Bernstein Bound is Tight

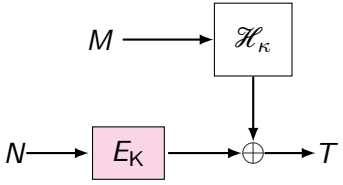
Repairing Luykx-Preneel Optimal Forgeries

Mridul Nandi

Indian Statistical Institute, Kolkata

CRYPTO 2018

Wegman-Carter-Shoup (WCS) MAC



- Nonce based Authenticator
- Initial variant (WC authenticator) due to Wegman and Carter [WC81]
- Use of Block cipher E_K due to [Sho96]

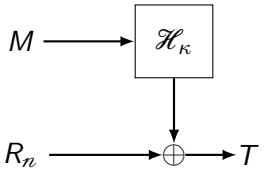
Brief History of WC Authenticator

- Code of Gilbert, MacWilliams and Sloane [GMS74]
  - one-time authentication protocol
  - Issue: a fresh key of size as large as message

Brief History of WC Authenticator

- Code of Gilbert, MacWilliams and Sloane [GMS74]
  - one-time authentication protocol
  - Issue: a fresh key of size as large as message
- WC authenticator uses strongly universal$_2$ hash function $\mathscr{H}_\kappa$ (based on [CW79]).
  - $R_1, R_2, \ldots,$ is a sequence of secret keys
  - message number n (unique) and a message M
  - Tag: $\mathscr{H}_\kappa(M) \oplus R_n$.

Brief History of WC Authenticator



- universal$_2$ is relaxed to a weaker hash AXU in [Kra94/Rog95]
  - $\Pr(\mathscr{H}_\kappa(M) \oplus \mathscr{H}_\kappa(M') = \delta)$ is small
- **polynomial hashing** over n-bits: $\mathrm{Poly}_\kappa(M) := m_d \cdot \kappa \oplus \cdots \oplus m_1 \cdot \kappa^d$ is $\frac{d}{2^n}$-AXU

Getting rid of onetime masking

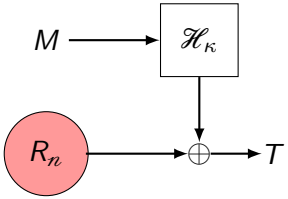


Figure: We can compute R_n directly from n and a secret key.

Getting rid of onetime masking

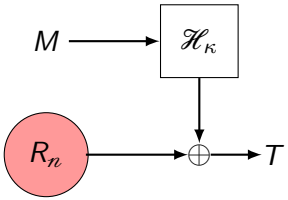
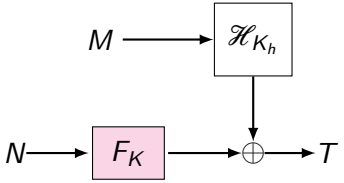


Figure: We can compute R_n directly from n and a secret key.

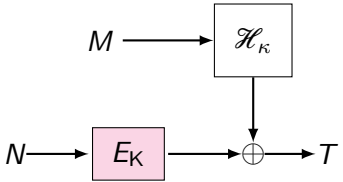
- Use PRBG (Brassard [Bra83]).
- Sequential in nature.
  - Direct efficient computation of R_n (Blum-Blum-Shub PRBG)
  - also modeled as pseudorandom function.

Getting rid of onetime masking



- Use pseudorandom function

Finally - We have WCS



- Use pseudorandom function
- The block cipher (pseudorandom permutation) is widely available. Shoup analyzed WC when PRF is replaced by PRP.

In this Talk

We briefly revisit the security analysis.

- Different attacks.
- Shoup's security guarantee.
- Bernstein's bound and interpretation.

In this Talk

We briefly revisit the security analysis.

- Different attacks.
- Shoup's security guarantee.
- Bernstein's bound and interpretation.

Recent development on WCS.

- Missing difference Problem [LS18].
- Luykx-Preneel "optimal" forgeries [LP18] using false key set.

Identify the issues of Luykx-Preneel forgeries.

In this Talk (contd.)

We resolve it here.

- We prove the optimality of Bernstein Bound.
- False-key based approach, but different analysis:

In this Talk (contd.)

We resolve it here.

- We prove the optimality of Bernstein Bound.
- False-key based approach, but different analysis:
  - messages are chosen random
  - messages are any fixed values

In this Talk (contd.)

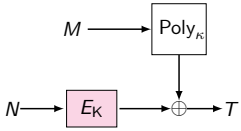
We resolve it here.

