DbHtS: A Paradigm for Constructing BBB Secure PRF

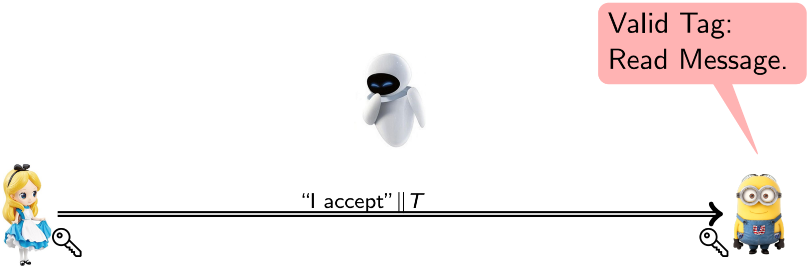
Nilanjan Datta, Avijit Dutta, **Mridul Nandi** and Goutam Paul

Indian Statistical Institute, Kolkata, India

FSE, 2019

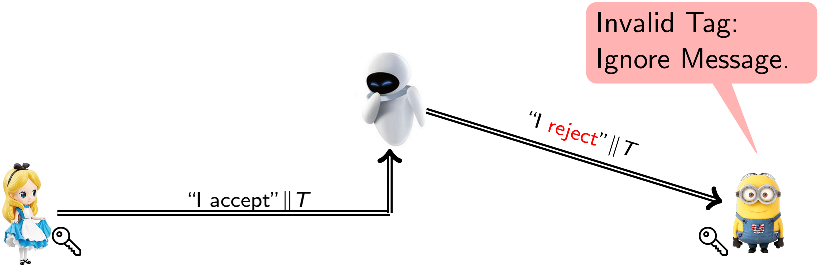
Introduction

- **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.



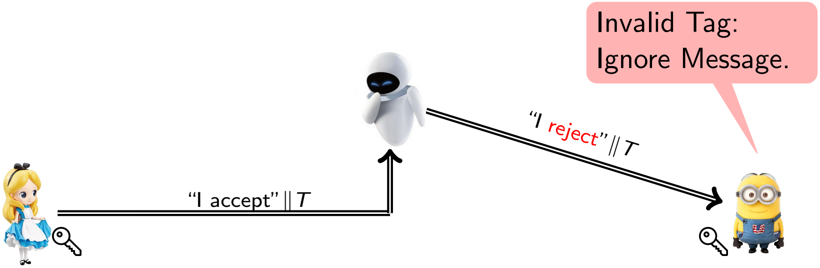
Introduction

- **Verifying:** Bob verifies the tag with the shared key and only reads the message if tags match.
- **Forgery:** Eve cannnot modify the message without forging a new and correct tag.



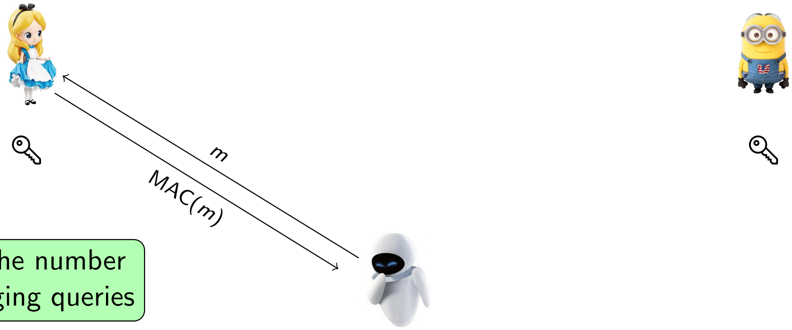
Introduction

- **Verifying:** Bob verifies the tag with the shared key and only reads the message if tags match.
- **Forgery:** Eve cannnot modify the message without forging a new and correct tag.



