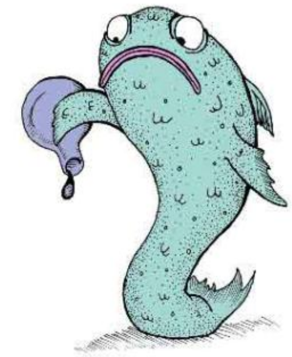
The OCB Authenticated-Encryption Algorithm
draft-krovetz-ocb-03

Ted Krovetz
California State University, Sacramento, USA

Phillip Rogaway
University of California, Davis, USA

IETF 83 – Paris, France – CFRG – 11:20-12:20 in 212/213 – March 30, 2012

Why am I here?



- I've not attended standards meetings
- Underused academic work of mine, 2001-11 (OCB – draft-krovetz-ocb-03)
- David McGrew explained that someone must present OCB for the RG sponsor it.
- Not clear it matters if the RFC is sponsored, but seems more consistent with the maturity and degree of review.

What is authenticated-encryption (AE)

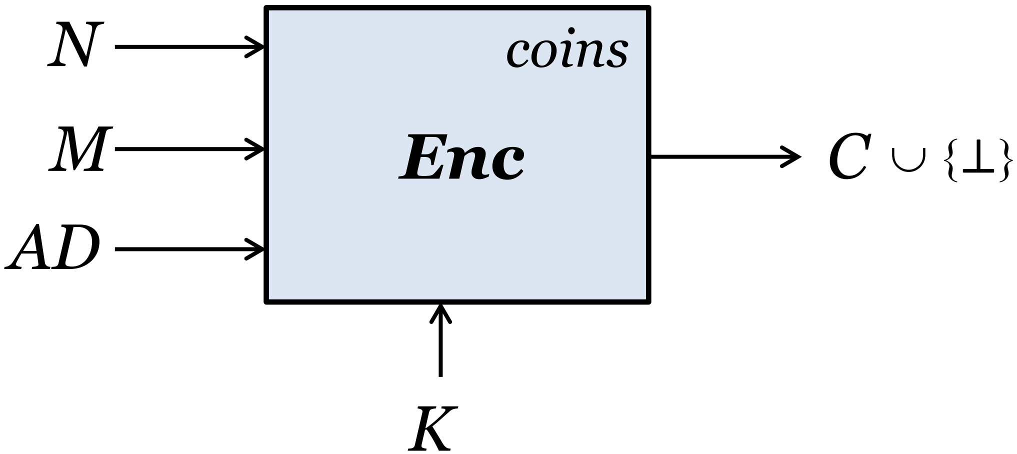
Symmetric encryption that simultaneously
provides privacy **and** authenticity

Historically: Encryption **only** for **privacy** – IND-CPA
Separate tool, a **MAC**, for **authenticity**

Why AE?

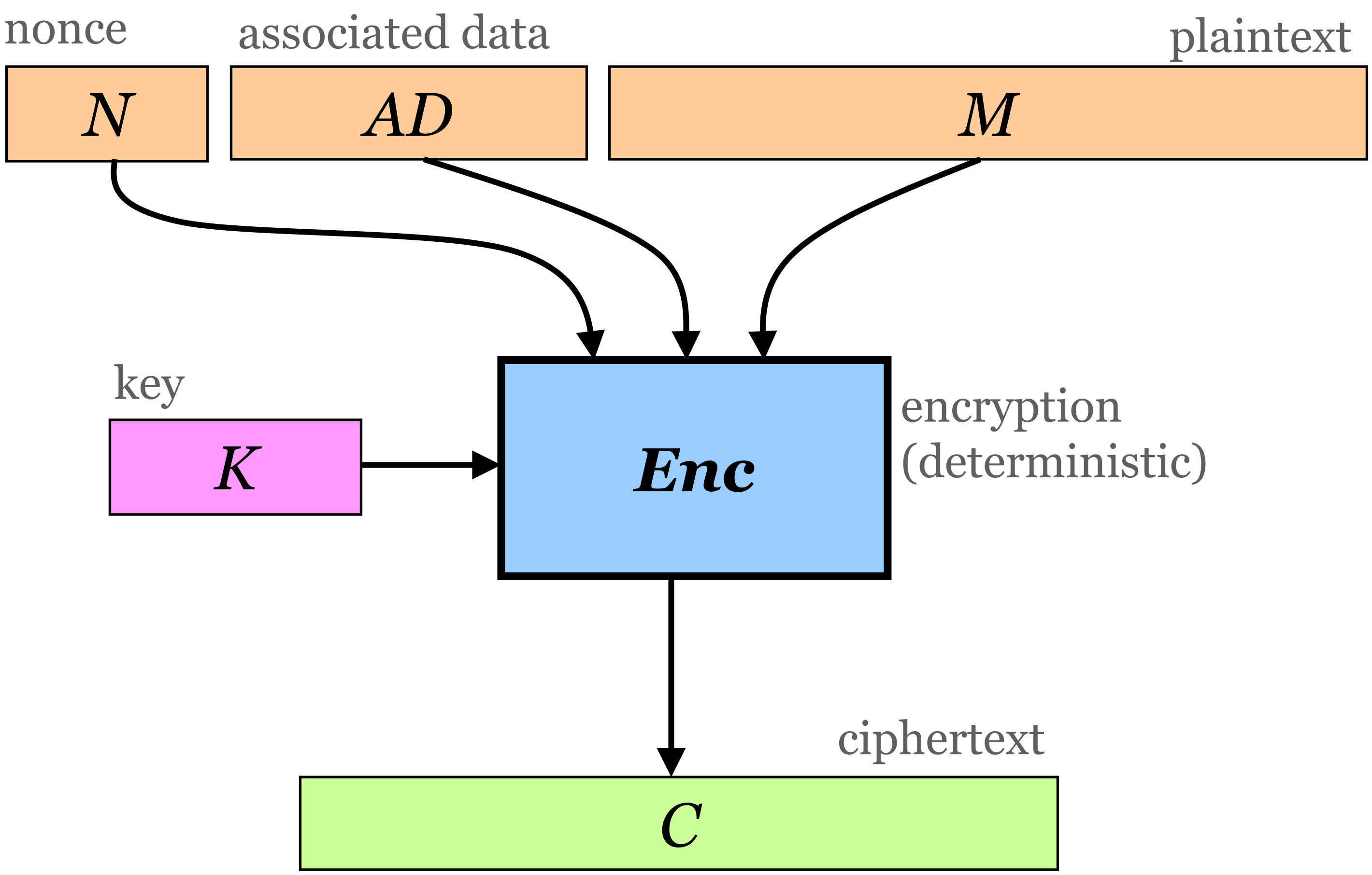
- Simper-to-correctly use
- Efficiency improvements possible