- We prove the optimality of Bernstein Bound.
- False-key based approach, but different analysis:
  - messages are chosen random
  - messages are any fixed values

Finally, extend this to show tightness of GCM security

Polynomial Hashing based WCS

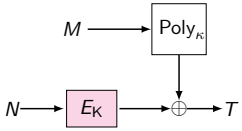
Nonce Misuse Forgery



- $\mathrm{P}_M(\kappa) := \mathrm{Poly}_\kappa(M) := m_d \cdot \kappa + \cdots + m_1 \cdot \kappa^d$
- nonce misuse (Joux's forbidden attack):

Polynomial Hashing based WCS

Nonce Misuse Forgery



- $\mathrm{P}_M(\kappa) := \mathrm{Poly}_\kappa(M) := m_d \cdot \kappa + \cdots + m_1 \cdot \kappa^d$
- nonce misuse (Joux's forbidden attack):
  1. T and T' tags of (N, M) and $(N, M') \Rightarrow$

$$\mathrm{P}_M(\kappa) \oplus \mathrm{P}_{M'}(\kappa) = T \oplus T'$$

  2. solve the hash key (solving polynomial equation).

Polynomial Hashing based WCS

Nonce Respecting Forgery

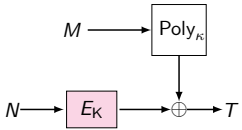


Figure: T is a tag of (N, M).

$$(N, M', T') \text{ is invalid } \kappa \notin \mathrm{Sol}(\mathrm{P}_M(\kappa) \oplus \mathrm{P}_{M'}(\kappa) = T \oplus T').$$

Polynomial Hashing based WCS

Nonce Respecting Forgery

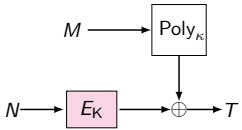


Figure: T is a tag of (N, M).

$$(N, M', T') \text{ is invalid } \kappa \notin \mathrm{Sol}(\mathrm{P}_M(\kappa) \oplus \mathrm{P}_{M'}(\kappa) = T \oplus T').$$

- d disjoint solutions for each forging attempt.

Polynomial Hashing based WCS

Nonce Respecting Forgery

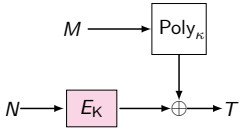


Figure: T is a tag of (N, M).

$$(N, M', T') \text{ is invalid } \kappa \notin \mathrm{Sol}(\mathrm{P}_M(\kappa) \oplus \mathrm{P}_{M'}(\kappa) = T \oplus T').$$

- d disjoint solutions for each forging attempt.
- success probability after v forging attempts: $v \cdot \epsilon = \frac{v \cdot d}{2^n}$.

Bernstein and Shoup's Bound on WCS

- Classical bound: $v \cdot \epsilon$ (based on RF or one time key)

Bernstein and Shoup's Bound on WCS

- Classical bound: $v \cdot \epsilon$ (based on RF or one time key)
- By PRP-PRF switching lemma:

$$v \cdot \epsilon + \frac{(q+v)^2}{2^{n+1}}. \tag{1}$$

Bernstein and Shoup's Bound on WCS

- Classical bound: $v \cdot \epsilon$ (based on RF or one time key)
- By PRP-PRF switching lemma:

$$v \cdot \epsilon + \frac{(q+v)^2}{2^{n+1}}. \tag{1}$$

- Shoup's bound:

$$2v \cdot \epsilon, \qquad \text{if } \epsilon q^2 \leq 1. \tag{2}$$

Bernstein and Shoup's Bound on WCS

- Classical bound: $v \cdot \epsilon$ (based on RF or one time key)
- By PRP-PRF switching lemma:

$$v \cdot \epsilon + \frac{(q+v)^2}{2^{n+1}}. \tag{1}$$

- Shoup's bound:

$$2v \cdot \epsilon, \qquad \text{if } \epsilon q^2 \leq 1. \tag{2}$$

- Bernstein Bound: For all q and v

$$v \cdot \epsilon \cdot (1 - \frac{q}{2^n})^{-\frac{q+1}{2}} \approx v \cdot \epsilon \cdot e^{q^2/2^n}. \tag{3}$$

Interpretation of Shoup's and Bernstein Bound polynomial hash ($\epsilon = d/2^{-n}$) and $v = 1$

