PMAC: A Parallelizable Message Authentication Code

John Black
Department of Computer Science
University of Nevada, Reno
jrb@cs.unr.edu
http://www.cs.unr.edu/~jrb

Phillip Rogaway
Department of Computer Science
UC Davis + CMU
rogaway@cs.ucdavis.edu
http://www.cs.ucdavis.edu/~rogaway
+66 1 530 7620   +1 530 753 0987

NIST Modes of Operation Workshop 2 – Aug 24, 2001 - Santa Barbara, California

What is a MAC

K K

A B

MAC^G: generate authentication tag

$\sigma = \mathrm{MAC}^{G}_{K}(\ [IV],\ M)$

$M\ .\ \sigma$

$\longrightarrow$

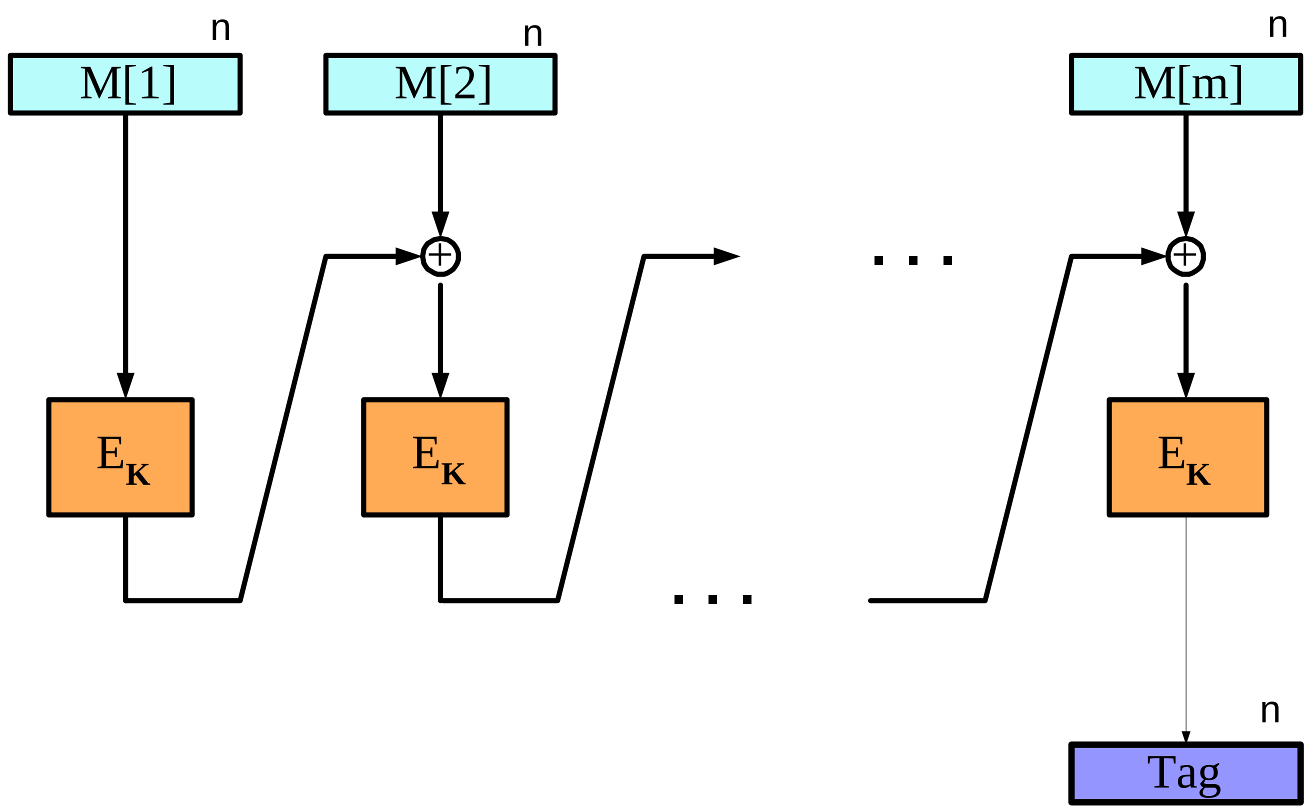
MAC^V: verify authentication tag:

$\mathrm{MAC}^{V}_{K}(M,\ \sigma)$

- Security addresses an adversary's **inability** to forge a **valid** authentication tag for some **new** message.
- Most MACs are **deterministic**–they need no nonce/state/IV/$. In practice, such MACs are preferable. Deterministic MACs are usually PRFs.