AE Scheme

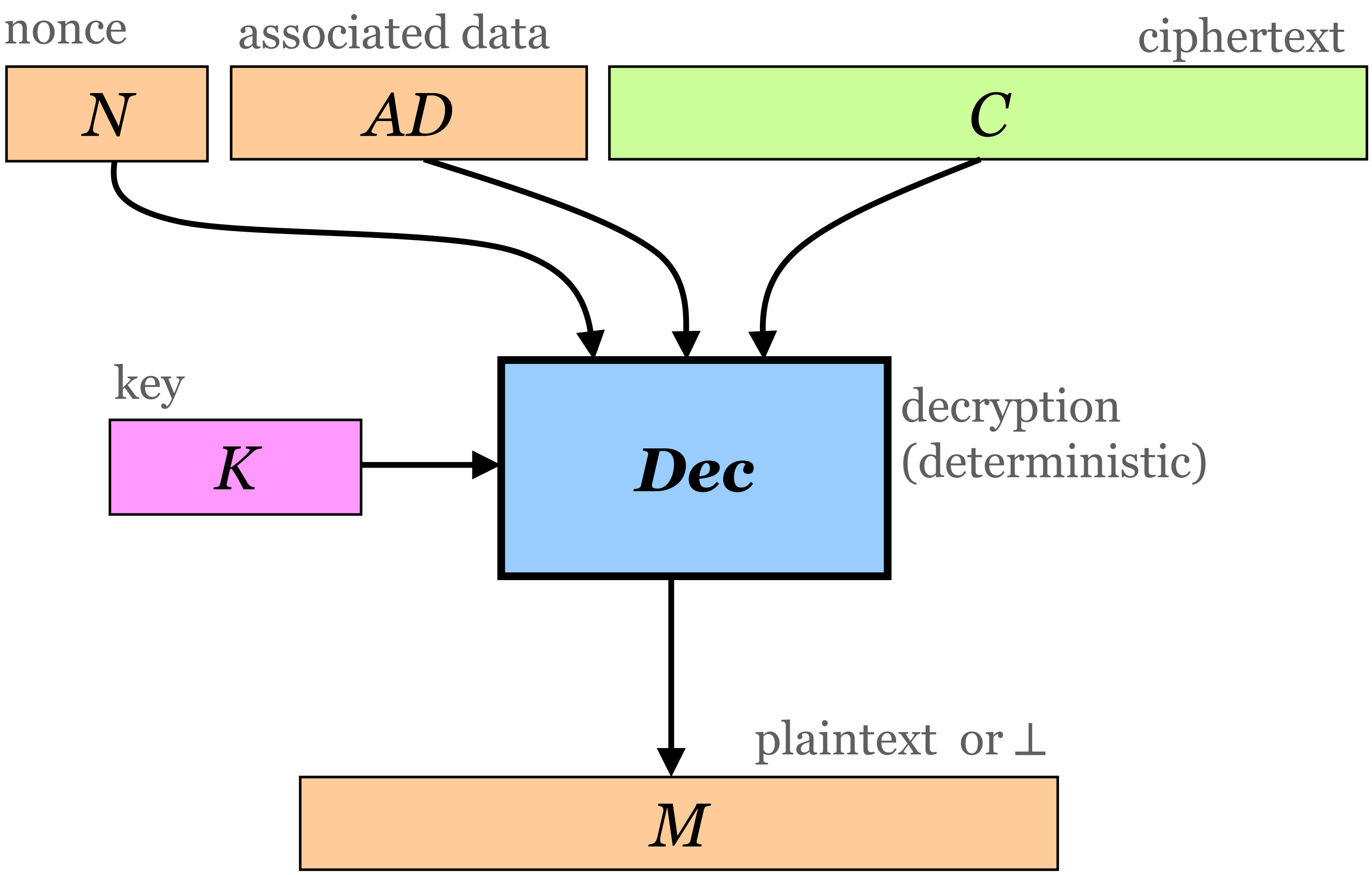


- Move the coins "out"
- Make "nonce" sufficient
- Build in authenticity
- Add "associated data"

