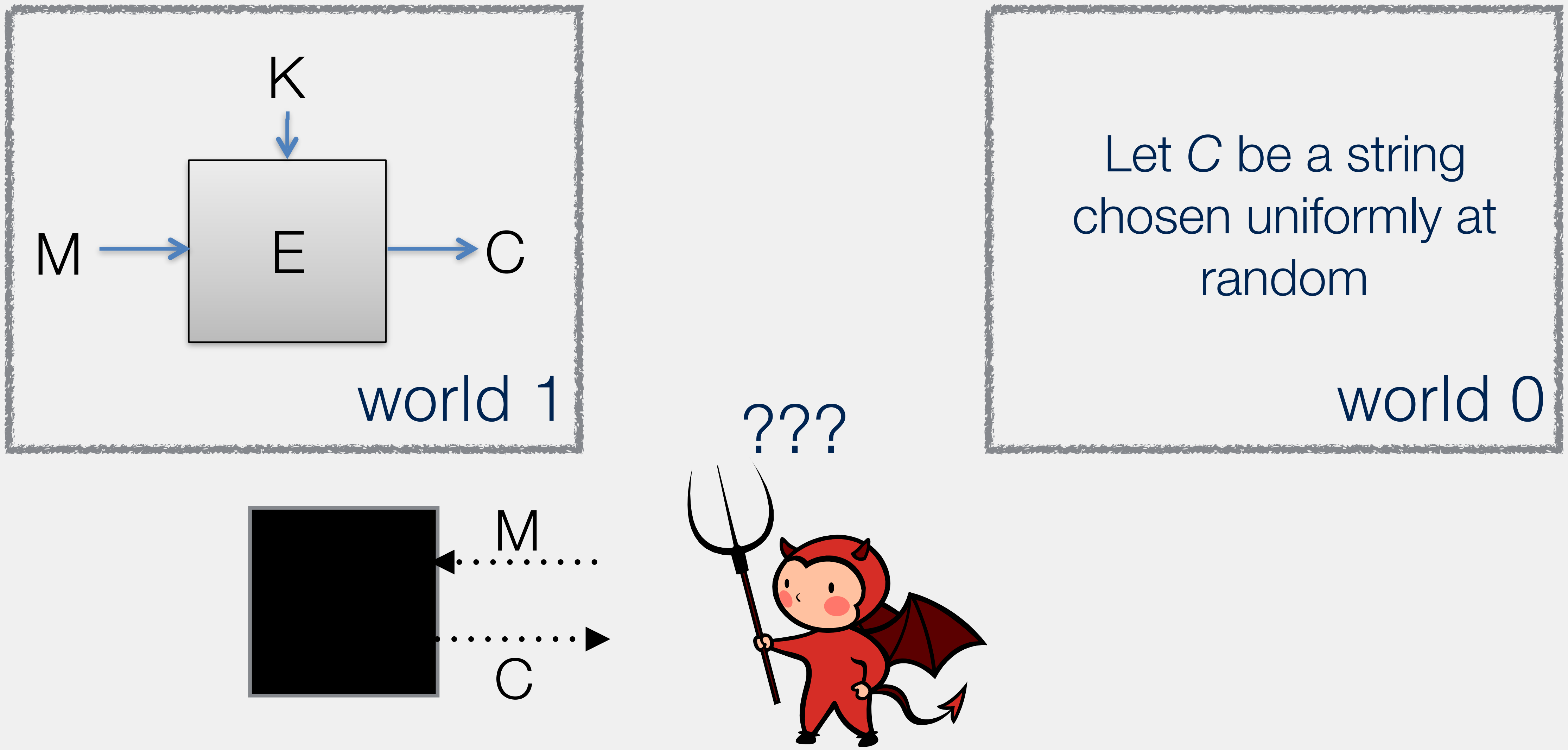
block ciphers



$$E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

Security goal: *E*(*K*,*M*) is indistinguishable from a random *n*-bit string for anyone that doesn't know *K*

$$E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

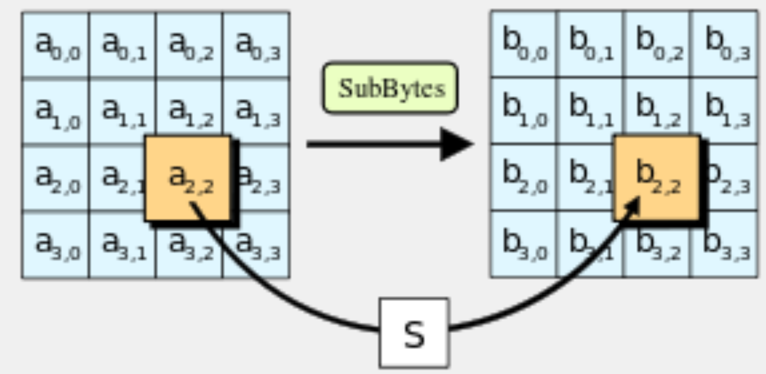


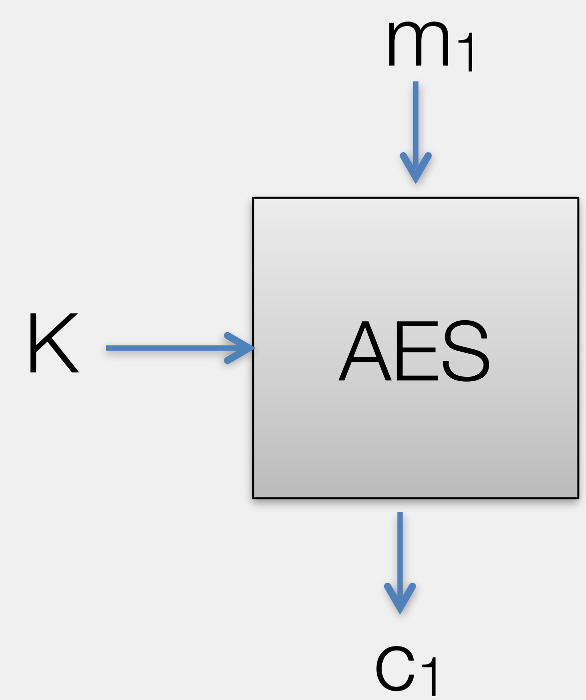
Can adversary distinguish between World 0 and World 1?