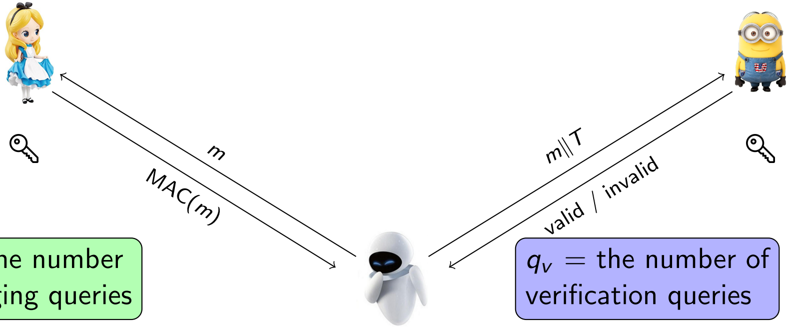
How can I forge ? Define the power and goal of a forgery

Forgery Security Game



q_t = the number of tagging queries

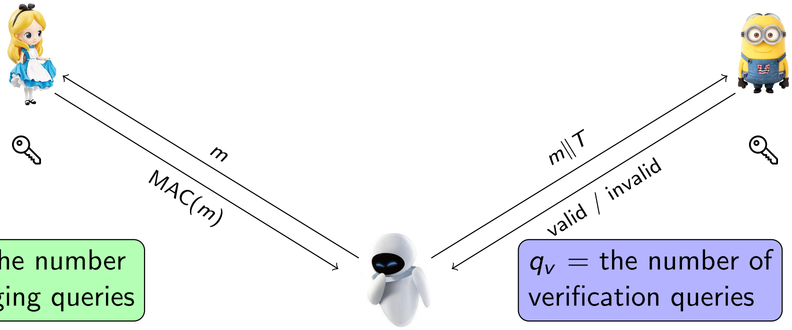
Forgery Security Game



q_t = the number of tagging queries

q_v = the number of verification queries

Forgery Security Game



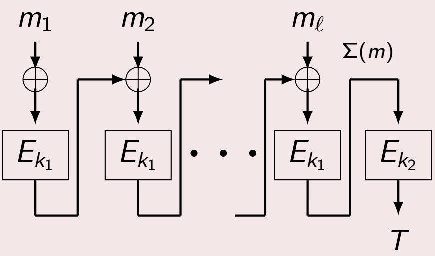
q_t = the number of tagging queries

q_v = the number of verification queries

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

Case of ECBC

ECBC Mode

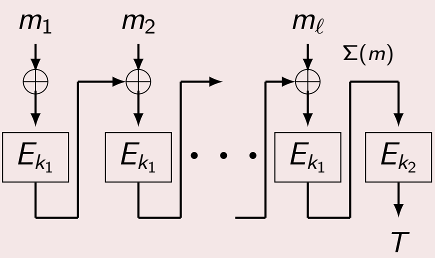


Properties of ECBC: For all messages m, m', c

$$
\begin{aligned}
& \mathsf{MAC}(m) = \mathsf{MAC}(m') \\
\Leftrightarrow\quad & E_{k_2}(\Sigma(m)) = E_{k_2}(\Sigma(m')) \\
\Leftrightarrow\quad & \Sigma(m) = \Sigma(m') \\
\Leftrightarrow\quad & \Sigma(m\|c) = \Sigma(m'\|c) \\
& \mathsf{MAC}(m\|c) = \mathsf{MAC}(m'\|c)
\end{aligned}
$$

Case of ECBC

ECBC Mode



Properties of ECBC: For all messages m, m', c

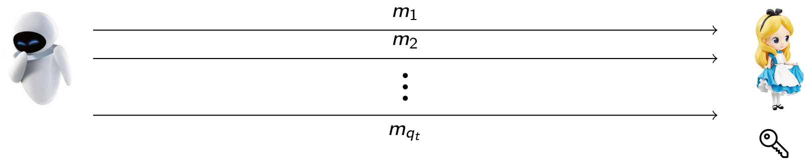
$$\begin{aligned}
& \mathrm{MAC}(m) = \mathrm{MAC}(m') \\
\Leftrightarrow\ & E_{k_2}(\Sigma(m)) = E_{k_2}(\Sigma(m')) \\
\Leftrightarrow\ & \Sigma(m) = \Sigma(m') \\
\Leftrightarrow\ & \Sigma(m\|c) = \Sigma(m'\|c) \\
& \mathrm{MAC}(m\|c) = \mathrm{MAC}(m'\|c)
\end{aligned}$$

