Optimally Secure Tweakable Blockciphers

Bart Mennink
KU Leuven (Belgium)

Fast Software Encryption
March 10, 2015
Introduction

- Tweaks: flexibility to the cipher
- Each tweak gives different permutation

Dedicated constructions:
- Hasty Pudding Cipher [Sch98]
- Mercy [Cro01]
- Threesh [FLS+07]
Introduction

- Tweak: flexibility to the cipher
- Each tweak gives different permutation
Introduction

- Tweak: flexibility to the cipher
- Each tweak gives different permutation

Dedicated constructions:
- Hasty Pudding Cipher [Sch98]
- Mercy [Cro01]
- Threesh [FLS+07]
Introduction: Modular Designs

- LRW1 and LRW2 by Liskov et al. [LRW02]:

\[
\begin{align*}
E(m, k, t) &= E(m, k) \\
E(m, k) &= c
\end{align*}
\]

- \( h \) is XOR-universal hash
- Related: XEX
- Secure up to \( 2^{n/2} \) queries
Introduction: Modular Designs

- LRW2[ρ]: concatenation of ρ LRW2’s
- $k_1, \ldots, k_\rho$ and $h_1, \ldots, h_\rho$ independent
Intro duction: Mod u lar Designs

- LRW2[ρ]: concatenation of ρ LRW2’s
- $k_1, \ldots, k_\rho$ and $h_1, \ldots, h_\rho$ independent

- $\rho = 2$: secure up to $2^{2n/3}$ queries [LST12, Pro14]
- $\rho \geq 2$ even: secure up to $2^{\rho n/(\rho+2)}$ queries [LS13]
- Conjecture: optimal $2^{\rho n/(\rho+1)}$ security
**Introduction: State of the Art**

<table>
<thead>
<tr>
<th>scheme</th>
<th>security ((\log_2))</th>
<th>key length</th>
<th>cost</th>
<th>(E)</th>
<th>(\otimes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRW1</td>
<td>(n/2)</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LRW2</td>
<td>(n/2)</td>
<td>2(n)</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>XEX</td>
<td>(n/2)</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LRW2[2]</td>
<td>(2n/3)</td>
<td>4(n)</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>LRW2[(\rho)]</td>
<td>(\rho n/(\rho+2))</td>
<td>2(\rho n)</td>
<td>(\rho)</td>
<td>(\rho)</td>
<td></td>
</tr>
</tbody>
</table>

Optimal \(2^n\) security only if **key length and cost \(\to\) \(\infty\)?
Introduction: Tweak-Dependent Keys

**Efficiency**

- Tweak schedule *lighter* than key schedule
Introduction: Tweak-Dependent Keys

**Efficiency**
- tweak schedule lighter than key schedule

**Security**
- tweak schedule stronger than key schedule

TWEAKEY [JNP14]
Introduction: Tweak-Dependent Keys

- **Efficiency**
  - tweak schedule lighter than key schedule

- **Security**
  - tweak schedule stronger than key schedule

Tweak and key change approximately **equally expensive**
Introduction: Tweak-Dependent Keys

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>tweak schedule lighter than key schedule</td>
<td>tweak schedule stronger than key schedule</td>
</tr>
</tbody>
</table>

Tweak and key change approximately equally expensive

- TWEAKEY [JNP14] key scheduling blends key and tweak
Introduction: Tweak-Dependent Keys

- Minematsu [Min09]:

- Secure up to \( \max\{2^{n/2}, 2^n - |t|\} \) queries
- Beyond birthday bound for \(|t| < n/2\)
## Introduction: State of the Art

<table>
<thead>
<tr>
<th>scheme</th>
<th>security (log(_2))</th>
<th>key length</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRW1</td>
<td>(n/2)</td>
<td>(n)</td>
<td>2</td>
</tr>
<tr>
<td>LRW2</td>
<td>(n/2)</td>
<td>(2n)</td>
<td>1</td>
</tr>
<tr>
<td>XEX</td>
<td>(n/2)</td>
<td>(n)</td>
<td>2</td>
</tr>
<tr>
<td>LRW2([2])</td>
<td>(2n/3)</td>
<td>(4n)</td>
<td>2</td>
</tr>
<tr>
<td>LRW2([\rho])</td>
<td>(\rho n/(\rho+2))</td>
<td>(2\rho n)</td>
<td>(\rho)</td>
</tr>
<tr>
<td>Min</td>
<td>(\max{n/2, n-</td>
<td>t</td>
<td>})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>scheme</th>
<th>(E)</th>
<th>(\otimes/h)</th>
<th>tdk</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRW1</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>LRW2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>XEX</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LRW2([2])</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>LRW2([\rho])</td>
<td>(\rho)</td>
<td>(\rho)</td>
<td>0</td>
</tr>
<tr>
<td>Min</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Our Goal