Compare: advantage $= \eta$

- Classical bound: $(v + q) \ll 2^{n/2} \Rightarrow \eta$ is small
- Shoup's bound: $q \leq \frac{2^{n/2}}{\sqrt{d}} \Rightarrow \eta \approx 2^{-n}$
- Bernstein bound: $q \leq 2^{n/2} \Rightarrow \eta \approx 2^{-n}$

Interpretation of Shoup's and Bernstein Bound polynomial hash ($\epsilon = d/2^{-n}$) and $v = 1$

Compare: advantage $= \eta$

- Classical bound: $(v + q) \ll 2^{n/2} \Rightarrow \eta$ is small
- Shoup's bound: $q \leq \frac{2^{n/2}}{\sqrt{d}} \Rightarrow \eta \approx 2^{-n}$
- Bernstein bound: $q \leq 2^{n/2} \Rightarrow \eta \approx 2^{-n}$

Example: $n = 128$ and $d = 2^{20}$. Data limit is set for advantage 2^{-32}.

- Classical bound: $(v + q) \leq 2^{48.5}$.
- Shoup's bound: $q \leq 2^{54}$.
- Bernstein bound: $q \leq 2^{64}$.

Missing Difference Problem

Missing Difference Problem

Let $\mathscr{L}$, $\mathscr{L}'$ and $\mathscr{S}$ be three lists of n-bit strings satisfying the missing condition:

$$\exists\, s \in \mathscr{S}, \quad s \notin \mathscr{L} \oplus \mathscr{L}'.$$

Find s.

Missing Difference Problem

Missing Difference Problem

Let $\mathscr{L}$, $\mathscr{L}'$ and $\mathcal{S}$ be three lists of n-bit strings satisfying the missing condition:

$$\exists\, s \in \mathcal{S}, \quad s \notin \mathscr{L} \oplus \mathscr{L}'.$$

Find s.

Complexity Finding Questions:

1. Let $\mathcal{S} = \{0,1\}^n$. How large the lists should be to ensure the missing condition?
2. How efficiently (both time and memory) we can compute s?

Missing Difference Problem

- LS18 constructed $2^{2n/3}$ (ignoring log factor) time and memory algorithm for missing difference when both list sizes are $2^{2n/3}$.

Missing Difference Problem

- LS18 constructed $2^{2n/3}$ (ignoring log factor) time and memory algorithm for missing difference when both list sizes are $2^{2n/3}$.
- Optimal list size: $2^{n/2}\sqrt{n}$.

Missing Difference Problem

- LS18 constructed $2^{2n/3}$ (ignoring log factor) time and memory algorithm for missing difference when both list sizes are $2^{2n/3}$.
- Optimal list size: $2^{n/2}\sqrt{n}$.
  1. Assumptions: for all $x \in \mathscr{L}, x' \in \mathscr{L}'$, $x \oplus x'$ values are uniform and independent from $\{0,1\}^n \setminus \{s\}$.

Missing Difference Problem

- LS18 constructed $2^{2n/3}$ (ignoring log factor) time and memory algorithm for missing difference when both list sizes are $2^{2n/3}$.
- Optimal list size: $2^{n/2}\sqrt{n}$.
  1. Assumptions: for all $x \in \mathscr{L}, x' \in \mathscr{L}'$, $x \oplus x'$ values are uniform and independent from $\{0,1\}^n \setminus \{s\}$.
  2. Number of pairs is $2^n \cdot n$.

Missing Difference Problem

- LS18 constructed $2^{2n/3}$ (ignoring log factor) time and memory algorithm for missing difference when both list sizes are $2^{2n/3}$.
- Optimal list size: $2^{n/2}\sqrt{n}$.
  1. Assumptions: for all $x \in \mathscr{L}, x' \in \mathscr{L}'$, $x \oplus x'$ values are uniform and independent from $\{0,1\}^n \setminus \{s\}$.
  2. Number of pairs is $2^n \cdot n$.
  3. Coupon collecting problem: Expected number of tries to collect all N coupons (here $2^n - 1$) is $N \log N$

Recovering Hash Key: Approach 1

- For single block message m, tag $T = E_K(N) \oplus \kappa \cdot m$.
- Hash-key recovery algorithm
  1. queries $(N_i, 0)$ and $(N_i', 1)$. Response T_i and T_i' $(1 \leq i \leq q)$
  2. Note, $T_i = E_K(N_i)$ and $T_i' = E_K(N_i') \oplus \kappa$.
  3. So, $\kappa \neq T_i \oplus T_j$ (as $N_i \neq N_j'$).
  4. κ is the missing number for the sum of the lists $\mathscr{L}$ (of T_i values) and $\mathscr{L}'$ (of T_i' values).

