DbHtS: A Paradigm for Constructing BBB Secure PRF

Nilanjan Datta, Avijit Dutta, Mridul Nandi and Goutam Paul

Indian Statistical Institute, Kolkata, India

FSE, 2019
Introduction

- **Symmetric cryptography**: Alice and Bob shares the same key.
- **Active attacker**: Eve might intercept and manipulate Alice’s message.
- **Authentication**: Alice computes and appends a tag. Bob recomputes tag and matches with the received tag.

```
"I accept" || T
```

Valid Tag: Read Message.
Introduction

- **Verifying:** Bob verifies the tag with the shared key and only reads the message if tags match.
- ** Forgery:** Eve cannot modify the message without forging a new and correct tag.
Introduction

- **Verifying:** Bob verifies the tag with the shared key and only reads the message if tags match.

- **Forgery:** Eve cannot modify the message without forging a new and correct tag.

Invalid Tag: Ignore Message.

How can I forge? Define the power and goal of a forgery.
Forgery Security Game

\[ q_t = \text{the number of tagging queries} \]
Forgery Security Game

$q_t = \text{the number of tagging queries}$

$q_v = \text{the number of verification queries}$
Forgery Security Game

Can Eve forge a valid tag for a message that Alice never saw?
Case of ECBC

ECBC Mode

\[ m_1 \oplus E_{k_1} \rightarrow m_2 \oplus E_{k_1} \rightarrow \cdots \rightarrow m_{\ell} \oplus E_{k_1} \rightarrow \Sigma(m) \oplus E_{k_2} \rightarrow T \]

Properties of ECBC: For all messages \( m, m', c \)

\[
\begin{align*}
\text{MAC}(m) &= \text{MAC}(m') \\
\iff E_{k_2}(\Sigma(m)) &= E_{k_2}(\Sigma(m')) \\
\iff \Sigma(m) &= \Sigma(m') \\
\iff \Sigma(m \parallel c) &= \Sigma(m' \parallel c) \\
\text{MAC}(m \parallel c) &= \text{MAC}(m' \parallel c)
\end{align*}
\]
Case of ECBC

**ECBC Mode**

\[ E_{k_1} \rightarrow E_{k_1} \rightarrow \cdots \rightarrow E_{k_1} \rightarrow E_{k_2} \]

\( m_1 \rightarrow m_2 \rightarrow \cdots \rightarrow m_\ell \rightarrow \Sigma(m) \rightarrow T \)

**Properties of ECBC:** For all messages \( m, m', c \)

\[
\begin{align*}
\text{MAC}(m) &= \text{MAC}(m') \\
\iff E_{k_2}(\Sigma(m)) &= E_{k_2}(\Sigma(m')) \\
\iff \Sigma(m) &= \Sigma(m') \\
\iff \Sigma(m || c) &= \Sigma(m' || c) \\
\text{MAC}(m || c) &= \text{MAC}(m' || c)
\end{align*}
\]

**Expansion Property**

Look for a pair of messages \( m, m' \) such that \( \text{MAC}(m) = \text{MAC}(m') \).
Then for all \( c \),

\[
\text{MAC}(m || c) = \text{MAC}(m' || c)
\]
Birthday Bound Attack

Looking for collision

Eve looks for $\text{MAC}(m_i) = \text{MAC}(m_j)$ for some $i \neq j$. She has $\sim q_t^2$ pairs for an $n$-bit relationship so chances grow as

$$\text{Adv}(A) \sim \frac{q_t^2}{2^n}$$
Forgery from collision

Expansion property

\[ \text{MAC}(m) = \text{MAC}(m') \Rightarrow \text{MAC}(m\|c) = \text{MAC}(m'\|c), \forall c. \]

Collision found:

\[ \text{MAC}(I \text{ accept}) = \text{MAC}(I \text{ reject}) \]
Forgery from collision

**Expansion property**

\[
\text{MAC}(m) = \text{MAC}(m') \Rightarrow \text{MAC}(m\|c) = \text{MAC}(m'\|c), \forall c.
\]

Tell Bob your review.

Oh You are right!

Collision found:

\[
\text{MAC(I accept)} = \text{MAC(I reject)}
\]
Forgery from collision

**Expansion property**

\[ \text{MAC}(m) = \text{MAC}(m') \Rightarrow \text{MAC}(m||c) = \text{MAC}(m'||c), \forall c. \]

Collision found:
\[ \text{MAC}(\text{I accept}) = \text{MAC}(\text{I reject}) \]

“I accept your paper” \( || \) \( T \)

“I reject your paper” \( || \) \( T \)
Forgery from collision

Expansion property

\[ \text{MAC}(m) = \text{MAC}(m') \Rightarrow \text{MAC}(m\|c) = \text{MAC}(m'\|c), \forall c. \]

Collision found:
\[ \text{MAC(}\text{I accept}) = \text{MAC(}\text{I reject}) \]

Forgery requires \( q_t \approx 2^{n/2} \) and \( q_v = 1 \)

Not secure beyond birthday bound \( (2^{n/2}) \)
Why Beyond Birthday Security?

