XCBC: A Version of the CBC MAC for Handling Arbitrary-Length Messages

(From our CRYPTO ’00 paper)

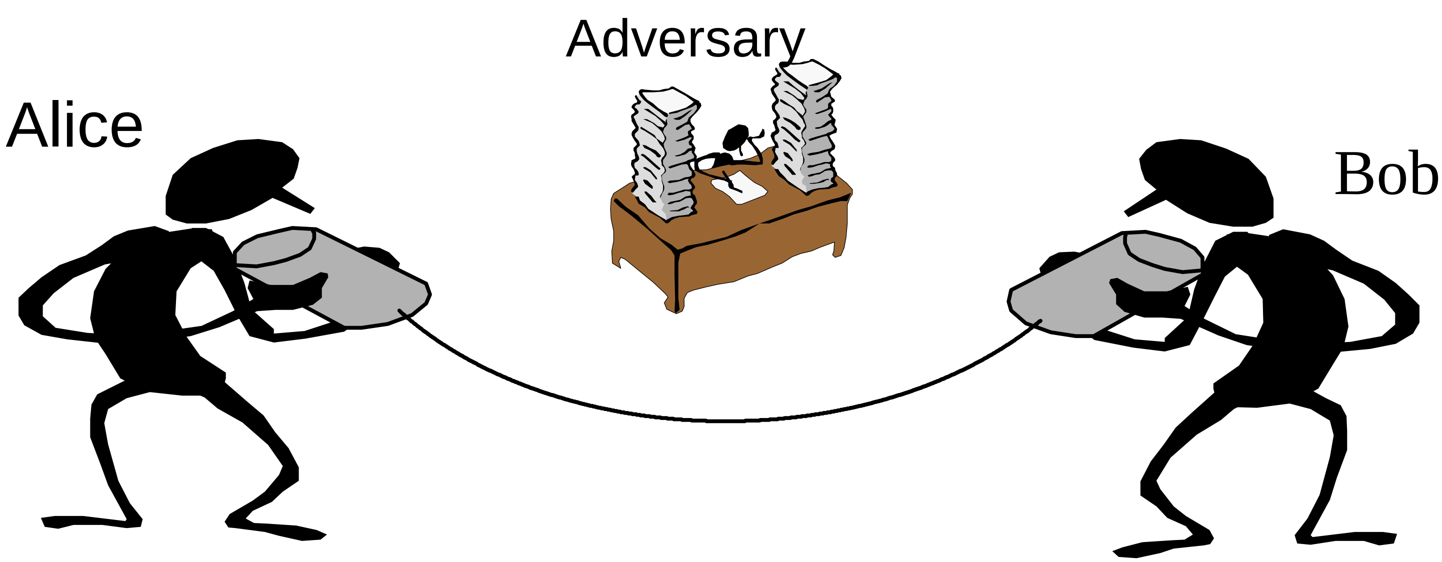
John Black
UNR
jrb@cs.unr.edu
www.cs.unr.edu/~jrb

Phillip Rogaway
UC Davis
rogaway@cs.ucdavis.edu
www.cs.ucdavis.edu/~rogaway

NIST Workshop 2 – Santa Barbara, California
August 24, 2001

What is a MAC?

Alice wishes to send Bob a message in such a way that Bob can be certain (with very high probability) that Alice was the true originator of the message.



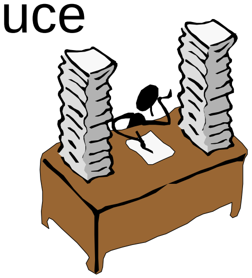
What is the Goal?

The adversary sees messages and their MACs, then attempts to produce a new message and valid MAC (aka a “forgery”).

[GMR, BKR]