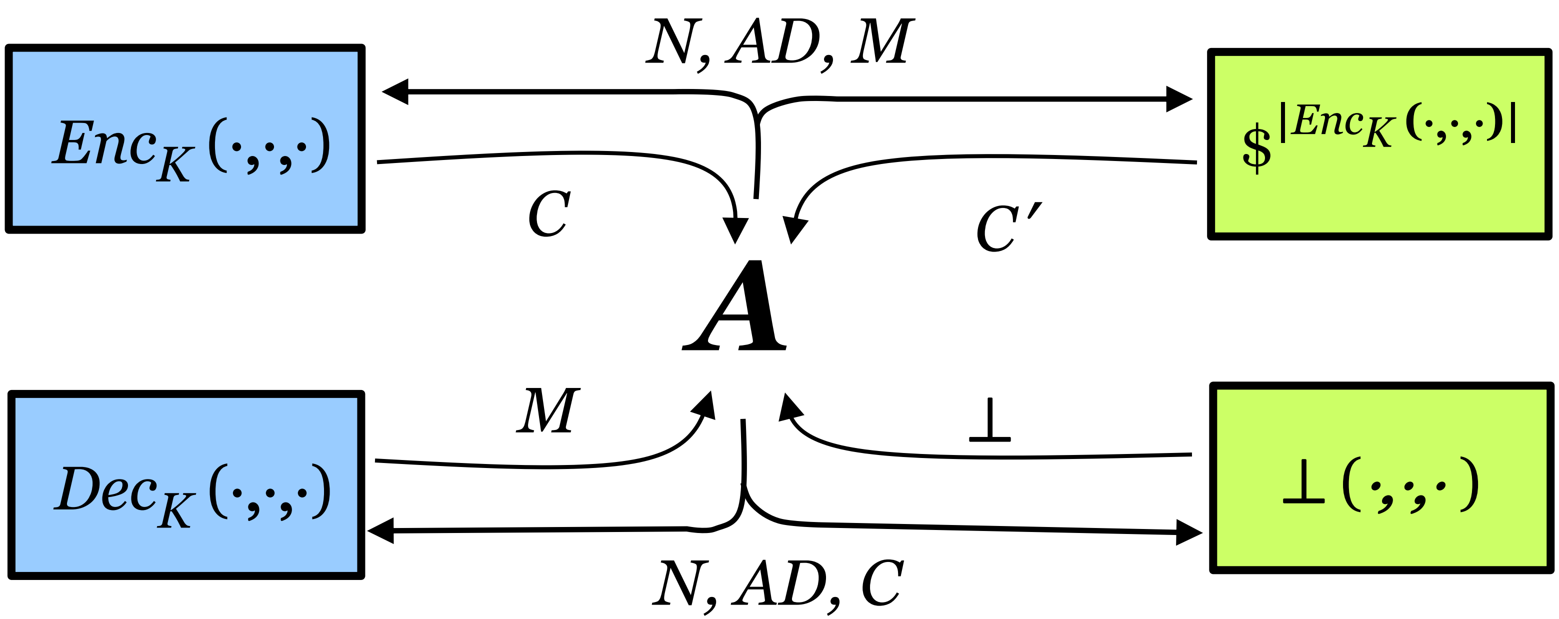
AE Scheme


nonce
associated data
plaintext
N
AD
M
key
K
Enc
encryption
(deterministic)
ciphertext
C

AE Scheme


nonce
associated data
ciphertext
N
AD
C
key
K
Dec
decryption
(deterministic)
plaintext or ⊥
M

AE Security



$$\mathbf{Adv}^{\mathrm{ae}}_{\Pi}(\boldsymbol{A}) = \Pr[\boldsymbol{A}^{Enc_K\ Dec_K} \rightarrow 1] - \Pr[\boldsymbol{A}^{\$\ \perp} \rightarrow 1]$$

A may not repeat an encryption query or ask a decryption query (*N*, *AD*, *C*) where *C* was previously returned by an (*N*, *AD*, ·) encryption query.

Approaches to achieving AE

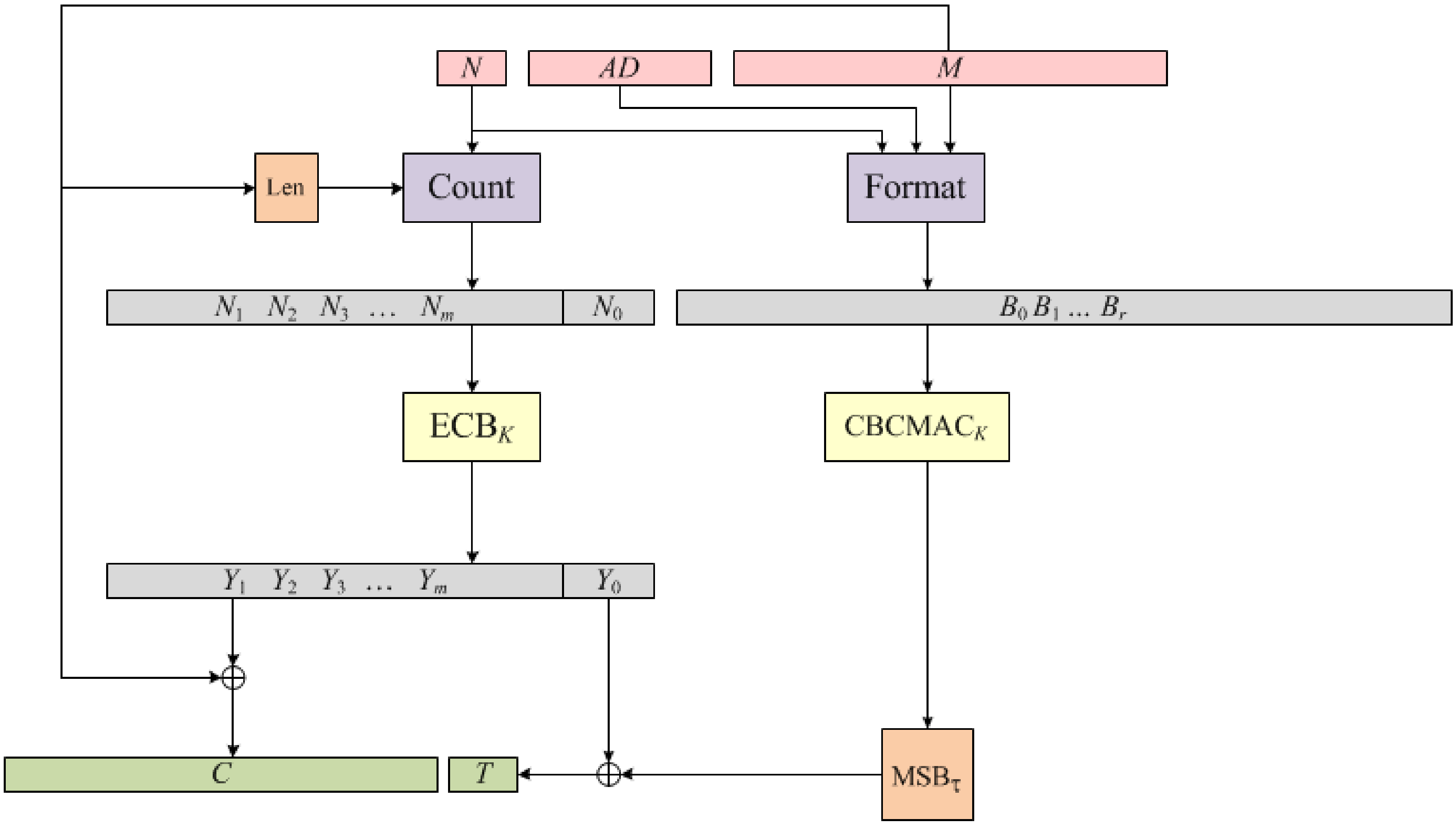
Confusion/diffusion: one atomic primitive
* **Helix**, **SOBER**, ...