Expansion Property

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

$$\mathrm{MAC}(m\|c) = \mathrm{MAC}(m'\|c)$$

Birthday Bound Attack



Looking for collsion

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

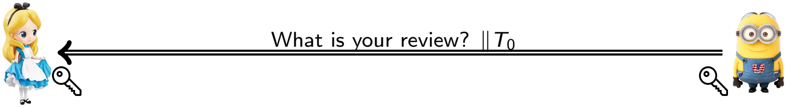
$$\mathbf{Adv}(\mathcal{A}) \simeq \frac{q_t^2}{2^n}$$

Forgery from collision

Expansion property

$$\mathrm{MAC}(m) = \mathrm{MAC}(m') \Rightarrow \mathrm{MAC}(m\|c) = \mathrm{MAC}(m'\|c),\ \forall c.$$

Collision found:
MAC(I accept) =
MAC(I reject)



Forgery from collision

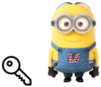
Expansion property

$\mathrm{MAC}(m) = \mathrm{MAC}(m') \Rightarrow \mathrm{MAC}(m \| c) = \mathrm{MAC}(m' \| c), \forall c.$

Collision found:
MAC(I accept) =
MAC(I reject)

Tell Bob your review.

Oh You are right!

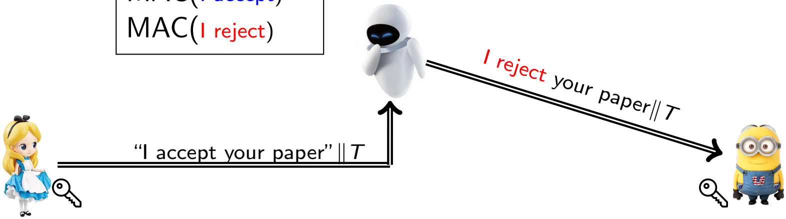


Forgery from collision

Expansion property

$$\mathrm{MAC}(m) = \mathrm{MAC}(m') \Rightarrow \mathrm{MAC}(m\|c) = \mathrm{MAC}(m'\|c),\ \forall c.$$

Collision found:
MAC(I accept) =
MAC(I reject)

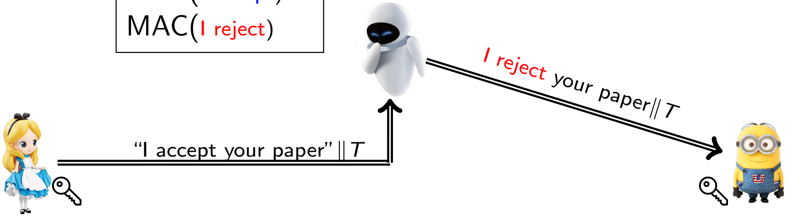


Forgery from collision

Expansion property

$$\mathsf{MAC}(m) = \mathsf{MAC}(m') \Rightarrow \mathsf{MAC}(m \| c) = \mathsf{MAC}(m' \| c),\ \forall c.$$

Collision found:
MAC(I accept) =
MAC(I reject)



Forgery requires $q_t \simeq 2^{n/2}$ 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.

Construction	Security	# of queries
ECBC	$16q_t^2/2^n$	$\approx 2^{25}$
PMAC	$5\ell q_t^2/2^n$	$\approx 2^{18}$

Table : Data limit of constructions acheiving birthday bound security.

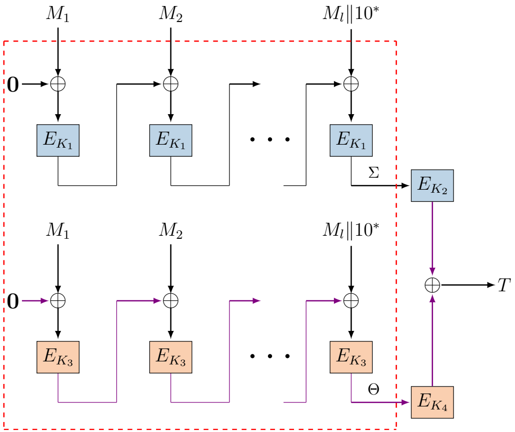
BBB security allows to process larger number of blocks per session key.