CBC MAC

Inherently sequential

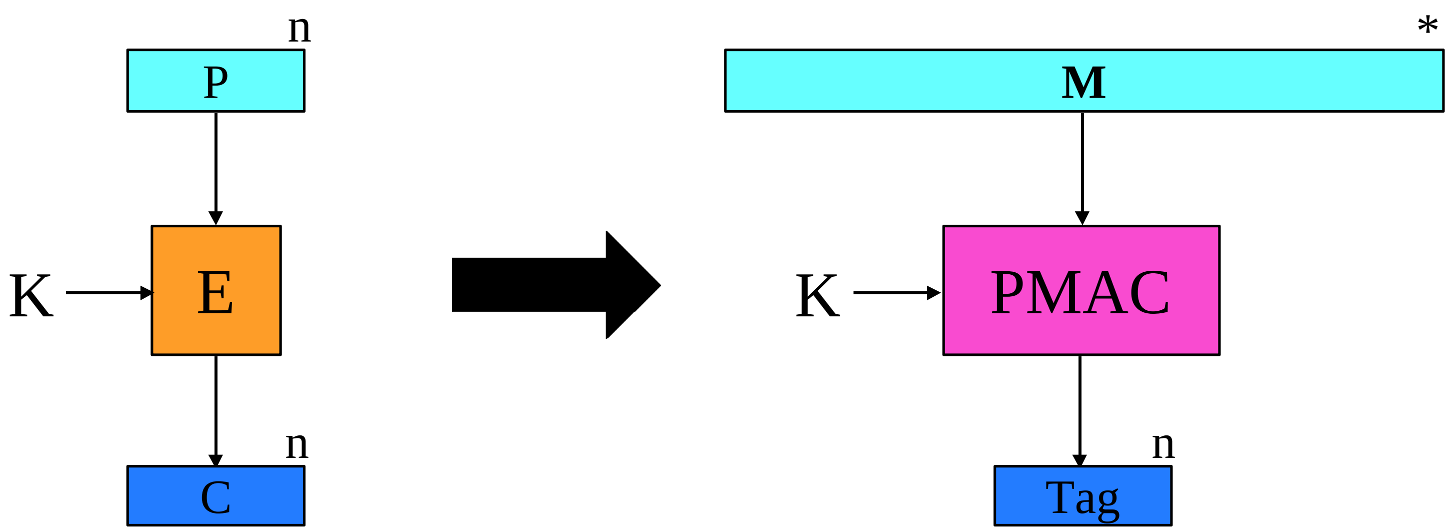


PMAC's Goals

- A fully parallelizable alternative to the CBC MAC
- But without paying much for parallelizability in terms of serial efficiency
- While we're at it, fix up other "problems" of the CBC MAC
    - Make sure PMAC applies to any bit string
    - Make sure it is correct across messages of different lengths

