Some thoughts on

AEZ

v. 4

Viet Tung Hoang
Florida State University
USA

Ted Krovetz
Sacramento State
USA

Phillip Rogaway
Univ of California, Davis
USA

DIAC 2016
Nagoya, Japan
27 September 2016

With thanks to
Tetsu Iwata and
Shiho Moriai
for organizing this workshop!
Reluctant to give a talk

- **No changes** for Round-3
- **Talks @ DIAC 2014**
  - EUROCRYPT 2015
  - Several AE survey talks

But some reasons to do so

- My view of the mode has evolved
- Attacks @ ASIACRYPT 2015 and at FSE 2017
- AEZ is already in use (should it be?)
What kind of object is AEZ?

An Robust-AE scheme

Encrypt($K, N, A, M, \lambda$)

M

$E$

C

K

N

A

\lambda

arbitrary expansion

 Should look like a uniform $\lambda$-expanding injection (ind for $N, A, \lambda$) (forward + backward oracles)

A Generalized Blockcipher

Encipher($K, T, X$)

X

Y

K

T

arbitrary length

 Should look like a uniform permutation (ind for all $T$) (forward and backward oracles)

- AIL / VIL blockcipher
- Wide-block blockcipher
- An enciphering scheme
Robust-AE $\Leftrightarrow$ Generalized Blockcipher

The natural construction, “enciphering-based AE,” to make an RAE scheme from a generalized blockcipher
Unifying MRAE and Blockciphers
(1) Enciphering-based AE is a great way to achieve AE: very strong properties – not necessarily expensive
a) If \((M, A)\) tuples are known not to repeat, no nonce is needed
b)Nonce repetitions: privacy loss is limited to revealing repetitions in \((N, A, M)\) tuples, authenticity not damaged at all.
c) Any authenticator-length can be selected, achieving best-possible authenticity for this amount of stretch.
d) If there’s redundancy in plaintexts whose presence is verified on decryption, this augments authenticity
e) By last two properties: one can minimize length-expansion for bandwidth-constrained apps
f) If a decrypting party leaks some or all of a putative plaintext that was supposed to be squelched because of an authenticity-check failure, no problem.

(2) A generalized blockcipher is a great tool to have around
Conceptual simplicity and versatility: it’s an AE scheme, a PRG, a MAC, a PRF, a hash function, an entropy extractor, ...
AEZ

The first concrete construction of a generalized blockcipher

(although VIL wide-block blockciphers like EME₂ [Halevi; Halevi-Rogaway] come very close)
AEZ-tiny
FFX-like (Feistel)
[NIST SP 800-38G]
For strings < 32B
AES4-based

Structure of AEZ

AEZ-core
Builds on EME [HR04] and OTR [M14]
For strings ≥ 32B
AES4 & AES based.
AEZ-tiny

Not shown: each round depends on the hashed tweak

1 byte: 24 rounds
2 bytes: 16 rounds
3-15 bytes: 10 rounds
16-31 bytes: 8 rounds

Not shown: we correct for Feistel networks only generating even permutations
AEZ-core
What’s to Like?

Defense-in-depth and good speed

• The security target. Robust-AE is a very strong notion – implies almost all security properties one might hope for. Very few MRAE schemes remain in round-3.

• Wonderful versatility, ease of use – arbitrary-length keys, arbitrary ciphertext expansion, single-version scheme

• Amazing speed (in SW with AES-NI: peak 0.63 cpb Skylake; 1.0 AES-equivalents/block) considering the goal. Two-pass schemes are not inherently slow. HW performance looks respectable. Quick-rejection of invalid messages

• A proof for AEZ-core, to the birthday bound, in the prove-then-prune paradigm
What’s not to Like?

• Scheme is very complex. Anything-but-EZ in HW... and not easy for the SW, either. 58 lines of dense pseudocode.

• Aggressively optimized – not a conservative design.

