Tight Security Analysis of EHtM MAC

Avijit Dutta, Ashwin Jha and Mridul Nandi

Presented By : Ritam Bhaumik

Indian Statistical Institute, Kolkata

7th March, 2018

Outline of the talk

- Definition and Security Game of MAC.
- Hash-then-Mask.
- Enhanced Hash-then-Mask
- Forgery Attack on Enhanced Hash-then-Mask
- Sketch of Security Proof
- Summary

Categories of MAC: Stateful or Probabilistic

$\Pi = (\mathrm{KG}, \mathrm{TG}, \mathrm{Ver})$ is a triplet of algorithms

- KG is called key-generation algorithm that outputs a key $K \xleftarrow{\$} \mathcal{K}$ (Key-space).
- $\mathrm{TG} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \rightarrow \mathcal{T}$ is called tag generation algorithm.
- $\mathrm{Ver} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \times \mathcal{T} \rightarrow \{0, 1\}$ such that

$$\mathrm{Ver}(K, IV, M, T) = \begin{cases} 1 & \text{if } \mathrm{TG}(K, IV, M) = T, \\ 0 & \text{otherwise} \end{cases}$$

Categories of MAC: Stateful or Probabilistic

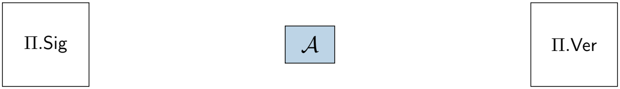
$\Pi = (\text{KG}, \text{TG}, \text{Ver})$ is a triplet of algorithms

- KG is called key-generation algorithm that outputs a key $K \xleftarrow{\$} \mathcal{K}$ (Key-space).
- $\text{TG} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \rightarrow \mathcal{T}$ is called tag generation algorithm.
- $\text{Ver} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \times \mathcal{T} \rightarrow \{0, 1\}$ such that

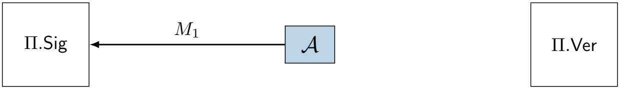
$$\text{Ver}(K, IV, M, T) = \begin{cases} 1 & \text{if } \text{TG}(K, IV, M) = T, \\ 0 & \text{otherwise} \end{cases}$$

- Stateful : IV is a counter / nonce (e.g XMACC, PCS)
- Probabilistic : IV is random (e.g XMACR, EHtM)

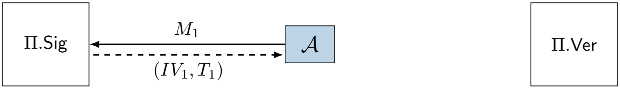
Security Game of Probabilistic MAC



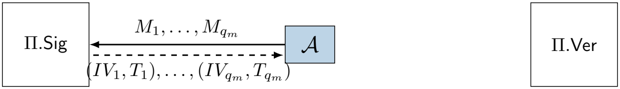
Security Game of Probabilistic MAC



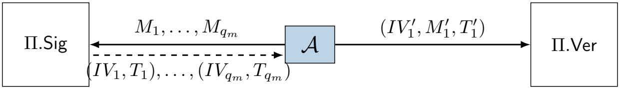
Security Game of Probabilistic MAC



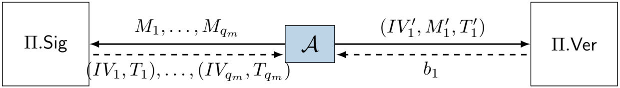
Security Game of Probabilistic MAC



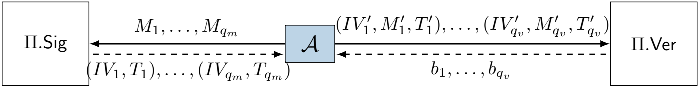
Security Game of Probabilistic MAC



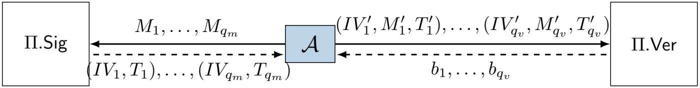
Security Game of Probabilistic MAC



Security Game of Probabilistic MAC

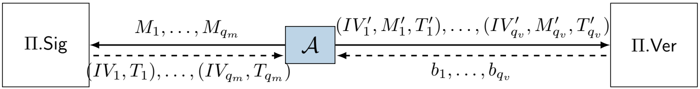


Security Game of Probabilistic MAC



Verification queries can be interleaved with MAC queries and should be fresh.