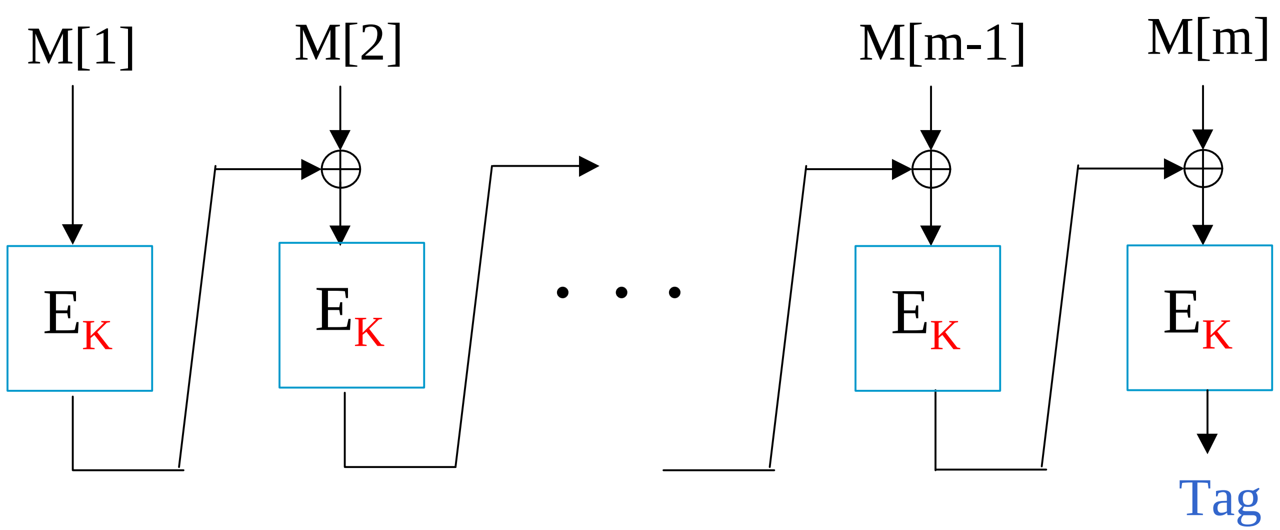
Can easily produce valid MACs

Cannot produce valid MACs



The CBC MAC

- Simple
- Widely used
- Secure (on messages of a fixed length) [BKR]
- Widely standardized: ANSI X9.19, FIPS 113, ISO 9797



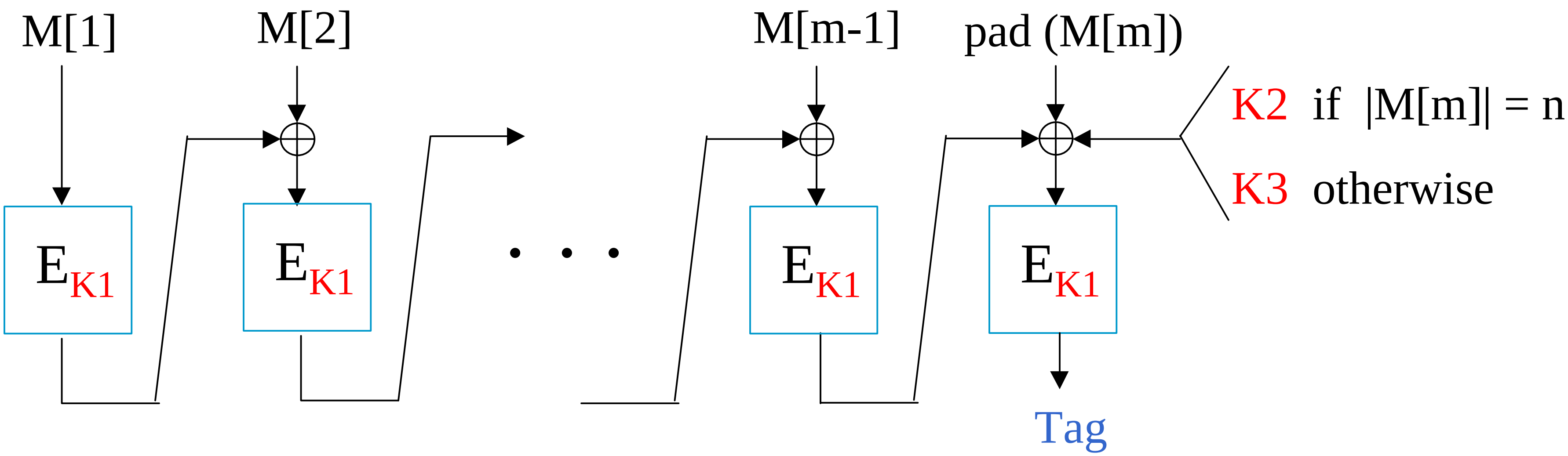
Extending the Message Domain

- The CBC MAC does not allow messages of arbitrary bit length

  // all messages must be a multiple of *n* bits

- The CBC MAC does not allow messages of varying lengths
- Several suggestions address these problems:
  - Various padding schemes
  - ANSI X9.19 (Optional Triple-DES)
  - Race Project (EMAC) (Analysis by [Petrank, Rackoff])
  - [Knudsen, Preneel] (MacDES)
  - [Black, Rogaway] (XCBC) ← Today

The XCBC MAC



$$\text{pad}(x) = \begin{cases} x & \text{if } |x| = n \\ x\,10\cdots0 & \text{if } |x| < n \end{cases}$$