Recovering Hash Key: Approach 1

- For single block message m, tag $T = E_K(N) \oplus \kappa \cdot m$.
- Hash-key recovery algorithm
  1. queries $(N_i, 0)$ and $(N_i', 1)$. Response T_i and T_i' $(1 \leq i \leq q)$
  2. Note, $T_i = E_K(N_i)$ and $T_i' = E_K(N_i') \oplus \kappa$.
  3. So, $\kappa \neq T_i \oplus T_j$ (as $N_i \neq N_j'$).
  4. κ is the missing number for the sum of the lists $\mathscr{L}$ (of T_i values) and $\mathscr{L}'$ (of T_i' values).
- The assumption on uniformity and independence is wrong.

Luykx-Preneel Forgery: Approach 2

- τ denotes transcript $((N_1, m_1, T_1), \ldots, (N_q, m_q, T_q))$
- Consider False-Key set $\mathscr{F}_\tau$.

$$\mathscr{F}_\tau = \{x : \mathscr{H}_x(M_i) \oplus T_i = \mathscr{H}_x(M_j) \oplus T_j,\ i \neq j\}$$

- $\kappa \notin \mathscr{F}_\tau$ (if not, $E_K(N_i) = E_K(N_j)$) Hope: false-key set almost exhaust the key-space
- Choose an element randomly from $\mathscr{F}_\tau^c$ as a guess of κ
- Key-recovery advantage is at least $\frac{1}{2^n - Ex(|\mathscr{F}_\tau|)}$.

Luykx-Preneel Forgery: Approach 2 (contd.)

Theorem(LP18)

1. $Ex(\mathscr{F}_\tau) \geq q^2/4$ for all $q < 2^{n/2}$.
2. KR advantage is at least $\frac{1}{2^n - q^2/4}$ for all $q < 2^{n/2}$.

Concluded from above that Bernstein Bound is Tight!

Luykx-Preneel Forgery: Approach 2 (contd.)

Theorem(LP18)

1. $Ex(\mathscr{F}_\tau) \geq q^2/4$ for all $q < 2^{n/2}$.
2. KR advantage is at least $\frac{1}{2^n - q^2/4}$ for all $q < 2^{n/2}$.

Concluded from above that Bernstein Bound is Tight!

Wait:

Luykx-Preneel Forgery: Approach 2 (contd.)

Theorem(LP18)

1. $Ex(\mathscr{F}_\tau) \geq q^2/4$ for all $q < 2^{n/2}$.
2. KR advantage is at least $\frac{1}{2^n - q^2/4}$ for all $q < 2^{n/2}$.

Concluded from above that Bernstein Bound is Tight!

Wait: The maximum guaranteed KR advantage is $\frac{1}{0.75 \times 2^n} \approx \frac{1}{2^n}$.

Luykx-Preneel Forgery: Approach 2 (contd.)

Theorem(LP18)

1. $Ex(\mathscr{F}_\tau) \geq q^2/4$ for all $q < 2^{n/2}$.
2. KR advantage is at least $\frac{1}{2^n - q^2/4}$ for all $q < 2^{n/2}$.

Concluded from above that Bernstein Bound is Tight!

Wait: The maximum guaranteed KR advantage is $\frac{1}{0.75 \times 2^n} \approx \frac{1}{2^n}$.

- In the range $q \leq 2^{n/2}$, the random guess shows the optimality.
- For $q \geq 2^{n/2}$, [LP18] did not show anything.

Our Work: Resolving The Issue

- Consider true key approach (complement of false key).
- We have shown KR-advantage is at least

$$\frac{1}{1 + 2^n e^{-\frac{q^2}{2^{n+1}}}}.$$

1. messages are chosen randomly and
2. messages are fixed.

Our Work: Resolving The Issue

- Consider true key approach (complement of false key).
- We have shown KR-advantage is at least

$$\frac{1}{1 + 2^n e^{-\frac{q^2}{2^{n+1}}}}.$$

  1. messages are chosen randomly and
  2. messages are fixed.

- KR is at least $1/2$ for $q = 2^{n/2} \cdot \sqrt{n}$.

Our Work: Resolving The Issue

- Consider true key approach (complement of false key).
- We have shown KR-advantage is at least

$$\frac{1}{1 + 2^n e^{-\frac{q^2}{2^{n+1}}}}.$$

  1. messages are chosen randomly and
  2. messages are fixed.