Composed: ind$-secure symmetric encryption + PRF
* **EtM**, **MtE**, **E&M** [folklore; BN 2000]
* **CCM** [WHF 2002; NIST 800-38c]
* **GCM** [MV 2004; NIST 800-38D]

Integrated: blend privacy/authenticity parts
* **OCB** [RBBK 2001, R2004, KR 2011]; following [Jutla 2001]

Whiting, Housley, Ferguson 2002
NIST SP 800-38C
RFC 3610, 4309, 5084

CCM Mode



Whiting, Housley, Ferguson 2002
NIST SP 800-38C
RFC 3610, 4309, 5084, 5116

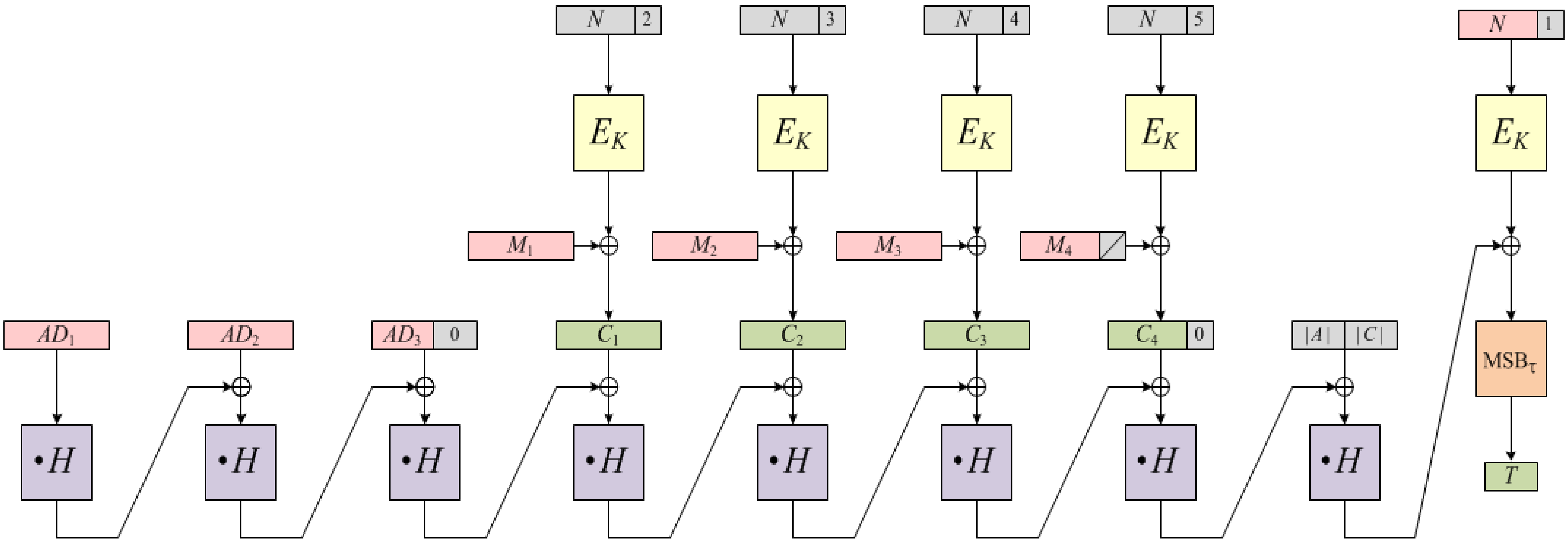
CCM Mode

- Provably secure AE if E is a good PRP
- Widely used, standardized (eg, in 802.11)
- About $2m$ blockcipher calls
- Half of them non-parallelizable
- Not "online" — need to know m in advance

McGrew, Viega 2004
NIST SP 800-38D
RFC 4106, 5084,
5116, 5288, 5647

GCM Mode

with 96-bit nonce



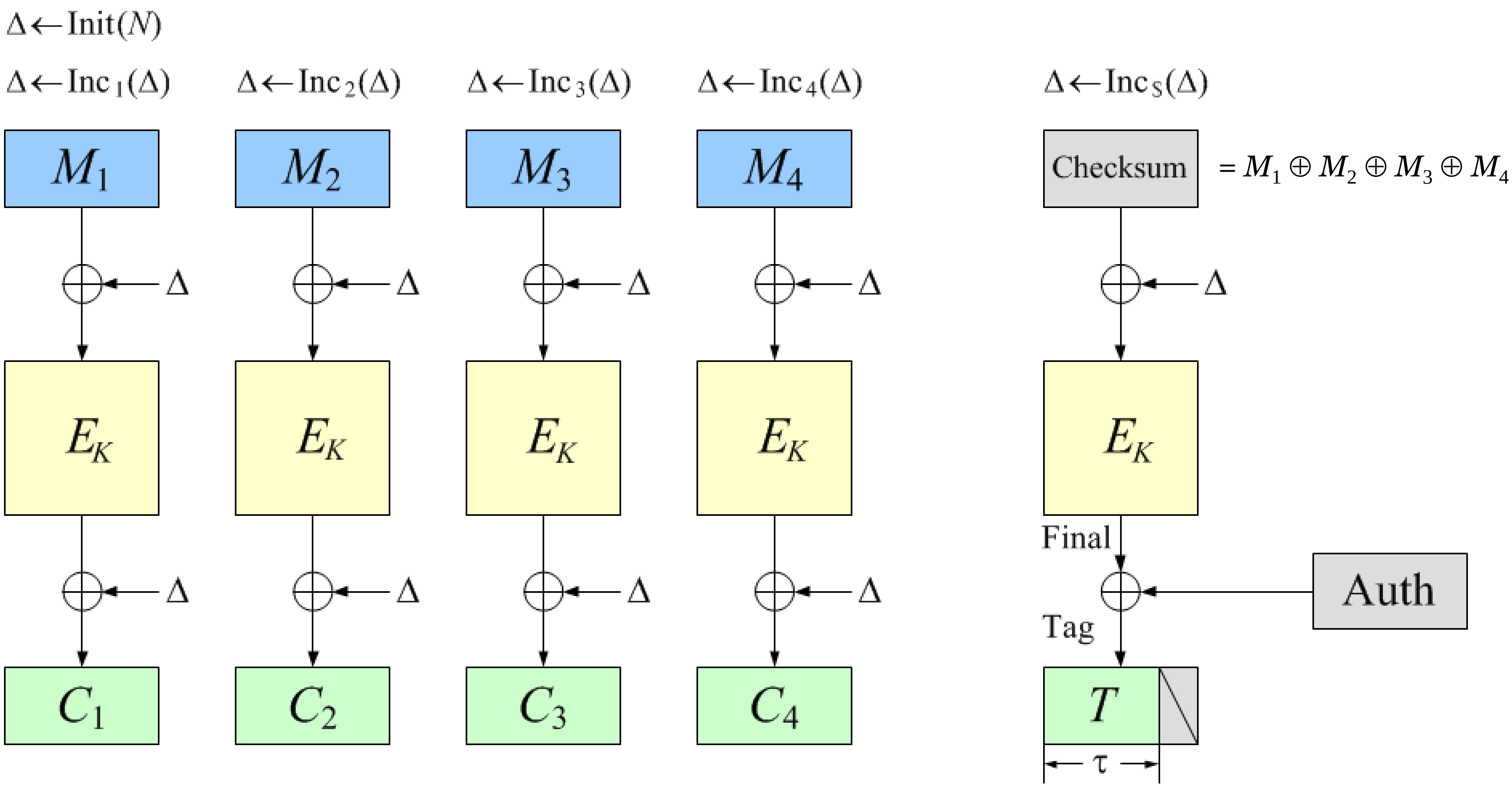
McGrew, Viega 2004
NIST SP 800-38D
RFC 4106, 5084, 5116, 5288, 5647

