A New Look at Counters: Don't Run Like Marathon in a Hundred Meter Race

Directions in Authenticated Ciphers '16, Nagoya

Avijit Dutta, Ashwin Jha and Mridul Nandi

September 27, 2016

Indian Statistical Institute Kolkata

Counters in Cryptography

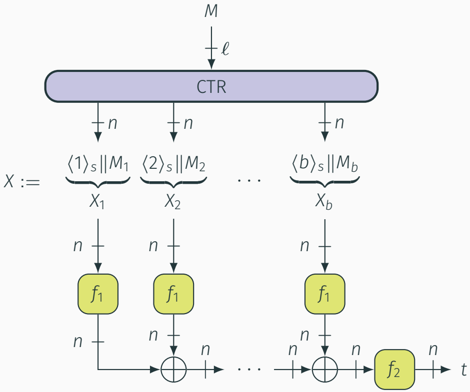
Classical View:

$$\langle 0\rangle_s, \langle 1\rangle_s, \langle 2\rangle_s, \langle 3\rangle_s, \ldots, \langle 2^s - 1\rangle_s$$

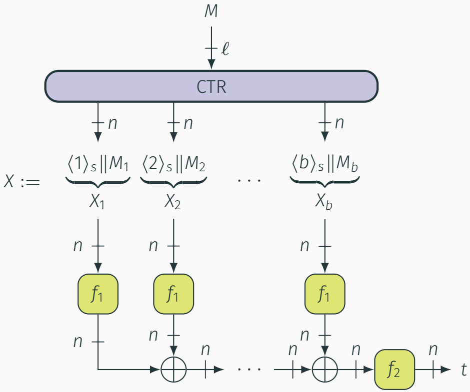
where $\langle i\rangle_s$ is the s-bits binary representation of i for some fixed s.

- Prevents collisions on the inputs to the underlying primitive.
- Standalone input: CTR mode, HAIFA, GCM, SIV.
- Encoded within message blocks: HAIFA, XORMAC, LightMAC.

Counter-Based Input Encoding



Counter-Based Input Encoding



Security Needs

Blockwise Collision-free:

$$\forall\, i \neq j, \quad X_i \neq X_j.$$

Injective:

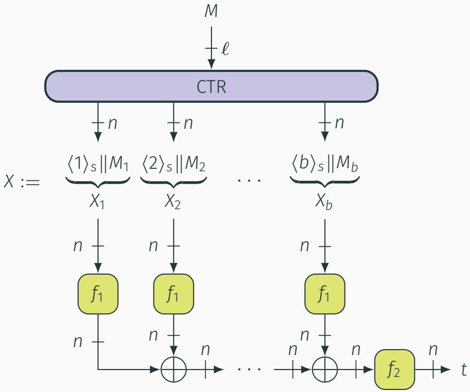
$$\forall\, M \neq M', \quad X \neq X'.$$

Rate signifies Efficiency

$$rate_{STD} = \frac{n-s}{n}$$

where $s = \log_2 L$, L being the maximum permissible message length.

Counter-Based Input Encoding



Security Needs

Blockwise Collision-free:

$$\forall\, i \neq j,\quad X_i \neq X_j.$$

Injective:

$$\forall\, M \neq M',\quad X \neq X'.$$

Rate signifies Efficiency

$$rate_{STD} = \frac{n - s}{n}$$

where $s = \log_2 L$, L being the maximum permissible message length.

Example

For $n = 128$ and $s = 64$, the rate is 0.5 for any message lengths.

Can we have better rate for smaller messages?

STD^{opt}: Length Dependent Counter Scheme

- Computes the optimal counter size ($\approx \log_2 \ell$) for the given message length ℓ.

$$rate_{\mathrm{STD}^{opt}} = \frac{n - \log_2 \ell}{n}$$

- For $\ell < L$, $rate_{\mathrm{STD}^{opt}} > rate_{STD}$.

Comparison

For $n = 128$ bits and $\ell = 2^{10}$ bits, the rate is 0.92.

