How to Build Fully Secure Tweakable Blockciphers from Classical Blockciphers

Lei Wang

(joint work with Jian Guo, Guoyan Zhang, Jingyuan Zhao, Dawu Gu)

Shanghai Jiao Tong University

ASK 2016 – Nagoya University, Japan

September 29, 2016
1 Motivation
Outline

1. Motivation

2. Target Construction
Outline

1. Motivation
2. Target Construction
3. Search among Instances

L. Wang, SJTU

How to Build Fully Secure TBCs

ASK 2016 — Nagoya
Outline

1. Motivation
2. Target Construction
3. Search among Instances
4. Provable Security
Outline

1. Motivation
2. Target Construction
3. Search among Instances
4. Provable Security
5. Conclusion
Tweakable Blockcipher (TBC)

- additional parameter: public tweak $t$
- more natural primitive for modes of operation
  - disk encryption, authenticated encryption, etc
- all wires have a size of $n$ bits

Goals of this work

- Find TBCs that can achieve full $2^n$ provable security

Classical blockcipher

Tweakable blockcipher
Tweakable Blockcipher (TBC)

- additional parameter: public tweak $t$
- more natural primitive for modes of operation
  - disk encryption, authenticated encryption, etc
- all wires have a size of $n$ bits

Goal of this work

Find TBCs that can achieve full $2^n$ provable security
Three Approaches to Build TBCs

from the scratch

- Hasty pudding cipher [S98], Mercy [C00], Threelfish [FLS+08]
- a drawback: no security proof
Three Approaches to Build TBCs

from the scratch

- Hasty pudding cipher [S98], Mercy [C00], Threefish [FLS+08]
- a drawback: no security proof

from blockcipher constructions

- tweak luby-rackoff [GHL+07], generalized feistel [MI08], key-alternating [JNP14,CLS15], etc
- provable security bound: (at most) $2^{2n/3}$ [CLS15]
- still far from full $2^n$ provable security
Three Approaches to Build TBCs

from blockcipher as a black-box

- tweak-dependent key (tdk): changing tweak values leads to rekeying blockciphers
- without using tdk
  - LRW1/2 [LRW02], XEX [R04], CLRW2 [LST12], etc
  - asymptotically approach full security [LS13]: $2^{sn/(s+2)}$ security with $s$ blockcipher calls (low efficiency)
  - in the standard model: blockcipher as PRP
- with using tdk
  - Minematsu’s design [M09], Mennink’s design [M15]
  - full $2^n$ provable security [M15]: the only TBC claiming full $2^n$ provable security
  - in the ideal blockcipher model [M15]
Mennink’s Design

- tweak-dependent key
- two blockcipher calls
- full $2^n$ provable security claimed
Mennink’s Design

- tweak-dependent key
- two blockcipher calls
- full $2^n$ provable security claimed

A key-recovery attack can be launched with a birthday-bound complexity
an observation

When \((t, c) = (0, 0)\), it has \(y_1 = y_2\), and in turn \(x_2 = 0\). Hence, by querying \((t = 0, c = 0)\) to decryption \(\widetilde{F}_2^{-1}\), the received \(p = y_1 = E_k(0)\).
Key-recovery Attack on Mennink’s Design $\tilde{F}_2$

an observation
When $(t, c) = (0, 0)$, it has $y_1 = y_2$, and in turn $x_2 = 0$. Hence, by querying $(t = 0, c = 0)$ to decryption $\tilde{F}_2^{-1}$, the received $p = y_1 = E_k(0)$.

recover $E(k \oplus t, \text{const})$ for any $t$

1. query $(0, E(k, 0) \oplus t)$ to $\tilde{F}_2$, get $c$, and compute $E(k, t) = c \oplus E(k, 0)$;
2. query $(t, E(k, t) \oplus \text{const})$ to $\tilde{F}_2$, get $c$ and compute $E(k \oplus t, \text{const}) = c \oplus E(k, t)$.
an observation

When \((t, c) = (0, 0)\), it has \(y_1 = y_2\), and in turn \(x_2 = 0\). Hence, by querying \((t = 0, c = 0)\) to decryption \(\tilde{F}_2^{-1}\), the received \(p = y_1 = E_k(0)\).

recover \(E(k \oplus t, \text{const})\) for any \(t\)

1. query \((0, E(k, 0) \oplus t)\) to \(\tilde{F}_2\), get \(c\), and compute \(E(k, t) = c \oplus E(k, 0)\);
2. query \((t, E(k, t) \oplus \text{const})\) to \(\tilde{F}_2\), get \(c\) and compute \(E(k \oplus t, \text{const}) = c \oplus E(k, t)\).