If this holds for all polynomial time adversaries, then *E* is called a secure pseudorandom function (PRF)

block cipher security

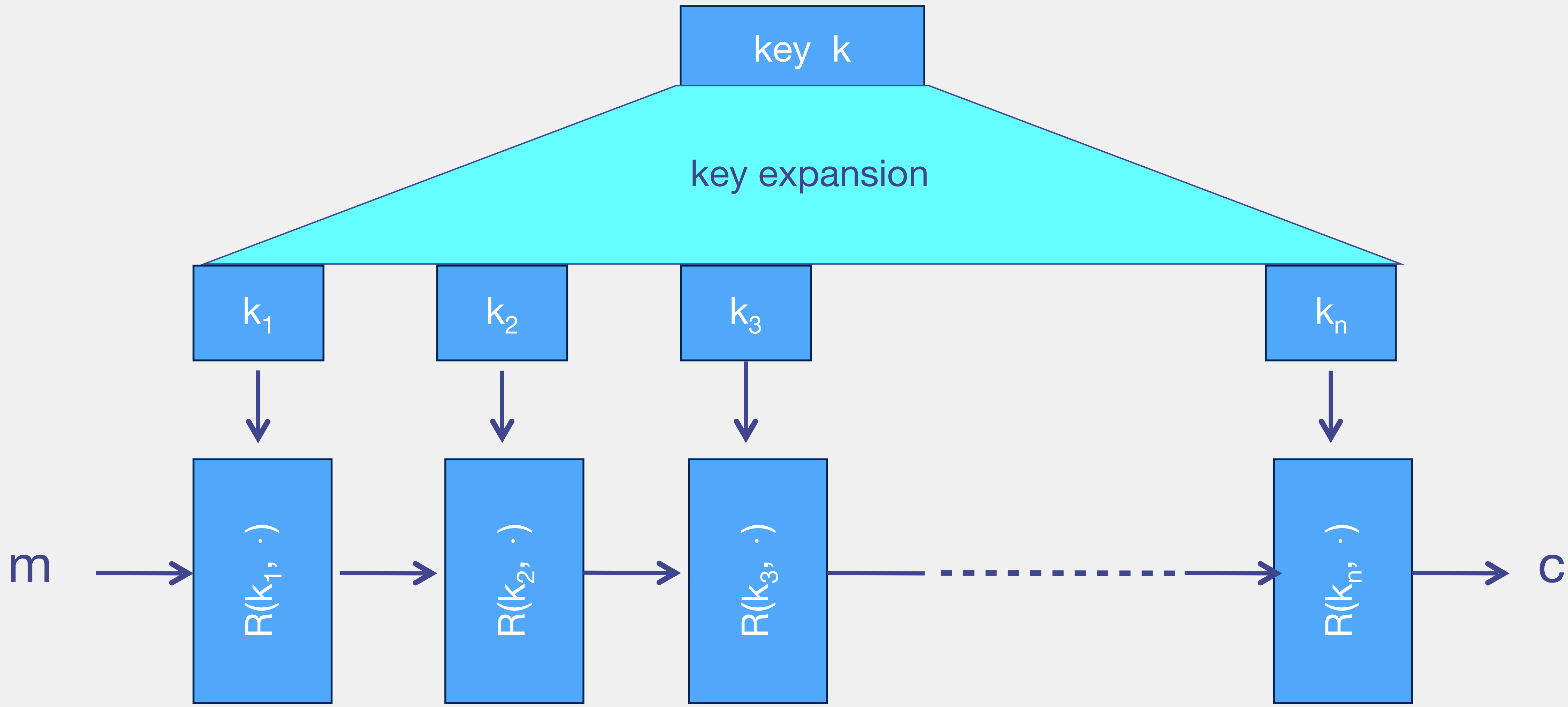
- Advanced Encryption Standard (AES)
- Current standard for a secure block cipher
- Chosen by public competition, run by NIST, academic cryptographers
- Key sizes: 128b, 192b, 256b
- Block size: 128b