STD^{opt}: Length Dependent Counter Scheme

- Computes the optimal counter size ($\approx \log_2 \ell$) for the given message length ℓ.

$$rate_{\mathrm{STD}^{opt}} = \frac{n - \log_2 \ell}{n}$$

- For $\ell < L$, $rate_{\mathrm{STD}^{opt}} > rate_{STD}$.

Comparison

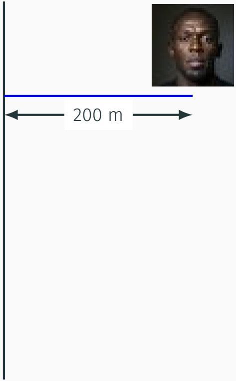
For $n = 128$ bits and $\ell = 2^{10}$ bits, the rate is 0.92.

Catch

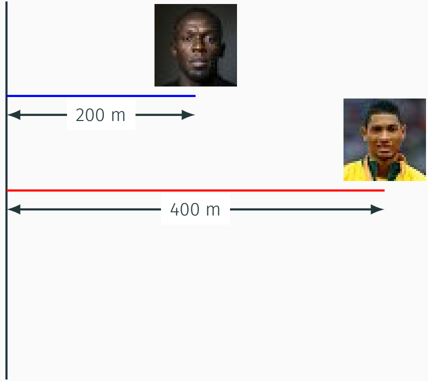
What if we don't know the length? Can we have a close approximation of STD^{opt} in this case?

A Race over Unknown Distance

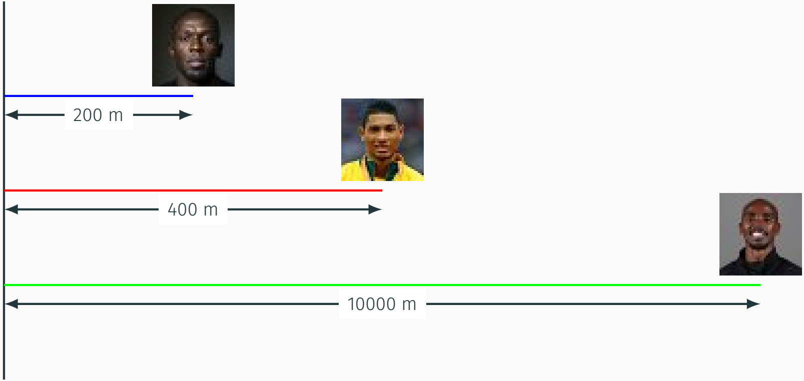
A Race over Unknown Distance



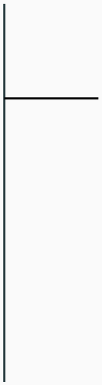
A Race over Unknown Distance



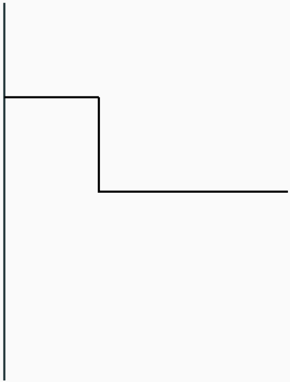
A Race over Unknown Distance



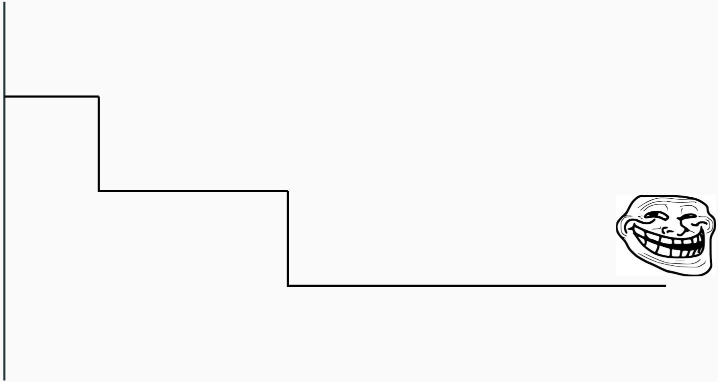
A Race over Unknown Distance



A Race over Unknown Distance



A Race over Unknown Distance



A Candidate Length Independent Counter

$0\ ,\ 1\ ,\ 00\ ,\ 01\ ,\ 10\ ,\ 11\ ,\ 000\ldots$

A Candidate Length Independent Counter

$0\ ,\ 1\ ,\ 00\ ,\ 01\ ,\ 10\ ,\ 11\ ,\ 000\ldots$

- Length Independent. ✓

A Candidate Length Independent Counter

0 , 1 , 00 , 01 , 10 , 11 , 000 . . .

