Tweakable HCTR: A BBB Secure Tweakable Enciphering Scheme

Name: Avijit Dutta and Mridul Nandi (Indian Statistical Institute, Kolkata, India.)
Date: December 10, 2018

INDOCRYPT, 2018
Outline

* Introduction

* HCTR Construction

* Tweakable HCTR

* Security Proof
Tweakable Enciphering Scheme

TES is a triplet of three algorithms: \( \mathcal{I} = (KGen, Enc, Dec) \).
TES is a triplet of three algorithms: $\mathcal{I} = (\text{KGen}, \text{Enc}, \text{Dec})$.

* Alice and Bob shares a secret key $K \leftarrow \text{KGen}(1^n)$.
* Alice generates the ciphertext $C \leftarrow \text{Enc}(K, M, T)$ and sends $C$ to Bob.
* Bob decrypts the ciphertext $C$ to $M \leftarrow \text{Dec}(K, C, T)$.
Tweakable Enciphering Scheme

TES is a triplet of three algorithms: $\mathcal{I} = (\text{KGen}, \text{Enc}, \text{Dec})$.

* Alice and Bob shares a secret key $K \leftarrow \text{KGen}(1^n)$.
* Alice generates the ciphertext $C \leftarrow \text{Enc}(K, M, T)$ and sends $C$ to Bob.
* Bob decrypts the ciphertext $C$ to $M \leftarrow \text{Dec}(K, C, T)$.

Correctness Condition

$$\forall K, M, T : \text{Dec}(K, \text{Enc}(K, M, T), T) = M.$$
Security Game of TES

Real World

Enc

Dec

Ideal World

\[ \Pi \]

\[ \Pi^{-1} \]

A

I is secure against all such computationally bounded adversary \( A \), if the advantage is small.
\[ \text{Adv}^\text{tes}_I(A) = \left| \Pr[A^{I,\text{Enc},I,\text{Dec}} = 1] - \Pr[A^{\Pi,\Pi^{-1}} = 1] \right|. \]
\[ \text{Adv}_{\mathcal{I}}^{\text{tes}}(A) = | \Pr[A^{\mathcal{I},\text{Enc},\mathcal{I},\text{Dec}} = 1] - \Pr[A^{\Pi,\Pi^{-1}} = 1] |. \]

\(\mathcal{I}\) is secure against all such computationally bounded adversary \(A\), if the advantage is small.
Types of TES

I. Hash-ECB-Hash: PEP, TET, HEH.
II. Encrypt-Mix-Encrypt: CMC, EME, Fmix.
III. Hash-Counter-Hash: XCB, HCTR, HCH, FAST.
Types of TES

1. Hash-ECB-Hash: PEP, TET, HEH.
2. Encrypt-Mix-Encrypt: CMC, EME, EME∗, FMix.
3. Hash-Counter-Hash: XCB, HCTR, HCH, FAST.
>>> Types of TES