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

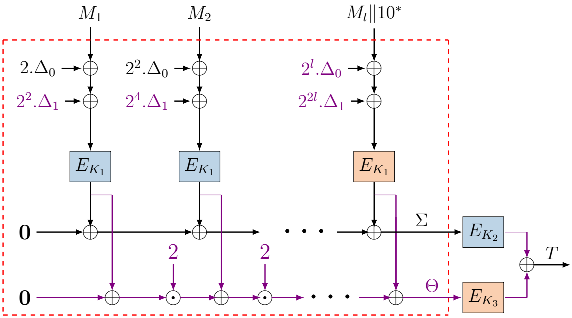
Coming Up: How to get BBB secure MAC.

SUM-ECBC [Yasuda, CT-RSA 2010]



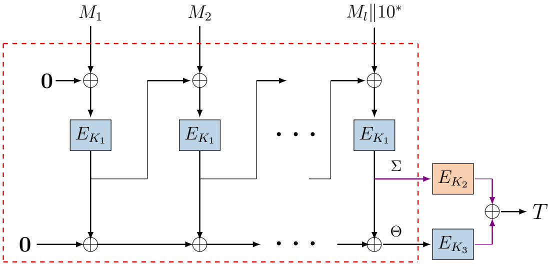
- Rate 1/2, sequential
- Four independent BC keys
- Security: $O(q^3\ell^3/2^{2n})$

PMAC_Plus [Yasuda, CRYPTO 2011]



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

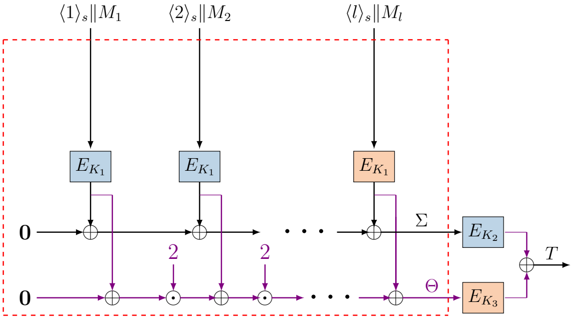
3kf9 [Zhang et al., ASIACRYPT 2012]



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

$\star$ 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.

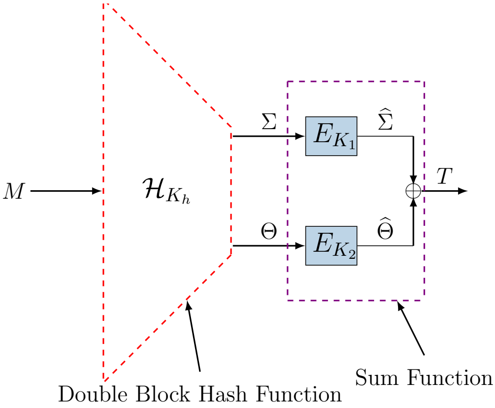
Summary so far

- 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.

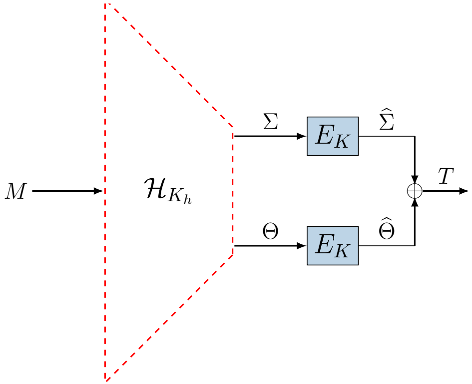
Abstract view of BBB Secure MACs : Double Block Hash-then-Sum (DbHtS)

Three Keyed



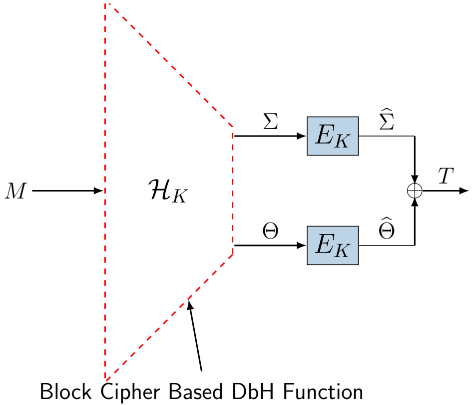
Abstract view of BBB Secure MACs : Double Block Hash-then-Sum (DbHtS)