aes

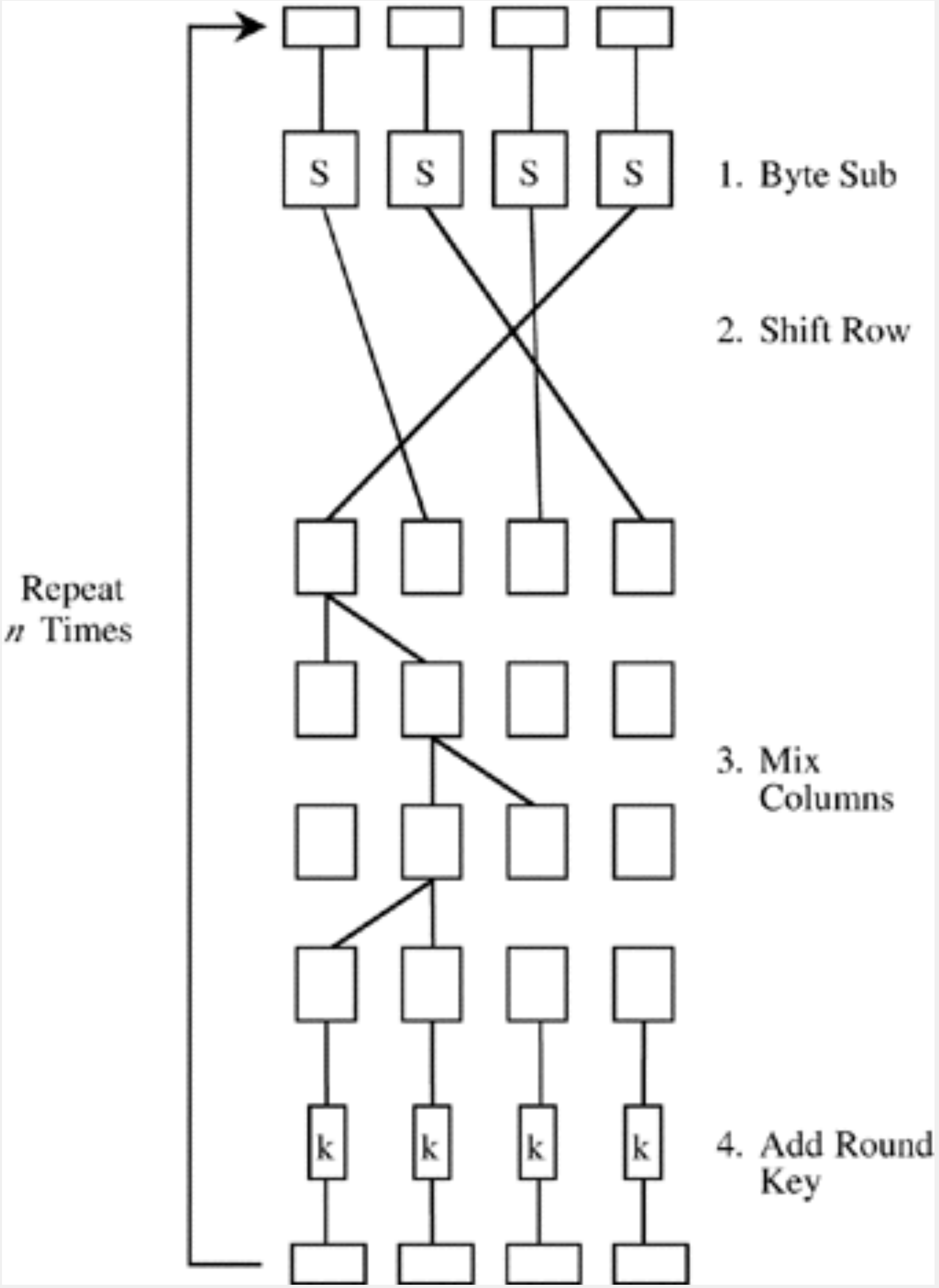
building a block cipher



R(k,m): round function

AES-128 n=10

[slide credit: Dan Boneh, CS155]



Designing good block ciphers is a dark art

Must resist subtle attacks: differential attack, linear attacks, others

Chosen through public design contests

Use build-*break*-build-*break* iteration

aes round function

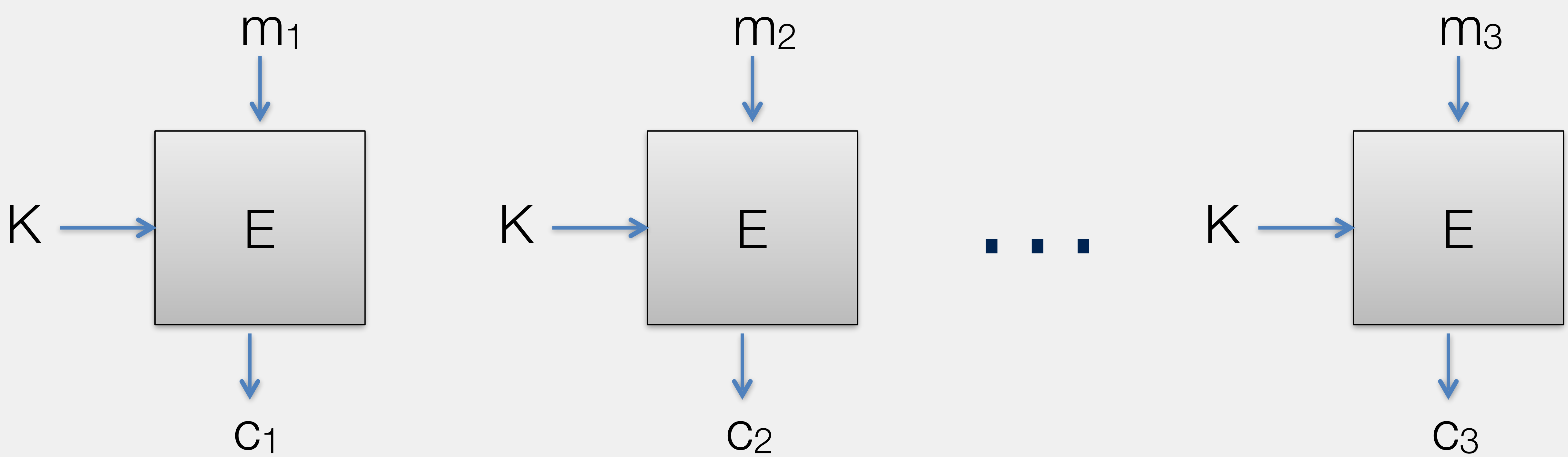
Attack	Attack type	Complexity	Year
Bogdanov, Khovratovich, Rechberger	chosen ciphertext, recovers key	$2^{126.1}$ time + some data overheads	2011