• There are birthday key-recovery attacks:
  [Chaigneau, Gilbert 2017] ($2^{66.5}$ chosen plaintexts) (v.4), following [Fuhr, Leurent, Suder 2015] (v.3).
  Note: $2^{48}$ byte usage cap.

• A prove-then-prune proof does not, by itself, imply security; cryptanalysis is still needed. Should not be treated as a proof in the same sense as assuming some primitive is a PRP.

• Are the RAE \ MRAE properties (particularly the possibility of small ciphertext expansion) useful?
Seduced by speed?

“Don’t worry about speed. An RAE scheme / generalized blockcipher is very strong goal, and a scheme achieving it based on aesenc is going to need to be $2\times - 3\times$ slower, per block, than AES.”

“No! We can **match** AES’s speed in an RAE scheme. We can even get features like fast-reject and encipher-direction only processing, at the same time.” \(\rightarrow\) AEZ

“No!! We should be able to **exceed** AES speed in an aesenc-based MRAE scheme, and even an RAE scheme. What goes for AEGIS/Tioxin can be made to fly here, too.”
For in the future, I’d like to see

A generalized blockcipher / RAE scheme

that’s much simpler than AEZ, yet

Maybe a healthier alternative:

Enjoys (good old-fashioned)
provable security

(DJB “boring crypto”)

Is just as fast, or faster

Feels more conservative

Apparently has BBB security

Has at least an ideal-permutation
model proof of security, with good bounds

But, for now: AEZ is the best there is for this
degree of versatility and defense in depth.
AEZ (v4)

100 \textbf{algorithm} Encrypt(K, N, A, \tau, M) \quad \text{// AEZ authenticated encryption}
101 \quad X \leftarrow M \parallel 0^\tau; (A_1, \ldots, A_a) \leftarrow A
102 \quad T \leftarrow ([\tau]_{128}, N, A_1, \ldots, A_a)
103 \quad \text{if } M = \varepsilon \text{ then return } \text{AEZ-prf}(K, T, \tau)
104 \quad \text{return } \text{Encipher}(K, T, X)

110 \textbf{algorithm} Decrypt(K, N, A, \tau, C') \quad \text{// AEZ authenticated decryption}
111 \quad (A_1, \ldots, A_a) \leftarrow A; \quad T \leftarrow ([\tau]_{128}, N, A_1, \ldots, A_a)
112 \quad \text{if } |C| < \tau \text{ then return } \bot
113 \quad \text{if } |C| = \tau \text{ then if } C = \text{AEZ-prf}(K, T, \tau) \text{ then return } \varepsilon \text{ else return } \bot \quad \text{fi}
114 \quad X \leftarrow \text{Decipher}(K, T, C')
115 \quad M \parallel Z \leftarrow X \text{ where } |Z| = \tau
116 \quad \text{if } (Z = 0^\tau) \text{ then return } M \text{ else return } \bot

200 \textbf{algorithm} Encipher(K, T, X) \quad \text{// AEZ enciphering}
201 \quad \text{if } |X| < 256 \text{ then return } \text{Encipher-AEZ-tiny}(K, T, X)
202 \quad \text{if } |X| \geq 256 \text{ then return } \text{Encipher-AEZ-core}(K, T, X)
algorithm Encipher-AEZ-tiny($K, T, M$) // AEZ-tiny enciphering
\[
\begin{align*}
\mu & \leftarrow |M|; \quad n \leftarrow \mu/2; \quad \Delta \leftarrow \text{AEZ-hash}(K,T) \\
\text{if } \mu = 8 \text{ then } k & \leftarrow 24 \quad \text{else if } \mu = 16 \text{ then } k \leftarrow 16 \quad \text{else if } \mu < 128 \text{ then } k \leftarrow 10 \quad \text{else } k \leftarrow 8 \quad \text{fi} \\
L & \leftarrow M[1..n]; \quad R \leftarrow M[n+1..\mu]; \quad \text{if } \mu \geq 128 \text{ then } i \leftarrow 6 \quad \text{else } i \leftarrow 7 \quad \text{fi} \\
\text{for } j \leftarrow 0 \text{ to } k - 1 \text{ do } R' & \leftarrow L \oplus ((E_{\Delta \oplus R_{10}^* \oplus [j]_{128}}^0)(1..n)); \quad L \leftarrow R; \quad R \leftarrow R' \quad \text{od} \\
C & \leftarrow R \parallel L; \quad \text{if } \mu < 128 \text{ then } C \leftarrow C \oplus (E_{\Delta \oplus (C_{0*} \lor 10^*)}^{0,3})(1..n) \quad \text{fi} \\
\text{return } C
\end{align*}
\]

