Beyond $2^{c/2}$ Security in Sponge-Based Authenticated Encryption Modes

Philipp Jovanovic[1], Atul Luykx[2], and Bart Mennink[2]

[1] Universität Passau [2] KU Leuven

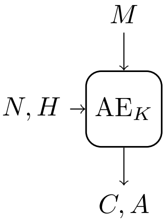
DIAC — August 23, 2014

Authenticated Encryption

- Encryption and authentication in one
- Applications: SSH, IPsec, TLS, IEEE 802.11
- CAESAR competition

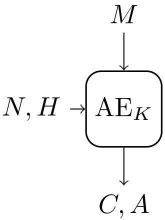
Authenticated Encryption

- Encryption and authentication in one
- Applications: SSH, IPsec, TLS, IEEE 802.11
- CAESAR competition



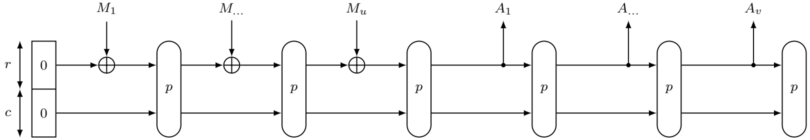
Authenticated Encryption

- Encryption and authentication in one
- Applications: SSH, IPsec, TLS, IEEE 802.11
- CAESAR competition



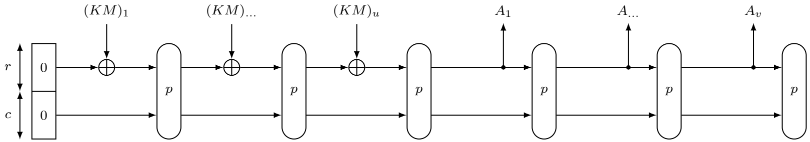
- Security goals: privacy + integrity
  - Nonce-dependent or security against nonce-reuse

Sponge Functions



- Bertoni, Daemen, Peeters, and Van Assche (2007)
- Based on permutation p
- $b = r + c$

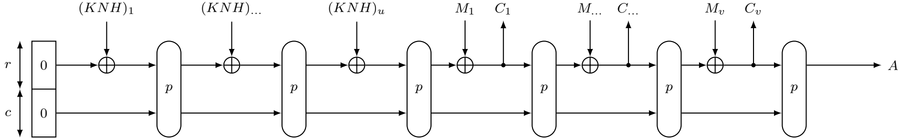
Sponge Functions



- Bertoni, Daemen, Peeters, and Van Assche (2007)
- Based on permutation p
- $b = r + c$

- MAC: Keyed sponge (secret key K prepended to M)

Sponge Functions



- Bertoni, Daemen, Peeters, and Van Assche (2007)
- Based on permutation p
- $b = r + c$

- MAC: Keyed sponge (secret key K prepended to M)
- AE: SpongeWrap (duplexing mode)

Sponge Functions

Sponge (hash) $2^{c/2}$ security

c = capacity κ = key size τ = tag size

Sponge Functions

Sponge (hash)	$2^{c/2}$ security
Keyed sponge (MAC)	$\min\{2^{c-a}, 2^{\kappa}\}$ security (2^a offline compl.)

c = capacity κ = key size τ = tag size

Sponge Functions

Sponge (hash)	$2^{c/2}$ security
Keyed sponge (MAC)	$\min\{2^{c-a}, 2^{\kappa}\}$ security (2^a offline compl.)
	$\approx \min\{2^{c/2}, 2^{\kappa}\}$ security