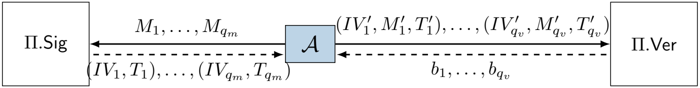
Security Game of Probabilistic MAC



Verification queries can be interleaved with MAC queries and should be fresh.

$\mathbf{Adv}_{\Pi}^{\mathrm{mac}}(\mathcal{A}) = \Pr[\exists i : b_i = 1]$.

Security Game of Probabilistic MAC



Verification queries can be interleaved with MAC queries and should be fresh.

$\mathbf{Adv}_{\Pi}^{\mathrm{mac}}(\mathcal{A}) = \Pr[\exists i : b_i = 1]$.

Π is secure against all such computationally bounded adversary $\mathcal{A}$, if the probability of obtaining $b_i = 1$ for any $i \in \{1, \ldots, q_v\}$ is small.

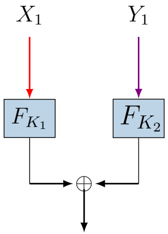
Birthday And Beyond the Birthday Bound (BBB) Security

- Birthday Bound: Security is void after $2^{n/2}$ queries (e.g CBC-MAC, LightMAC)

- Drawback: Not practical when block size is small (e.g PRINCE, HEIGHT, LED etc.)

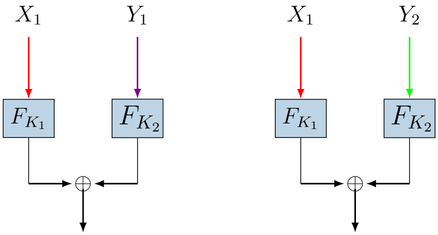
- Beyond Birthday Bound: Security remains even after $2^{n/2}$ queries ("**Beyond Birthday Bound Security**") without increasing the output length (e,g SUM-ECBC, PMAC_Plus, 3kf9).

Abstraction: Sum function is not a PRF



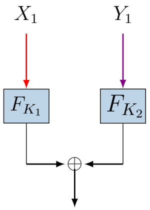
$$T_1 = F_{K_1}(X_1) \oplus F_{K_2}(Y_1)$$

Abstraction: Sum function is not a PRF

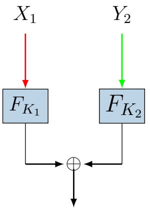


$T_1 = F_{K_1}(X_1) \oplus F_{K_2}(Y_1)$ $\quad$ $T_2 = F_{K_1}(X_1) \oplus F_{K_2}(Y_2)$

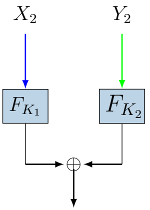
Abstraction: Sum function is not a PRF



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

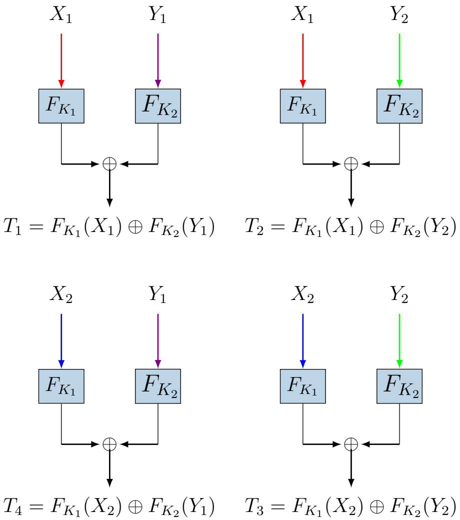


$T_2 = F_{K_1}(X_1) \oplus F_{K_2}(Y_2)$



$T_3 = F_{K_1}(X_2) \oplus F_{K_2}(Y_2)$

Abstraction: Sum function is not a PRF



Alternating Cycle (AC)