Given a blockcipher $E$, construct optimally secure tweakable blockcipher $\widetilde{E}$

\begin{align*}
\text{all wires} & \quad \text{carry } n \text{ bits}
\end{align*}
\[ \tilde{E}[\rho] \text{ (for } \rho \geq 1) \]
Generic Design

\[ \tilde{E}[\rho] \text{ (for } \rho \geq 1) \]

- Mixing functions \( A_i, B_i \)
  - should be such that \( \tilde{E}[\rho] \) is invertible
  - but can be anything otherwise
Security Model

\[ \tilde{E}[\rho]_{k}^{\pm} \rightarrow E^{\pm} \rightarrow \tilde{\pi}^{\pm} \rightarrow E^{\pm} \]

- Information-theoretic indistinguishability
  - \( \tilde{\pi} \) ideal tweakable cipher
  - \( E \) ideal cipher
Security Model

- Information-theoretic indistinguishability
  - $\pi$ ideal tweakable cipher
  - $E$ ideal cipher
- Complexity-theoretic indistinguishability?
One \( E \)-Call with Linear Mixing

\[ \begin{align*}
  \text{Diagram:} & \\
  m & \rightarrow A_1 & x_1 & \rightarrow E & y_1 & \rightarrow A_2 & c \\
  k, t & \rightarrow B_1 & & \rightarrow & & \rightarrow \\
  l_1 & \rightarrow E & & \rightarrow & & \rightarrow \\
  \end{align*} \]
One $E$-Call with Linear Mixing

**Theorem**

- If $A_1, B_1, A_2$ are linear, $\tilde{E}[1]$ can be distinguished from $\tilde{\pi}$ in at most about $2^{n/2}$ queries
One $E$-Call with Linear Mixing

Theorem

- If $A_1, B_1, A_2$ are linear, $\tilde{E}[1]$ can be distinguished from $\tilde{\pi}$ in at most about $2^{n/2}$ queries

Proof idea

- Relation among queries to $\tilde{E}[1]$?
- Case distinction based on how $k, t, m$ are processed
One $E$-Call with Polynomial Mixing

Idea
- Subkey $k \oplus t$
- Masking $k \otimes t$

\[ \tilde{F}[1](k, t, m) = c \]
One $E$-Call with Polynomial Mixing

\[ \tilde{F}[1](k, t, m) = c \]

**Idea**
- Subkey $k \oplus t$
- Masking $k \otimes t$

**Security**
- Up to $2^{2n/3}$ queries
One $E$-Call with Polynomial Mixing

Idea
- Subkey $k \oplus t$
- Masking $k \otimes t$

Security
- Up to $2^{2n/3}$ queries

Cost
- One $E$-call
- One $\otimes$-evaluation
- One re-key

$$\tilde{F}[1](k, t, m) = c$$
One $E$-Call with Polynomial Mixing: Proof Idea

- Key $k$ is secret
One $E$-Call with Polynomial Mixing: Proof Idea

- Key $k$ is secret
- Consider any construction query $(t, m, c)$
One $E$-Call with Polynomial Mixing: Proof Idea

- Key $k$ is secret
- Consider any construction query $(t, m, c)$
One $E$-Call with Polynomial Mixing: Proof Idea

- Key $k$ is secret
- Consider any construction query $(t, m, c)$
- May “hit” any primitive query $(l, x, y)$
One $E$-Call with Polynomial Mixing: Proof Idea

- Key $k$ is secret
- Consider any construction query $(t, m, c)$
- May “hit” any primitive query $(l, x, y)$

$k \oplus t = l$ and $m \oplus k \otimes t = x$
One $E$-Call with Polynomial Mixing: Proof Idea

- Key $k$ is secret
- Consider any construction query $(t, m, c)$
- May “hit” any primitive query $(l, x, y)$

$k \oplus t = l$ and $m \oplus k \otimes t = x$

or

$k \oplus t = l$ and $c \oplus k \otimes t = y$
One $E$-Call with Polynomial Mixing: Proof Idea

- Key $k$ is secret
- Consider any construction query $(t, m, c)$
- May “hit” any primitive query $(l, x, y)$

\[
\begin{align*}
    k \oplus t &= l \quad \text{and} \quad m \oplus k \otimes t = x & \iff & & k = l \oplus t \quad \text{and} \quad m \oplus (l \oplus t) \otimes t = x \\
    \text{or} & & & & \text{or} \\
    k \oplus t &= l \quad \text{and} \quad c \oplus k \otimes t = y & \iff & & k = l \oplus t \quad \text{and} \quad c \oplus (l \oplus t) \otimes t = y
\end{align*}
\]
One $E$-Call with Polynomial Mixing: Proof Idea