Two Keyed

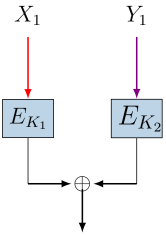


Abstract view of BBB Secure MACs : Double Block Hash-then-Sum (DbHtS)

Single Keyed

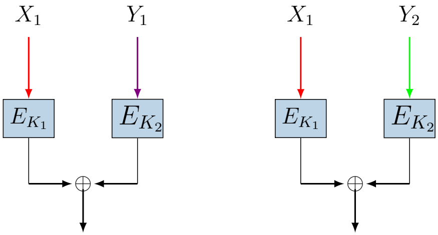


Sum function is not a PRF



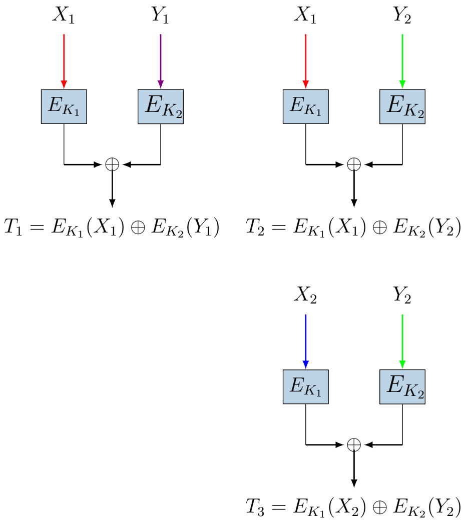
$$T_1 = E_{K_1}(X_1) \oplus E_{K_2}(Y_1)$$

Sum function is not a PRF

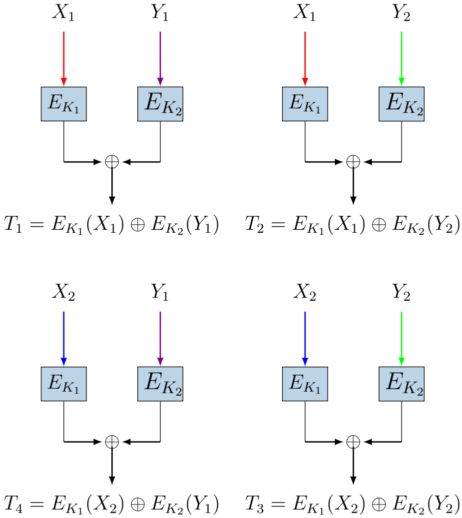


$T_1 = E_{K_1}(X_1) \oplus E_{K_2}(Y_1) \quad T_2 = E_{K_1}(X_1) \oplus E_{K_2}(Y_2)$

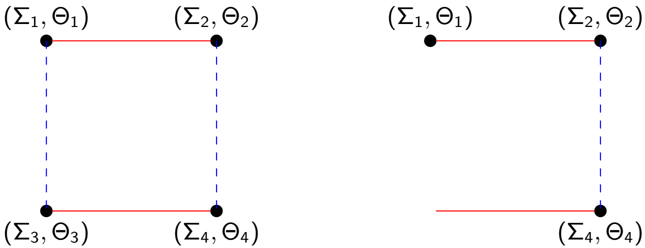
Sum function is not a PRF



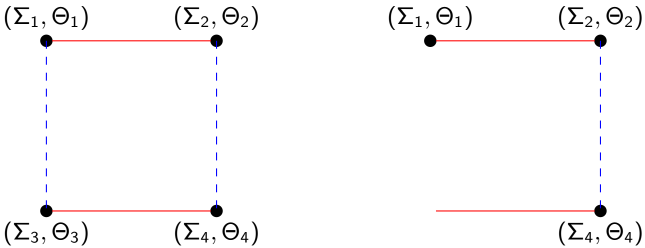
Sum function is not a PRF



(Alternating) Cycle and Path

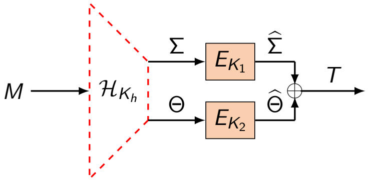


(Alternating) Cycle and Path



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

Security of DbHtS

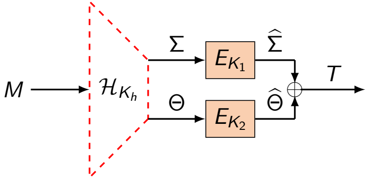


$\widetilde{\Sigma} = (\Sigma_1, \ldots, \Sigma_q)$, $\widetilde{\Theta} = (\Theta_1, \ldots, \Theta_q)$ is called covered if $\exists i \neq j, i \neq k$ such that

