Authenticated ciphers

D. J. Bernstein
University of Illinois at Chicago

Joint work with:
Tanja Lange
Technische Universiteit Eindhoven

Advertisement: SHARCS 2012
(Special-Purpose Hardware for
Attacking Cryptographic Systems)
is right before FSE+SHA-3.
2012.01.23 deadline
to submit extended abstracts.
`2012.sharcs.org`

Multiple-year SHA-3 competition has produced a natural focus for security analysis and performance analysis.

Community shares an interest in selecting best hash as SHA-3. Intensive analysis of candidates: hash conferences, hash workshops, active SHA-3 mailing list, etc.

Would have been harder to absorb same work spread over more conferences, more time. Focus improves community's understanding and confidence.

This is a familiar pattern.

June 1998: AES block-cipher submissions from 50 people $\Rightarrow$ community focus.

April 2005: eSTREAM stream-cipher submissions from 100 people $\Rightarrow$ community focus.

October 2008: SHA-3 hash-function submissions from 200 people $\Rightarrow$ community focus.

This is a familiar pattern.

June 1998: AES block-cipher submissions from 50 people $\Rightarrow$ community focus.

April 2005: eSTREAM stream-cipher submissions from 100 people $\Rightarrow$ community focus.

October 2008: SHA-3 hash-function submissions from 200 people $\Rightarrow$ community focus.

NESSIE was much less focused and ended up in more trouble: e.g., only two MAC submissions.

The next community focus

What's next after block ciphers, stream ciphers, hash functions?

Proposal: authenticated ciphers.

Basic security goal: two users start with a shared secret key; then want to protect messages against espionage and forgery.

The usual competition:
maximize security subject to performance constraints;
i.e.: maximize performance subject to security constraints.

"Isn't authenticated encryption done already?"

"Isn't authenticated encryption done already?"

FSE 2011 Krovetz–Rogaway cite EtM, RPC, IAPM, XCBC, OCB1, TAE, CCM, CWC, GCM, EAX, OCB2, CCFB, CHM, SIV, CIP, HBS, BTM; and propose OCB3.

Same paper reports
various timings for AES-GCM;
better timings for AES-OCB3,
"the fastest reported times for AE" (authenticated encryption);
within ϵ of AES.

"Isn't authenticated encryption done already?"

FSE 2011 Krovetz–Rogaway cite EtM, RPC, IAPM, XCBC, OCB1, TAE, CCM, CWC, GCM, EAX, OCB2, CCFB, CHM, SIV, CIP, HBS, BTM; and propose OCB3.

Same paper reports
various timings for AES-GCM;
better timings for AES-OCB3,
"the fastest reported times for AE" (authenticated encryption);
within ϵ of AES.

"That's the end! AES-OCB3!"

General themes of next several slides in this talk:

1. Is AES-OCB3 the best way to build an authenticated cipher? Many reasons to be skeptical.

General themes of next several slides in this talk:

1. Is AES-OCB3 the best way to build an authenticated cipher? Many reasons to be skeptical.

2. Examples of how earlier authenticated ciphers already beat AES-OCB3

General themes of next several slides in this talk:

1. Is AES-OCB3 the best way to build an authenticated cipher? Many reasons to be skeptical.

2. Examples of how earlier authenticated ciphers already beat AES-OCB3 . . . in some respects.

General themes of next several slides in this talk:

1. Is AES-OCB3 the best way to build an authenticated cipher? Many reasons to be skeptical.

2. Examples of how earlier authenticated ciphers already beat AES-OCB3 . . . in some respects.

Conclusion: No reason to think that existing work is optimal. Ample room for competition.

Changing the components

AES-GCM uses AES-CTR.

Many bits of AES input
thus end up as constants,
invalidating many differentials.

Can AES-GCM get away with
one or two fewer AES rounds
while still providing security
against differential attacks?

AES-OCB3 doesn't use CTR.
Can it be safely modified
to use some constant bits?

We know more about ciphers
in 2012 than we did in 1998.

Can we obtain better speeds
by replacing AES
with another block cipher?

We know more about ciphers
in 2012 than we did in 1998.

Can we obtain better speeds
by replacing AES
with another block cipher?

Can we obtain better speeds
by replacing AES-CTR
with another stream cipher?

We know more about ciphers in 2012 than we did in 1998.

Can we obtain better speeds by replacing AES with another block cipher?

Can we obtain better speeds by replacing AES-CTR with another stream cipher?

Yes, course! See eSTREAM.
Example, ARM Cortex A8:
28.9 cycles/byte for AES-OCB3.
25.4 cycles/byte for AES-CTR.
8.53 cycles/byte for Salsa20/20.
5.53 cycles/byte for Salsa20/12.