recover the key by a meet-in-the-middle procedure

**Online.** recover \(E(k \oplus t, \text{const})\) for \(2^{n/2}\) tweaks \(t\);

**Offline.** compute \(E(l, \text{const})\) for \(2^{n/2}\) values \(l\);

**MitM.** recover \(k = l \oplus t\) from \(E(k \oplus t, \text{const}) = E(l, \text{const})\).
Motivation of this work

Are there tweakable blockciphers that can achieve full $2^n$ provable security (even in the ideal blockcipher model)?
Remark on Flaw and Patch of $\tilde{F}_2$

a small flaw in the original proof

In the proof, under the condition that the attacker cannot guess the key correctly (that is, (12a) defined in [M15] is not set), it claimed that the distribution of $y_1$ is independent from $y_2$. However, when the tweak $t = 0$, both the two blockcipher calls share the same key, and therefore the distribution of their outputs are highly related.

patched $\tilde{F}_2$ by the designer: full $2^n$ provable security
Outline

1. Motivation
2. Target Construction
3. Search among Instances
4. Provable Security
5. Conclusion
The Target Construction

- \( a_{i,j}, b_{i,j} \in \{0, 1\} \)
- simple XORs as linear mixing
- this talk focuses on the case of two blockcipher calls
  - one blockcipher call with linear mixings can reach at most birthday-bound security [M15]

\[
\begin{align*}
  a_{1,1} \cdot k & \quad a_{1,2} \cdot t \\
  a_{2,1} \cdot k & \quad a_{2,2} \cdot t \quad a_{2,3} \cdot y_1 \\
  b_{1,1} \cdot k & \quad b_{2,4} \cdot y_1 \\
  b_{1,2} \cdot t & \quad b_{2,1} \cdot k \\
  b_{1,3} \cdot p & \quad b_{2,2} \cdot t \\
  b_{2,3} \cdot p & \\
  \quad b_{3,1} \cdot k \\
  \quad b_{3,2} \cdot t \\
  \quad b_{3,3} \cdot p \\
  \quad b_{3,4} \cdot y_1 \\
  \quad c
\end{align*}
\]
Invertibility of Target Construction

Constraint 1

plaintext $p$ must be used in exactly one linear mixing. Thus, one of $\{b_{3,1}, b_{3,2}, b_{3,3}\}$ is 1, and the other two are 0.
Invertibility of Target Construction

**Constraint 1**

plaintext $p$ must be used in exactly one linear mixing. Thus, one of \{ $b_{3,1}$, $b_{3,2}$, $b_{3,3}$ \} is 1, and the other two are 0.

**Constraint 2**

if $y_1$ is computed depending on plaintext $p$, it must not be used to compute $z_2$. Thus, if $b_{1,3} = 1$, $a_{2,3}$ must be 0.
Invertibility of Target Construction

**Constraint 1**

plaintext \( p \) must be used in exactly one linear mixing. Thus, one of \( \{b_{3,1}, b_{3,2}, b_{3,3}\} \) is 1, and the other two are 0.

**Constraint 2**

if \( y_1 \) is computed depending on plaintext \( p \), it must not be used to compute \( z_2 \). Thus, if \( b_{1,3} = 1 \), \( a_{2,3} \) must be 0.

**Constraint 3**

if both \( y_1 \) and \( y_2 \) are computed depending on plaintext \( p \), they must not be used both as inputs to the final linear mixing. Thus, if \( b_{1,3} \) and \( b_{2,4} \) are 1, \( b_{3,4} \) must be 0.
Invertibility of Target Construction

Constraint 1

plaintext $p$ must be used in exactly one linear mixing. Thus, one of \{ $b_{3,1}$, $b_{3,2}$, $b_{3,3}$ \} is 1, and the other two are 0.

Constraint 2

if $y_1$ is computed depending on plaintext $p$, it must not be used to compute $z_2$. Thus, if $b_{1,3} = 1$, $a_{2,3}$ must be 0.

Constraint 3

if both $y_1$ and $y_2$ are computed depending on plaintext $p$, they must not be used both as inputs to the final linear mixing. Thus, if $b_{1,3}$ and $b_{2,4}$ are 1, $b_{3,4}$ must be 0.