- Length Independent. ✓
- $rate > rate_{\mathrm{STD}^{opt}}$. ✓

A Candidate Length Independent Counter

$$0\ ,\ 1\ ,\ 00\ ,\ 01\ ,\ 10\ ,\ 11\ ,\ 000\ldots$$

- Length Independent. ✓
- $rate > rate_{\mathrm{STD}^{opt}}$. ✓
- But, is this blockwise collision-free? ✗

A Candidate Length Independent Counter

$$0\ ,\ 1\ ,\ 00\ ,\ 01\ ,\ 10\ ,\ 11\ ,\ 000\ldots$$

- Length Independent. ✓
- $rate > rate_{\mathsf{STD}^{opt}}$. ✓
- But, is this blockwise collision-free? ✗

Trivial Collision

For $n = 8$ and $M := 0abcdefghijklmabcdef$ we have $X_1 = 00abcdef,\ X_2 = 1ghijklm,$ and $X_3 = 00abcdef$. Clearly, $X_1 = X_3$.

VAR: Message Length Independent Counter

- Add a small fixed length (r) counter that gets updated with the change in counter size.

$000\ ,\ 001\ ,\ 010\underline{0}\ ,\ \ldots\ ,\ 0111\ ,\ 1000\underline{0}\ ,\ \ldots\ ,\ 10111\ ,\ 11000\underline{0}\ ,\ \ldots$

VAR: Message Length Independent Counter

- Add a small fixed length (r) counter that gets updated with the change in counter size.

$$000\ ,\ 001\ ,\ 010\underline{0}\ ,\ \ldots\ ,\ 0111\ ,\ 1000\underline{0}\ ,\ \ldots\ ,\ 10111\ ,\ 11000\underline{0}\ ,\ \ldots$$

- Length Independent. ✓
- Blockwise Collision-free and Injective. ✓

VAR: Message Length Independent Counter

- Add a small fixed length (r) counter that gets updated with the change in counter size.

$$000\ ,\ 001\ ,\ 010\underline{0}\ ,\ \ldots\ ,\ 0111\ ,\ 1000\underline{0}\ ,\ \ldots\ ,\ 10111\ ,\ 11000\underline{0}\ ,\ \ldots$$

- Length Independent. ✓
- Blockwise Collision-free and Injective. ✓
- $r \approx \log_2 \log_2 L$, for $L < 2^{c(n)}$, $\frac{n}{2} \leq c(n) < n$.

$$rate_{VAR} \approx \frac{n - r + 2 - \log_2 \ell}{n}$$

Comparison

For $n = 128$ bits, $L = 2^{64}$ bits, and $\ell = 2^{10}$ bits, the rate is 0.89.

Counter Function Family (CFF)

Definition:

CTR is a family of counter functions $\{\text{ctr}_\ell : \ell \leq L\}$ where

$$\forall\, \ell \leq L, \;\; \text{ctr}_\ell : \mathbb{N} \to \{0,1\}^{<n}.$$

- Length Independent: For STD counter function family $\text{std}_\ell(i) = \langle i \rangle_s, \forall\, \ell,\; i.$
- Length Dependent: For STD^{opt} counter function family $\text{opt}_\ell(i) = \langle i \rangle_{\log_2 \ell}, \forall\, \ell,\; i.$
- For a given ℓ, if $\forall\, i \neq j,\; |\text{ctr}_\ell(i)| = |\text{ctr}_\ell(j)|$, we say that CTR is a fixed length CFF; variable length CFF otherwise.