GCM Mode

- Provably secure AE if E is a good PRP
- Poor bound if truncate tag too much (Ferguson, 2005) (don't truncate <96 bits)
- Published proof is buggy [Iwata, 2012]
- Used in: IPSec, P1619.1, TLS, …
- About m blockcipher calls, all of them parallelizable
- Efficient implementation in HW
- Efficient implementation in SW with preprocessing & tables, or HW support
- Timing attacks may be possible

OCB Mode

[RBBK01, R04, KR10]
following [J01,GD01,LR02]



OCB, in full

$\textbf{algorithm}\ \mathcal{E}_K^{N,A}(M)$
$\textbf{if}\ |N| \geq 128\ \textbf{then return}\ \textsc{Invalid}$
$M_1 \cdots M_m\ M_* \leftarrow M$ where each
$\quad |M_i| = 128$ and $|M_*| < 128$
$\text{Checksum} \leftarrow 0^{128}; \quad C \leftarrow \varepsilon$
$\text{Nonce} \leftarrow 0^{127-|N|}\ 1\ N$
$\text{Top} \leftarrow \text{Nonce} \wedge 1^{122}\, 0^6$
$\text{Bottom} \leftarrow \text{Nonce} \wedge 0^{122}\, 1^6$
$\text{Ktop} \leftarrow E_K(\text{Top})$
$\text{Stretch} \leftarrow \text{Ktop} \parallel (\text{Ktop} \oplus (\text{Ktop} \ll 8))$
$\Delta \leftarrow (\text{Stretch} \ll \text{Bottom})[1..128]$
$\textbf{for}\ i \leftarrow 1\ \textbf{to}\ m\ \textbf{do}$
$\quad \Delta \leftarrow \Delta \oplus L[\text{ntz}(i)]$
$\quad C \stackrel{\|}{\leftarrow} E_K(M_i \oplus \Delta) \oplus \Delta$
$\quad \text{Checksum} \leftarrow \text{Checksum} \oplus M_i$
$\textbf{if}\ M_* \neq \varepsilon\ \textbf{then}$
$\quad \Delta \leftarrow \Delta \oplus L_*$
$\quad \text{Pad} \leftarrow E_K(\Delta)$
$\quad C \stackrel{\|}{\leftarrow} M_* \oplus \text{Pad}[1..|M_*|]$
$\quad \text{Checksum} \leftarrow \text{Checksum} \oplus M_*\, 10^*$
$\Delta \leftarrow \Delta \oplus L_\$$
$\text{Final} \leftarrow E_K(\text{Checksum} \oplus \Delta)$
$\text{Auth} \leftarrow \text{Hash}_K(A)$
$\text{Tag} \leftarrow \text{Final} \oplus \text{Auth}$
$T \leftarrow \text{Tag}[1..\tau]$
$\textbf{return}\ C \parallel T$

$\textbf{algorithm}\ \mathcal{D}_K^{N,A}(\mathcal{C})$
$\textbf{if}\ |N| \geq 128\ \textbf{or}\ |\mathcal{C}| < \tau\ \textbf{then return}\ \textsc{Invalid}$
$C_1 \cdots C_m\ C_*\ T \leftarrow \mathcal{C}$ where each
$\quad |C_i| = 128$ and $|C_*| < 128$ and $|T| = \tau$
$\text{Checksum} \leftarrow 0^{128}; \quad M \leftarrow \varepsilon$
$\text{Nonce} \leftarrow 0^{127-|N|}\ 1\ N$
$\text{Top} \leftarrow \text{Nonce} \wedge 1^{122}\, 0^6$
$\text{Bottom} \leftarrow \text{Nonce} \wedge 0^{122}\, 1^6$
$\text{Ktop} \leftarrow E_K(\text{Top})$
$\text{Stretch} \leftarrow \text{Ktop} \parallel (\text{Ktop} \oplus (\text{Ktop} \ll 8))$
$\Delta \leftarrow (\text{Stretch} \ll \text{Bottom})[1..128]$
$\textbf{for}\ i \leftarrow 1\ \textbf{to}\ m\ \textbf{do}$
$\quad \Delta \leftarrow \Delta \oplus L[\text{ntz}(i)]$
$\quad M \stackrel{\|}{\leftarrow} D_K(C_i \oplus \Delta) \oplus \Delta$
$\quad \text{Checksum} \leftarrow \text{Checksum} \oplus M_i$
$\textbf{if}\ C_* \neq \varepsilon\ \textbf{then}$
$\quad \Delta \leftarrow \Delta \oplus L_*$
$\quad \text{Pad} \leftarrow E_K(\Delta)$
$\quad M \stackrel{\|}{\leftarrow} M_* \leftarrow C_* \oplus \text{Pad}[1..|C_*|])$
$\quad \text{Checksum} \leftarrow \text{Checksum} \oplus M_*\, 10^*$
$\Delta \leftarrow \Delta \oplus L_\$$
$\text{Final} \leftarrow E_K(\text{Checksum} \oplus \Delta)$
$\text{Auth} \leftarrow \text{Hash}_K(A)$
$\text{Tag} \leftarrow \text{Final} \oplus \text{Auth}$
$T' \leftarrow \text{Tag}[1..\tau]$
$\textbf{if}\ T = T'\ \textbf{then return}\ M$
$\qquad \textbf{else return}\ \textsc{Invalid}$