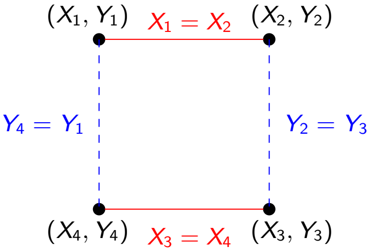


Figure : Alternating Cycle (AC) of length 4

Alternating Cycle (AC)

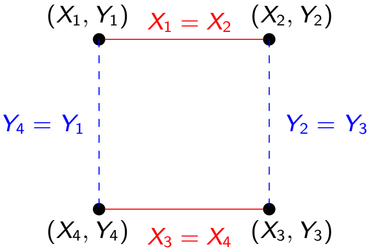
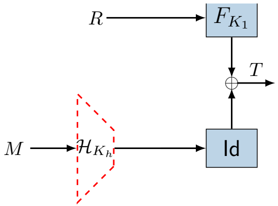


Figure : Alternating Cycle (AC) of length 4

AC in the input of sum function makes the sum of its output zero, i.e. $T_1 \oplus T_2 \oplus T_3 \oplus T_4 = 0$.

Attacks on most of the Probabilistic MAC is based on the formation of Alternating Cycle!

Hash-then-Mask (HtM): First instantiation of Probabilistic MAC

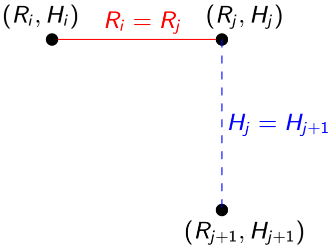


- Id is the identity function.
- Birthday bound security and the bound is also tight.

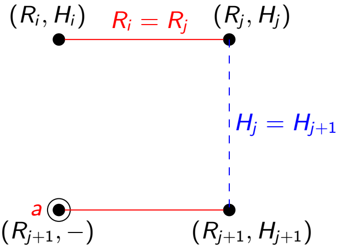
Birthday Attack of HtM

(R_i, H_i) $\quad R_i = R_j \quad$ (R_j, H_j)

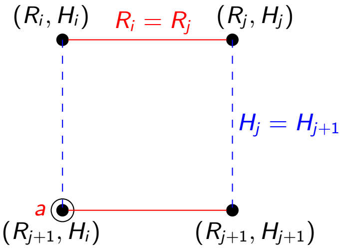
Birthday Attack of HtM



Birthday Attack of HtM

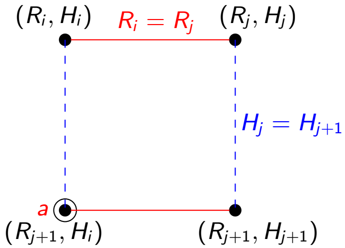


Birthday Attack of HtM



For valid verification attempt, we need to set T_a to $T_i \oplus T_j \oplus T_{j+1}$.

Birthday Attack of HtM



For valid verification attempt, we need to set T_a to $T_i \oplus T_j \oplus T_{j+1}$.

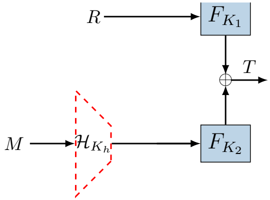
Query Complexity

If we make roughly $2^{n/2}$ many MAC queries, then the top right edge holds w.h.p

Hash-then-Mask offers upto birthday security.

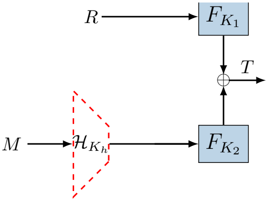
Hash-then-Mask offers upto birthday security.

How can we beat the birthday barrier ? (Replacing Id with F_{K_2} !)



Hash-then-Mask offers upto birthday security.

How can we beat the birthday barrier ? (Replacing Id with F_{K_2} !)



- The previous attack for Hash-then-Mask works here.

Existing BBB Secure Probabilistic MAC.