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

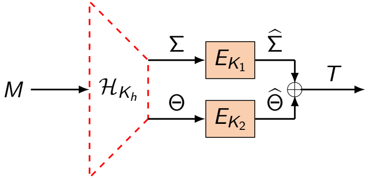
If $\mathcal{H}$ holds either of the above two conditions, it is called covered DbH.

Security of DbHtS



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

Security of DbHtS



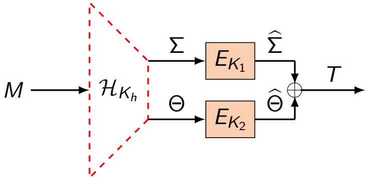
Alternating cycle in $\widetilde{\Sigma}, \widetilde{\Theta}$ makes the sum of T_i's zero.

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

Bad Event (CF)

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

Security of DbHtS



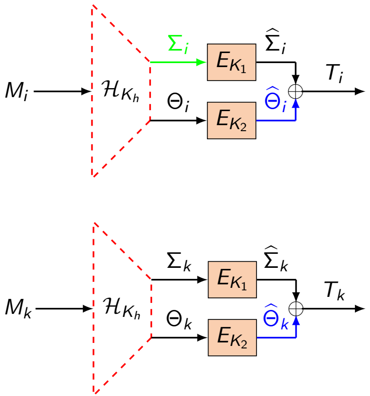
Alternating cycle in $\widetilde{\Sigma}, \widetilde{\Theta}$ makes the sum of T_i's zero.

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

Bad Event (CF)

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

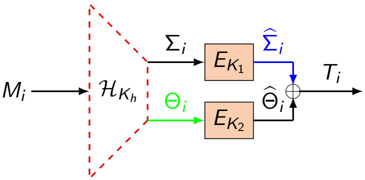
Security of DbHtS

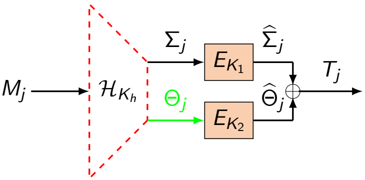


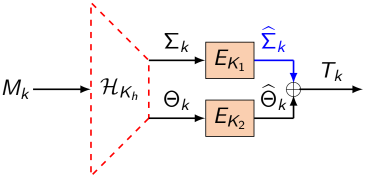
Bad Event (RC1)

$\exists i \neq j, i \neq k$ such that
$\Sigma_i = \Sigma_j$ and $\widehat{\Theta_i} = \widehat{\Theta}_k$.

Security of DbHtS







Bad Event (RC2)

$\exists i \neq j, i \neq k$ such that $\Theta_i = \Theta_j$ and $\widehat{\Sigma_i} = \widehat{\Sigma_k}$.

Security of DbHtS

Bad Tuple

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

Security of DbHtS

Bad Tuple

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

Probability of Bad Events.

- $\Pr[\mathrm{CF}] \leq \binom{q}{3} \cdot \underbrace{\Pr[\Sigma_i = \Sigma_j, \Theta_i = \Theta_k]}_{\epsilon_{\mathrm{cf}}(3,\ell)} + \binom{q}{2} \cdot \underbrace{\Pr[\Sigma_i = \Sigma_j, \Theta_i = \Theta_j]}_{\epsilon_{\mathrm{coll}}}$
- $\Pr[\mathrm{RC}] \leq \frac{q^3}{2^n} \cdot \underbrace{\max\Big(\Pr[\Sigma_i = \Sigma_j], \Pr[\Theta_i = \Theta_j]\Big)}_{\epsilon_{\mathrm{univ}}(2,\ell)}$

Security of DbHtS

Bad Tuple

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

Probability of Bad Events.