$\textbf{algorithm}\ \text{Setup}(K)$
$L_* \leftarrow E_K(0^{128})$
$L_\$ \leftarrow \text{double}(L_*)$
$L[0] \leftarrow \text{double}(L_\$)$
$\textbf{for}\ i \leftarrow 1, 2, \cdots\ \textbf{do}\ L[i] \leftarrow \text{double}(L[i-1])$
$\textbf{return}$

$\textbf{algorithm}\ \text{double}(X)$
$\textbf{return}\ (X \ll 1) \oplus (\text{msb}(X) \cdot 135)$

$\textbf{algorithm}\ \text{Hash}_K(A)$
$A_1 \cdots A_m\ A_* \leftarrow A$ where each
$\quad |A_i| = 128$ and $|A_*| < 128$
$\text{Sum} \leftarrow 0^{128}$
$\Delta \leftarrow 0^{128}$
$\textbf{for}\ i \leftarrow 1\ \textbf{to}\ m\ \textbf{do}$
$\quad \Delta \leftarrow \Delta \oplus L[\text{ntz}(i)]$
$\quad \text{Sum} \leftarrow \text{Sum} \oplus E_K(A_i \oplus \Delta)$
$\textbf{if}\ A_* \neq \varepsilon\ \textbf{then}$
$\quad \Delta \leftarrow \Delta \oplus L_*$
$\quad \text{Sum} \leftarrow \text{Sum} \oplus E_K(A_*\, 10^* \oplus \Delta)$
$\textbf{return}\ \text{Sum}$

OCB Mode

- Provably secure AE (if blockcipher a strong PRP)
- Good bound (no problem to truncate tag)
- Most software-efficient AE scheme
- No timing attacks (if underlying blockcipher immune)
- Comprehensive literature

  RBBK01 – *CCS 2001* – A blockcipher mode of operation for efficient AE

  Ro02 – *CCS 2002* – Authenticated-encryption with associated data

  Ro04 – *Asiacrypt 2004* – Efficient instantiations of TBCs and refinements to OCB

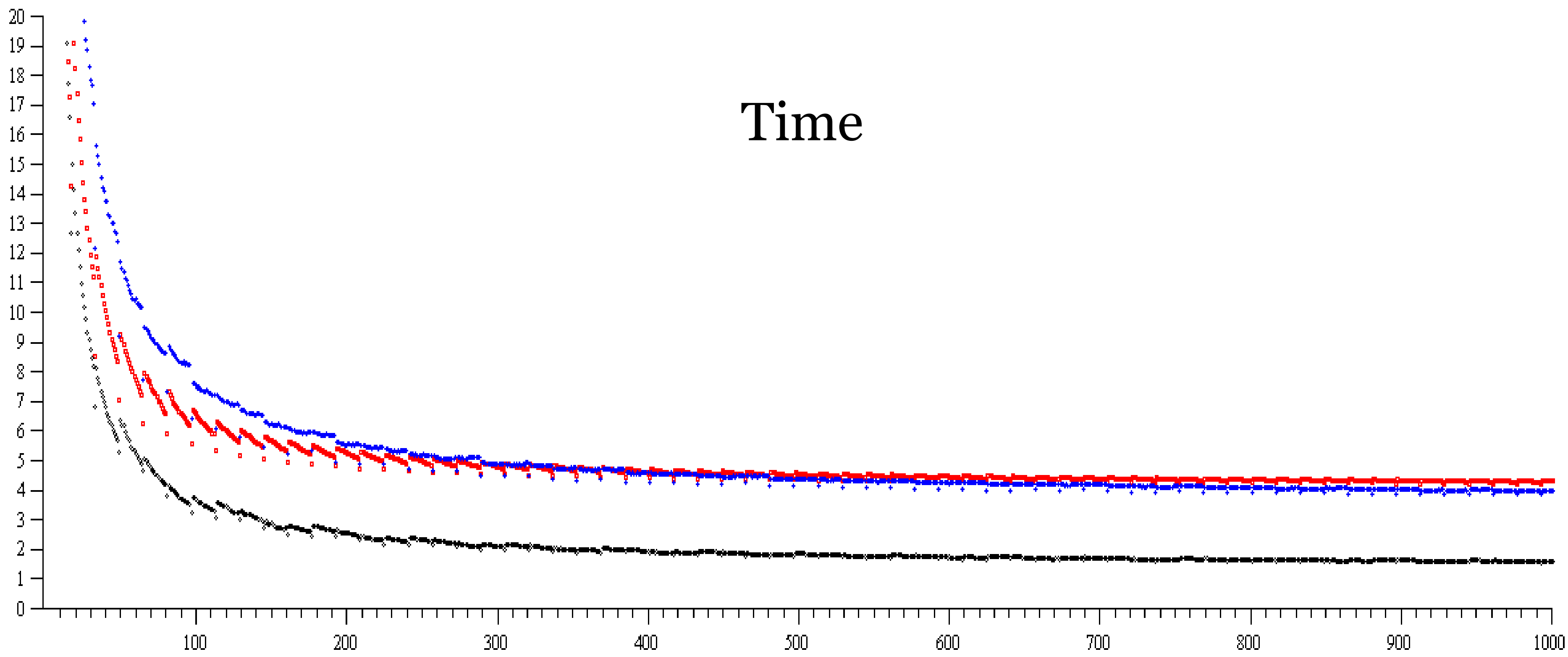
  KR11 – *FSE 2011* – The software performance of AE modes