Others

we always assume both blockciphers are indeed involved in the encryption/decryption process.
Design Goal

- first and top-priority goal: full $2^n$ provable security
- second goal: the minimum number of blockcipher calls
- third goal: (comparably) high efficiency of changing a tweak
  - start with (at most) one tweak-dependent key
1. Motivation
2. Target Construction
3. Search among Instances
4. Provable Security
5. Conclusion
Three Types of Instances

According to the position of plaintext $p$ (Constraint 1)

- Type I: $b_{1,3} = 1$, $b_{2,3} = 0$, $b_{3,3} = 0$
- Type II: $b_{1,3} = 0$, $b_{2,3} = 1$, $b_{3,3} = 0$
- Type III: $b_{1,3} = 0$, $b_{2,3} = 0$, $b_{3,3} = 1$

Constraint 1

plaintext $p$ must be used in exactly one linear mixing. Thus, one of \{ $b_{3,1}$, $b_{3,2}$, $b_{3,3}$ \} is 1, and the other two are 0.
Type I

divided into two cases

Case (1). \( z_1 \) is a tweak-dependent key

Case (2). \( z_2 \) is a tweak-dependent key

* each case is divided into 4 subcases depending on \((a_{1,1}, b_{1,1})\).

\[
\begin{align*}
\oplus & \quad a_{1,1} \cdot k & \oplus & \quad a_{1,2} \cdot t \\
\oplus & \quad a_{2,1} \cdot k & \oplus & \quad a_{2,2} \cdot t \\
\oplus & \quad b_{1,1} \cdot k & \oplus & \quad b_{1,1} \cdot k \\
\oplus & \quad b_{2,1} \cdot k & \oplus & \quad b_{3,1} \cdot k \\
\oplus & \quad b_{2,2} \cdot t & \oplus & \quad b_{3,2} \cdot t \\
\oplus & \quad \text{E} & \oplus & \quad \text{E} \\
x_1 & \quad z_1 & y_1 & \quad x_2 & \quad z_2 & y_2 \\
p & \quad & & & & c
\end{align*}
\]
Type I

divided into two cases

Case (1). \( z_1 \) is a tweak-dependent key
Case (2). \( z_2 \) is a tweak-dependent key

* each case is divided into 4 subcases depending on \((a_{1,1}, b_{1,1})\).

\[
\begin{align*}
\text{Case (1):} & \quad a_1,1 \cdot k + a_1,2 \cdot t \quad \text{and} \quad b_1,1 \cdot k + b_1,2 \cdot t \\
\text{Case (2):} & \quad a_2,1 \cdot k + a_2,2 \cdot t \\
& \quad b_2,1 \cdot k + b_2,2 \cdot t \\
& \quad b_3,1 \cdot k
\end{align*}
\]

search result

Type I instances with one tweak-dependent key have at most birthday-bound security.
Subcase (1.1) as an example

- \((a_{1,1}, b_{1,1}) = (0, 0)\);
- the first blockcipher call is independent from \(k\);
- \(y_1\) can be obtained by querying \(E(\cdot, \cdot)\), and hence essentially one blockcipher call in attackers’ view;
- at most birthday-bound security [M15]
Subcase (1.2) as an example

- \((a_{1,1}, b_{1,1}) = (0, 1)\)

an observation

For any pair \((t, p, c)\) and \((t', p', c')\), it has that \(c = c'\) implies \(y_1 \oplus y'_1 = b_{2,2} \cdot (t \oplus t')\).
Subcase (1.2) as an example

recover $k$ by a meet-in-the-middle procedure

- fix two distinct tweaks $t$ and $t'$;
- **Online.** collect $E(t, p \oplus k) \oplus E(t', p \oplus k)$ for $2^{n/2}$ distinct plaintexts $p$;
- **Offline.** collect $E(t, l) \oplus E(t', l)$ for $2^{n/2}$ distinct values $l$;
- **MitM.** compute $k = p \oplus l$ from an online/offline collision

![Diagram showing the process](image-url)
Type II

- two cases depending on $z_1$ or $z_2$ as a tweak-dependent key;
- each case is further divided into several subcases;
- 32 instances that no attack can be found
32 Plausible TBCs

\( \tilde{E}_1 \)

\( \tilde{E}_2 \)

\( \tilde{E}_3 \)

\( \tilde{E}_4 \)

\( \tilde{E}_5 \)

\( \tilde{E}_6 \)

\( \tilde{E}_7 \)