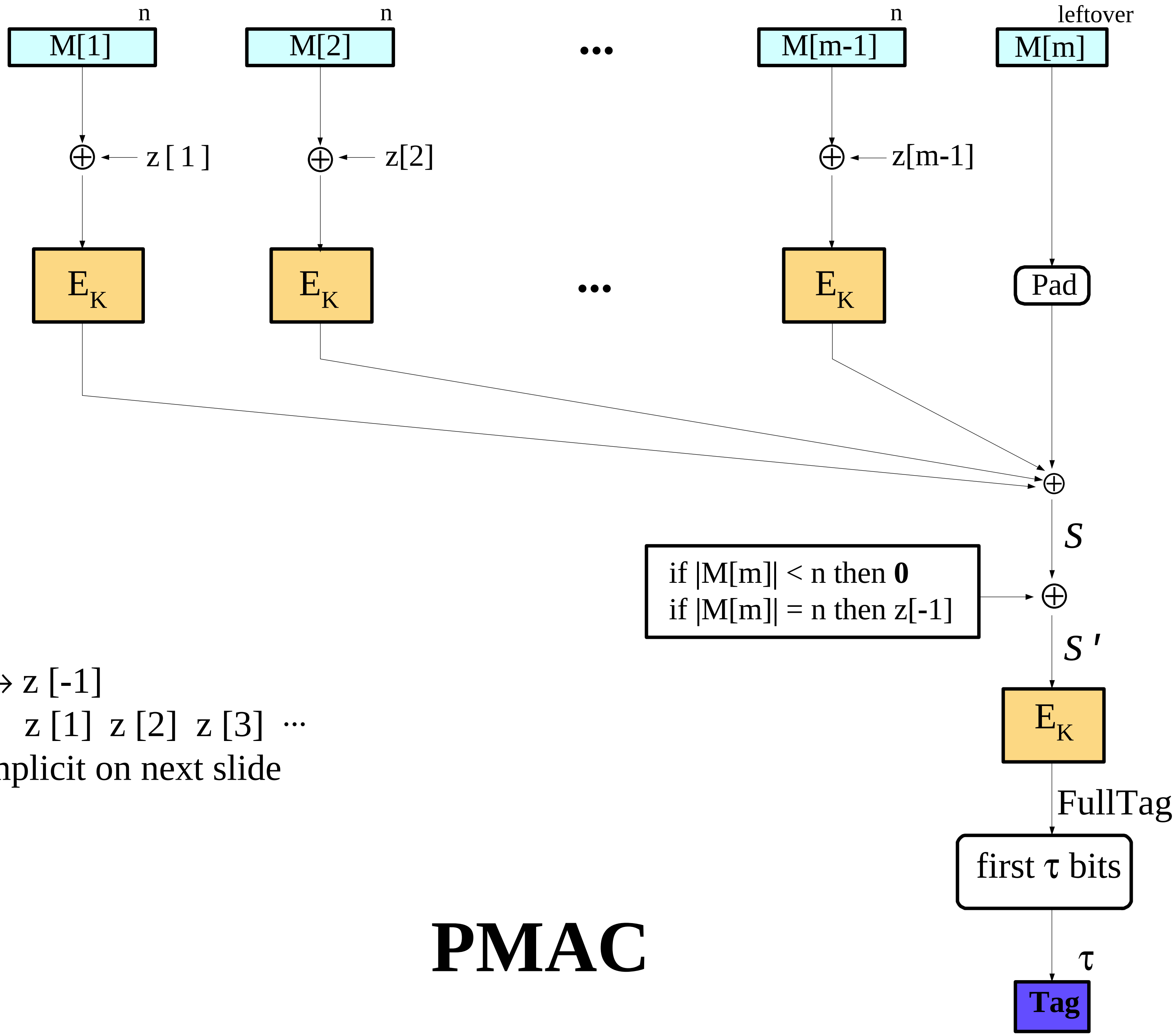
What *is* PMAC ?

- A **variable-input-length pseudorandom function** (VIL PRF):
  PMAC: $\{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^n$
- That you make from
  a **fixed-input-length pseudorandom function** (FIL PRF) –
  invariably a block cipher such as E=AES:
  E: $\{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$



PMAC's Properties

- Functionality: **VIL PRF**: $\{0,1\}^* \rightarrow \{0,1\}^n$
  Can't distinguish $\mathrm{PMAC}_K(\cdot)$ from a random function $\mathbf{R}(\cdot)$
- Customary use of a VIL PRF:
  A (stateless, deterministic) **Message Authentication Code** (MAC)
- PRFs make the most pleasant MACs because they are deterministic and stateless.
- Few block-cipher calls**:** $\lceil |M| / n \rceil$ to PMAC message M
- Low session-setup cost: about **one block-cipher call**
- **Fully parallelizable**
- No n-bit addition or mod p operations – just xors and shifts
- Uses a **single block-cipher key**
- **Provably secure**: If E is a secure block cipher
  then PMAC-E is a good PRF



