OMD
A Compression Function Mode of Operation for Authenticated Encryption

Simon Cogliani, Diana Maimut, David Naccache (ENS, France)

Rodrigo Portella do Canto (Paris II - Panthéon-Assas University, France)

Reza Reyhanitabar, Serge Vaudenay, Damian Vizár (EPFL, Switzerland)
Outline

- Authenticated Encryption
  - Nonce-based Authenticated Encryption with Associated Data
  - The Security Goal(s)

- OMD:
  - Description
  - Security Analysis
  - Performance

- Conclusion
Authenticated Encryption

Privacy (Confidentiality) + Integrity (Authenticity)
Authenticated Encryption

Privacy (Confidentiality) + Integrity(Authenticity)

Privacy-Only Encryption Schemes
- Probabilistic
- IV-based
- Nonce-based
- Deterministic
Authenticated Encryption

Privacy (Confidentiality) + Integrity (Authenticity)

Privacy-Only Encryption Schemes
- Probabilistic
- IV-based
- Nonce-based
- Deterministic

Message Authentication Schemes
- Deterministic (MAC)
- (Randomized) IV-based
- (Counter) Nonce-based
## Authenticated Encryption

### Privacy (Confidentiality) + Integrity/Authenticity

<table>
<thead>
<tr>
<th>Privacy-Only Encryption Schemes</th>
<th>Message Authentication Schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>❖ Probabilistic</td>
<td>❖ Deterministic (MAC)</td>
</tr>
<tr>
<td>❖ IV-based</td>
<td>❖ (Randomized) IV-based</td>
</tr>
<tr>
<td>❖ Nonce-based</td>
<td>❖ (Counter) Nonce-based</td>
</tr>
<tr>
<td>❖ Deterministic</td>
<td></td>
</tr>
</tbody>
</table>

### Formalizations:

- Indistinguishability and Non-malleability notions (IND-CPA, IND-CCA1, IND-CCA2, NM-CCA, PRP)
- Unforgeability and PRF
Authenticated Encryption

Privacy (Confidentiality) + Integrity (Authenticity)

Privacy-Only Encryption Schemes
- Probabilistic
- IV-based
- Nonce-based
- Deterministic

Message Authentication Schemes
- Deterministic (MAC)
- (Randomized) IV-based
- (Counter) Nonce-based

Indistinguishability and Non-malleability notions (IND-CPA, IND-CCA1, IND-CCA2, NM-CCA, PRP)

Unforgeability and PRF

Authenticated Encryption?
**Nonce-based** Authenticated Encryption **with Associated Data**

- $N$: Nonce (public message number)
- $M$: Plaintext that needs to be encrypted and authenticated
- $AD$: Associated data that needs to be authenticated, but must not be encrypted
- $C$: Ciphertext
- $K$: Secret Key
The Security Goal(s)
The Security Goal(s)

\[
\text{Adv}^\text{priv}_\Pi (A) = \Pr[A^{\text{Enc}_K(\cdot, \cdot, \cdot)} \Rightarrow 1] - \Pr[A^{\$ (\cdot, \cdot, \cdot)} \Rightarrow 1]
\]
The Security Goal(s)

\[ \text{Adv}^\text{priv}_\Pi (A) = \Pr[A^{\text{Enc}}_K(\ldots) \Rightarrow 1] - \Pr[A^{\$}(\ldots) \Rightarrow 1] \]

\[ \text{Adv}^\text{auth}_\Pi (A) = \Pr[A^{\text{Enc}}_K(\ldots), D^{\text{Dec}}_K(\ldots) \text{ forges}] \]

A forges if: \( \exists (N, AD, C) \text{ such that } D^{\text{Dec}}_K(N, AD, C) \neq \bot \text{ AND no previous query } E^{\text{Enc}}_K(N, AD, M) \text{ returned } C \)
The **OMD** Mode of Operation

- OMD stands for **Offset Merkle–Damgård**.
- OMD is **a mode of operation that converts a compression function to a nonce-based AEAD**.
The **OMD** Mode of Operation

- OMD stands for *Offset Merkle–Damgård*.
- OMD is *a mode of operation that converts a compression function to a nonce-based AEAD*.
- **Design rationale:**
The OMD Mode of Operation

- OMD stands for Offset Merkle-Damgård.
- OMD is a mode of operation that converts a compression function to a nonce-based AEAD.
- Design rationale:
  - High security level (beyond the classical 64-bit security by AES-based designs).
The **OMD** Mode of Operation

- OMD stands for **Offset Merkle–Damgård**.
- OMD is a mode of operation that converts a compression function to a nonce-based AEAD.

**Design rationale:**
- High security level (beyond the classical 64-bit security by AES-based designs).
- Provable security based on a well-studied standard property of a widely-used primitive.
The **OMD** Mode of Operation

- OMD stands for **Offset Merkle–Damgård**.
- OMD is a mode of operation that converts a compression function to a nonce-based AEAD.