algorithm Encipher-AEZ-core($K, T, M$) // AEZ-core enciphering
\[
\begin{align*}
\Delta & \leftarrow \text{AEZ-hash}(K,T) \\
M_1M'_1 \cdots M_mM'_m & \leftarrow \text{ such that } |M_1| = \cdots = |M'_m| = |M_x| = |M_y| = 128 \text{ and } |M_{uv}| < 256 \\
d & \leftarrow |M_{uv}|; \quad \text{if } d \leq 127 \text{ then } M_u \leftarrow M_{uv}; \quad M_v \leftarrow \varepsilon \quad \text{else } M_u \leftarrow M_{uv}[1..128]; \quad M_v \leftarrow M_{uv}[129..|M_{uv}|] \quad \text{fi} \\
\text{for } i \leftarrow 1 \text{ to } m \text{ do } W_i & \leftarrow M_i \oplus E_{\Delta}^{1,i}(M'_i); \quad X_i \leftarrow M'_i \oplus E_{\Delta}^{0,0}(W_i) \quad \text{od} \\
\text{if } d = 0 \text{ then } X & \leftarrow X_1 \oplus \cdots \oplus X_m \oplus 0 \quad \text{else if } d \leq 127 \text{ then } X \leftarrow X_1 \oplus \cdots \oplus X_m \oplus E_{\Delta}^{0,4}(M_{u10^*}) \quad \text{fi} \\
\text{else } X & \leftarrow X_1 \oplus \cdots \oplus X_m \oplus E_{\Delta}^{0,4}(M_u) \oplus E_{\Delta}^{0,5}(M_{v10^*}) \quad \text{fi} \\
S_x & \leftarrow M_x \oplus \Delta \oplus X \oplus E_{\Delta}^{0,1}(M_y); \quad S_y \leftarrow M_y \oplus E_{\Delta}^{0,1}(S_x); \quad S \leftarrow S_x \oplus S_y \\
\text{for } i \leftarrow 1 \text{ to } m \text{ do } S' & \leftarrow E_{\Delta}^{2,i}(S); \quad Y_i \leftarrow W_i \oplus S'; \quad Z_i \leftarrow X_i \oplus S'; \quad C'_i \leftarrow Y_i \oplus E_{\Delta}^{0,0}(Z_i); \quad C_i \leftarrow Z_i \oplus E_{\Delta}^{1,i}(C'_i) \quad \text{od} \\
\text{if } d = 0 \text{ then } C_u & \leftarrow C_v \leftarrow \varepsilon; \quad Y \leftarrow Y_1 \oplus \cdots \oplus Y_m \oplus 0 \\
\text{else if } d \leq 127 \text{ then } C_u & \leftarrow M_u \oplus E_{\Delta}^{1,4}(S); \quad C_v \leftarrow \varepsilon; \quad Y \leftarrow Y_1 \oplus \cdots \oplus Y_m \oplus E_{\Delta}^{0,4}(C_{u10^*}) \quad \text{fi} \\
\text{else } C_u & \leftarrow M_u \oplus E_{\Delta}^{1,4}(S); \quad C_v \leftarrow M_v \oplus E_{\Delta}^{1,5}(S); \quad Y \leftarrow Y_1 \oplus \cdots \oplus Y_m \oplus E_{\Delta}^{0,4}(C_u) \oplus E_{\Delta}^{0,5}(C_{v10^*}) \quad \text{fi} \\
C_y & \leftarrow S_x \oplus E_{\Delta}^{1,2}(S_y); \quad C_x & \leftarrow S_y \oplus \Delta \oplus Y \oplus E_{\Delta}^{0,2}(C_y) \\
\text{return } C_1C'_1 \cdots C_mC'_m; \quad C_uC_vC_xC_y
\end{align*}
\]
algorithm AEZ-hash($K, T$) \quad // AXU hash. $T$ is a vector of strings
(T_1, \ldots, T_t) \leftarrow T
for $i \leftarrow 1$ to $t$
\quad $\ell \leftarrow \max(1, \lceil |T_i|/128 \rceil)$; \quad $j \leftarrow i + 2$; \quad $Z_1 \cdots Z_\ell \leftarrow T_i$ where $|Z_1| = \cdots = |Z_{\ell-1}| = 128$
\quad if $|Z_\ell| = 128$ then $\Delta_i \leftarrow E_{K}^{j,1}(Z_1) \oplus \cdots \oplus E_{K}^{j,\ell}(Z_\ell)$ fi
\quad if $|Z_\ell| < 128$ then $\Delta_i \leftarrow E_{K}^{j,1}(Z_1) \oplus \cdots \oplus E_{K}^{j,\ell-1}(Z_{\ell-1}) \oplus E_{K}^{j,0}(Z_{\ell}10^*)$ fi
\quad return $\Delta_1 \oplus \cdots \oplus \Delta_t \oplus 0$