- KR is at least $1/2$ for $q = 2^{n/2} \cdot \sqrt{n}$.
- We extend our analysis for GCM.

Analysis for random messages

Hash-key recovery algorithm

1. queries:(N_i, m_i) / Response:T_i – m_i's are chosen randomly
2. $\kappa \in \mathscr{T}_\tau := \{x : \underbrace{T_i \oplus x \cdot m_i}_{R_{i,x}} \neq \underbrace{T_j \oplus x \cdot m_j}_{R_{j,x}} \text{ for all } i \neq j\}$.
3. return an element randomly from $\mathscr{T}_\tau$.

Analysis for random messages

Hash-key recovery algorithm

1. queries:(N_i, m_i) / Response:T_i – m_i's are chosen randomly
2. $\kappa \in \mathscr{T}_\tau := \{x : \underbrace{T_i \oplus x \cdot m_i}_{R_{i,x}} \neq \underbrace{T_j \oplus x \cdot m_j}_{R_{j,x}} \text{ for all } i \neq j\}$.
3. return an element randomly from $\mathscr{T}_\tau$.

Observations

1. for all $x \neq \kappa$, $R_{i,x} := E_K(N_i) \oplus (\kappa \oplus x) \cdot m_i$ – uniform and independent.

Analysis for random messages

Hash-key recovery algorithm

1. queries:(N_i, m_i) / Response:T_i – m_i's are chosen randomly
2. $\kappa \in \mathscr{T}_\tau := \{x : \underbrace{T_i \oplus x \cdot m_i}_{R_{i,x}} \neq \underbrace{T_j \oplus x \cdot m_j}_{R_{j,x}} \text{ for all } i \neq j\}$.
3. return an element randomly from $\mathscr{T}_\tau$.

Observations

1. for all $x \neq \kappa$, $R_{i,x} := E_K(N_i) \oplus (\kappa \oplus x) \cdot m_i$ – uniform and independent.
2. $x \notin \mathscr{T}_\tau$ if and only if $R_{i,x}$ values are distinct.

Analysis for random messages

Hash-key recovery algorithm

1. queries:(N_i, m_i) / Response:T_i – m_i's are chosen randomly
2. $\kappa \in \mathscr{T}_\tau := \{x : \underbrace{T_i \oplus x \cdot m_i}_{R_{i,x}} \neq \underbrace{T_j \oplus x \cdot m_j}_{R_{j,x}} \text{ for all } i \neq j\}$.
3. return an element randomly from $\mathscr{T}_\tau$.

Observations

1. for all $x \neq \kappa$, $R_{i,x} := E_K(N_i) \oplus (\kappa \oplus x) \cdot m_i$ – uniform and independent.
2. $x \notin \mathscr{T}_\tau$ if and only if $R_{i,x}$ values are distinct.
3. $|\mathscr{T}_\tau| = \sum_x I_x$ where I_x is indicator r.v. taking 1 if $R_{i,x}$'s are distinct.

Analysis for random messages

- $\Pr(I_x = 1) = \prod_{i=1}^{q-1}(1 - \frac{i}{2^n}) \approx e^{-\frac{q^2}{2^{n+1}}}$ (birthday paradox bound)

$$\mathbb{E}(|\mathscr{T}_\tau|) = 1 + \sum_{x \neq \kappa} \mathbb{E}(I_x)$$
$$\leq 1 + (H-1)e^{-\frac{q^2}{2^{n+1}}}$$

where H is the size of hash key space (here 2^n).

Analysis for random messages

- $\Pr(I_x = 1) = \prod_{i=1}^{q-1}(1 - \frac{i}{2^n}) \approx e^{-\frac{q^2}{2^{n+1}}}$ (birthday paradox bound)

$$\mathbb{E}(|\mathscr{T}_\tau|) = 1 + \sum_{x \neq \kappa} \mathbb{E}(I_x)$$
$$\leq 1 + (H-1)e^{-\frac{q^2}{2^{n+1}}}$$

  where H is the size of hash key space (here 2^n).

- KR advantage is at least $\frac{1}{1+(H-1)e^{-\frac{q^2}{2^{n+1}}}}$.