Counter Function Family (CFF)

Definition:

CTR is a family of counter functions $\{\mathrm{ctr}_\ell : \ell \leq L\}$ where

$$\forall\, \ell \leq L, \;\; \mathrm{ctr}_\ell : \mathbb{N} \to \{0,1\}^{<n}.$$

- Length Independent: For STD counter function family $\mathrm{std}_\ell(i) = \langle i \rangle_s, \forall\, \ell,\; i.$
- Length Dependent: For STD^{opt} counter function family $\mathrm{opt}_\ell(i) = \langle i \rangle_{\log_2 \ell}, \forall\, \ell,\; i.$
- For a given ℓ, if $\forall\, i \neq j,\; |\mathrm{ctr}_\ell(i)| = |\mathrm{ctr}_\ell(j)|$, we say that CTR is a fixed length CFF; variable length CFF otherwise.

What can we say about the security relevant properties?

Prefix-free and Injective CFFs

Prefix-free:

CTR is prefix-free if

$$\forall\, \ell \leq L,\ \forall\, i \neq j \in b(\ell),\ \mathrm{ctr}_\ell(i) \text{ is not a prefix of } \mathrm{ctr}_\ell(j).$$

Prefix-free and Injective CFFs

Prefix-free:

CTR is prefix-free if

$$\forall\, \ell \le L,\ \forall\, i \neq j \in b(\ell),\ \mathrm{ctr}_\ell(i) \text{ is not a prefix of } \mathrm{ctr}_\ell(j).$$

CFF as an Encoding Function:

For any ℓ length message M, $\mathrm{CTR}(M) = (X_1, \ldots, X_{b(\ell)})$, where each $X_i = \mathrm{ctr}_\ell(i) \| M_i$ and $b(\ell)$ is the least integer b that satisfies,

$$\ell + 1 \ \le\ \sum_{i=1}^{b} (n - |\mathrm{ctr}_\ell(i)|) \ \le\ \ell + n.$$

Prefix-free and Injective CFFs

Prefix-free:

CTR is prefix-free if

$$\forall\, \ell \leq L,\ \forall\, i \neq j \in b(\ell),\ \mathrm{ctr}_\ell(i) \text{ is not a prefix of } \mathrm{ctr}_\ell(j).$$

CFF as an Encoding Function:

For any ℓ length message M, $\mathrm{CTR}(M) = (X_1, \ldots, X_{b(\ell)})$, where each $X_i = \mathrm{ctr}_\ell(i) \| M_i$ and $b(\ell)$ is the least integer b that satisfies,

$$\ell + 1 \ \leq\ \sum_{i=1}^{b} (n - |\mathrm{ctr}_\ell(i)|) \ \leq\ \ell + n.$$

Lemma: Prefix-free $\Leftrightarrow$Blockwise Collision-free

CTR is a blockwise collision-free encoding if and only if it is CTR is a prefix-free CFF.

Prefix-free and Injective CFFs

Prefix-free:

CTR is prefix-free if

$$\forall\, \ell \leq L,\ \forall\, i \neq j \in b(\ell),\ \mathrm{ctr}_\ell(i) \text{ is not a prefix of } \mathrm{ctr}_\ell(j).$$

CFF as an Encoding Function:

For any ℓ length message M, $\mathrm{CTR}(M) = (X_1, \ldots, X_{b(\ell)})$, where each $X_i = \mathrm{ctr}_\ell(i) \| M_i$ and $b(\ell)$ is the least integer b that satisfies,

$$\ell + 1 \ \leq\ \sum_{i=1}^{b} (n - |\mathrm{ctr}_\ell(i)|) \ \leq\ \ell + n.$$

Lemma: Prefix-free ⇔Blockwise Collision-free

CTR is a blockwise collision-free encoding if and only if it is CTR is a prefix-free CFF.

What about injective property?

Prefix-free and Injective CFFs

Injective:

CTR is *injective* if $\forall\, M \neq M'$, $\mathrm{CTR}(M) \neq \mathrm{CTR}(M')$ (as sets, i.e. $\mathrm{CTR}(M) = \{X_i \;:\; 1 \leq i \leq b(\ell)\}$).

