Chaskey: a Lightweight MAC Algorithm for Microcontrollers

Nicky Mouha\textsuperscript{1}, Bart Mennink\textsuperscript{1}, Anthony Van Herrewege\textsuperscript{1}, Dai Watanabe\textsuperscript{2}, Bart Preneel\textsuperscript{1}, Ingrid Verbauwhede\textsuperscript{1}

\textsuperscript{1}ESAT/COSIC, KU Leuven and iMinds, Belgium
\textsuperscript{2}Yokohama Research Laboratory, Hitachi, Japan

NIST Lightweight Cryptography Workshop
July 20, 2015
MAC Algorithm for Microcontrollers

Message Authentication Code (MAC)

- $\text{MAC}_K(m) = \tau$
- Authenticity, no confidentiality
- Same key for MAC generation and verification
MAC Algorithm for Microcontrollers

Message Authentication Code (MAC)

- $\text{MAC}_K(m) = \tau$
- Authenticity, no confidentiality
- Same key for MAC generation and verification

Microcontroller

- Cheap 8/16/32-bit processor: USD 25-50¢
- Applications: home, medical, industrial,...
- Ubiquitous: 30-100 in any recent car
Design

Requirements

- Drop-in replacement for AES-CMAC
  (variant of CBC-MAC for variable-length messages)
- Same functionality and security
Design

Requirements

- Drop-in replacement for AES-CMAC (variant of CBC-MAC for variable-length messages)
- Same functionality and security

Speed

- “Ten times faster than AES”
Design

Requirements
- Drop-in replacement for AES-CMAC (variant of CBC-MAC for variable-length messages)
- Same functionality and security

Speed
- “Ten times faster than AES”

Approach
- Dedicated design for microcontrollers
Commonly used MACs

Based on (cryptographic) hash function

- Example: HMAC, SHA3-MAC
- Large block size, collision resistance unnecessary
Commonly used MACs

Based on (cryptographic) hash function

- **Example**: HMAC, SHA3-MAC
- Large block size, collision resistance unnecessary

Based on universal hashing

- **Examples**: UMAC, GMAC, Poly1305
- **Requires**: nonce, constant-time multiply, long tags
Commonly used MACs

**Based on (cryptographic) hash function**
- **Example:** HMAC, SHA3-MAC
- Large block size, collision resistance unnecessary

**Based on universal hashing**
- **Examples:** UMAC, GMAC, Poly1305
- **Requires:** nonce, constant-time multiply, long tags

**Based on block cipher**
- **Example:** CMAC
Commonly used MACs

Based on (cryptographic) hash function
- **Example:** HMAC, SHA3-MAC
- Large block size, collision resistance unnecessary

Based on universal hashing
- **Examples:** UMAC, GMAC, Poly1305
- **Requires:** nonce, constant-time multiply, long tags

Based on block cipher
- **Example:** CMAC
- **Problem:** ten times too slow!
Our Approach

Every cycle counts!

- Avoid load/store: keep data in registers
- Avoid bit masking
- Make optimal use of instruction set
Our Approach

Every cycle counts!
- Avoid load/store: keep data in registers
- Avoid bit masking
- Make optimal use of instruction set

Bridging the gap
- Provable security
- Cryptanalysis
- Implementation
Primitive

Which primitive?
- Cryptographic hash function $\checkmark$
Primitive

Which primitive?

- Cryptographic hash function ×
- Universal hash function ×
Primitive

Which primitive?

- Cryptographic hash function ✗
- Universal hash function ✗
- Block cipher ✗
Primitive

Which primitive?

- Cryptographic hash function ✗
- Universal hash function ✗
- Block cipher ✗
- Ideal permutation ✗
Which primitive?

- Cryptographic hash function $\times$
- Universal hash function $\times$
- Block cipher $\times$
- Ideal permutation $\times$

$\rightarrow$ Even-Mansour Block Cipher $\checkmark$
Primitive

Which primitive?