Analysis for random messages

- $\Pr(I_x = 1) = \prod_{i=1}^{q-1}(1 - \frac{i}{2^n}) \approx e^{-\frac{q^2}{2^{n+1}}}$ (birthday paradox bound)

$$\mathbb{E}(|\mathscr{T}_\tau|) = 1 + \sum_{x \neq \kappa} \mathbb{E}(I_x)$$
$$\leq 1 + (H-1)e^{-\frac{q^2}{2^{n+1}}}$$

  where H is the size of hash key space (here 2^n).

- KR advantage is at least $\frac{1}{1+(H-1)e^{-\frac{q^2}{2^{n+1}}}}$.
- Choose $q = 2^{n/2}\sqrt{\log H}$.

Analysis for fixed messages

- $R_{i,x} := E_K(N_i) \oplus (\kappa \oplus x) \cdot m_i$ are no longer independent and uniform.
- Apply WOR distribution of $V_i = E_K(N_i)$

Analysis for fixed messages

- $R_{i,x} := E_K(N_i) \oplus (\kappa \oplus x) \cdot m_i$ are no longer independent and uniform.
- Apply WOR distribution of $V_i = E_K(N_i)$

Lemma

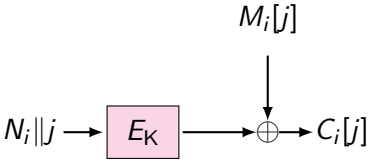
Let $\mathsf{V}_1, \ldots, \mathsf{V}_q$ *be a uniform without replacement sample from* B *of size* N *and* $a_1, \ldots, a_q \in B$ *be some distinct elements, for some* $q \leq N/6$*. Then,*

$$p_x := \Pr(\mathsf{V}_1 + a_1, \ldots, \mathsf{V}_q + a_q \text{ are distinct}) \leq e^{-q^2/4N}.$$

- Apply this lemma with a_i as $(\kappa \oplus x) \cdot m_i$.
- Rest is similar.

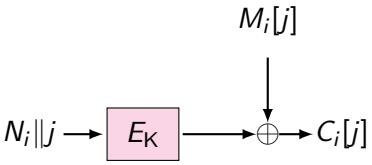
Overview of GCM and Its Attack

ciphertext generation

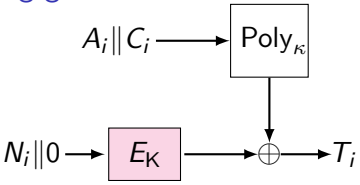


Overview of GCM and Its Attack

ciphertext generation

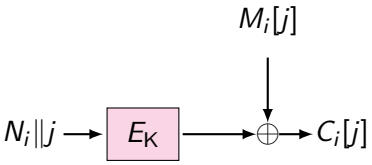


tag generation

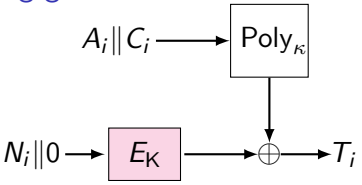


Overview of GCM and Its Attack

ciphertext generation



tag generation



- Encryption query for random messages M_i (same A_i) and let C_i and T_i be responses.
- Construct false key set by comparing $E_K(N_i \| j)$ and $E_K(N_i \| 0)$.
- Hash key recovery whenever ℓq^2 is about $2^n \cdot n$.

More Formally –

- Let M_i be ℓ blocks messages and C_i, T_i be responses.
- True key set $\mathscr{T}_\tau := \{x : \underbrace{T_i \oplus Poly_x(A \| C_i)}_{R_{i,x}} \neq M_i[j] \oplus C_i[j] \text{ for all } i \neq j\}$.
- Clearly, $\kappa \in \mathscr{T}_\tau$.
- As M_i's are random, C_i's are random and using random message analysis, the key recovery advantage is at least

$$\frac{1}{1 + 2^n \cdot e^{\frac{-\ell q^2}{2^n}}}$$

- Query complexity: $\ell q^2 = n \cdot 2^n$.

Conclusion

- Luykx-Preneel forgeries (as it is) is not better than random guess.
- Idea of false key set or missing difference problem is useful.
- We provide correct analysis with higher complexities.
- Applicable for more general set up.
    1. arbitrary hash functions.
    2. applicable for both random and any fixed messages.
- Extend the result for GCM.

Conclusion

- Luykx-Preneel forgeries (as it is) is not better than random guess.
- Idea of false key set or missing difference problem is useful.
- We provide correct analysis with higher complexities.
- Applicable for more general set up.
  1. arbitrary hash functions.
  2. applicable for both random and any fixed messages.
- Extend the result for GCM.

Thank You.