MAC	Randomness	PRF	Security Model
MACRX_3 [CRYPTO 99]	$3n$	n	Standard
RWMAC [FSE 2010]	n	$2n$ to n	Standard
RMAC, FRMAC [FSE 2002]	n	n	Ideal Cipher

Existing BBB Secure Probabilistic MAC.

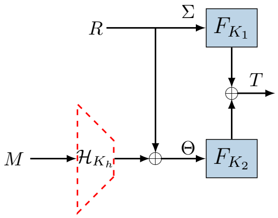
MAC	Randomness	PRF	Security Model
MACRX_3 [CRYPTO 99]	$3n$	n	Standard
RWMAC [FSE 2010]	n	$2n$ to n	Standard
RMAC, FRMAC [FSE 2002]	n	n	Ideal Cipher

Can we design a Probabilistic MAC with n-bit PRF and n-bit randomness with BBB security in standard model?

Enhanced Hash-then-Mask: Minematsu, FSE 2010

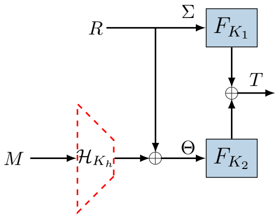
Enhanced Hash-then-Mask (EHtM) by Minematsu, FSE 2010

- EHtM is the first BBB secure (i.e. $2^{2n/3}$-MAC security) probabilistic MAC with *n*-bit PRF and *n*-bit randomness.



Enhanced Hash-then-Mask (EHtM) by Minematsu, FSE 2010

- EHtM is the first BBB secure (i.e. $2^{2n/3}$-MAC security) probabilistic MAC with n-bit PRF and n-bit randomness.



Contribution

We have improved the MAC security bound of EHtM to $2^{3n/4}$ and shown that the bound is tight.

The Fundamental Result.

If $(\widetilde{\Sigma}, \widetilde{\Theta})$ does not contain any alternating cycle, then the distribution of $F_{K_1}(\Sigma_i) \oplus F_{K_2}(\Theta_i)$ is perfectly random.

The Fundamental Result.

If $(\widetilde{\Sigma}, \widetilde{\Theta})$ does not contain any alternating cycle, then the distribution of $F_{K_1}(\Sigma_i) \oplus F_{K_2}(\Theta_i)$ is perfectly random.

We use this result to mount the attack with $2^{3n/4}$ MAC queries, i.e. we'll try to form an alternating cycle!

$2^{3n/4}$ Forging Complexity of EHtM

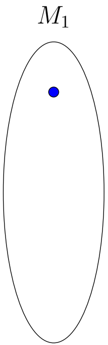
Forging attack is based on the formation of AC4 that consists of two phases:

Part I: Estimation of Hash Difference.

Part II: Forging Attempt for Fixed Estimated Hash Difference.

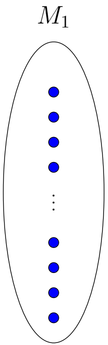
$2^{3n/4}$ Forging Complexity of EHtM

Part I : Estimation of Hash Difference.



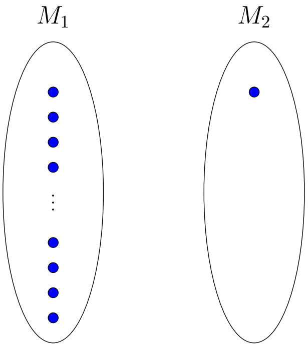
$2^{3n/4}$ Forging Complexity of EHtM

Part I : Estimation of Hash Difference.



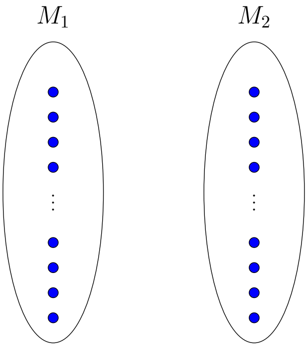
$2^{3n/4}$ Forging Complexity of EHtM

Part I : Estimation of Hash Difference.