- Brute force attack against AES: 2^{128}
- ~4x speedup

best attacks

Electronic Code Book (ECB) mode

$M = m_1\, m_2\, m_3\, m_4 \ldots m_L$

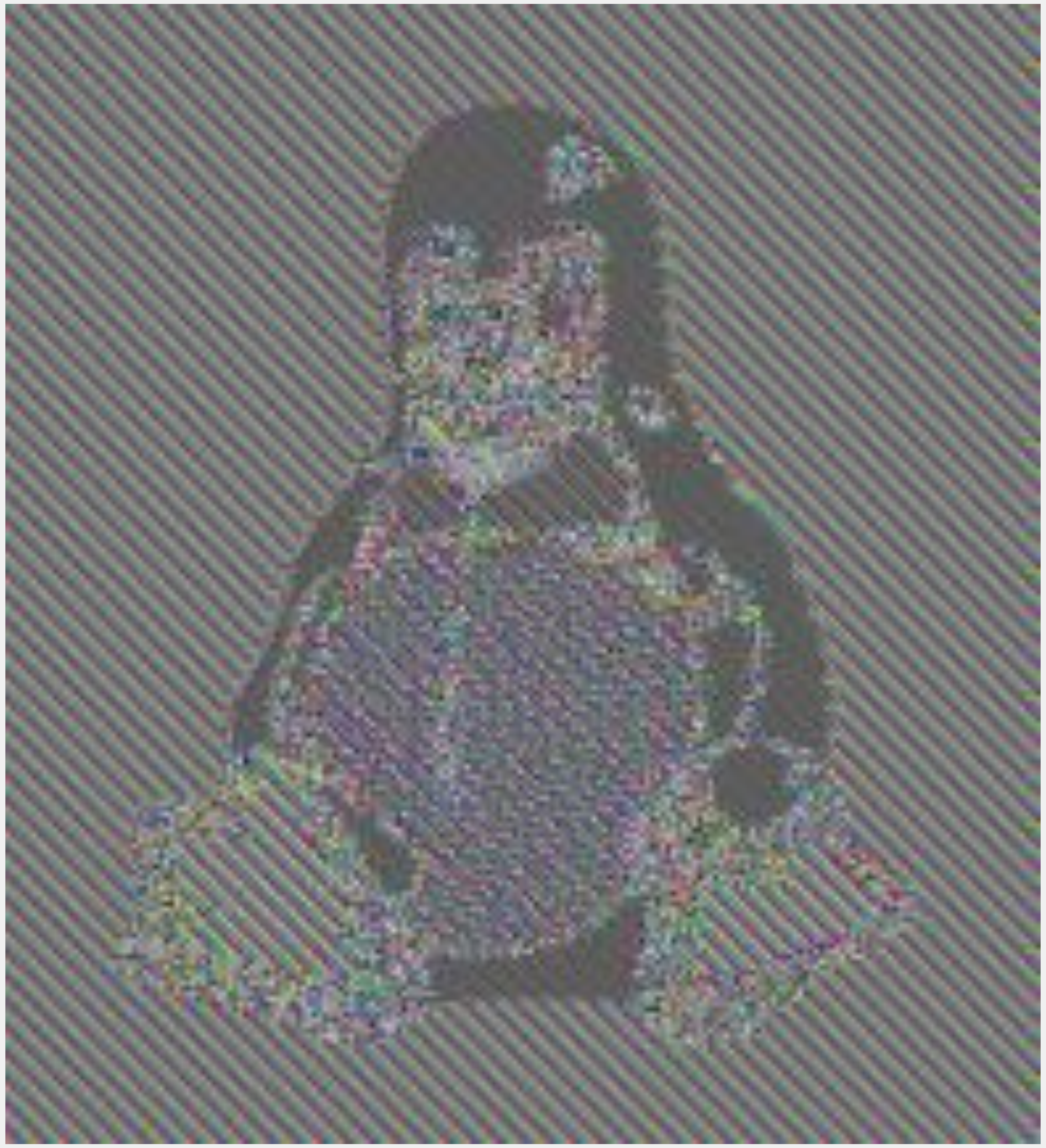


$C = c_1\, c_2\, c_3\, c_4 \ldots c_L$

modes of operation

[images from wikipedia]

image encrypted with ECB



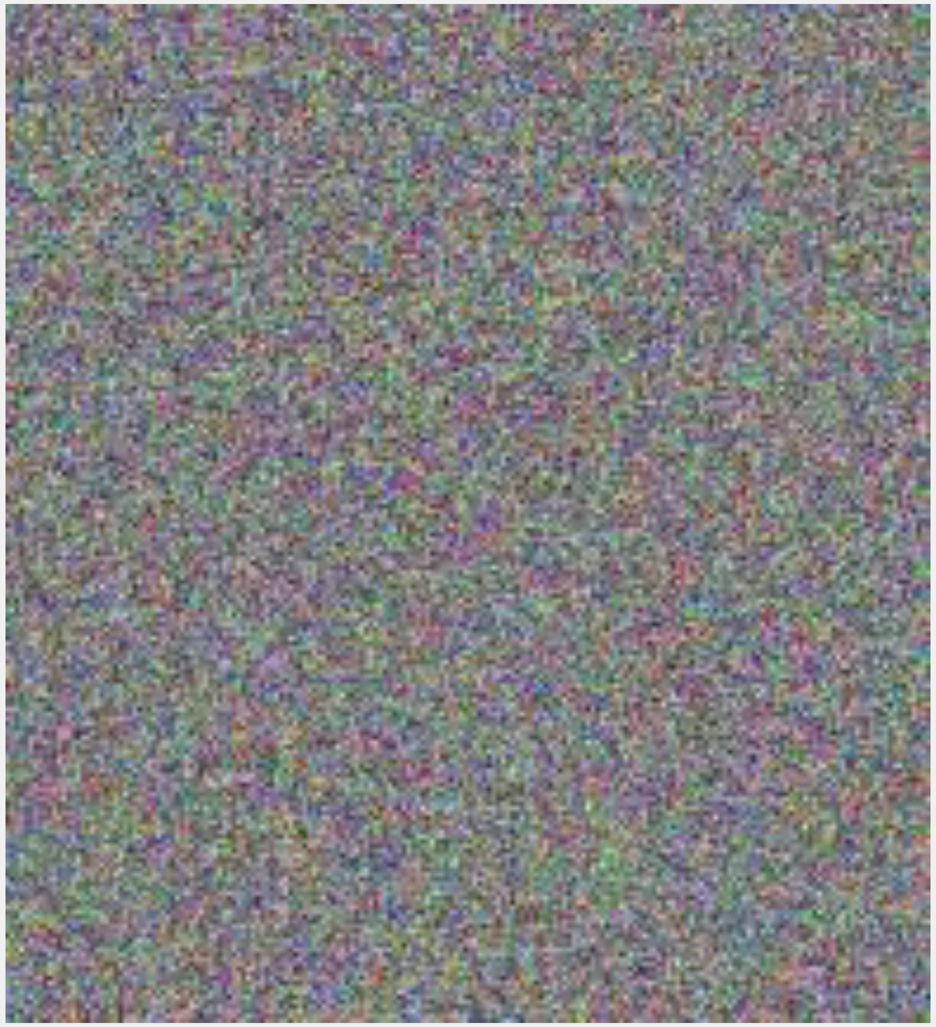
ECB is the *natural way* to implement encryption with block ciphers