- Standardized in ISO/IEC 19772
- Not widely used
- Complies with RFC 5116

[KR11]

Software Performance
Intel Core x86 i5-650 – "**Clarkdale**"
64-bit OS, using AES/GCM NIs

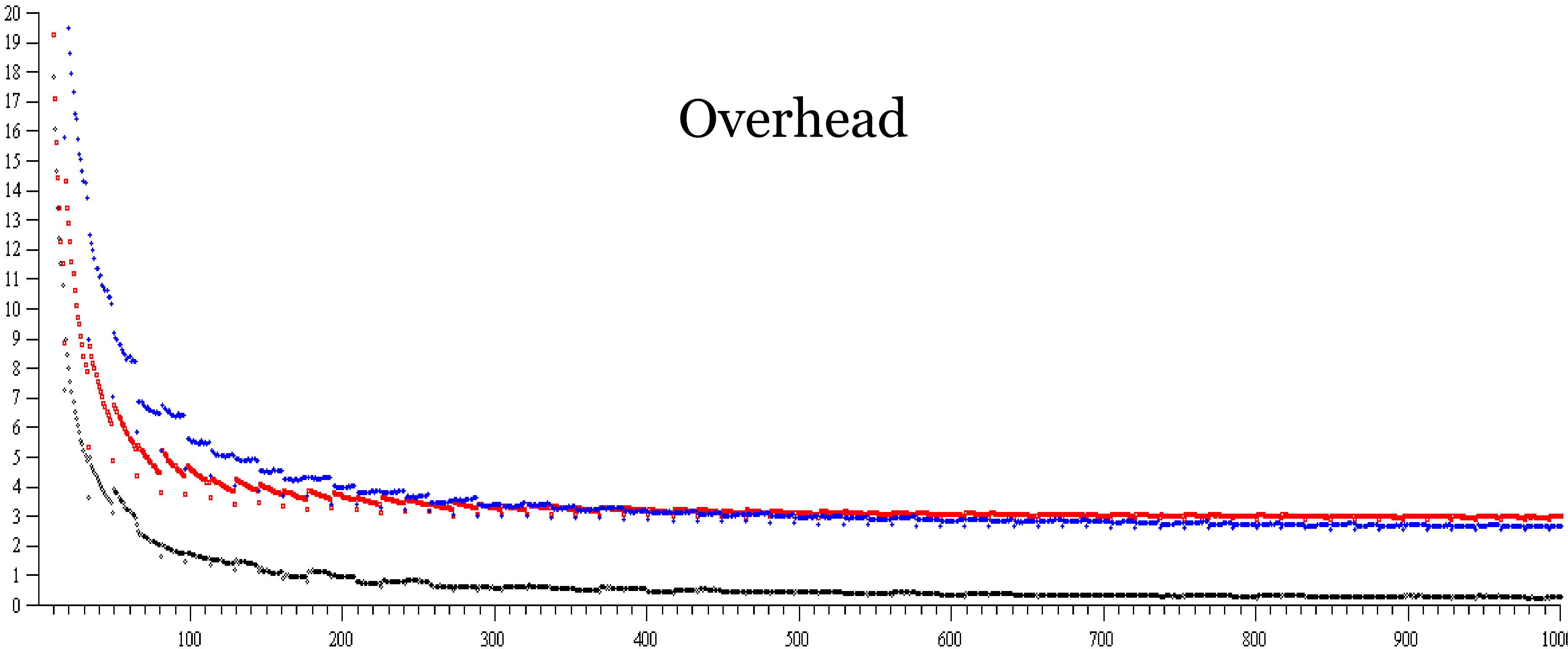
Mode	Peak cpb
CCM	4.17
GCM	3.73
OCB	1.48
CTR	1.27



[KR11]

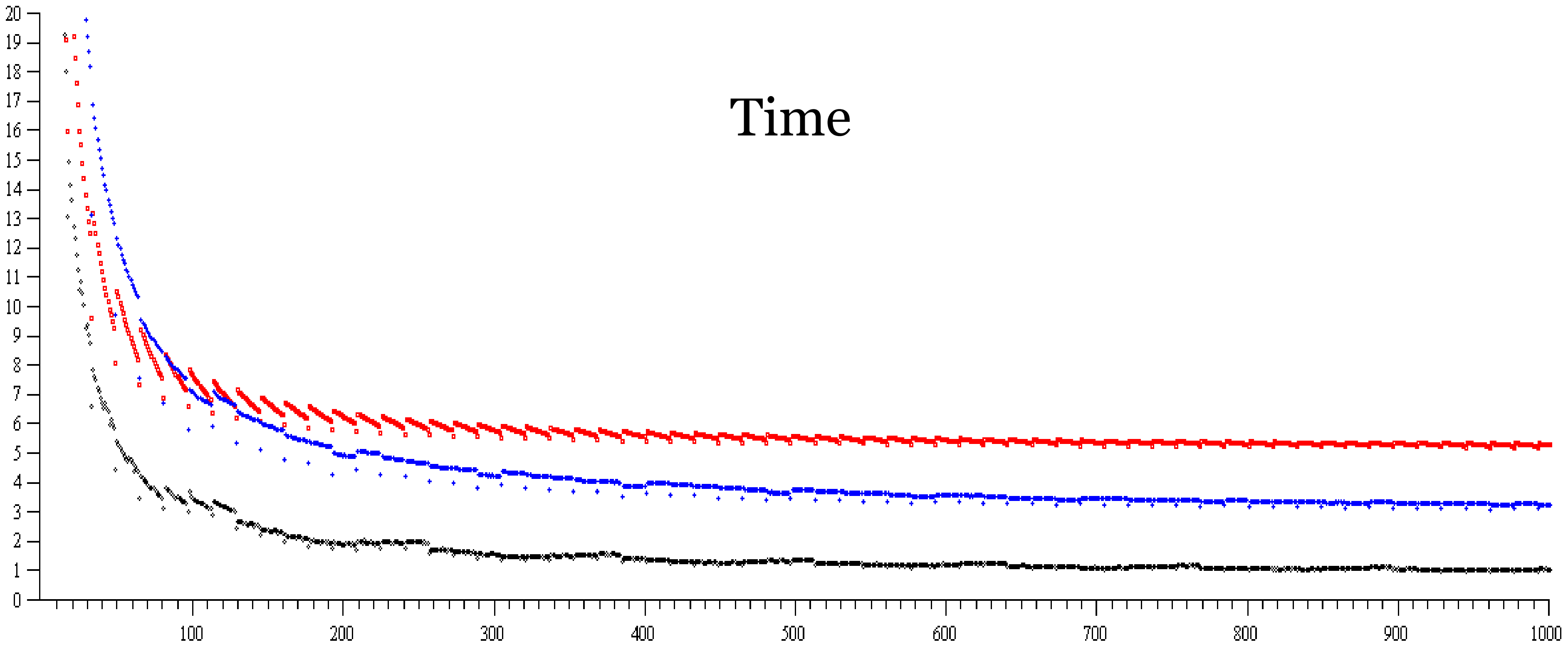
Software Performance
Intel Core x86 i5-650 – "**Clarkdale**"
64-bit OS, using AES/GCM NIs