- Cryptographic hash function \( \times \)
- Universal hash function \( \times \)
- Block cipher \( \times \)
- Ideal permutation \( \times \)

\( \rightarrow \) Even-Mansour Block Cipher \( \checkmark \)

Related-key attacks

- Insecure, so choose uniformly random keys!
Chaskey: Mode of Operation

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$
- $K_1 = 2K$
Chaskey: Mode of Operation

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$, bottom: $0 \leq |m_\ell| < n$
- $K_1 = 2K$, $K_2 = 4K$
Chaskey: Mode of Operation: Phantom XORs

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$, bottom: $0 \leq |m_\ell| < n$
- $K_1 = 2K$, $K_2 = 4K$
Chaskey: Mode of Operation: Phantom XORs

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$, bottom: $0 \leq |m_\ell| < n$
- $K_1 = 2K$, $K_2 = 4K$
Chaskey: Mode of Operation: Block-cipher-based

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$, bottom: $0 \leq |m_\ell| < n$
- $K_1 = 2K$, $K_2 = 4K$
Chaskey: Mode of Operation: Block-cipher-based

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$, bottom: $0 \leq |m_\ell| < n$
- $K_1 = 2K$, $K_2 = 4K$

variant of FCBC [BR’00]
Chaskey: Mode of Operation: Compared to CMAC

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$, bottom: $0 \leq |m_\ell| < n$
- $K_1 = 2K$, $K_2 = 4K$

variant of CMAC [IK’03]
Chaskey: Mode of Operation: Compared to CMAC

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$, bottom: $0 \leq |m_\ell| < n$
- $K_1 = 2K$, $K_2 = 4K$

\[
\begin{align*}
\text{variant of CMAC [IK'03]}
\end{align*}
\]

\[
\begin{array}{c}
E_K || K \\
E_K || K \\
\vdots
E_K || K \\
E_K || K \\
\end{array}
\]
Chaskey: Mode of Operation: Compared to CMAC

- Split $m$ into $\ell$ blocks of $n$ bits
- Top: $|m_\ell| = n$, bottom: $0 \leq |m_\ell| < n$
- $K_1 = 2K$, $K_2 = 4K$
  \[ E_K(0^n) \rightarrow K \]

variant of CMAC [IK’03]

![Diagram](image-url)

1. Even-Mansour

$0 \rightarrow E_{K\parallel K} \rightarrow E_{K\parallel K} \rightarrow \ldots \rightarrow E_{K\parallel K} \rightarrow E_{K\parallel K} \rightarrow K \oplus K_1 \rightarrow \tau$

2. Even-Mansour

$0 \rightarrow E_{K\parallel K} \rightarrow E_{K\parallel K} \rightarrow \ldots \rightarrow E_{K\parallel K} \rightarrow E_{K\parallel K} \rightarrow K \oplus K_2 \rightarrow \tau$
Chaskey: Mode of Operation: Compared to CMAC

- Split \( m \) into \( \ell \) blocks of \( n \) bits
- Top: \( |m_\ell| = n \), bottom: \( 0 \leq |m_\ell| < n \)
- \( K_1 = 2K \), \( K_2 = 4K \)

1. \( E_K(0^n) \to K \)

<table>
<thead>
<tr>
<th>variant of CMAC [IK’03]</th>
</tr>
</thead>
<tbody>
<tr>
<td>not in CMAC</td>
</tr>
</tbody>
</table>

\[ 0 \rightarrow m_1 \rightarrow E_K||K \rightarrow m_2 \rightarrow E_K||K \rightarrow \ldots \rightarrow E_K||K \rightarrow m_\ell \rightarrow K \oplus K_1 \rightarrow K \oplus K_2 \rightarrow \tau \]

- 2. Even-Mansour

\[ 0 \rightarrow m_1 \rightarrow E_K||K \rightarrow m_2 \rightarrow E_K||K \rightarrow \ldots \rightarrow E_K||K \rightarrow m_\ell \rightarrow 10*K_2 \rightarrow K \oplus K_2 \rightarrow \tau \]
Cryptanalysis

**MAC forgery:** find new valid \((m, \tau)\)
- \(D\): data complexity (\# blocks of chosen messages)
- \(T\): time complexity (\# permutation evaluations)

**Attacks**
- Internal collision: \(D \approx 2^{n/2}\)
- Key recovery: \(T \approx 2^n/D\)
- Tag guessing: \(\approx 2^t\) guesses

**Chaskey parameters**
- Key size, block size: \(n = 128\), tag length: \(t \geq 64\)
Permutation

Design
- Add-Rot-XOR (ARX)
- Inspired by SipHash
- 32-bit words
- 8 rounds

Properties
- Rotations by 8, 16: faster on 8-bit μC
- Fixed point: $0 \rightarrow 0$
- Cryptanalysis: rotational, (truncated) differential, MitM, slide,... see paper!
## Chaskey: Speed Optimized (gcc -O2)