But it's *insecure*

Basically → it's a complicated substitution cipher

If $m_i = m_j$ then $E(k, m_i) = E(k, m_j)$

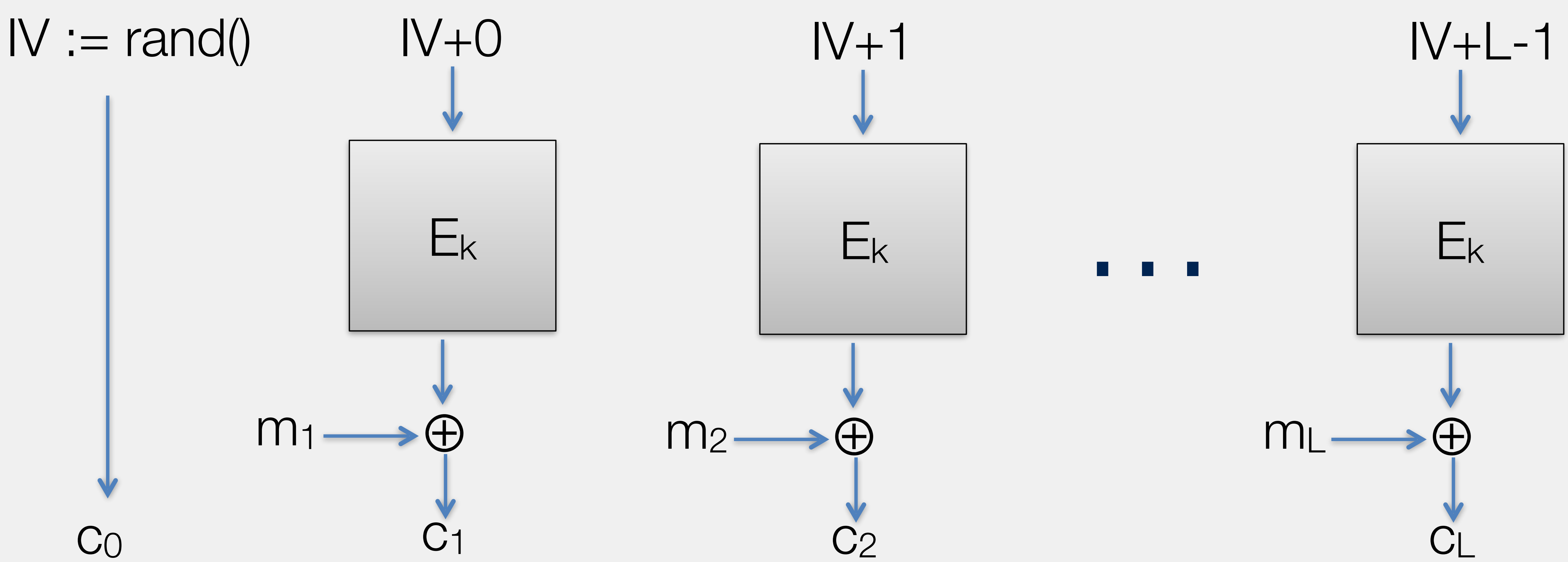
ecb



CTR, GCM, any randomized mode

secure modes

$M = m_1\ m_2\ m_3\ m_4\ \ldots\ m_L$



$C = c_0\ c_1\ c_2\ c_3\ c_4\ \ldots\ c_L$

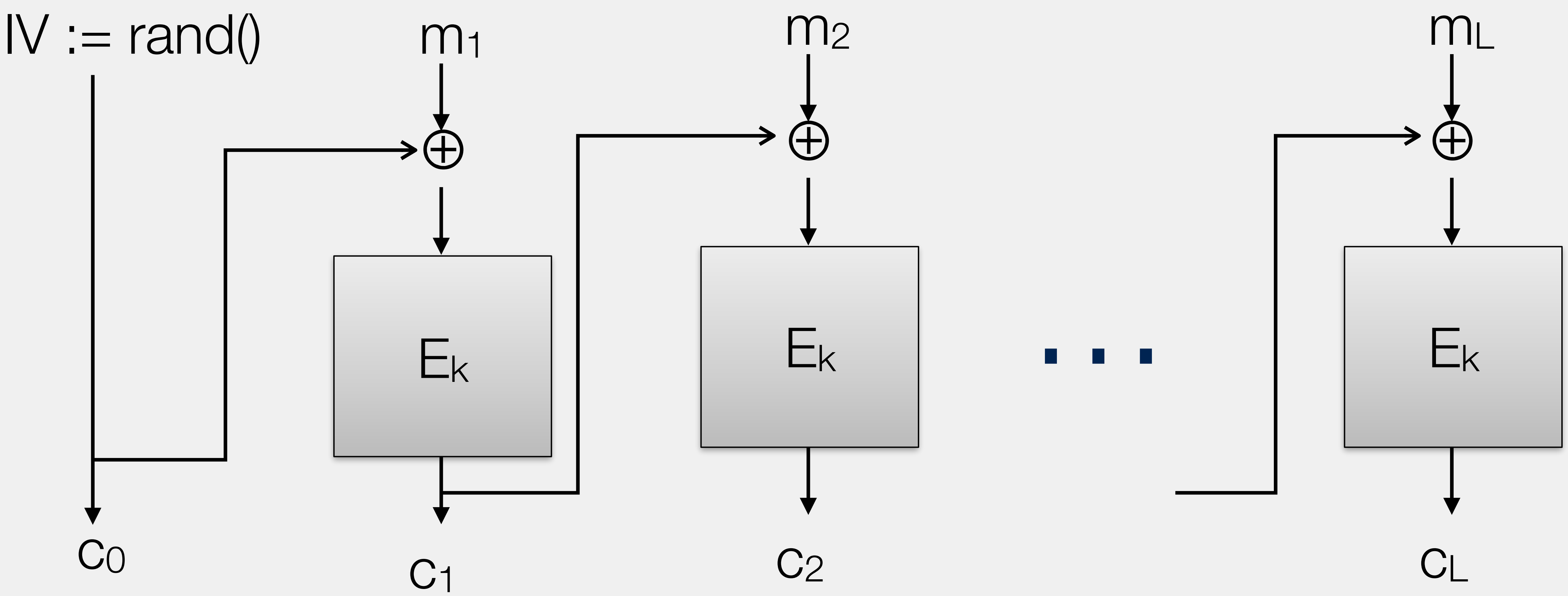
How do we do decryption?

think-*pair*-share

counter mode (CTR)