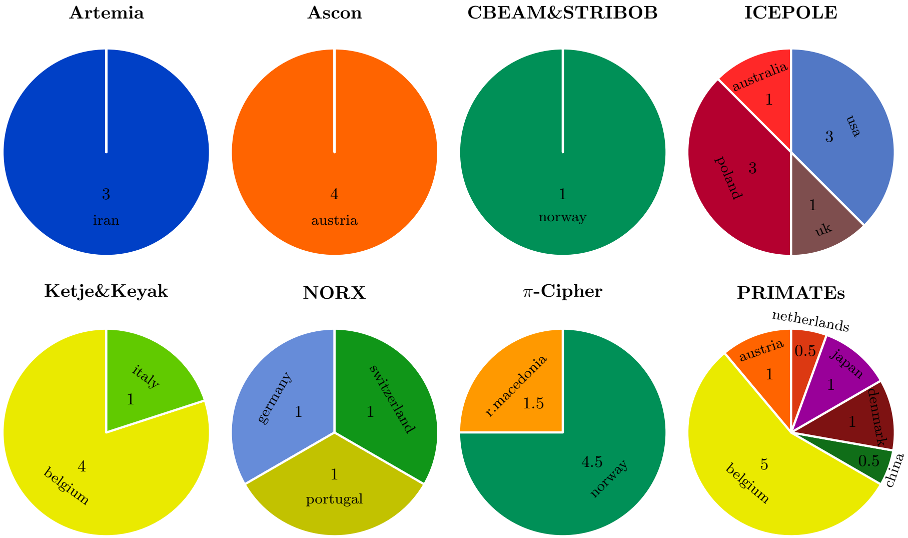
c = capacity κ = key size τ = tag size

Sponge Functions

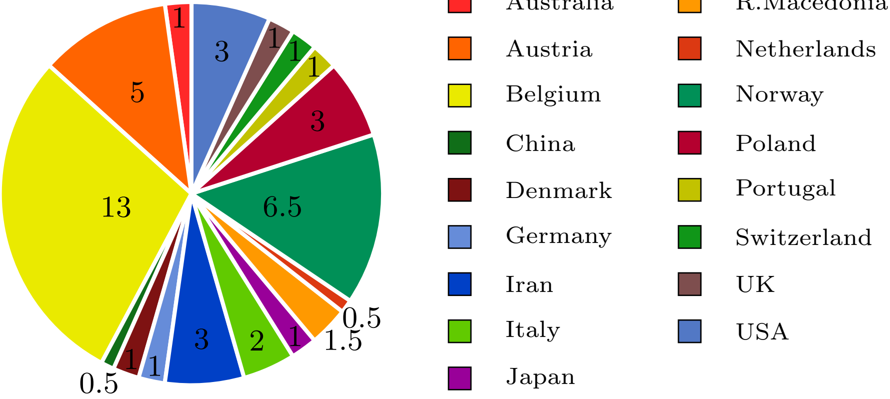
Sponge (hash)	$2^{c/2}$ security
Keyed sponge (MAC)	$\min\{2^{c-a}, 2^{\kappa}\}$ security (2^a offline compl.)
	$\approx \min\{2^{c/2}, 2^{\kappa}\}$ security
SpongeWrap (AE)	$\min\{2^{c/2}, 2^{\kappa}\}$ security (privacy)
	$\min\{2^{c/2}, 2^{\kappa}, 2^{\tau}\}$ security (integrity)