**Design rationale:**
- **High security level** (beyond the classical 64-bit security by AES-based designs).
- **Provable security** based on a well-studied standard property of a widely-used primitive.
- **Simplicity** (using only a single primitive).
The **OMD** Mode of Operation

- **OMD** stands for **Offset Merkle–Damgård**.
- **OMD** is a mode of operation that converts a compression function to a nonce-based **AEAD**.
- **Design rationale:**
  - **High security level** (beyond the classical 64-bit security by AES-based designs).
  - **Provable security** based on a well-studied standard property of a widely-used primitive.
  - **Simplicity** (using only a single primitive).
  - **Patent-freeness** (avoid using any patented algorithms, such as PMAC, as a subroutine).
The **OMD** Mode of Operation

- OMD stands for **Offset Merkle–Damgård**.

- OMD is a **mode of operation that converts a compression function to a nonce-based AEAD**.

- **Design rationale:**
  - High security level (beyond the classical 64-bit security by AES-based designs).
  - Provable security based on a well-studied standard property of a widely-used primitive.
  - Simplicity (using only a single primitive).
  - Patent-freeness (avoid using any patented algorithms, such as PMAC, as a subroutine).
  - Not being a blockcipher-based or permutation-based design (*Don't Put All Your “Security” Eggs in One or Two Baskets!*).
The **OMD** Mode of Operation

- OMD stands for **Offset Merkle–Damgård**.
- OMD is a mode of operation that converts a compression function to a nonce-based AEAD.

**Design rationale:**
- **High security level** (beyond the classical 64-bit security by AES-based designs).
- **Provable security** based on a well-studied standard property of a widely-used primitive.
- **Simplicity** (using only a single primitive).
- **Patent-freeness** (avoid using any patented algorithms, such as PMAC, as a subroutine).
- **Not being a blockcipher-based or permutation-based design** (Don't Put All Your “Security” Eggs in One or Two Baskets!)
- **Acceptable performance**, comparable with that of the standardized AES-GCM scheme.
We Assume that: the keyed compression function $F$ is a PRF.

We know that: MD Preserves PRF. (Bellare and Ristenpart, ICALP 2007)
Toward making an AE out of the MD iteration
OMD: A Secure Nonce-based AE Algorithm

(Encrypting a message whose length is a multiple of the block length)
OMD: A Secure Nonce-based AE Algorithm

(Encrypting a message whose length is **not** a multiple of the block length)
Handling Associated Data when the length of the data is a multiple of the input length.
Handling Associated Data when the length of the data is not a multiple of the input length.
OMD: A Secure Nonce-based **AEAD** Algorithm

(The Case shown: encrypting a message and associated data whose lengths are a multiple of the block length and input length, respectively.)
Security Analysis

\[
\text{Adv}^{\text{priv}}_{\text{OMD}[F,\tau]}(t, \sigma_e, \ell_{\text{max}}) = \text{Adv}^{\text{prf}}_F (t', 2\sigma_e) + \frac{3\sigma_e^2}{2^n}
\]

\[
\text{Adv}^{\text{auth}}_{\text{OMD}[F,\tau]}(t, q_v, \sigma, \ell_{\text{max}}) = \text{Adv}^{\text{prf}}_F (t', 2\sigma) + \frac{3\sigma^2}{2^n} + \frac{q_v\ell_{\text{max}}}{2^n} + \frac{q_v}{2^\tau}
\]

\(\sigma_e\): total number of calls to the compression function in encryption queries
\(\sigma\): total number of calls to the compression function in all (encryption and verification) queries
\(q_v\): the number of decryption (verification) queries
\(\ell_{\text{max}}\): the maximum number of message blocks in any query
\(n\): the output length of the compression function in bits
\(\tau\): the tag length
\(t' = t + cn\sigma\)
Modular, Simple Proof:

- **Step 1**: (Idealized) Generalized OMD using a tweakable random function

- **Step 2**: Realization of the tweakable random function using a tweakable PRF

- **Step 3**: Instantiation of the tweakable PRF using a PRF
Proof Step 1: Generalized OMD (G-OMD)
Each call to the tweakable random function uses a new distinct tweak.
Proof Step 2

Proof Step 3

\( \Delta_K(T) \)

The XE method
(Rogaway, ASIACRYPT 2004):

\( \text{Cause of } \frac{3\sigma^2}{2^n} \text{ in the security bounds} \)
Performance

- We have done some preliminary performance measurements on Intel Core i5-2415M.

- The results show that OMD-sha256 and OMD-sha512 have reasonable software performance comparable to AES-GCM (while providing much higher security levels).

- More optimized implementations and performance measurements will be available through the CAESAR website.
Timing Measurements for some different implementations of OMD-sha256
Timing Measurements for some different implementations of OMD-sha512
Performance of GCM and OCB (without AES-NI)

Performance and Security Comparison

Performance (rate) measurements are in “cycles per byte” (cpb).