- Key $k$ is secret
- Consider any construction query $(t, m, c)$
- May “hit” any primitive query $(l, x, y)$

\[ k \oplus t = l \quad \text{and} \quad m \oplus k \otimes t = x \quad \iff \quad k = l \oplus t \quad \text{and} \quad m \oplus (l \oplus t) \otimes t = x \]

or

\[ k \oplus t = l \quad \text{and} \quad c \oplus k \otimes t = y \quad \iff \quad k = l \oplus t \quad \text{and} \quad c \oplus (l \oplus t) \otimes t = y \]
One $E$-Call with Polynomial Mixing: Proof Idea

\[ k = l \oplus t \text{ and } m \oplus (l \oplus t) \otimes t = x \]
One $E$-Call with Polynomial Mixing: Proof Idea

$k = l \oplus t$ and $m \oplus (l \oplus t) \otimes t = x$

**Szemerédi-Trotter theorem [ST83]**

Consider a finite field $\mathbb{F}$. Let

- $L \subseteq \mathbb{F}^2$ be a set of lines
- $P \subseteq \mathbb{F}^2$ be a set of points

# point-line incidences \(\leq\) \(\min\{ |L|^{1/2} |P| + |L|, |L||P|^{1/2} + |P| \} \)
One $E$-Call with Polynomial Mixing: Proof Idea

\[ k = l \oplus t \text{ and } m \oplus (l \oplus t) \otimes t = x \]

**Szemerédi-Trotter theorem [ST83]**

Consider a finite field $\mathbb{F}$. Let

- $L \subseteq \mathbb{F}^2$ be a set of lines
- $P \subseteq \mathbb{F}^2$ be a set of points

\# point-line incidences $\leq \min\{|L|^{1/2}|P| + |L|, |L||P|^{1/2} + |P|\}$

- Construction queries = lines
- Primitive queries = points
One $E$-Call with Polynomial Mixing: Proof Idea

\[ k = l \oplus t \text{ and } m \oplus (l \oplus t) \otimes t = x \]

**Szemerédi-Trotter theorem [ST83]**

Consider a finite field $\mathbb{F}$. Let
- $L \subseteq \mathbb{F}^2$ be a set of lines
- $P \subseteq \mathbb{F}^2$ be a set of points

\[ \# \text{ point-line incidences} \leq \min\{|L|^{1/2}|P| + |L|, |L||P|^{1/2} + |P|\} \]

- Construction queries = lines
- Primitive queries = points
- About $q^{3/2}$ solutions to $m \oplus (l \oplus t) \otimes t = x$
One $E$-Call with Polynomial Mixing: Proof Idea

\[ k = l \oplus t \text{ and } m \oplus (l \oplus t) \otimes t = x \]

**Szemerédi-Trotter theorem [ST83]**

Consider a finite field $\mathbb{F}$. Let
- $L \subseteq \mathbb{F}^2$ be a set of lines
- $P \subseteq \mathbb{F}^2$ be a set of points

\[ \text{\# point-line incidences} \leq \min\{|L|^{1/2}|P| + |L|, |L||P|^{1/2} + |P|\} \]

- Construction queries = lines
- Primitive queries = points
- About $q^{3/2}$ solutions to $m \oplus (l \oplus t) \otimes t = x$
- Every solution fixes one $l \oplus t$
One $E$-Call with Polynomial Mixing: Proof Idea

\[ k = l \oplus t \text{ and } m \oplus (l \oplus t) \otimes t = x \]

**Szemerédi-Trotter theorem [ST83]**

Consider a finite field $F$. Let

- $L \subseteq F^2$ be a set of lines
- $P \subseteq F^2$ be a set of points

\[
\# \text{ point-line incidences} \leq \min\{ |L|^{1/2} |P| + |L|, |L||P|^{1/2} + |P| \}
\]

- Construction queries = lines
- Primitive queries = points
- About $q^{3/2}$ solutions to $m \oplus (l \oplus t) \otimes t = x$
- Every solution fixes one $l \oplus t$
- $k$ is random $n$-bit key
Two $E$-Calls with Linear Mixing

\[ \tilde{F}[2](k, t, m) = c \]

**Idea**
- Subkey $k \oplus t$
- Masking $E(k, t)$
Two $E$-Calls with Linear Mixing

\[ \tilde{F}[2](k, t, m) = c \]

**Idea**
- Subkey $k \oplus t$
- Masking $E(k, t)$

**Security**
- Up to $2^n$ queries
Two $E$-Calls with Linear Mixing

Idea
- Subkey $k \oplus t$
- Masking $E(k, t)$

Security
- Up to $2^n$ queries

Cost
- Two $E$-calls
- Zero $\otimes$-evaluations
- One re-key

\[
\tilde{F}[2](k, t, m) = c
\]
Two $E$-Calls with Linear Mixing: Proof Idea
Two $E$-Calls with Linear Mixing: Proof Idea