I. Hash-ECB-Hash: PEP, TET, HEH.

II. Encrypt-Mix-Encrypt: CMC, EME, EME*, FMix.

III. Hash-Counter-Hash: XCB, HCTR, HCH, FAST.
Types of TES

I. Hash-ECB-Hash: PEP, TET, HEH.

II. Encrypt-Mix-Encrypt: CMC, EME, EME*, FMix.

III. Hash-Counter-Hash: XCB, HCTR, HCH, FAST.
### Comparison Chart for various TES

<table>
<thead>
<tr>
<th>Type</th>
<th>Constructions</th>
<th>Prim. Calls</th>
<th>Mult. Calls</th>
<th>Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>PEP ((\ell + 5))</td>
<td>(4(\ell - 6))</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TET ((\ell + 2))</td>
<td>((\ell - 1))</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HEH ((\ell + 2))</td>
<td>2((\ell - 1))</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>CMC (2(\ell + 1))</td>
<td>0</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EME (2(\ell + 2))</td>
<td>0</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EME* (2(\ell + \ell/n + 1))</td>
<td>0</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FMix (2(\ell + 1))</td>
<td>0</td>
<td>(q^2 + \ell^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>XCB ((\ell + 6))</td>
<td>2((\ell + 1))</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HCTR ((\ell))</td>
<td>2((\ell - 1))</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HCH ((\ell + 3))</td>
<td>2((\ell - 2))</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FAST ((\ast)) ((\ell))</td>
<td>2((\ell - 3))</td>
<td>(\sigma^2 / 2^n)</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Constructions</td>
<td>Prim. Calls</td>
<td>Mult. Calls</td>
<td>Bound</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>I</td>
<td>PEP</td>
<td>$(\ell + 5)$</td>
<td>$(4\ell - 6)$</td>
<td>$\sigma^2/2^n$</td>
</tr>
<tr>
<td></td>
<td>TET</td>
<td>$(\ell + 2)$</td>
<td>$(\ell - 1)$</td>
<td>$\sigma^2/2^n$</td>
</tr>
<tr>
<td></td>
<td>HEH</td>
<td>$(\ell + 2)$</td>
<td>$2(\ell - 1)$</td>
<td>$\sigma^2/2^n$</td>
</tr>
<tr>
<td>II</td>
<td>CMC</td>
<td>$(2\ell + 1)$</td>
<td>0</td>
<td>$\sigma^2/2^n$</td>
</tr>
<tr>
<td></td>
<td>EME</td>
<td>$(2\ell + 2)$</td>
<td>0</td>
<td>$\sigma^2/2^n$</td>
</tr>
<tr>
<td></td>
<td>EME*</td>
<td>$(2\ell + \ell/n + 1)$</td>
<td>0</td>
<td>$\sigma^2/2^n$</td>
</tr>
<tr>
<td></td>
<td>FMix</td>
<td>$(2\ell + 1)$</td>
<td>0</td>
<td>$q^2 + \ell^2/2^n$</td>
</tr>
<tr>
<td>III</td>
<td>XCB</td>
<td>$(\ell + 6)$</td>
<td>$2(\ell + 1)$</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>HCTR</td>
<td>$\ell$</td>
<td>$2(\ell - 1)$</td>
<td>$\sigma^2/2^n$</td>
</tr>
<tr>
<td></td>
<td>HCH</td>
<td>$(\ell + 3)$</td>
<td>$2(\ell - 2)$</td>
<td>$\sigma^2/2^n$</td>
</tr>
<tr>
<td></td>
<td>FAST (⋆)</td>
<td>$\ell$</td>
<td>$2(\ell - 3)$</td>
<td>$\sigma^2/2^n$</td>
</tr>
</tbody>
</table>

⋆ FAST is a PRF based TES.
*** Key Points

* Only CMC, EME and EME* are based only on block ciphers.
* Field Multiplication for other constructions.
* HCTR is the efficient one \(^1\).
* Secured upto the birthday bound.

\(^1\)Lopez et al., INDOCRYPT 07.
* Most efficient candidate.
Most efficient candidate.

Sec. bound $\sigma^3/2^n$ [Wang et al.].
* Most efficient candidate.

* Sec. bound $\sigma^3/2^n$ [Wang et al.].

* Improved bound $\sigma^2/2^n$ [Chakraborty and Nandi, FSE 08].
* Most efficient candidate.

* Sec. bound $\sigma^3/2^n$ [Wang et al.].

* Improved bound $\sigma^2/2^n$ [Chakraborty and Nandi, FSE 08].

* The bound is tight (due to hash collision).
* Most efficient candidate.

* Sec. bound $\sigma^3/2^n$ [Wang et al.].

* Improved bound $\sigma^2/2^n$ [Chakraborty and Nandi, FSE 08].

* The bound is tight (due to hash collision).

Can we construct a BBB secure variant of HCTR?
Naive Approach to construct BBB Secure variant of HCTR

[2. HCTR Construction]
Drawback of the Scheme

* 2 * field multiplications.
>>> Drawback of the Scheme

* 2 * field multiplications.

* 2 * state size.
* 2 * field multiplications.
* 2 * state size.
* Optimally secure SPRP requires at least 6 BC calls.
>>> Drawback of the Scheme

* 2 * field multiplications.
* 2 * state size.
* Optimally secure SPRP requires at least 6 BC calls.
* Ctr mode requires 2 BC calls per block of message.
>>> Drawback of the Scheme

* 2 * field multiplications.
* 2 * state size.
* Optimally secure SPRP requires at least 6 BC calls.
* Ctr mode requires 2 BC calls per block of message.