The XCBC MAC

algorithm XCBCMAC$_{K1\ K2\ K3}$(M)
partition M into M[1] … M[m]
C[0] = 0^n
for i=1 **to** m-1 **do**
  C[i] = E_{K1}(C[i-1] ⊕ M[i])
if |M[m]|=n **then** Tag = E_{K1}(C[m-1] ⊕ M[m] ⊕ K2)
          else Tag = E_{K1}(C[m-1] ⊕ M[m] 10···0 ⊕ K3)
return Tag

Advantages of XCBC

- Uses minimal number of block cipher invocations for this style of MAC
- Correctly handles messages of any bit-length
- Block cipher is invoked with only one key: K1
- Block cipher invoked only in forward direction
- Allows on-line processing
- Easy to implement, familiar to users
- Patent-free

Advantages of XCBC (cont.)

- XCBC is a PRF (not just a MAC)
  - A secure PRF is always a secure MAC [GGM, BKR]
  - No nonce/IV is used
  - Tags are shorter
  - Tags may be truncated
  - Other applications
    - Key separation
    - PRG
    - Handshake protocols
- Provably secure (assuming E is a PRP)

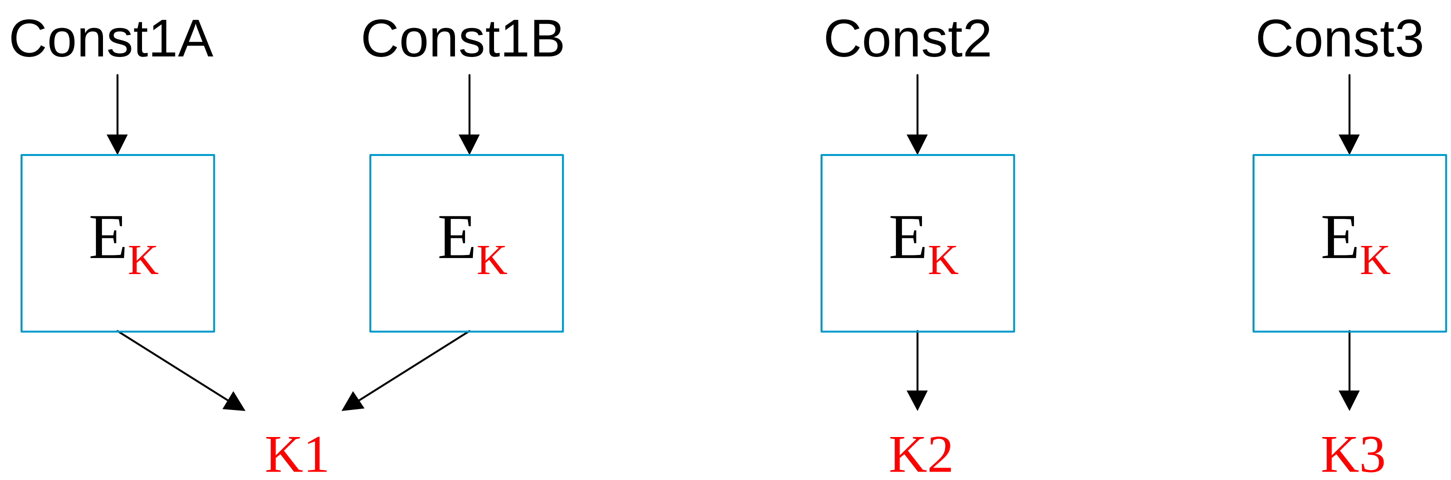
Disadvantages of XCBC

- Limited parallelism

  (Inherent in CBC MAC)
- Key of length $k + 2n$

A Note on Deriving K1, K2, K3

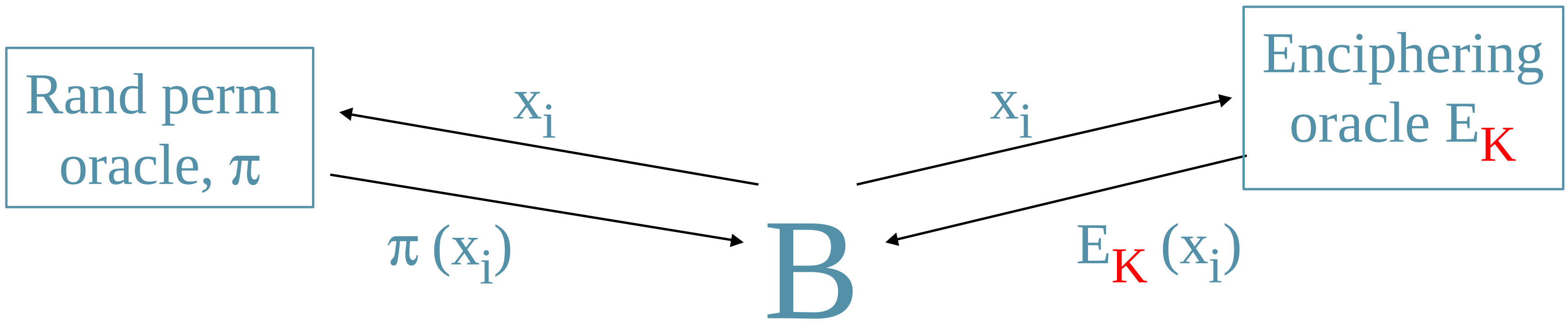
- Under standard assumptions (ie, that E is a PRP) we can derive K1, K2, and K3 in the standard way from a single key K.