Prefix-free and Injective CFFs

Injective:

CTR is injective if $\forall\, M \neq M'$, $\mathrm{CTR}(M) \neq \mathrm{CTR}(M')$ (as sets, i.e. $\mathrm{CTR}(M) = \{X_i \;:\; 1 \leq i \leq b(\ell)\}$).

Lemma: Prefix-free++ $\implies$ Injective

Let CTR be a prefix-free CFF. It is injective if it satisfies the following condition,

$$\forall\, \ell, \ell',\; b(\ell) = b(\ell') \Rightarrow \mathrm{ctr}_\ell = \mathrm{ctr}_{\ell'}.$$

Prefix-free and Injective CFFs

Injective:

CTR is injective if $\forall\, M \neq M'$, $\mathrm{CTR}(M) \neq \mathrm{CTR}(M')$ (as sets, i.e. $\mathrm{CTR}(M) = \{X_i \;:\; 1 \leq i \leq b(\ell)\}$).

Lemma: Prefix-free++ $\implies$ Injective

Let CTR be a prefix-free CFF. It is injective if it satisfies the following condition,

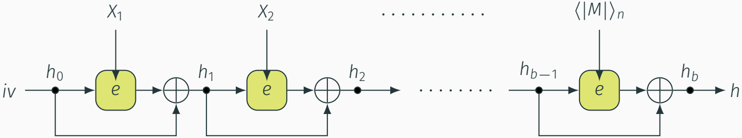
$$\forall\, \ell, \ell',\; b(\ell) = b(\ell') \Rightarrow \mathrm{ctr}_\ell = \mathrm{ctr}_{\ell'}.$$

STD, STD^{opt}, and VAR are prefix-free and injective CFFs.

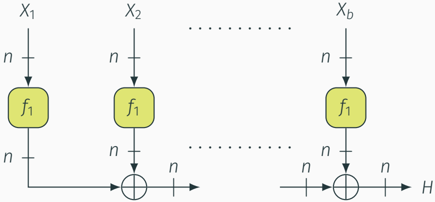
Summary of Candidate CFFs

	STD	STD^{opt}	VAR
Length Dependent	✗	✓	✗
Length Independent	✓	✗	✓
Fixed Length	✓	✓	✗
Variable Length	✗	✗	✓
Rate	$\frac{n-s}{n}$	$\frac{n-\log_2 \ell}{n}$	$\frac{n-r+2-\log_2 \ell}{n}$
Prefix-free	✓	✓	✓
Injective	✓	✓	✓