- Construction query $(t, m, c)$ “hits” primitive query $(l, x, y)$ if

\[
\begin{align*}
k \oplus t &= l \quad \text{and} \quad z \oplus m = x \\
\text{or} \\
k \oplus t &= l \quad \text{and} \quad z \oplus c = y
\end{align*}
\]
Two $E$-Calls with Linear Mixing: Proof Idea

- Construction query $(t, m, c)$ “hits” primitive query $(l, x, y)$ if

\[ k \oplus t = l \quad \text{and} \quad z \oplus m = x \]

or

\[ k \oplus t = l \quad \text{and} \quad z \oplus c = y \]

- $k$ is random key, $z$ is almost-random subkey
## Comparison

<table>
<thead>
<tr>
<th>scheme</th>
<th>security (log₂)</th>
<th>key length</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>E</td>
</tr>
<tr>
<td>LRW1</td>
<td>n/2</td>
<td>n</td>
<td>2</td>
</tr>
<tr>
<td>LRW2</td>
<td>n/2</td>
<td>2n</td>
<td>1</td>
</tr>
<tr>
<td>XEX</td>
<td>n/2</td>
<td>n</td>
<td>2</td>
</tr>
<tr>
<td>LRW2[2]</td>
<td>2n/3</td>
<td>4n</td>
<td>2</td>
</tr>
<tr>
<td>LRW2[ρ]</td>
<td>ρn/(ρ+2)</td>
<td>2ρn</td>
<td>ρ</td>
</tr>
<tr>
<td>Min</td>
<td>max{ n/2, n−</td>
<td>t</td>
<td>}</td>
</tr>
<tr>
<td>( \tilde{F}[1] )</td>
<td>2n/3 *</td>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>( \tilde{F}[2] )</td>
<td>n *</td>
<td>n</td>
<td>2</td>
</tr>
</tbody>
</table>

* Information-theoretic model
Towards Complexity-Theoretic Model

\[ \tilde{F}[\alpha] \text{ with ideal cipher } E \quad \text{ideal tweakable cipher } \tilde{\pi} \]

current proof
Towards Complexity-Theoretic Model

$\tilde{F}[\alpha]$ with any cipher $E$

$\tilde{F}[\alpha]$ with ideal cipher $E$

ideal tweakable cipher $\tilde{\pi}$

current proof
Towards Complexity-Theoretic Model

\[ \tilde{F}[\alpha] \] with any cipher \( E \)

\[ \tilde{F}[\alpha] \] with ideal cipher \( E \)

ideal tweakable cipher \( \tilde{\pi} \)

security of \( E \)

current proof
Towards Complexity-Theoretic Model

\[ \tilde{F}[\alpha] \text{ with any cipher } E \]

\[ \tilde{F}[\alpha] \text{ with ideal cipher } E \]

ideal tweakable cipher \( \tilde{\pi} \)

\( \oplus \text{-rk security of } E \)

current proof
Towards Complexity-Theoretic Model

- $\tilde{F}[\alpha]$ with any cipher $E$
- $\tilde{F}[\alpha]$ with ideal cipher $E$
- Ideal tweakable cipher $\tilde{\pi}$

- First step unnecessarily loose
- Tweak change influences key and message input
- Details in paper
Conclusions

\( \tilde{F}[1] \) and \( \tilde{F}[2] \)

- Simple and few primitive calls
- High security level
- Efficient if key renewal is relatively cheap

Future Research

- One-call weakable cipher with improved security?
- Avoiding related-key security condition?
- Implementations?

Thank you for your attention!
Conclusions

$\tilde{F}[1]$ and $\tilde{F}[2]$

- Simple and few primitive calls
- High security level
- Efficient if key renewal is relatively cheap

Future Research

- One-call tweakable cipher with improved security?
- Avoiding related-key security condition?
- Implementations?
Conclusions

\( \tilde{F}[1] \) and \( \tilde{F}[2] \)

- Simple and few primitive calls
- High security level
- Efficient if key renewal is relatively cheap

Future Research

- One-call tweakable cipher with improved security?
- Avoiding related-key security condition?
- Implementations?

Thank you for your attention!
Supporting Slides

SUPPORTING SLIDES
Generic Design: Inverse

Valid Mixing Functions (informal)

$A_i, B_i$ are valid if there is one $A_{i^*}$ that processes $m$, s.t.

- first $i^* - 1$ rounds computable in forward direction
- last $\rho - (i^* - 1)$ rounds computable in inverse direction

both without usage of $m$

Example for $i^* = 2$
Both Designs on One Slide

\[ \tilde{F}[1](k, t, m) = c \]

\[ \tilde{F}[2](k, t, m) = c \]