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>Algorithm</th>
<th>Data [byte]</th>
<th>ROM [byte]</th>
<th>Speed [cycles/byte]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortex-M0</td>
<td>AES-128-CMAC</td>
<td>16</td>
<td>13492</td>
<td>173.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>13492</td>
<td>136.5</td>
</tr>
<tr>
<td></td>
<td>Chaskey</td>
<td>16</td>
<td>1308</td>
<td>21.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>1308</td>
<td>18.3</td>
</tr>
<tr>
<td>Cortex-M4</td>
<td>AES-128-CMAC</td>
<td>16</td>
<td>28524</td>
<td>118.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>28524</td>
<td>105.0</td>
</tr>
<tr>
<td></td>
<td>Chaskey</td>
<td>16</td>
<td>908</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>908</td>
<td>7.0</td>
</tr>
<tr>
<td>Microcontroller</td>
<td>Algorithm</td>
<td>Data [byte]</td>
<td>ROM [byte]</td>
<td>Speed [cycles/byte]</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------</td>
<td>-------------</td>
<td>------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Cortex-M0</td>
<td>AES-128-CMAC</td>
<td>16</td>
<td>13492</td>
<td>173.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>13492</td>
<td>136.5</td>
</tr>
<tr>
<td></td>
<td>Chaskey</td>
<td>16</td>
<td>1308</td>
<td>21.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>1308</td>
<td>18.3</td>
</tr>
<tr>
<td>Cortex-M4</td>
<td>AES-128-CMAC</td>
<td>16</td>
<td>28524</td>
<td>118.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>28524</td>
<td>105.0</td>
</tr>
<tr>
<td></td>
<td>Chaskey</td>
<td>16</td>
<td>908</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>908</td>
<td>7.0</td>
</tr>
</tbody>
</table>
Summary

Chaskey:
MAC algorithm for 32-bit microcontrollers

- Addition-Rotation-XOR (ARX)
- Even-Mansour block cipher
- ARM Cortex-M: 7-15× faster than AES-128-CMAC
Summary

Chaskey:
MAC algorithm for 32-bit microcontrollers

- Addition-Rotation-XOR (ARX)
- Even-Mansour block cipher
- ARM Cortex-M: $7\times$ faster than AES-128-CMAC
Summary

Chaskey:
MAC algorithm for 32-bit microcontrollers

- Addition-Rotation-XOR (ARX)
- Even-Mansour block cipher
- ARM Cortex-M: $7-15 \times$ faster than AES-128-CMAC

More info, updates:
http://mouha.be/chaskey
Questions?
Supporting Slides
## Chaskey: Size Optimized (gcc -Os)

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>Algorithm</th>
<th>Data [byte]</th>
<th>ROM [byte]</th>
<th>Speed [cycles/byte]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortex-M0</td>
<td>AES-128-CMAC</td>
<td>16</td>
<td>11664</td>
<td>176.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>11664</td>
<td>140.0</td>
</tr>
<tr>
<td></td>
<td>Chaskey</td>
<td>16</td>
<td>414</td>
<td>21.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>414</td>
<td>16.9</td>
</tr>
<tr>
<td>Cortex-M4</td>
<td>AES-128-CMAC</td>
<td>16</td>
<td>10925</td>
<td>127.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>10925</td>
<td>89.4</td>
</tr>
<tr>
<td></td>
<td>Chaskey</td>
<td>16</td>
<td>402</td>
<td>16.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>402</td>
<td>11.2</td>
</tr>
</tbody>
</table>
Security Proof

**MAC forgery**: find new valid \((m, \tau)\)
- \(D\): block cipher (PRP) queries
- \(T\): permutation queries

**Standard Model**
- \(\text{Adv}^{\text{mac}}_{\text{Chaskey-B}}(q, D, r) \leq \frac{2D^2}{2^n} + \frac{1}{2^t} + \text{Adv}^{3\text{prp}}_E(D, r)\)

**Ideal Permutation Model**
- \(\text{Adv}^{\text{mac}}_{\text{Chaskey}}(q, D, r) \leq \frac{2D^2}{2^n} + \frac{1}{2^t} + \frac{D^2 + 2DT}{2^n}\)