Deoxys

A Proposal for Beyond-Birthday Nonce-Misuse Authenticated Encryption

Thomas Peyrin

NTU - Singapore

CFRG - IETF 106 Meeting
Singapore - November 20, 2019
Outline

1. The Deoxys-BC TBC
2. The SCT Authenticated Encryption Mode
3. The ZMAC MAC and ZAE AEAD modes
What is a authenticated encryption?

Authenticated Encryption = Authentication + Encryption

Goal of authenticated encryption:

- avoid numerous issues that can arise when using separate authentication and encryption primitives (https://competitions.cr.yp.to/disasters.html)
- efficiency gain
- add feature of having authenticated-only data: Authenticated Encryption with Associated Data (AEAD)

Hot topic:

**What is beyond-birthday security?**

**Problem:** Most cipher modes have security bounds in $q^2/2^{128}$ for a 128-bit cipher (birthday bounds), $q$ is number of queries.

This means that after about $2^{64}$ data all security is lost.

**Examples:** OCB, AES–GCM, etc. provide only birthday bounds.

**Solution:** Beyond birthday modes provide beyond $2^n/2$ security, potentially up to full $2^n$.

This effectively avoids strong data constraints issues.
Security claims - a comparison of the nonce-respecting case

\[-\log_2(Adv)\]

\[-\log_2(\sigma)\]
What is nonce-misuse resistance?

**Problem:** Most cipher modes will have their security completely removed if the nonce is repeated just a single time.

This creates a lot of problems:

- if the nonce is generated randomly:
  need to make sure of proper randomness source

- if the nonce is a counter:
  need to constantly maintain a state

- other mechanisms required to make sure no repetition

**Examples:** OCB, AES-GCM, etc. are completely broken if the nonce is repeated just once (universal forgery and decryption)

**Solution:** Nonce-misuse resistant modes will maintain security even if the nonce is repeated, a really robust defence in depth feature.
Outline

1. The Deoxys-BC TBC
2. The SCT Authenticated Encryption Mode
3. The ZMAC MAC and ZAE AEAD modes
- DEOXYS-BC -

J. Jean, I. Nikolic, T. Peyrin

ASIACRYPT 2014

WINNER OF THE CAESAR COMPETITION
(Defense in depth portfolio)

Paper, Specifications, Results and Updates available at:
https://sites.google.com/view/deoxyscipher/
What is a (tweakable) block cipher?

**Block cipher (BC)**: a family of permutations parametrized by a secret key $K$. **Example**: AES

![Block cipher diagram](image)

**Tweakable block cipher (TBC)**: a family of permutations parametrized by a secret key $K$ and a tweak value $T$ [LRW02]. **Example**: Deoxys-BC

![Tweakable block cipher diagram](image)
The **Deoxys-BC** tweakable block ciphers

- **The round function is exactly** the AES round function
- **$h'$** is a simple permutation of the bytes positions
- The LFSRs can be clocked with a single XOR
- Constant additions to break symmetries (**RCON** from AES **KS**)
Deoxys-BC security and efficiency

2 versions: Deoxys-BC-256 and Deoxys-BC-384

128-bit tweakable block ciphers

- Deoxys-BC-256: 14 rounds and 256-bit tweakey
- Deoxys-BC-384: 16 rounds and 384-bit tweakey

- Security guarantees for differential/linear cryptanalysis (both single and related-key)
- A lot of 3rd party cryptanalysis since 2014, still comfortable security margin
- Reuses analysis already performed on AES
- Accepts 256-bit keys (post-quantum security)
- Very efficient software implementations (mostly AES round function), on Skylake (avx2) for fixed key:
  - 0.87 c/B for Deoxys-BC-256
  - 0.99 c/B for Deoxys-BC-384

- no patent
Outline

1. The Deoxys-BC TBC
2. The SCT Authenticated Encryption Mode
3. The ZMAC MAC and ZAE AEAD modes
- SCT AEAD mode -

T. Peyrin, Y. Seurin
CRYPTO 2016

WINNER OF THE CAESAR COMPETITION
(Defense in depth portfolio)

Deoxys-II = Deoxys-BC + SCT

Paper, Specifications, Results and Updates available at:
https://sites.google.com/view/deoxyscipher/
The **SCT** authenticated encryption mode

**SCT is:**
- a **simple** TBC-based AEAD mode
- 2 pass mode (because of nonce-misuse resistance)
- with **full** \(n\)-bit security in nonce-respecting scenario
- with \(n/2\)-bit security in nonce-misuse scenario (but linear degradation of security with the maximal number of nonce repetition, so in practice \(\sim n\)-bit security).
- Strong MRAE security notion.
- when instantiated with **Deoxys-BC**, it is very efficient
- no precomputation, almost no overhead for small messages
- fully parallel, inverse-free
- extra tweak input for other purposes (leakage resilience)
- provably secure (security proofs in the article)
- no patent
The SCT AEAD mode
Outline

1. The Deoxys-BC TBC
2. The SCT Authenticated Encryption Mode
3. The ZMAC MAC and ZAE AEAD modes
T. Iwata, K. Minematsu, T. Peyrin and Y. Seurin
CRYPTO 2017

Paper and Specifications available at:
The ZMAC MAC mode

**ZHASH**

- Input: \(X[1], X[2], X[m]\)
- Output: \(V, U\)

\[
\begin{align*}
X[1] & \\
X[2] & \\
X[m] & \\
\end{align*}
\]

\[
\begin{align*}
\tilde{E}_K^8 & \\
\tilde{E}_K^8 & \\
\tilde{E}_K^8 & \\
\end{align*}
\]

\[
\begin{align*}
L_\ell & \\
2 \cdot L_\ell & \\
2 \cdot L_r & \\
\end{align*}
\]

\[
\begin{align*}
0^t & \\
2 & \\
2 & \\
\end{align*}
\]

\[
\begin{align*}
2^m & \\
2^m & \\
2^m & \\
\end{align*}
\]