How expensive are MACs?

Can take any modern hash
(or design another one!),
plug into HMAC.

How expensive are MACs?

Can take any modern hash
(or design another one!),
plug into HMAC.

Are universal hashes better?

GCM's universal hash:
faster than HMAC in hardware
but much slower in software.

How expensive are MACs?

Can take any modern hash
(or design another one!),
plug into HMAC.

Are universal hashes better?

GCM's universal hash:
faster than HMAC in hardware
but much slower in software.

UMAC, VMAC, etc.:
faster than HMAC in software;
what about hardware?

(I'm doing a new PEMA design.)

Improving security

AES-GCM, AES-OCB3, etc.
advertise "provable security"
if AES is secure.

Improving security

AES-GCM, AES-OCB3, etc.
advertise “provable security”
if AES is secure.

But is AES actually secure?

Are the latest AES-cryptanalysis
papers reason for concern?
(I don’t think so,
but maybe you disagree.)

Improving security

AES-GCM, AES-OCB3, etc.
advertise "provable security"
if AES is secure.

But is AES actually secure?

Are the latest AES-cryptanalysis
papers reason for concern?
(I don't think so,
but maybe you disagree.)

Does efficiency force ciphers
to have a scary key schedule?

What happens to security
if there are many messages?

What happens to security
if there are many messages?

Usually the security proofs
become meaningless.

e.g. AES-OCB3 theorems allow
attack probability $6q^2/2^{128}$
after q blocks of AES input.
Is $q \approx 2^{60}$ so hard to imagine?

What happens to security
if there are many messages?

Usually the security proofs
become meaningless.

e.g. AES-OCB3 theorems allow
attack probability $6q^2/2^{128}$
after q blocks of AES input.
Is $q \approx 2^{60}$ so hard to imagine?

128-bit block size for AES
is beginning to look rather small.
Wouldn't it be more comfortable
to have 256-bit blocks?

What happens to security
if the attacker is lucky
and succeeds at one forgery?

AES-GCM answer: key recovery.
AES-OCB3 answer: ?

What happens to security
if the attacker is lucky
and succeeds at one forgery?

AES-GCM answer: key recovery.
AES-OCB3 answer: ?

Can limit the damage
by rejecting old nonces
and deriving key from nonce;
but this creates
speed problems for AES,
bigger speed problems for GCM.

What happens to security
if the attacker is lucky
and succeeds at one forgery?

AES-GCM answer: key recovery.
AES-OCB3 answer: ?

Can limit the damage
by rejecting old nonces
and deriving key from nonce;
but this creates
speed problems for AES,
bigger speed problems for GCM.

How important is this?
Do we need high key agility?

What about side-channel attacks?

Not a strong point for AES.
Not a strong point for GCM.

What about side-channel attacks?

Not a strong point for AES.
Not a strong point for GCM.

We understand reasonably well
how to design primitives
to avoid software side channels.

What about side-channel attacks?

Not a strong point for AES.
Not a strong point for GCM.

We understand reasonably well
how to design primitives
to avoid software side channels.

How can we design primitives
to reduce cost of
avoiding hardware side channels?
One approach (e.g., Keccak):
maximize bit-level parallelism,
minimize degree over $\mathbf{F}_2$.

Cost metrics

Is time the most important metric for performance?

Cost metrics

Is time the most important
metric for performance?

Does your cryptography fit
onto an RFID, or
into a small corner of a CPU?

What is the smallest area
for an authenticated cipher?

Cost metrics

Is time the most important metric for performance?

Does your cryptography fit onto an RFID, or into a small corner of a CPU?

What is the smallest area for an authenticated cipher?

For each A: How fast is an authenticated cipher that fits into area A?

Is AES-OCB3 actually faster than AES-GCM at *rejecting forgeries*?

Is AES-OCB3 actually faster than AES-GCM at *rejecting forgeries*?

AES-GCM rejects forgery with no decryption time.

AES-OCB3 is faster than AES-GCM, but is it faster than just the MAC in AES-GCM?

Is AES-OCB3 actually faster than AES-GCM at *rejecting forgeries*?

AES-GCM rejects forgery with no decryption time.

AES-OCB3 is faster than AES-GCM, but is it faster than just the MAC in AES-GCM?

Many other MACs are clearly faster than AES-OCB3.

Is AES-OCB3 actually faster than AES-GCM at *rejecting forgeries*?

AES-GCM rejects forgery with no decryption time.

AES-OCB3 is faster than AES-GCM, but is it faster than just the MAC in AES-GCM?

Many other MACs are clearly faster than AES-OCB3.