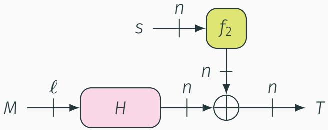
Counter-Based Constructions



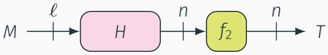
CtHAIFA



CtH

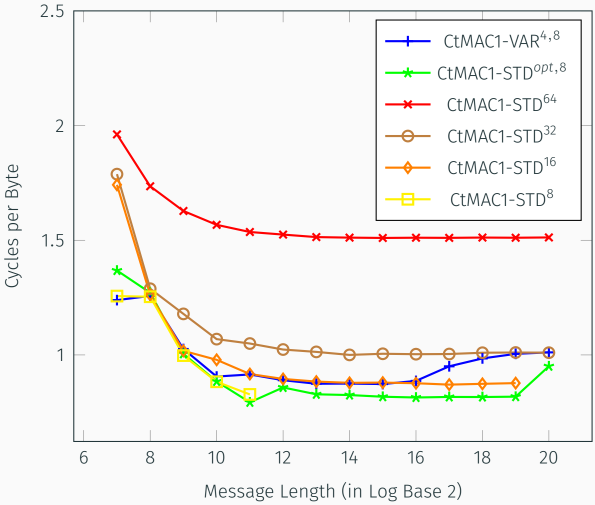


CtMAC2

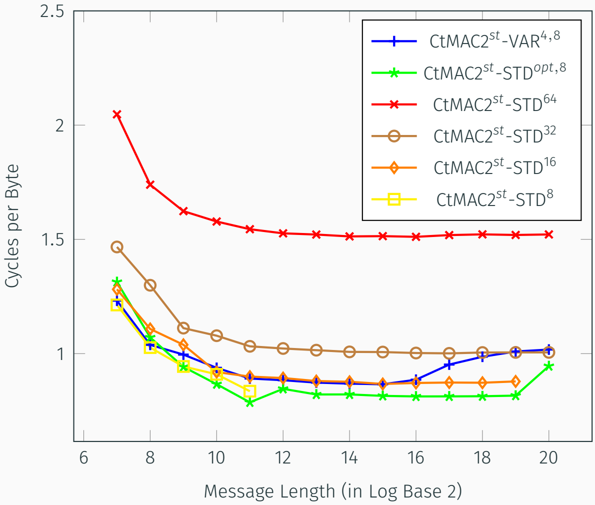


CtMAC1

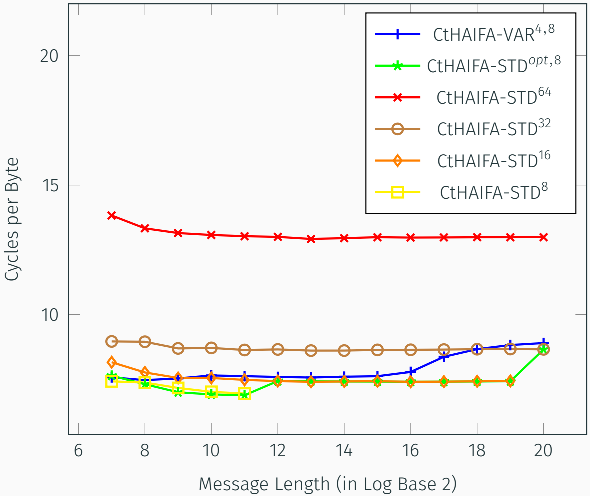
Performance Comparison: CtMAC1



Performance Comparison: CtMAC2



Performance Comparison: CtHAIFA



Summary of Security Results: CtHAIFA and CtH

Theorem: Second Preimage Security of CtHAIFA

CtHAIFA has full second preimage security. More specifically, for any second preimage adversary $\mathcal{A}$ that makes at most q queries, we have

$$\mathsf{Adv}^{\mathrm{2PI}}_{\mathrm{CtHAIFA}}(q) \leq \frac{3q}{2^n}.$$

Theorem: AXU Security of CtH

$\mathrm{CtH}_{\Pi,\mathrm{CTR}}$ is $1/(2^n - b)$-AXU where $b = b(L)$ (the number of blocks for the largest message).

Summary of Security Results: CtMAC1 and CtMAC2

Theorem: PRF Security of CtMAC1

Let $\text{CtMac1} := \text{CtMac1}_{E_{K_1}, E_{K_2}}$ be defined based on two independently chosen keyed blockcipher. Then,

$$\mathbf{Adv}^{\mathrm{prf}}_{\text{CtMac1}}(t, q, \ell) \leq \frac{1.5q^2}{2^n} + \mathbf{Adv}^{\mathrm{prp}}_{E}(t', \ell q)$$

Theorem: MAC Security of CtMAC2

Let $\text{CtMac2}_{E_{K_1}, E_{K_2}}(s, M)$ be defined on two independently chosen keyed block ciphers. Then,

1. $\mathbf{Adv}^{\mathrm{forge}}_{\text{CtMac2}^{st}}(t, q_m, q_v, \ell) \leq \frac{0.5q^2}{2^n} + \mathbf{Adv}^{\mathrm{prp}}_{E}(t', \ell(q_m + q_v)) + \frac{q_v}{2^n}$
2. $\mathbf{Adv}^{\mathrm{forge}}_{\text{CtMac2}^{\$}}(t, q_m, q_v, \ell) \leq \frac{q^2}{2^n} + \mathbf{Adv}^{\mathrm{prp}}_{E}(t', \ell(q_m + q_v)) + \frac{q_v}{2^n}$

Conclusion

- Two efficient alternatives for the standard counter scheme.
- A general notion for counters and counter based encoding.
- Counter property based security results for some schemes.
- Software performance comparison between the three counter schemes.

Thank you.