$K \rightarrow z[-1]$

$z[1]\ \ z[2]\ \ z[3]\ \cdots$

as implicit on next slide

PMAC

Definition of PMAC [E, t]

algorithm $\text{PMAC}_K(M)$
$L(0) = E_K(\mathbf{0})$
$L(-1) = \text{lsb}(L(0))?\ (L(0) \gg 1) \oplus \text{Const43} : (L(0) \gg 1)$
for $i = 1, 2, \ldots$ **do** $L(i) = \text{msb}(L(i-1))?\ (L(i-1) \ll 1) \oplus \text{Const87} : (L(i-1) \ll 1)$
Partition M into $M[1] \cdots M[m]$ // each 128 bits, except M[m] may be shorter
Offset = $\mathbf{0}$
for i=1 **to** m-1 **do**
    $\text{Offset} = \text{Offset} \oplus L(\text{ntz}(i))$
    $S = S \oplus E_K(M[i] \oplus \text{Offset})$
$S = S \oplus \text{pad}(M[m])$
if $|M[m]| = n$ **then** $S = S \oplus L(-1)$
$\text{FullTag} = E_K(S)$
Tag = first t bits of FullTag
return Tag

Related Work

- [Bellare, Guerin, Rogaway 95] – the XOR MAC.
  Not a PRF, but introduced central element of the construction
- [Bernstein 99] – A PRF-variant of the XOR MAC
- [Gligor, Donescu 00, 01] – Another descendent of the XOR MAC.
  Introduced the idea of combining message blocks with a sequence of offsets as an alternative to encoding. Not a PRF

- [Black, Rogaway 00] – Tricks for optimal handing of arbitrary input lengths (XCBC method you have just seen)

- [Carter-Wegman 79, 81] – A completely different approach that can achieve the same basic goals.
- Tree MAC (a la Merkle) – Another approach, not fully parallelizable.

Speed

Data courtesy of **Ted Krovetz**

PMAC-AES	**18.4** cpb	8 % slower
CBCMAC-AES	**17.1** cpb	

The CBC MAC is in its "raw" form. Code is Pentium 3 assembly under gcc.
This CBC MAC figure is **inferior** to Lipmaa's **OCB** results, indicating that PMAC and OCB add so little overhead that quality-of-code differences contribute more to measured timing differences than algorithmic differences across
CBC – CBCMAC – PMAC – OCB.
Since Lipmaa obtained **15.5** cpb for the CBC MAC,
adding 8% to this, **16.7** cpb,
is a conservative estimate for well-optimized Pentium code.

Provable Security

- Provable security begins with [Goldwasser, Micali 82]
- Despite the name, one doesn't really *prove* security
- Instead, one gives *reductions*: theorems of the form

  If a certain primitive is secure
  then the scheme based on it is secure

  For us:

  If AES is a secure block cipher
  then PMAC-AES is a secure authenticated-encryption scheme

  Equivalently:

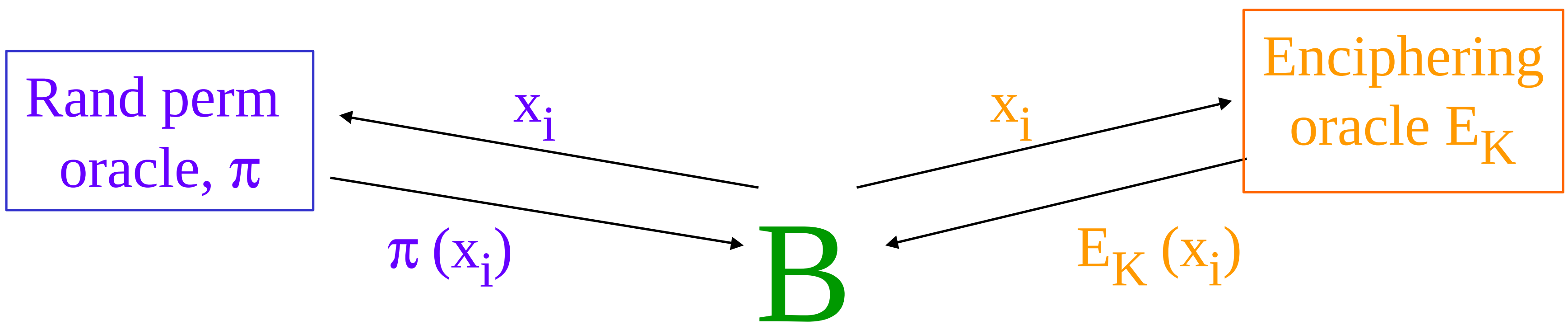
  If some adversary **A** does a good job at breaking PMAC-AES
  then some comparably efficient **B** does a good job to break AES

- Actual theorems quantitative: they measure how much security is "lost" across the reduction.