algorithm AEZ-prf($K, T, \tau$) \quad // PRF used when $M = \varepsilon$
\quad $\Delta \leftarrow$ AEZ-hash($K, T$)
\quad return $(E_{K}^{-1,3}(\Delta) \parallel E_{K}^{-1,3}(\Delta \oplus [1]_{128}) \parallel E_{K}^{-1,3}(\Delta \oplus [2]_{128}) \parallel \cdots)[1..\tau]$

algorithm $E_{K}^{j,i}(X)$ \quad // Scaled-down TBC
\quad $I \parallel J \parallel L \leftarrow$ Extract($K$) where $|I| = |J| = |L| = 128$
\quad $K \leftarrow (0, I, J, L, I, J, L, I, J, L, I)$
\quad if $j = -1$ then $\Delta \leftarrow iJ$; \quad return AES10$_K(X \oplus \Delta)$ fi
\quad $k \leftarrow k_1 \leftarrow (0, J, I, L, 0)$; \quad $k_2 \leftarrow (0, L, I, J, L)$
\quad if $j = 0$ then $\Delta \leftarrow iI$; \quad return AES4$_k(X \oplus \Delta)$ fi
\quad if $1 \leq j \leq 2$ then $\Delta \leftarrow (2^{3+[j/(i-1)/8]} + (i - 1 \mod 8))I$; \quad return AES4$_{k,j}(X \oplus \Delta)$ fi
\quad if $j \geq 3$ and $i = 0$ then $\Delta \leftarrow 2^{j-3} \cdot L$; \quad return AES4$_k(X \oplus \Delta) \oplus \Delta$ fi
\quad if $j \geq 3$ and $i \geq 1$ then $\Delta \leftarrow 2^{j-3} \cdot L \oplus (2^{3+[j/(i-1)/8]} + (i - 1 \mod 8))J$; \quad return AES4$_k(X \oplus \Delta) \oplus \Delta$ fi

algorithm Extract($K$) \quad // Map key to subkeys
\quad if $|K| = 384$ then return $K$
\quad else return BLAKE2b($K$)