Mode	Peak cpb
CCM	2.09
GCM	2.46
OCB	0.21



Software Performance
Intel Core x86 i7 – "**Sandy Bridge**"
64-bit OS, using AES/GCM NIs

Mode	Peak cpb
CCM	5.14
GCM	2.95
OCB	0.87



Key Differences

	Increment	AD	Cipher calls	Stalls
OCB1 (2001)	Table	No	m+2	2
OCB2 (2004)	shift, xor	Yes	m+2	2
OCB3 (20011)	Table	Yes	m+1.02	0

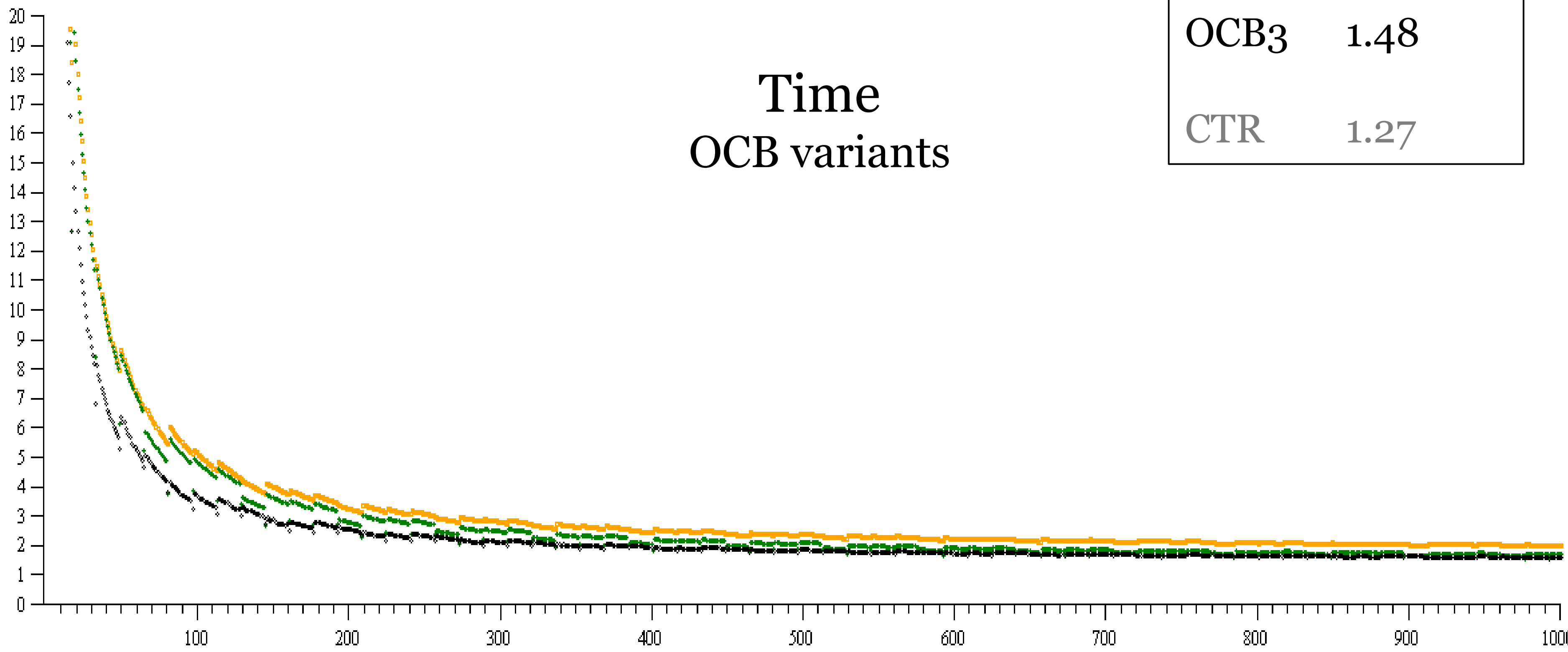
Non-Differences

Bounds, ciphertext length, parallelizability, timing-attack resistance.

Software Performance
Intel Core x86 i5-650 – "**Clarkdale**"
64-bit OS, using AES/GCM NIs

Mode	Peak cpb
CCM	4.17
GCM	3.73
OCB1	1.48
OCB2	1.80
OCB3	1.48
CTR	1.27

Time
OCB variants



Final Comments

- Very mature algorithm. No further refinements
- Significant advantages to CCM and GCM
  - software speed (CCM, GCM)
  - parallelizability (CCM)
  - key agility (GCM)
  - online (CCM)
  - tag truncation (GCM)
- Trying to get all parties to agree to free licensing for all SW (or at least all open-source SW)
- www.cs.ucdavis.edu/~rogaway/ocb
  - optimized C code
  - performance graphs
  - …

Questions?