Block-Cipher Security

Security as a FIL PRP

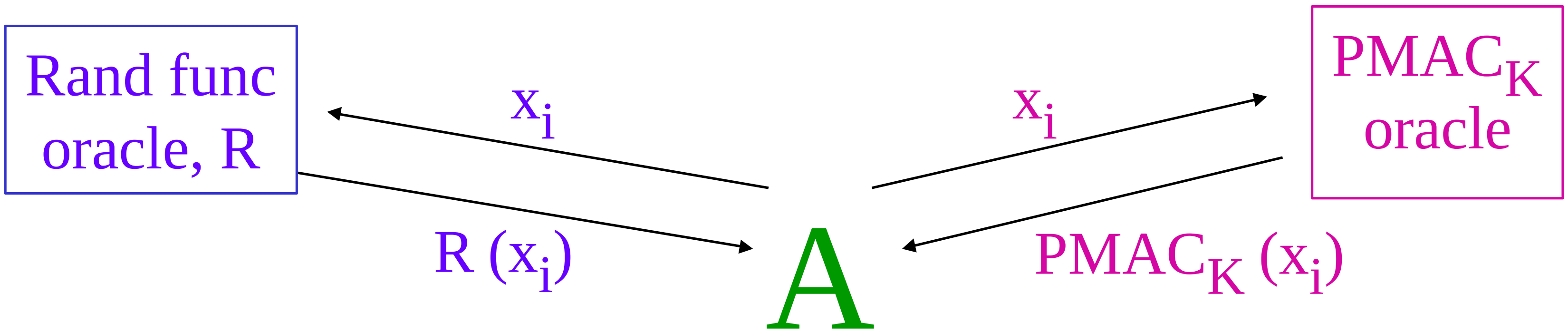
[Goldreich, Goldwasser, Micali]
[Luby, Rackoff]
[Bellare, Kilian, Rogaway]



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

PMAC’s Security
Security as a VIL PRF

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



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

PMAC Theorem

Suppose ∃ an adversary **A**
that breaks **PMAC-E** with:
time = t
total-num-of-blocks = σ
adv = $\mathbf{Adv}^{\mathbf{prf}}(\mathbf{A})$ $\sigma^2 / 2^n$

Then ∃ an adversary **B**
that breaks block cipher **E** with:
time ≈ t
num-of-queries ≈ σ
$\mathbf{Adv}^{\mathbf{prp}}(\mathbf{B}) \approx \mathbf{Adv}^{\mathbf{prf}}(\mathbf{A}) - \sigma^2 / 2^{n-1}$

(To wrap up,
it is a standard result that any τ-bit-output PRF
can be used as a MAC, where the forging probability
will be at most $\mathbf{Adv}^{\mathbf{prf}}(\mathbf{A}) + 2^{-\tau}$)

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

	Domain	PRF	MAC length	Parallelizable	#calls	Key bits	/ blk overhead
CBCMAC	$(\{0,1\}^n)^m$	✓	τ		$\|M\| / n$	k	1 xor
XCBC [BR 00]	$\{0,1\}^*$	✓	τ		$\lceil \|M\| / n \rceil$	k + 2n	1 xor
XECB-MAC (3 versions) [GD 00,01]	$\{0,1\}^*$		$\tau+\nu$	✓	$\lceil \|M\| / n \rceil +$ varies	varies	1 xor 2 add
PMAC [BR 00,01]	$\{0,1\}^*$	✓	τ	✓	$\lceil \|M\| / n \rceil$	k	3 xor

For More Information

- PMAC web page → www.cs.ucdavis.edu/~rogaway
  Contains FAQ, papers, reference code, test vectors...
- Feel free to call or send email
- Or grab me now!