- $\Pr[\mathrm{CF}] \leq \binom{q}{3} \cdot \underbrace{\Pr[\Sigma_i = \Sigma_j, \Theta_i = \Theta_k]}_{\epsilon_{\mathrm{cf}}(3,\ell)} + \binom{q}{2} \cdot \underbrace{\Pr[\Sigma_i = \Sigma_j, \Theta_i = \Theta_j]}_{\epsilon_{\mathrm{coll}}}$
- $\Pr[\mathrm{RC}] \leq \frac{q^3}{2^n} \cdot \underbrace{\max\Big(\Pr[\Sigma_i = \Sigma_j], \Pr[\Theta_i = \Theta_j]\Big)}_{\epsilon_{\mathrm{univ}}(2,\ell)}$

Analysis of Good Transcript

We use Sum of Permtation result by [Lucks, Eurocrypt 00].

Security of DbHtS

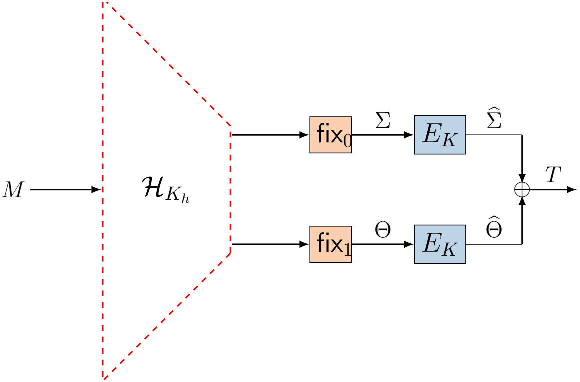
Summarizing the security:

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

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{HtS}}(q,\ell) \leq \binom{q}{3} \cdot \epsilon_{\mathrm{cf}}(3,\ell) + \binom{q}{2} \cdot \epsilon_{\mathrm{coll}} + \frac{q^3}{2^n} \cdot \epsilon_{\mathrm{univ}}(2,\ell) + \frac{4q^3}{3 \cdot 2^{2n}}.$$

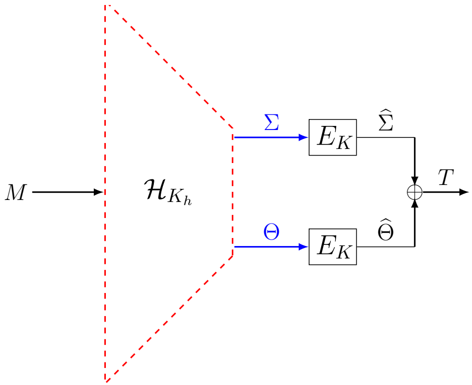
NOTE: $\frac{4q^3}{3 \cdot 2^{2n}}$ comes from sum of permutation result.

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 $\mathsf{fix}_0, \mathsf{fix}_1$

Summarizing the security:

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

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{2K\text{-}HtS}}(q,\ell) \leq \binom{q}{3}\cdot\epsilon_{\mathrm{cf}}(3,\ell) + \binom{q}{2}\cdot\epsilon_{\mathrm{coll}} + \frac{q^3}{2^n}\cdot\epsilon_{\mathrm{univ}}(2,\ell) + \frac{q}{2^n} + \frac{6q^3}{2^{2n}}$$

NOTE: $\frac{6q^3}{3\cdot 2^{2n}}$ comes from sum of permutation result.

Instantiations of three-keyed and two-keyed DbHtS

Type	Instantiattions	Old Bound	New Bound
3-key DbHtS	SUM-ECBC	$q^3\ell^4/2^{2n}$	$q\ell^2/2^n + q^3/2^{2n}$
	PMAC_Plus	$q^3\ell^3/2^{2n} + q\ell/2^n$	$q^3\ell/2^{2n} + q^2\ell^2/2^{2n}$
	3kf9	$q^3\ell^3/2^{2n} + q\ell/2^n$	$q^3\ell^4/2^{2n}$
	LightMAC_Plus	$q^3/2^{2n}$	$q^3/2^{2n}$
2-key DbHtS	2K-SUM-ECBC	-	$q\ell^2/2^n + q^3\ell^2/2^{2n}$
	2K-PMAC_Plus	-	$q^3\ell/2^{2n} + q^2\ell^2/2^{2n}$
	2kf9	-	$q^3\ell^4/2^{2n}$
	2K-LightMAC_Plus	-	$q^3/2^{2n} + q/2^n$

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!