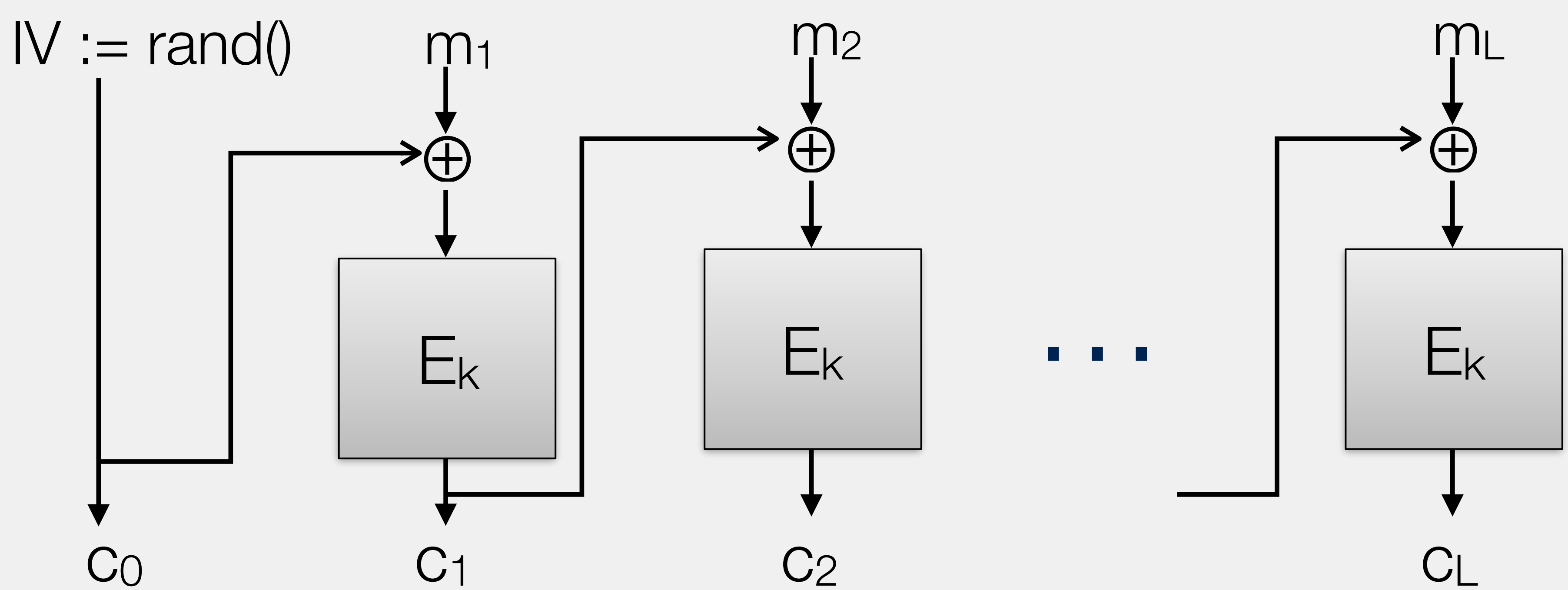
Cipher Block Chaining (CBC) mode

$M = m_1\ m_2\ m_3\ m_4 \ldots m_L$



$C = c_0\ c_1\ c_2\ c_3\ c_4 \ldots c_L$

cbc



Eve

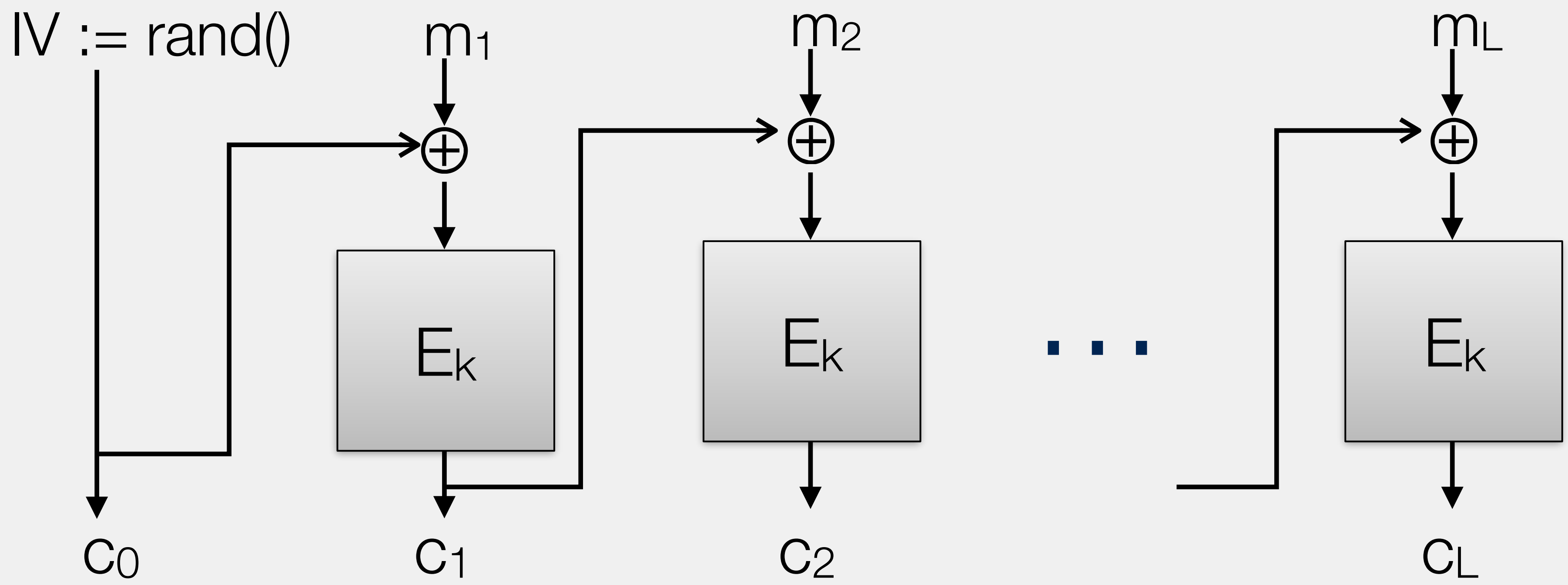
Can attacker learn K from just c_0,c_1,c_2?
 Implies attacker can break E (recover block cipher key)

Can attacker m_1,m_2,m_3 from c_0,c_1,c_2?