\( \tilde{E}_8 \)
32 Plausible TBCs

E9

E10

E11

E12

E13

E14

E15

E16
32 Plausible TBCs

- **E17**
  
- **E18**
  
- **E19**
  
- **E20**
  
- **E21**
  
- **E22**
  
- **E23**
  
- **E24**
32 Plausible TBCs

$k \oplus t \oplus y$

$E25$

$k \oplus y$

$E26$

$k \oplus y$

$E27$

$k \oplus y$

$E28$

$k \oplus y$

$E29$

$k \oplus y$

$E30$

$k \oplus y$

$E31$

$k \oplus y$

$E32$
Type III

- plaintext \( p \) and ciphertext \( c \) are \textit{linearly} related. Hence Type III instances are not secure.

\[
\begin{align*}
\oplus & \quad a_{1,1} \cdot k \quad a_{1,2} \cdot t \\
& \quad \downarrow \quad \downarrow \\
& \quad x_1 \quad b_{2,4} \cdot y_1 \\
& \quad \downarrow \quad \downarrow \\
& \quad b_{1,1} \cdot k \quad b_{2,1} \cdot k \\
& \quad \downarrow \quad \downarrow \\
& \quad b_{1,2} \cdot t \quad b_{2,2} \cdot t \\
\oplus & \quad a_{2,1} \cdot k \\
& \quad \downarrow \\
& \quad a_{2,2} \cdot t \\
& \quad \downarrow \\
& \quad a_{2,3} \cdot y_1 \\
\oplus & \quad b_{3,1} \cdot k \\
& \quad \downarrow \\
& \quad b_{3,2} \cdot t \\
& \quad \downarrow \\
& \quad b_{3,4} \cdot y_1 \\
\oplus & \quad b_{3,5} \cdot y_2 \\
& \quad \downarrow \\
& \quad c
\end{align*}
\]
Outline

1. Motivation
2. Target Construction
3. Search among Instances
4. Provable Security
5. Conclusion
Theorem

Let $\tilde{E}$ be any tweakable blockcipher construction from the set of $\tilde{E}_1, \ldots, \tilde{E}_{32}$. Let $q$ be an integer such that $q < 2^{n-1}$. Then the following bound holds.

$$\text{Adv}_{\tilde{E}}^{\text{sprp}}(q) \leq \frac{10q}{2^n}.$$
Proof Sketch for $\tilde{E}_1$

- the h-coefficient technique [P08, CS14]
- release $k$ and $y = E(k, 0)$ to the distinguisher after the interaction and before the final decision
- distinguisher gets all the input-output tuples of $E$ during the interaction, including
  - $\{(z, x, y) : E(z, x) = y\}$ from queries to $\tilde{E}_1$
  - $\{(l, u, v) : E(l, u) = v\}$ from queries to $E$
- if there is no $(z, x, y) = (l, u, v)$, the distinguisher fails.

\[ E_k \oplus t \oplus p \oplus k \rightarrow c \]

\[ E_l \rightarrow v \]
Outline

1. Motivation
2. Target Construction
3. Search among Instances
4. Provable Security
5. Conclusion
We find 32 TBCs with full $2^n$ provable security

- each TBC uses two blockcipher calls
- save one blockcipher call by precomputing and storing the subkey
- in the ideal blockcipher model

<table>
<thead>
<tr>
<th>tweakable blockciphers</th>
<th>key size</th>
<th>security $(\log_2)$</th>
<th>cost $E$ $\otimes/h$</th>
<th>tdk</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRW1</td>
<td>$n$</td>
<td>$n/2$</td>
<td>1</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>LRW2</td>
<td>$2n$</td>
<td>$n/2$</td>
<td>1</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>XEX</td>
<td>$n$</td>
<td>$n/2$</td>
<td>1</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>LRW2[2]</td>
<td>$4n$</td>
<td>$2n/3$</td>
<td>2</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>LRW2[s]</td>
<td>$2sn$</td>
<td>$sn/(s+2)$</td>
<td>$s$</td>
<td>$s$</td>
<td>N</td>
</tr>
<tr>
<td>Min</td>
<td>$n$</td>
<td>$\max{n/2, n -</td>
<td>t</td>
<td>}$</td>
<td>2</td>
</tr>
<tr>
<td>$\tilde{F}[1]$</td>
<td>$n$</td>
<td>$2n/3$</td>
<td>1</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>$\tilde{F}[2]$</td>
<td>$n$</td>
<td>$n/2$</td>
<td>2</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>patched $\tilde{F}[2]$</td>
<td>$n$</td>
<td>$n$</td>
<td>2</td>
<td>0</td>
<td>Y</td>
</tr>
</tbody>
</table>

$\otimes/h$ stands for multiplications or universal hashes;

tdk stands for the tweak-dependent key. ‘N’ refers to not using tdk, and ‘Y’ refers to using tdk;

$|t|$ stands for the bit length of the tweak;
thank you for your attention