<table>
<thead>
<tr>
<th>Message length (bytes)</th>
<th>AES-OCB cpb</th>
<th>AES-GCM cpb</th>
<th>OMD-sha256 cpb</th>
<th>OMD-sha512 cpb</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>16.14</td>
<td>32.31</td>
<td>44.56</td>
<td>45.93</td>
</tr>
<tr>
<td>256</td>
<td>11.94</td>
<td>27.12</td>
<td>36.37</td>
<td>34.11</td>
</tr>
<tr>
<td>512</td>
<td>9.84</td>
<td>24.61</td>
<td>32.28</td>
<td>28.11</td>
</tr>
<tr>
<td>1024</td>
<td>8.80</td>
<td>23.36</td>
<td>30.34</td>
<td>25.18</td>
</tr>
<tr>
<td>4096</td>
<td>8.05</td>
<td>22.40</td>
<td>28.77</td>
<td>23.28</td>
</tr>
</tbody>
</table>

Without AES-NI and Intel SHA Extensions

With AES-NI and Intel SHA Extensions

Security: 64 bits, 64 bits, 127 bits, 255 bits

(Projected)
More recent results on performance of SHA-256 presented at the SHA3 2014 Workshop by Shay Gueron

OMD-sha256 performs about 3 times slower than SHA-256, so we expect about 2.7 cpb for OMD-sha256.
OMD-sha256 Performance with Associated Data
OMD-sha512 Performance with Associated Data
OMD offers:

- **High security level** beyond the classical 64-bit security by AES-based designs (e.g. 127 bits for OMD-sha256 and 255 bits for OMD-sha512).
- **Provable security** based on a well-studied standard property of a widely-used primitive.
- **Simplicity.**
- **Patent-freeness** (does not use any patented algorithm structures; such as, PMAC or OCB as a subroutine).
- **Not relying on a blockcipher or ideal permutation** (Don't Put All Your “Security” Eggs in One or Two Baskets!)
- **Acceptable performance**, comparable with that of the standardized AES-GCM scheme.
  - On future processors with Intel SHA Extensions, OMD-sha256 will offer an appealing combination of high performance (about 3 cpb) and high security level (127 bits).
Nonce-Misuse Resistance?

OMD, as submitted to CAESAR, is not aimed to be misuse resistant because we want to have an online encryption process!

It has some weak level of misuse resistance (e.g. authenticity of the message is preserved) but we do not claim any security beyond nonce reuse.
Nonce-Misuse Resistance?

Can an online AEAD provide any useful privacy after nonce is reused?

I have posted a note answering this question negatively to the Google discussion group of CAESAR titled “Carefull with Misuse-Resistance of Online AEAD Schemes”.
Nonce-Misuse Resistance?

“Misuse Resistant variants of the OMD Authenticated Encryption Scheme” can be found in our paper in ProvSec 2014.

These variants, as expected, are two-pass unlike OMD which is one-pass.
References:


References...


References...


Appendix: Criteria for the Masking Function $\Delta_K(T)$

The masking function $\Delta_K(T) = \Delta_K(\alpha, i, j)$ outputs an $n$-bit mask such that the following two properties hold for any fixed string $H \in \{0, 1\}^n$:

1. $\Pr[\Delta_K(\alpha, i, j) = H] \leq 2^{-n}$ for any $(\alpha, i, j)$
2. $\Pr[\Delta_K(\alpha, i, j) \oplus \Delta_K(\alpha', i', j') = H] \leq 2^{-n}$ for $(\alpha, i, j) \neq (\alpha', i', j')$

where the probabilities are taken over random selection of the key.
Appendix: Computing the Masking Function

There are different ways to compute the masking values to satisfy these criteria. In OMD, we use the method proposed by Krovetz and Rogaway in FSE 2011 [16].

Initialization:

\[
\Delta_{N,0,0} = F_K(N||10^{n-1}-|N|, 0^m); \quad \tilde{\Delta}_{0,0} = 0^n; \quad L_* = F_K(0^n, 0^m); \quad L(0) = 4 \cdot L_*,
\]

and \( L(i) = 2 \cdot L(i - 1) \) for \( i \geq 1 \).

Masking sequence for processing the message:

For \( i \geq 1 \):

\[
\Delta_{N,i,0} = \Delta_{N,i-1,0} \oplus L(ntz(i)); \quad \Delta_{N,i,1} = \Delta_{N,i,0} \oplus 2 \cdot L_*; \quad \text{and} \quad \Delta_{N,i,2} = \Delta_{N,i,0} \oplus 3 \cdot L_*.
\]

Masking sequence for processing the associated data:

\[
\tilde{\Delta}_{i,0} = \Delta_{i-1,0} \oplus L(ntz(i)) \quad \text{for} \quad i \geq 1; \quad \text{and} \quad \tilde{\Delta}_{i,1} = \tilde{\Delta}_{i,0} \oplus L_* \quad \text{for} \quad i \geq 0.
\]