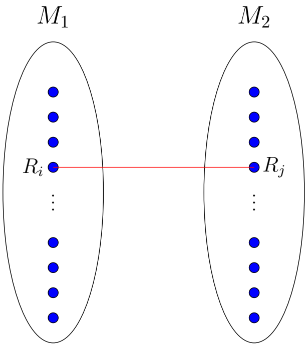
$2^{3n/4}$ Forging Complexity of EHtM

Part I : Estimation of Hash Difference.



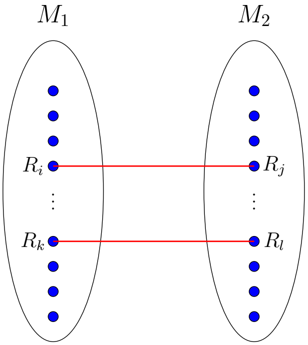
$2^{3n/4}$ Forging Complexity of EHtM

Part I : Estimation of Hash Difference.



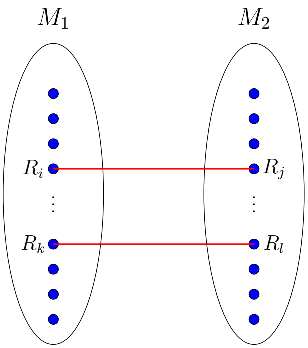
$2^{3n/4}$ Forging Complexity of EHtM

Part I : Estimation of Hash Difference.



$2^{3n/4}$ Forging Complexity of EHtM

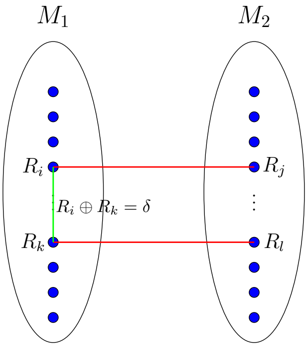
Part I : Estimation of Hash Difference.



If $T_i \oplus T_j \oplus T_k \oplus T_l = 0$ then, $F_{K_2}(R_i \oplus H(M_1)) \oplus F_{K_2}(R_j \oplus H(M_2)) \oplus F_{K_2}(R_k \oplus H(M_1)) \oplus F_{K_2}(R_l \oplus H(M_2)) = 0$

$2^{3n/4}$ Forging Complexity of EHtM

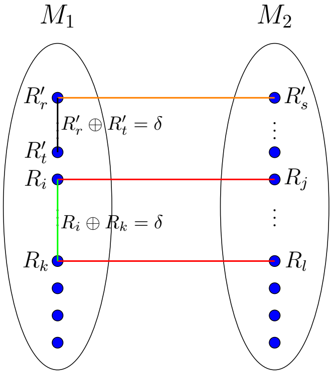
Part I : Estimation of Hash Difference.



Compute $R_i \oplus R_k = \delta$; estimated hash difference. (False positives may also occur)

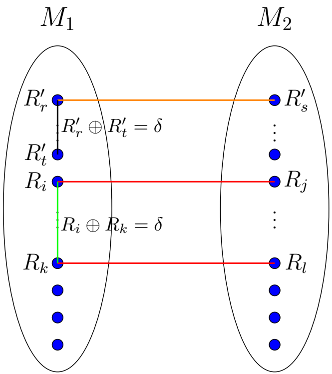
$2^{3n/4}$ Forging Complexity of EHtM

Part II : Forging Attempt for a correct guess of δ.



Verification query : $(R'_t, M_2, T'_r \oplus T'_s \oplus T'_t)$.

Part II : Forging Attempt for a correct guess of δ.



Verification query : $(R'_t, M_2, T'_r \oplus T'_s \oplus T'_t)$.

Attack Complexity is $2^{3n/4}$.