 Implies attacker can invert block cipher without K

Can attacker learn one of M from c_0,c_1,c_2?
 Implies attacker can break PRF security of E

Provably: passive adversaries cannot learn anything about M if E is secure

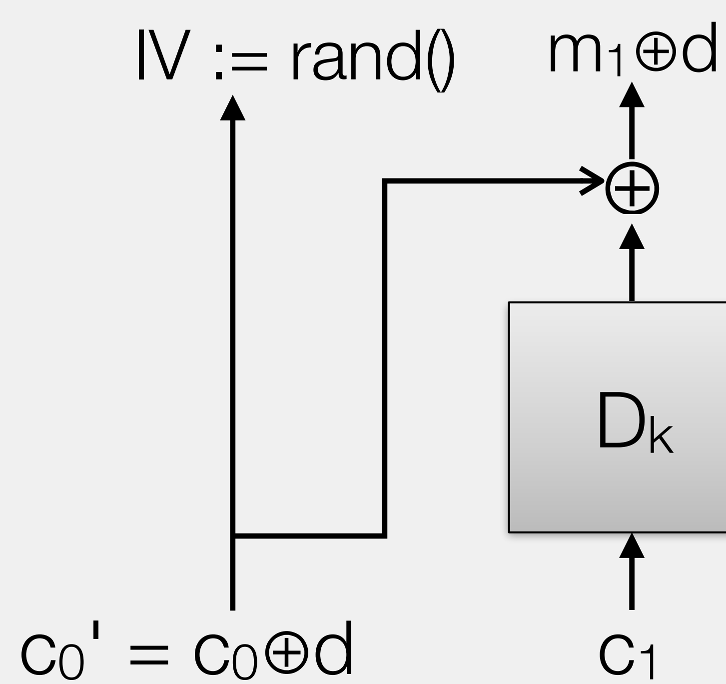
passive security



Mallory

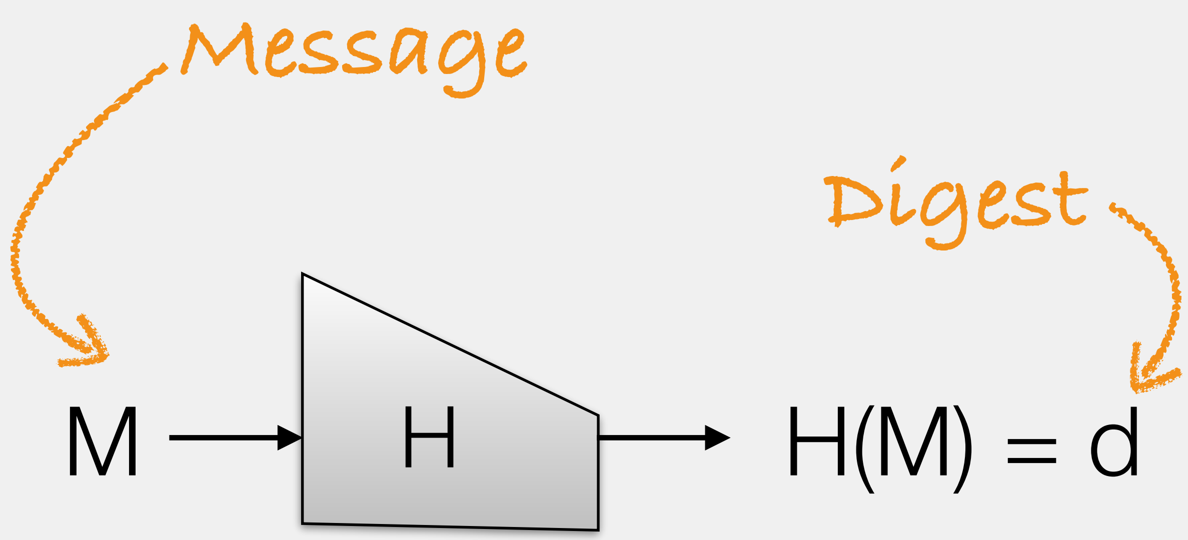
What about forging messages?