- BBB security is useful in lightweight cryptography
- Consider the security advantage $\epsilon = 2^{-10}$, $n = 64$ and $\ell = 2^{16}$ blocks.

<table>
<thead>
<tr>
<th>Construction</th>
<th>Security</th>
<th># of queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECBC</td>
<td>$16q_t^2/2^n$</td>
<td>$\approx 2^{25}$</td>
</tr>
<tr>
<td>PMAC</td>
<td>$5\ell q_t^2/2^n$</td>
<td>$\approx 2^{18}$</td>
</tr>
</tbody>
</table>

Table: Data limit of constructions achieving birthday bound security.

BBB security allows to process larger number of blocks per session key.
Summary So Far

- Forgery Game of Message Authentication Code
- Birthday Bound Forgery for ECBC MAC.
- Birthday Bound is not suitable for small block cipher based MAC

Coming Up: How to get BBB secure MAC.
SUM-ECBC [Yasuda, CT-RSA 2010]

- Rate 1/2, sequential
- Four independent BC keys
- Security: $O(q^3\ell^3/2^{2n})$
PMAC_Plus [Yasuda, CRYPTO 2011]

- Rate 1, parallel
- Three independent BC keys
- Security: $O(q^3 \ell^3 / 2^{2n})$
3kf9 [Zhang et al., ASIACRYPT 2012]

Rate 1, sequential
Three independent BC keys
Security: $O(q^3 \ell^3 / 2^{2n})$

* We found the security bound of 3kf9 is incorrect!
LightMAC_Plus [Naito, ASIACRYPT 2017]

- Rate 1, parallel
- Three independent BC keys
- Security: $O(q^3/2^{2n})$

※ First BBB Secure MAC whose security bound is independent of the message length.
Summary so far

- Beyond Birthday Bound deterministic MACs
- These constructions use three block cipher keys.
- All the constructions share a similar design principle

Coming Up: How to get unify the design and give a generic security proof.
Abstract view of BBB Secure MACs: Double Block Hash-then-Sum (DbHtS)

Three Keyed

\[ M \xrightarrow{\mathcal{H}_K} \mathcal{H}_K \xrightarrow{\Sigma} E_{K_1} \xrightarrow{\hat{\Sigma}} T \]

Double Block Hash Function

Sum Function
Abstract view of BBB Secure MACs: Double Block Hash-then-Sum (DbHtS)

Two Keyed

\[
\begin{align*}
M & \xrightarrow{\mathcal{H}_{K_h}} \\
\mathcal{H}_{K_h} & \xrightarrow{\sum} E_K \\
\Theta & \xrightarrow{\hat{\Theta}} E_K
\end{align*}
\]
Abstract view of BBB Secure MACs: Double Block Hash-then-Sum (DbHtS)

Single Keyed
Sum function is not a PRF

\[ T_1 = E_{K_1}(X_1) \oplus E_{K_2}(Y_1) \]
Sum function is not a PRF

\[ T_1 = E_{K_1}(X_1) \oplus E_{K_2}(Y_1) \quad T_2 = E_{K_1}(X_1) \oplus E_{K_2}(Y_2) \]
Sum function is not a PRF

\[ T_1 = E_{K_1}(X_1) \oplus E_{K_2}(Y_1) \quad T_2 = E_{K_1}(X_1) \oplus E_{K_2}(Y_2) \]

\[ T_3 = E_{K_1}(X_2) \oplus E_{K_2}(Y_2) \]
Sum function is not a PRF

\[ T_1 = E_{K_1}(X_1) \oplus E_{K_2}(Y_1) \]
\[ T_2 = E_{K_1}(X_1) \oplus E_{K_2}(Y_2) \]
\[ T_4 = E_{K_1}(X_2) \oplus E_{K_2}(Y_1) \]
\[ T_3 = E_{K_1}(X_2) \oplus E_{K_2}(Y_2) \]
(Alternating) Cycle and Path

\((\Sigma_1, \Theta_1)\) \hspace{1cm} \((\Sigma_2, \Theta_2)\)

\((\Sigma_3, \Theta_3)\) \hspace{1cm} \((\Sigma_4, \Theta_4)\)

\((\Sigma_1, \Theta_1)\) \hspace{1cm} \((\Sigma_2, \Theta_2)\)