Can we make the scheme BBB secure without increasing state size?
Tweakable HCTR
* BC → m-bit TBC.
* BC → m-bit TBC.
* $H'_L(T)$ for processing variable length tweak.
* BC → m-bit TBC.
* $H'_L(T)$ for processing variable length tweak.
* First $n$ bit of $H'_L(T)$ is xor-ed with the state.
* BC → m-bit TBC.
* $H'_L(T)$ for processing variable length tweak.
* First $n$ bit of $H'_L(T)$ is xor-ed with the state.
* Remaining $m$ bit of $H'_L(T)$ is the tweak to the underlying TBC.
* Tweakable HCTR

* BC → m-bit TBC.
* $H'_L(T)$ for processing variable length tweak.
* First $n$ bit of $H'_L(T)$ is xor-ed with the state.
* Remaining $m$ bit of $H'_L(T)$ is the tweak to the underlying TBC.
* $n$ bit state.
**Tweakable HCTR**

* BC → $m$-bit TBC.
* $H'_L(T)$ for processing variable length tweak.
* First $n$ bit of $H'_L(T)$ is xor-ed with the state.
* Remaining $m$ bit of $H'_L(T)$ is the tweak to the underlying TBC.
* $n$ bit state.
* Provides graceful security degradation.
Block Cipher vs Tweakable Block Cipher

\[ X \xrightarrow{E_K} Y \approx X \xrightarrow{\Pi} Y \]

Indistinguishable from random permutation.
### Block Cipher vs Tweakable Block Cipher

\[ X \xrightarrow{E_K} Y \quad \approx \quad X \xrightarrow{\Pi} Y \]

Indistinguishable from random permutation.

\[ X \xrightarrow{\tilde{E}_K} Y \quad \approx \quad X \xrightarrow{\tilde{\Pi}} Y \]

Indistinguishable from tweakable random permutation.
### Application of Tweakable Block Cipher

<table>
<thead>
<tr>
<th>Author</th>
<th>Construction</th>
<th>Bound</th>
<th>Publication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rogaway</td>
<td>PMAC1</td>
<td>$n/2$</td>
<td>ASIACRYPT, 04</td>
</tr>
<tr>
<td>Naito</td>
<td>PMAC_TBC3k, PMAC_TBC1k</td>
<td>$n$</td>
<td>Provsec, 15</td>
</tr>
<tr>
<td>Peyrin and Seurin</td>
<td>SCT</td>
<td>Graceful BBB</td>
<td>CRYPTO, 16</td>
</tr>
<tr>
<td>List and Nandi</td>
<td>PMACx, PMAC2x, SIVx</td>
<td></td>
<td>CT-RSA, 17</td>
</tr>
<tr>
<td>Iwata et al.</td>
<td>ZMAC, ZAE</td>
<td></td>
<td>CRYPTO, 17</td>
</tr>
<tr>
<td>List and Nandi</td>
<td>ZMAC+</td>
<td></td>
<td>ToSC, 17</td>
</tr>
<tr>
<td>Chen et al.</td>
<td>LDT</td>
<td>$q(q + \sigma)/2^n + \min{n,t}$</td>
<td>FSE, 18</td>
</tr>
<tr>
<td>Chen et al.</td>
<td>LDT</td>
<td>$n/2$ (*)</td>
<td>ASIACRYPT, 18</td>
</tr>
</tbody>
</table>

(\*) Beyond the birthday bound for domain $[n,3n/2)$.
How $H'_L$ is different from $H_{K_h}$?
How $H'_L$ is different from $H_{K_h}$?

* $H_{K_h}$ requires to be $n$ bit $\epsilon$-AXU and $\epsilon$-almost regular hash function.
How $H'_L$ is different from $H_{K_h}$?

* $H_{K_h}$ requires to be $n$ bit $\epsilon$-AXU and $\epsilon$-almost regular hash function.

  * AXU:

    \[
    \forall M \neq M', \forall \delta, \Pr[H_{K_h}(M) \oplus H_{K_h}(M') = \delta] \leq \epsilon.
    \]

  * Almost Regular:

    \[
    \forall M, \forall \delta, \Pr[H_{K_h}(M) = \delta] \leq \epsilon.
    \]
How $H'_L$ is different from $H_{K_h}$?

* $H_{K_h}$ requires to be $n$ bit $\epsilon$-AXU and $\epsilon$-almost regular hash function.
  
  * AXU:

  \[
  \forall M \neq M', \forall \delta, \Pr[H_{K_h}(M) \oplus H_{K_h}(M') = \delta] \leq \epsilon.
  \]

  * Almost Regular:

  \[
  \forall M, \forall \delta, \Pr[H_{K_h}(M) = \delta] \leq \epsilon.
  \]

* $H'_L$ requires to be $(n + m)$ bit $\epsilon$-partial AXU and $H'_L[2]$ requires to be $\epsilon$-almost universal hash function.
How $H'_L$ is different from $H_{K_h}$?