Security Result of EHtM

Theorem

EHtM *is secure upto* $\Theta(q_m^4/2^{3n} + q_v/2^n)$

Security Result of EHtM

Theorem

EHtM *is secure upto* $\Theta(q_m^4/2^{3n} + q_v/2^n)$

- Lower bound is already proved using the previous attack ✓

Security Result of EHtM

Theorem

EHtM *is secure upto* $\Theta(q_m^4/2^{3n} + q_v/2^n)$

- Lower bound is already proved using the previous attack ✓
- Now we show the upper bound in subsequent slides

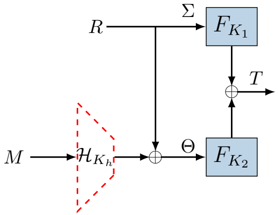
Security Result of EHtM

Theorem

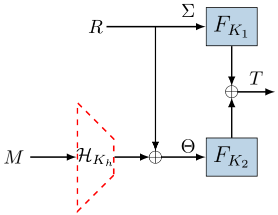
EHtM *is secure upto* $\Theta(q_m^4/2^{3n} + q_v/2^n)$

- Lower bound is already proved using the previous attack ✓
- Now we show the upper bound in subsequent slides
  - We prove using Coefficients-H Technique.
  - For this, we identify the set of bad transcripts (or bad events).
  - Realizing a good transcript is almost as likely as real and the ideal world.

Security Proof of EHtM



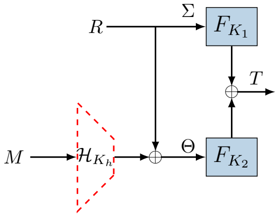
Security Proof of EHtM



Key Point of Bad Events

Avoid alternating cycles in $(\widetilde{\Sigma}, \widetilde{\Theta})$.

Security Proof of EHtM

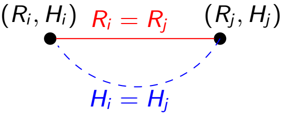


Key Point of Bad Events

Avoid alternating cycles in $(\widetilde{\Sigma}, \widetilde{\Theta})$.

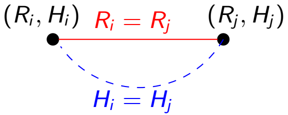
If there is no alternating cycle in $(\widetilde{\Sigma}, \widetilde{\Theta})$ then the output of EHtM is perfectly random.

Security Proof of EHtM: Bad Events

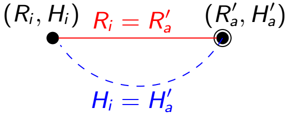


Prob: $q_m^2 \epsilon / 2^n$

Security Proof of EHtM: Bad Events

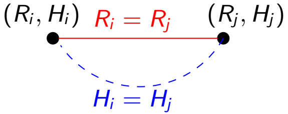


Prob: $q_m^2 \epsilon / 2^n$

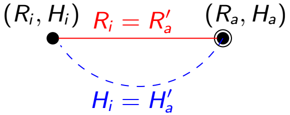


Prob: $q_m^2 / 2^{2n+1} + q_v \epsilon$

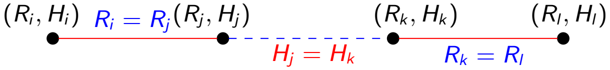
Security Proof of EHtM: Bad Events



Prob: $q_m^2 \epsilon / 2^n$

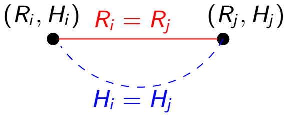


Prob: $q_m^2 / 2^{2n+1} + q_v \epsilon$

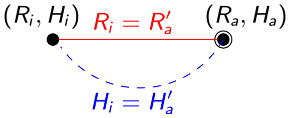


Prob: $q_m^4 \epsilon / 2^{2n}$

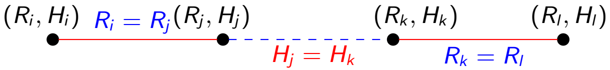
Security Proof of EHtM: Bad Events



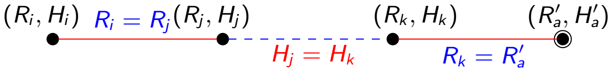
Prob: $q_m^2 \epsilon / 2^n$



Prob: $q_m^2 / 2^{2n+1} + q_v \epsilon$



Prob: $q_m^4 \epsilon / 2^{2n}$



Prob: $q_m^4 / 2^{3n} + q_v \epsilon$

Summary

Till date, EHtM is the best probabilistic MAC in terms of offering security with n-bit randomness and n-bit primitive.

Open Problem

Can we design a probabilistic MAC with n-bit randomness and n-bit primitive that offers optimal security ?

Thank You for your Attention!