**ZFIN**

- Input: \(U\)
- Output: \(V\)

\[
\begin{align*}
U & \\
U & \\
U & \\
\end{align*}
\]

\[
\begin{align*}
\tilde{E}_K^i & \\
\tilde{E}_K^{i+1} & \\
\tilde{E}_K^{i+2} & \\
\tilde{E}_K^{i+3} & \\
\end{align*}
\]

\[
\begin{align*}
Y[1] & \\
Y[2] &
\end{align*}
\]
The ZAE AEAD mode

ZAE

\[ AD \xrightarrow{encode} ZMAC \]

\[ \text{encoding} \]

\[ IV[1] \]

\[ \tilde{E}_K^{10} \]

\[ M_1 \]

\[ C_1 \]

\[ IV[2] \]

\[ \tilde{E}_K^{10} \]

\[ M_2 \]

\[ C_2 \]

\[ \ldots \]

\[ IV[2] \]

\[ \tilde{E}_K^{10} \]

\[ M_m \]

\[ C_m \]
The **ZMAC** MAC mode and **ZAE** AEAD mode

**ZMAC and ZAE are:**

- a TBC-based MAC mode and a TBC-based AEAD mode
- with **full** \( n \)-bit security for both the **nonce-respecting** and **nonce-misuse** scenario (strong MRAE sense)
- can handle \( n + t \) bits of message per TBC call (optimal)
- when instantiated with **Deoxys-BC**, it is **faster** than **PMAC-AES**, with a **much higher security**!
- fully parallel, inverse-free
- extra tweak input for other purposes (leakage resilience)
- provably secure (security proofs in the article)
- no patent
Comparison

AES-GCM-SIV - Deoxys-II - ZAE
Comparison of Deoxys-II and ZAE with AES-GCM-SIV

- **winner of the CAESAR competition**, well scrutinized
- **much simpler and flexible** than AES-GCM-SIV
- GCM family very sensitive to **timing attacks**, while trivial and efficient constant time impl. for Deoxys-II and ZAE
- **higher security**: for $2^{32}$ messages of 64 KB each, attacker advantage for authenticity is
  - $2^{-37}$ for OCB (1 in nonce-misuse)
  - $2^{-73}$ for AES-GCM-SIV ($2^{-41}$ in nonce-misuse)
  - $2^{-94}$ for Deoxys-II ($2^{-51}$ in nonce-misuse)
  - $2^{-144}$ for ZAE ($2^{-144}$ in nonce-misuse)
- more efficient in **hardware**, inverse-free
- can easily offer the **Deoxys-I mode**
  (twice faster, full 128-bit security for nonce-respecting)
- tweak input can be used for many other things:
  disk encryption, leakage resilience, hashing, sessions, etc.
### Comparison of Deoxys-II and ZAE with AES-GCM-SIV

**Software efficiency estimations (in AES rounds):**

- 1 $GF(2^{128})$ mult. $\approx 6$ AES rounds - actually more on ARM
- 1 AES Key schedule $\approx 10$ AES rounds

<table>
<thead>
<tr>
<th></th>
<th>$M$ block</th>
<th>$A$ block</th>
<th>init/tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-GCM-SIV</td>
<td>16</td>
<td>6</td>
<td>66</td>
</tr>
<tr>
<td>Deoxys-II</td>
<td>28</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>ZAE</td>
<td>21</td>
<td>7</td>
<td>56</td>
</tr>
</tbody>
</table>

### Internet Mix efficiency estimations (in AES rounds):

- 7 packets of 40B, 4 packets of 576B, 1 packet of 1500B

<table>
<thead>
<tr>
<th></th>
<th>40 Bytes</th>
<th>576 Bytes</th>
<th>1500 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-GCM-SIV</td>
<td>114</td>
<td>642</td>
<td>1570</td>
</tr>
<tr>
<td>Deoxys-II</td>
<td>98</td>
<td>1022</td>
<td>2646</td>
</tr>
<tr>
<td>ZAE</td>
<td>119</td>
<td>812</td>
<td>2030</td>
</tr>
</tbody>
</table>
Thank you!
Nonce-respecting AEAD: Deoxys-I

Deoxys-I is similar to TAE or OCB

For associated data authentication:

\[ A_1 \]
\[ A_2 \]
\[ A_{l_a} \]
\[ A_{*10^*} \]
\[ E^2_{K}||0 \]
\[ E^2_{K}||1 \]
\[ \ldots \]
\[ E^2_{K}||l_{a-1} \]
\[ E^6_{K}||l_{a} \]
\[ 0 \]
\[ \oplus \]
\[ \oplus \]
\[ \oplus \]
\[ \text{Auth} \]

For plaintext:

\[ M_1 \]
\[ M_2 \]
\[ M_{l} \]
\[ M_{*10^{*}} \]
\[ \Sigma \]
\[ E^0_{K}||N||0 \]
\[ E^0_{K}||N||1 \]
\[ \ldots \]
\[ E^0_{K}||N||l-1 \]
\[ E^4_{K}||N||l \]
\[ 0^{\mu} \]
\[ \oplus \]
\[ \text{tag} \]
\[ \oplus \]
\[ \text{Auth} \]