* $H_{K_h}$ requires to be $n$ bit $\epsilon$-AXU and $\epsilon$-almost regular hash function.
  
  * AXU:
    \[
    \forall M \neq M', \forall \delta, \Pr[H_{K_h}(M) \oplus H_{K_h}(M') = \delta] \leq \epsilon.
    \]
  
  * Almost Regular:
    \[
    \forall M, \forall \delta, \Pr[H_{K_h}(M) = \delta] \leq \epsilon.
    \]

* $H'_L$ requires to be $(n+m)$ bit $\epsilon$-partial AXU and $H'_L[2]$ requires to be $\epsilon$-almost universal hash function.
  
  * partial AXU:
    \[
    \forall M \neq M', \forall \delta, \Pr[H_{K_h}(M) \oplus H_{K_h}(M') = (\delta,0)] \leq \epsilon.
    \]
  
  * Almost Universal:
    \[
    \forall M \neq M', \Pr[H'_L[2](M) = H'_L[2](M')] \leq \epsilon.
    \]
Processing of tweak with $m$ bit Almost Universal Hash

A makes $2^{n/2}$ single block message query with distinct tweaks, i.e., $(M,T_1),\ldots,(M,T_{2^{n/2}})$. 

\[
\begin{align*}
M & \rightarrow \oplus \rightarrow 0 \\
X & \rightarrow H \rightarrow Y \\
T & \rightarrow \tilde{E}_K \\
C & \rightarrow 0
\end{align*}
\]
Processing of tweak with $m$ bit Almost Universal Hash

* A makes $2^{n/2}$ single block message query with distinct tweaks, i.e.,
  $(M, T_1), \ldots, (M, T_{2^{n/2}})$.

* Let $C_i = C_j$ (w.h.p due to $m$-bit hash collision).
Processing of tweak with $m$ bit Almost Universal Hash

Is $C_1 || C_2 = C_1' || C_2'$ ?
Security Result of Tweakable HCTR

Theorem

If $H_{K_h}$ is $\epsilon$-AXU and $\epsilon_1$-almost regular hash function and $H'_L$ be an $(n,m,\delta)$-partial AXU hash function and $H'[2]$ is a $\delta_{au}$-almost universal hash function. Then,

$$\text{Adv}^{\text{tsprp}}_{\text{HCTR}} \leq 2(\mu - 1)(q\epsilon + \sigma/2^n) + 2q\sigma\delta_{au}/2^n + q^2\delta + 2\max\{q\ell(\mu - 1)/2^n + q\sigma\delta_{au}/2^n, \sigma\epsilon_1\}.$$ 

$\mu := \# \text{ of repetition of tweaks}.$
Security Result of Tweakable HCTR

**Theorem**

If $H_{K_h}$ is $\epsilon$-AXU and $\epsilon_1$-almost regular hash function and $H'_L$ be an $(n,m,\delta)$-partial AXU hash function and $H'[2]$ is a $\delta_{au}$-almost universal hash function. Then,

$$\text{Adv}_{\text{HCTR}}^{\text{tSprp}} \leq 2(\mu - 1)(q\epsilon + \sigma/2^n) + 2q\sigma\delta_{au}/2^n + q^2\delta + 2\max\{q\ell(\mu - 1)/2^n + q\sigma\delta_{au}/2^n, \sigma\epsilon_1\}.$$  

$\mu := \# \text{ of repetition of tweaks.}$

**Corollary**

* Assuming $\epsilon, \epsilon_1 \approx 2^{-n}$, $\delta_{au} \approx 2^{-m}, \delta \approx 2^{-(n+m)}$ and $m > n$, security $\approx 2^n/\mu\ell$ queries.
Theorem
If $H_K$ is $\epsilon$-AXU and $\epsilon_1$-almost regular hash function and $H'_L$ be an $(n,m,\delta)$-partial AXU hash function and $H'_2$ is a $\delta_{au}$-almost universal hash function. Then,

$$\text{Adv}^{\text{tsprp}}_{\text{HCTR}} \leq 2(\mu - 1)(q\epsilon + \sigma/2^n) + 2q\sigma\delta_{au}/2^n + q^2\delta$$

$$+ 2 \max\{q\ell(\mu - 1)/2^n + q\sigma\delta_{au}/2^n, \sigma\epsilon_1\}.$$

$\mu := \# \text{ of repetition of tweaks}.$

Corollary

* Assuming $\epsilon, \epsilon_1 \approx 2^{-n}$, $\delta_{au} \approx 2^{-m}, \delta \approx 2^{-(n+m)}$ and $m > n$, security $\approx 2^n/\mu\ell$ queries.