What is most important for performance of authenticated ciphers: normal traffic, or floods of forged traffic?

AES-OCB3 saves time in encryption and decryption by building a MAC that "accidentally" also computes a ciphertext.

AES-OCB3 saves time in encryption and decryption by building a MAC that "accidentally" also computes a ciphertext.

Can we build a cipher that "accidentally" also computes a fast MAC?

AES-OCB3 saves time in
encryption and decryption
by building a MAC
that "accidentally" also
computes a ciphertext.

Can we build a cipher
that "accidentally" also
computes a fast MAC?

Fast MAC of $m_0, m_1, \ldots$
typically looks like
$k_0 m_0 + k_1 m_1 + \cdots$.
Use $k_i m_i$ in computing
ith block of ciphertext?
Compare to 1996 Lucks HFF.

Another approach (e.g.,
Helix, Phelix, Keccak):
map state, plaintext block
to new state, ciphertext block.

Another approach (e.g.,
Helix, Phelix, Keccak):
map state, plaintext block
to new state, ciphertext block.

Complaint about Helix/Phelix:
state-recovery attack
if user repeats nonces
for different plaintexts
chosen by the attacker.

Another approach (e.g., Helix, Phelix, Keccak): map state, plaintext block to new state, ciphertext block.

Complaint about Helix/Phelix: state-recovery attack if user repeats nonces for different plaintexts chosen by the attacker.

Does this actually matter?
Fix 1: Give up, and stop feeding plaintext into state.
Fix 2: Use much larger blocks, much stronger map.

Isn't this fun?

Authenticated-cipher competition, like hash-function competition, is much more than mode competition.

Isn't this fun?

Authenticated-cipher competition, like hash-function competition, is much more than mode competition.

Want to build a better cipher? Combine with any standard MAC, submit to the competition.

Isn't this fun?

Authenticated-cipher competition, like hash-function competition, is much more than mode competition.

Want to build a better cipher? Combine with any standard MAC, submit to the competition.

Want to build a better MAC? Combine with AES, submit.

Isn't this fun?

Authenticated-cipher competition, like hash-function competition, is much more than mode competition.

Want to build a better cipher? Combine with any standard MAC, submit to the competition.

Want to build a better MAC? Combine with AES, submit.

Oh, you *are* a mode designer? Take standard components, submit.

Suggested timeline

Suggested timeline

First and most important:

Stop thinking about SHA-3.

Suggested timeline

First and most important:
Stop thinking about SHA-3.

Second half of 2012:
Public discussion of requirements.
Much easier than for hashing,
but still some real questions:
e.g., how much damage is okay
if nonces are reused?

Suggested timeline

First and most important:
Stop thinking about SHA-3.

Second half of 2012:
Public discussion of requirements.
Much easier than for hashing,
but still some real questions:
e.g., how much damage is okay
if nonces are reused?

Mid-2012: ECRYPT workshop.
Mid-2013: Submission deadline.
Mid-2014: Second round.
Mid-2015: Third round.

Most work is volunteered
by cryptographers+cryptanalysts
designing+attacking submissions.
(And we'll do benchmarking.)

Also need central committee
of experienced cryptologists
evaluating cryptanalyses and
selecting the best submissions.

Is this committee work
so much fun that
the right people
will volunteer for it? Maybe!

Competition name

"AES": "Authenticated Encryption Standard"?

Competition name

"AES": "Authenticated Encryption Standard"?

"AACS": "Advanced Authenticated Cipher Standard"?

Competition name

"AES": "Authenticated Encryption Standard"?

"AACS": "Advanced Authenticated Cipher Standard"?

"aSTREAM"?

Competition name

“AES”: “Authenticated Encryption Standard”?

“AACS”: “Advanced Authenticated Cipher Standard”?

“aSTREAM”? “YACC”?

Competition name

“AES”: “Authenticated Encryption Standard”?

“AACS”: “Advanced Authenticated Cipher Standard”?

“aSTREAM”? “YACC”? “AEAAEADAOIAIP”?

Competition name

“AES”: “Authenticated Encryption Standard”?

“AACS”: “Advanced Authenticated Cipher Standard”?

“aSTREAM”? “YACC”? “AEAAEADAOIAIP”?

Greg Rose has suggested “eSAFE”. Maybe “ECRYPT Secure Authenticated Fast Encryption”?

Competition name

“AES”: “Authenticated Encryption Standard”?

“AACS”: “Advanced Authenticated Cipher Standard”?

“aSTREAM”? “YACC”? “AEAAEADAOIAIP”?

Greg Rose has suggested “eSAFE”. Maybe “ECRYPT Secure Authenticated Fast Encryption”?

Orr: “FEAR”? “SHÆ-3”?