For any change *d*:
Send C' = $c_0 \oplus d, c_1$



active security

hash functions



$$H: \{0,1\}^* \rightarrow \{0,1\}^m$$

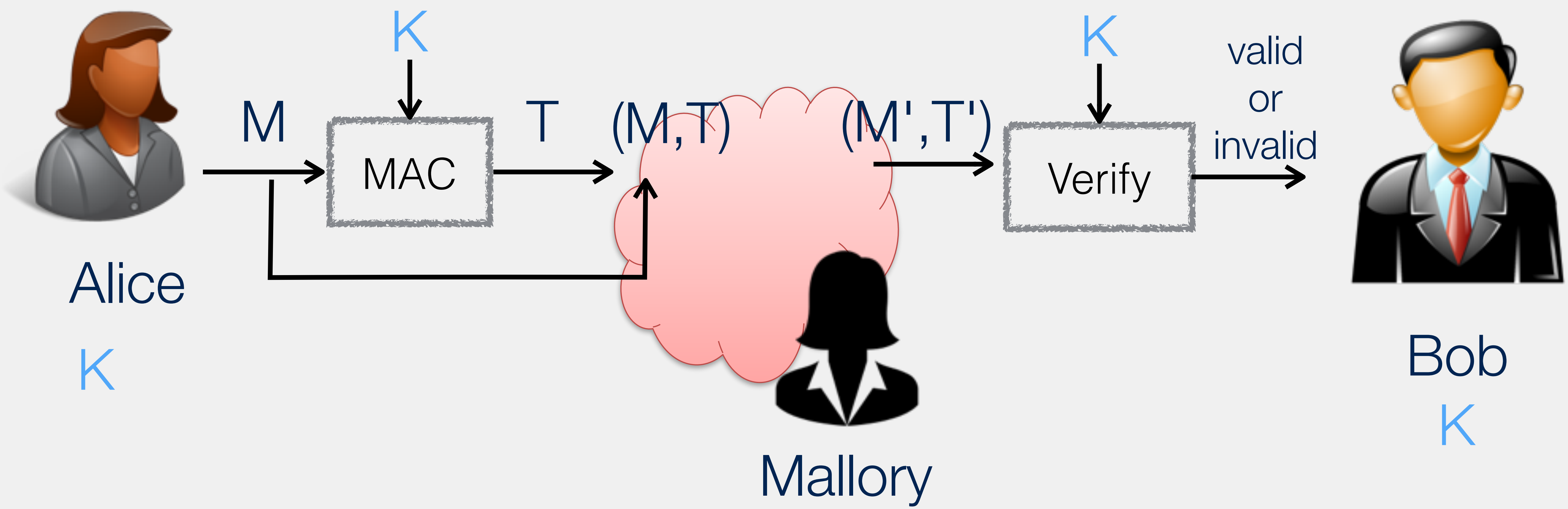
Broken:
- MD5 m=128
- SHA-1 m=160

Current:
- SHA-256 m=256
- SHA-512 m=512
- SHA3-256/512

Security goals

* **Collision resistance**
  / Hard to find any two messages: m != m', H(m) = H(m')
* **Second pre-image resistance**
  / Given H(m), hard to find m' where H(m) = H(m')
* **One-way**
  / Given H(m), hard to find m

hash function



Message Authentication Code (MAC)

message integrity & authenticity / symmetric

* Hashed Message Authentication Code (HMAC)

* Standard method to construct a secure MAC from a hash function H and a key

HMAC(K,M) =

H(K⊕opad || **H(K⊕ipad || M)**)

Fixed constants

(not tablets made by Apple)

hmac

authenticated encryption

k_1 - encryption key
k_2 - MAC key

$C=E_{k1}(M)$, $T=MAC_{k2}(C)$

C,T →

encrypt-then-mac
secure for all secure primitives

encrypt-and-mac $C=E_{k1}(M)$, $T=MAC_{k2}(M)$
may be insecure

C,T →

$T=MAC_{k2}(M)$, $C=E_{k1}(M,T)$

mac-then-encrypt
may be insecure

C →

Even better: use a dedicated AE mode

authenticated encryption

Dedicated authenticated encryption schemes

Attack	Inventors	Notes
OCB (Offset Codebook)	Rogaway	One-pass
GCM (Galios Counter Mode)	McGrew, Viega	CTR mode plus specialized MAC
CWC	Kohno, Viega, Whiting	CTR mode plus Carter-Wegman MAC
CCM	Housley, Ferguson, Whiting	CTR mode plus CBC-MAC
EAX	Wagner, Bellare, Rogaway	CTR mode plus OMAC

AES-GCM - most common,

built-in instructions in Intel chips (very fast)

ae modes

* Block ciphers (AES)

* Block cipher modes of operations
  / ECB - obvious, but insecure!!
  / CTR, CBC

* Hash functions, HMAC

* Authenticated encryption
  / Encrypt-then-MAC
  / AES-GCM and others

* Exit slips
  / 1 thing you learned
  / 1 thing you didn't understand

recap