Block-Cipher Security

Security as a PRP

[Goldreich, Goldwasser, Micali]
[Luby, Rackoff]
[Bellare, Kilian, Rogaway]
[Bellare, Guerin, Rogaway]

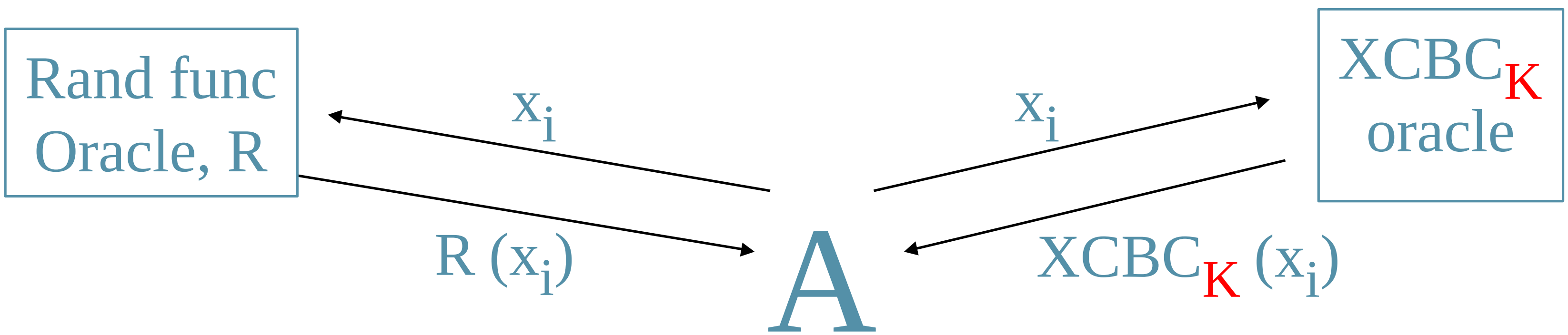


$$\mathbf{Adv}^{\mathbf{prp}}(\mathbf{B}) = \Pr[\mathbf{B}^{E_K} = 1] - \Pr[\mathbf{B}^{\pi} = 1]$$

XCBC's Security

Security as a PRF

[Goldreich, Goldwasser, Micali]
[Bellare, Kilian, Rogaway]
[Bellare, Guerin, Rogaway]



$$\mathbf{Adv}^{\mathbf{prf}}(\mathbf{A}) = \Pr[\mathbf{A}^{\mathbf{XCBC}_K} = 1] - \Pr[\mathbf{A}^R = 1]$$

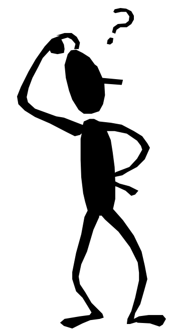
Security

Thm: Assume E is a random block cipher. Then an adversary A who makes at most q queries, each of at most mn bits ($m \le 2^{n-2}$), can distinguish XCBC from a random function with advantage

$$\mathrm{Adv}^{\mathrm{prf}}(A) \le \frac{(4m^2 + 1)\, q^2}{2^n}$$

When E is a real block cipher (eg, AES) one adds a term $\mathrm{Adv}^{\mathrm{prp}}$ to the above bound

What Did That Mean?



- Concrete Example:
  - Say our max message length is 10Kb
  - An adversary watches 1,000 MAC tags go by every second for a month
  - Adversary's chance of forgery is less than one in a trillion

Any Questions?

John Black
UNR

jrb@cs.unr.edu
www.cs.unr.edu/~jrb

Phillip Rogaway
UC Davis

rogaway@cs.ucdavis.edu
www.cs.ucdavis.edu/~rogaway

NIST Workshop 2 – Santa Barbara, California
August 24, 2001