\((\Sigma_4, \Theta_4)\)
(Alternating) Cycle and Path

AC in the input of sum function makes the sum of its output zero.
Security of DbHtS

\[ \hat{\Sigma} = (\Sigma_1, \ldots, \Sigma_q), \quad \hat{\Theta} = (\Theta_1, \ldots, \Theta_q) \] is called covered if \( \exists i \neq j, i \neq k \) such that

- \( \Sigma_i = \Sigma_j \) and \( \Theta_i = \Theta_j \) \( \Rightarrow \) AC2
- \( \Sigma_i = \Sigma_j \) and \( \Theta_i = \Theta_k \) \( \Rightarrow \) AP3

If \( \mathcal{H} \) holds either of the above two conditions, it is called covered DbH.
Security of DbHtS

Alternating cycle in $\tilde{\Sigma}$, $\tilde{\Theta}$ makes the sum of $T_i$’s zero.
Security of DbHtS

Alternating cycle in $\tilde{\Sigma}, \tilde{\Theta}$ makes the sum of $T_i$’s zero.

Avoid alternating cycle in $\tilde{\Sigma}, \tilde{\Theta}$.

**Bad Event (CF)**

- $\exists i \neq j$ such that $\Sigma_i = \Sigma_j$ and $\Theta_i = \Theta_j$ (AC2).
Security of DbHtS

Alternating cycle in $\tilde{\Sigma}$, $\tilde{\Theta}$ makes the sum of $T_i$’s zero.

Avoid alternating cycle in $\tilde{\Sigma}$, $\tilde{\Theta}$.

Bad Event (CF)

- $\exists i \neq j \neq k$ such that $\Sigma_i = \Sigma_j$ and $\Theta_i = \Theta_k$. (AP3)
Security of DbHtS

\[ M_i \xrightarrow{\mathcal{H}_{K_h}} E_{K_1} \xrightarrow{\Sigma_i} E_{K_2} \xrightarrow{\Theta_i} T_i \]

\[ M_j \xrightarrow{\mathcal{H}_{K_h}} E_{K_1} \xrightarrow{\Sigma_j} E_{K_2} \xrightarrow{\Theta_j} T_j \]

\[ M_k \xrightarrow{\mathcal{H}_{K_h}} E_{K_1} \xrightarrow{\Sigma_k} E_{K_2} \xrightarrow{\Theta_k} T_k \]

**Bad Event (RC1)**

\[ \exists i \neq j, i \neq k \text{ such that } \Sigma_i = \Sigma_j \text{ and } \hat{\Theta}_i = \hat{\Theta}_k. \]
Security of DbHtS

\[ M_i \rightarrow H_{K_h} \rightarrow E_{K_1} \rightarrow \Theta_i \rightarrow \widehat{\Theta}_i \rightarrow T_i \]

\[ M_j \rightarrow H_{K_h} \rightarrow E_{K_1} \rightarrow \Theta_j \rightarrow \widehat{\Theta}_j \rightarrow T_j \]

\[ M_k \rightarrow H_{K_h} \rightarrow E_{K_1} \rightarrow \Theta_k \rightarrow \widehat{\Theta}_k \rightarrow T_k \]

Bad Event (RC2)

\[ \exists i \neq j, i \neq k \text{ such that } \Theta_i = \Theta_j \text{ and } \widehat{\Sigma}_i = \widehat{\Sigma}_k. \]
Security of DbHtS

**Bad Tuple**

\((\tilde{\Sigma}, \tilde{\Theta}, \tilde{\Sigma}, \tilde{\Theta})\) is a **bad** tuple if either of CF or RC1 or RC2 holds.
Bad Tuple

\((\tilde{\Sigma}, \tilde{\Theta}, \tilde{\Sigma}, \tilde{\Theta})\) is a bad tuple if either of CF or RC1 or RC2 holds.

Probability of Bad Events.

- \(\Pr[CF] \leq \binom{q}{3} \cdot \Pr[\Sigma_i = \Sigma_j, \Theta_i = \Theta_k] + \frac{q}{2} \cdot \Pr[\Sigma_i = \Sigma_j, \Theta_i = \Theta_j]\)
  
  \[\epsilon_{cf}(3,\ell)\]

- \(\Pr[RC] \leq \frac{q^3}{2^n} \cdot \max\left( \Pr[\Sigma_i = \Sigma_j], \Pr[\Theta_i = \Theta_j] \right)\)
  
  \[\epsilon_{univ}(2,\ell)\]
Security of DbHtS

Bad Tuple

$(\tilde{\Sigma}, \tilde{\Theta}, \tilde{\Sigma}, \tilde{\Theta})$ is a bad tuple if either of CF or RC1 or RC2 holds.

Probability of Bad Events.