* Moreover, when $\mu = 1$, security $\approx 2^n$ many message blocks.
H-Coefficient Technique

\[
\text{Adv}_{\text{ideal}}^{\text{real}}(A) = | \Pr[A^{T,\text{Enc},T,\text{Dec}_K} = 1] - \Pr[A^{\Pi,\Pi^{-1}} = 1] |.
\]
**H-Coefficient Technique**

\[
\text{Adv}^\text{real}_{\text{ideal}}(A) = | \Pr[A^I.\text{Enc},I.\text{Dec}_K = 1] - \Pr[A^{I,I^{-1}} = 1]|.
\]

* Transcript: \( \tau = (M_1, T_1, C_1), \ldots, (M_q, T_q, C_q) \).
* \( X_{\text{re/id}} := \) probability distribution of transcript in real / ideal world.
* \( \mathcal{V} = \text{GoodT} \sqcup \text{BadT} \).
Main Theorem (H-Coefficient Technique)

If there exists $\epsilon_{\text{ratio}}, \epsilon_{\text{bad}} \geq 0$ such that

(i) for all $\tau \in \text{GoodT}$, $\frac{\Pr[X_{\text{re}}=\tau]}{\Pr[X_{\text{id}}=\tau]} \geq 1 - \epsilon_{\text{ratio}}$ and

(ii) $\Pr[X_{\text{id}} \in \text{BadT}] \leq \epsilon_{\text{bad}}$,

then

$$\text{Adv}_{\text{ideal}}^{\text{real}}(A) \leq \epsilon_{\text{ratio}} + \epsilon_{\text{bad}}.$$
Identifying Bad Event

Input or output collision for same tweak is bad.

* 1. $H_{2,i} = H_{2,j}, X_i = X_j$ or (b) $H_{2,i} = H_{2,j}, Y_i = Y_j$.

* 2. $H_{2,i} = H_{2,j}, IV_a^i = IV_b^j$.

* 3. $H_{2,i} = H_{2,j}, M_a^i \oplus C_a^i = M_b^j \oplus C_b^j$.

* 4. $H_{2,i} = H_{2,j}, X_i = IV_a^j$.

* 5. $H_{2,i} = H_{2,j}, Y_i = M_a^j \oplus C_a^j$. 

[4. Security Proof]
Analysing Bad Event

Analysing Bad Event I:

\[ B := H_{2,i} = H_{2,j}, X_i = X_j \iff H_{2,i} = H_{2,j}, H_{K_h}(\hat{M}_i) \oplus H_{1,i} \oplus M_i = H_{K_h}(\hat{M}_j) \oplus H_{1,j} \oplus M_j \]

where \( \hat{M}_i := M^i_2 \| \ldots \| M^i_{l_i} \), \( \hat{M}_j := M^j_2 \| \ldots \| M^j_{l_j} \).
Analysing Bad Event

Breaking $B := H_{2,i} = H_{2,j}, X_i = X_j \leftrightarrow H_{2,i} = H_{2,j}, H_{K_h}(\hat{M}_i) \oplus H_{1,i} \oplus M_i = H_{K_h}(\hat{M}_j) \oplus H_{1,j} \oplus M_j$

where $\hat{M}_i := M^i_2 \parallel \cdots \parallel M^i_{l_i}, \hat{M}_j := M^j_2 \parallel \cdots \parallel M^j_{l_j}$.

* Case a: $T_i = T_j \leftrightarrow (H_{1,i}, H_{2,i}) = (H_{1,j}, H_{2,j})$.

$$\Pr[B] \leq q(\mu - 1)\epsilon,$$

# of $(i, j) = (q, (\mu - 1))$.  

[4. Security Proof]$ _ { _ { _ { _ } } } $
Analysing Bad Event

B := $H_{2,i} = H_{2,j}, X_i = X_j \Leftrightarrow H_{2,i} = H_{2,j}, H_{K_h}(\hat{M}_i) \oplus H_{1,i} \oplus M_i = H_{K_h}(\hat{M}_j) \oplus H_{1,j} \oplus M_j$

where $\hat{M}_i := M_2^i \parallel \ldots \parallel M_{l_i}^i$, $\hat{M}_j := M_2^j \parallel \ldots \parallel M_{l_j}^j$.