- $\Pr[CF] \leq \left( \frac{q}{3} \right) \cdot \Pr[\Sigma_i = \Sigma_j, \Theta_i = \Theta_k] + \left( \frac{q}{2} \right) \cdot \Pr[\Sigma_i = \Sigma_j, \Theta_i = \Theta_j]$
  
  $\epsilon_{\text{cf}}(3,\ell)$

- $\Pr[RC] \leq \frac{q^3}{2^n} \cdot \max \left( \Pr[\Sigma_i = \Sigma_j], \Pr[\Theta_i = \Theta_j] \right)$
  
  $\epsilon_{\text{coll}}$ and $\epsilon_{\text{univ}}(2,\ell)$

Analysis of Good Transcript

We use Sum of Permutation result by [Lucks, Eurocrypt 00].
Security of DbHtS

Summarizing the security:

- if \( \mathcal{H} \) is a \( \epsilon_{\text{cf}}(3, \ell) \) cover free and
- \( \epsilon_{\text{univ}}(2, \ell) \) block-wise universal hash function, then

\[
\text{Adv}_{\text{HtS}}^{\text{prf}}(q, \ell) \leq \binom{q}{3} \cdot \epsilon_{\text{cf}}(3, \ell) + \binom{q}{2} \cdot \epsilon_{\text{coll}} + \frac{q^3}{2^n} \cdot \epsilon_{\text{univ}}(2, \ell) + \frac{4q^3}{3 \cdot 2^{2n}}.
\]

**NOTE:** \( \frac{4q^3}{3 \cdot 2^{2n}} \) comes from sum of permutation result.
Two-keyed DbHtS (with domain separation)

Domain separation enables us to deal with less bad events
Two-keyed DbHtS without domain separation

Without domain separation, one needs to consider the cross collision
Security of two-keyed DbHtS with fix₀, fix₁

Summarizing the security:

- if \( \mathcal{H} \) is a \( \epsilon_{\text{cf}}(3, \ell) \) cover free and
- \( \epsilon_{\text{univ}}(2, \ell) \) block-wise universal block-separated hash function,

then

\[
\text{Adv}^{\text{prf}}_{2K-\text{HtS}}(q, \ell) \leq \left( \frac{q}{3} \right) \cdot \epsilon_{\text{cf}}(3, \ell) + \left( \frac{q}{2} \right) \cdot \epsilon_{\text{coll}} + \frac{q^3}{2^n} \cdot \epsilon_{\text{univ}}(2, \ell) + \frac{q}{2^n} + \frac{6q^3}{2^{2n}}
\]

**Note:** \( \frac{6q^3}{3 \cdot 2^{2n}} \) comes from sum of permutation result.
## Instantiations of three-keyed and two-keyed DbHtS

<table>
<thead>
<tr>
<th>Type</th>
<th>Instantiations</th>
<th>Old Bound</th>
<th>New Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-key DbHtS</td>
<td>SUM-ECBC</td>
<td>$q^3 \ell^4/2^{2n}$</td>
<td>$q\ell^2/2^n + q^3/2^{2n}$</td>
</tr>
<tr>
<td></td>
<td>PMAC_Plus</td>
<td>$q^3 \ell^3/2^{2n} + q\ell/2^n$</td>
<td>$q^3 \ell^3/2^{2n} + q\ell/2^n$</td>
</tr>
<tr>
<td></td>
<td>3kf9</td>
<td>$q^3 \ell^3/2^{2n}$</td>
<td>$q^3 \ell^3/2^{2n}$</td>
</tr>
<tr>
<td></td>
<td>LightMAC_Plus</td>
<td>$q^3/2^{2n}$</td>
<td>$q^3/2^{2n}$</td>
</tr>
<tr>
<td>2-key DbHtS</td>
<td>2K-SUM-ECBC</td>
<td>-</td>
<td>$q\ell^2/2^n + q^3 \ell^2/2^{2n}$</td>
</tr>
<tr>
<td></td>
<td>2K-PMAC_Plus</td>
<td>-</td>
<td>$q^3 \ell/2^{2n} + q^2 \ell^2/2^{2n}$</td>
</tr>
<tr>
<td></td>
<td>2kf9</td>
<td>-</td>
<td>$q^3 \ell^4/2^{2n}$</td>
</tr>
<tr>
<td></td>
<td>2K-LightMAC_Plus</td>
<td>-</td>
<td>$q^3/2^{2n} + q/2^n$</td>
</tr>
</tbody>
</table>
Tightness of the bound

- We have shown security of all the constructions upto $2^{2n/3}$.
- Leurent et al. have shown attack on all these constructions with $2^{3n/4}$ query complexity.
- We believe that the security of all these constructions can be improved upto $2^{3n/4}$. 
Tightness of the bound

- We have shown security of all the constructions upto $2^{2n/3}$.
- Leurent et al. have shown attack on all these constructions with $2^{3n/4}$ query complexity.
- We believe that the security of all these constructions can be improved upto $2^{3n/4}$.

Thank You!