* Case a: $T_i = T_j \Leftrightarrow (H_{1,i}, H_{2,i}) = (H_{1,j}, H_{2,j})$.

$$\Pr[B] \leq q(\mu - 1)\epsilon, \quad \# \text{ of } (i, j) = (q, (\mu - 1)).$$

* Case (b): $T_i \neq T_j$, then

$$\Pr[B] \leq \binom{q}{2}\delta, \quad \# \text{ of } (i, j) = \binom{q}{2}.$$
Analysing Bad Event I:

\[ B := H_{2,i} = H_{2,j}, X_i = X_j \iff H_{2,i} = H_{2,j}, H_{K_h}(\widehat{M_i}) \oplus H_{1,i} \oplus M_i = H_{K_h}(\widehat{M_j}) \oplus H_{1,j} \oplus M_j \]

where \( \widehat{M_i} := M^i_2 \| \ldots \| M^i_{l_i}, \widehat{M_j} := M^j_2 \| \ldots \| M^j_{l_j} \).

* Case a: \( T_i = T_j \iff (H_{1,i}, H_{2,i}) = (H_{1,j}, H_{2,j}) \).

\[ \Pr[B] \leq q(\mu - 1)\epsilon, \quad \# \text{ of } (i, j) = (q, (\mu - 1)). \]

* Case (b): \( T_i \neq T_j \), then

\[ \Pr[B] \leq \binom{q}{2}\delta, \quad \# \text{ of } (i, j) = \binom{q}{2}. \]

Similarly, we can bound the other bad events too.
## Summarizing the proof

### Summary of the Bad Events Bound

<table>
<thead>
<tr>
<th>Bad Event</th>
<th>Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>$2(\mu - 1)q\epsilon + q^2\delta$</td>
</tr>
<tr>
<td>B.2</td>
<td>$\sigma(\mu - 1)/2^n + q\sigma\delta_{au}/2^n$</td>
</tr>
<tr>
<td>B.3</td>
<td>$\sigma(\mu - 1)/2^n + q\sigma\delta_{au}/2^n$</td>
</tr>
<tr>
<td>B.4</td>
<td>$\max{q\ell(\mu - 1)/2^n + q\sigma\delta_{au}/2^n, \sigma\epsilon_1}$</td>
</tr>
<tr>
<td>B.5</td>
<td>$\max{q\ell(\mu - 1)/2^n + q\sigma\delta_{au}/2^n, \sigma\epsilon_1}$</td>
</tr>
</tbody>
</table>
Summarizing the proof

Summary of the Bad Events Bound

<table>
<thead>
<tr>
<th>Bad Event</th>
<th>Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>$2(\mu - 1)q\epsilon + q^2\delta$</td>
</tr>
<tr>
<td>B.2</td>
<td>$\sigma(\mu - 1)/2^n + q\sigma\delta_{au}/2^n$</td>
</tr>
<tr>
<td>B.3</td>
<td>$\sigma(\mu - 1)/2^n + q\sigma\delta_{au}/2^n$</td>
</tr>
<tr>
<td>B.4</td>
<td>$\max{q\ell(\mu - 1)/2^n + q\sigma\delta_{au}/2^n, \sigma\epsilon_1}$</td>
</tr>
<tr>
<td>B.5</td>
<td>$\max{q\ell(\mu - 1)/2^n + q\sigma\delta_{au}/2^n, \sigma\epsilon_1}$</td>
</tr>
</tbody>
</table>

If Bad Events do not happen then for two same tweak to the BC, either its input or output is fresh. Therefore,

$$\frac{\Pr[X_{re} = \tau]}{\Pr[X_{id} = \tau]} \geq 1.$$
Conclusion & Future Work

* Tweakable Enciphering Scheme with Tweakable Block Cipher

* Mode with $n$ bit state.

* Provides BBB security with graceful degradation

BBB Secure TES using BC (other than trivial option) is still open.
Conclusion & Future Work

* Tweakable Enciphering Scheme with Tweakable Block Cipher
* Mode with $n$ bit state.
* Provides BBB security with graceful degradation

BBB Secure TES using BC (other than trivial option) is still open.

Thank You