c = capacity $\quad$ κ = key size $\quad$ τ = tag size

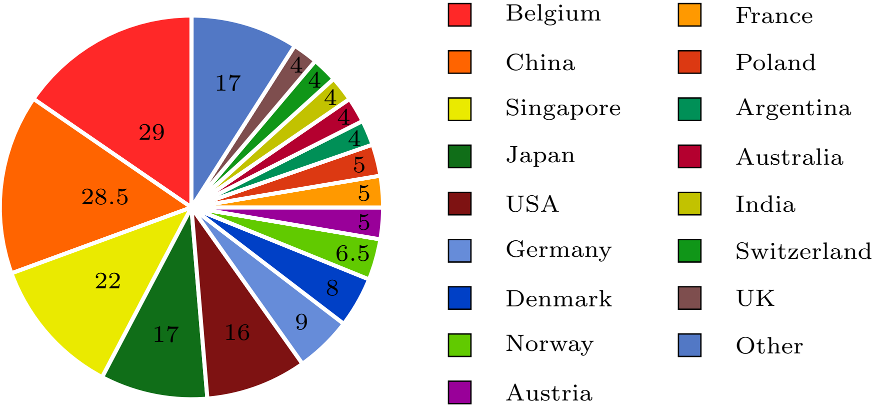
Sponge-Based CAESAR Modes



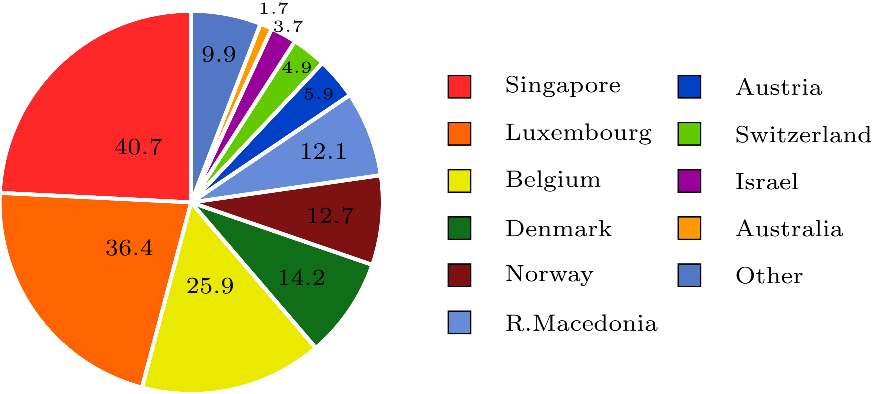
Sponge-Based CAESAR Modes



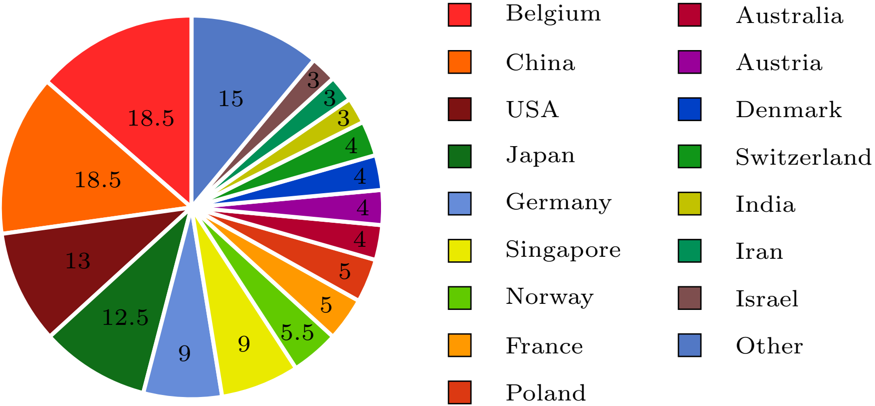
Intermezzo – All CAESAR Contributors



Intermezzo – All CAESAR Contributors (10.000.000/capita)



Intermezzo – All CAESAR Contributors (no duplicate)



Sponge-Based CAESAR Modes

nonce-dependent	security against nonce-reuse
Artemia	APE[2,3]
Ascon	
CBEAM/STRIBOB[1]	
ICEPOLE	
Ketje	
Keyak	
NORX	
π-Cipher	
GIBBON/HANUMAN[2]	

[1] CBEAM and STRIBOB use BLNK sponge mode

[2] PRIMATEs = {GIBBON, HANUMAN, APE}

[3] also used in submission Prøst

Sponge-Based CAESAR Modes

nonce-dependent	**security against nonce-reuse**
Artemia	APE[2,3]
Ascon	
CBEAM/STRIBOB[1]	
ICEPOLE	
Ketje	
Keyak	
NORX	
π-Cipher	
GIBBON/HANUMAN[2]	

[1] CBEAM and STRIBOB use BLNK sponge mode

[2] PRIMATEs = {GIBBON, HANUMAN, APE}

[3] also used in submission Prøst

$2^{c/2}$ security (tight)

Sponge-Based CAESAR Modes

nonce-dependent	security against nonce-reuse
Artemia	APE[2,3]
Ascon	
CBEAM/STRIBOB[1]	
ICEPOLE	
Ketje	
Keyak	
NORX	
π-Cipher	
GIBBON/HANUMAN[2]	

parameters based on $2^{c/2}$ and $(2^a, 2^{c-a})$ results

$2^{c/2}$ security (tight)

[1] CBEAM and STRIBOB use BLNK sponge mode

[2] PRIMATEs $=$ {GIBBON, HANUMAN, APE}

[3] also used in submission Prøst

Sponge-Based CAESAR Modes

nonce-dependent	security against nonce-reuse
Artemia	APE [2,3]
Ascon	
CBEAM/STRIBOB [1]	
ICEPOLE	
Ketje	
Keyak	
NORX	
π-Cipher	
GIBBON/HANUMAN [2]	

[1] CBEAM and STRIBOB use BLNK sponge mode

[2] PRIMATEs $= \{\text{GIBBON}, \text{HANUMAN}, \text{APE}\}$

[3] also used in submission Prøst

parameters based on $2^{c/2}$ and $(2^a, 2^{c-a})$ results

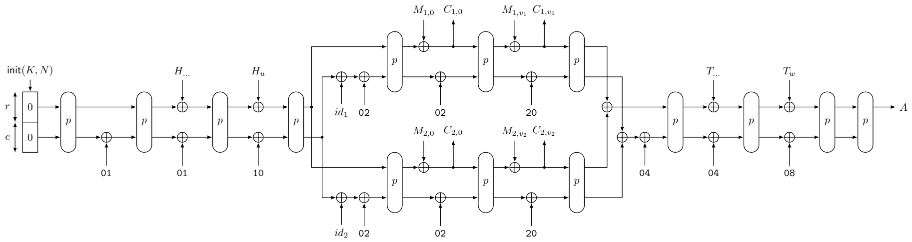
$2^{c/2}$ security (tight)

Nonce changes everything!

Sponge-Based CAESAR Modes

	b	c	r	κ	**security**
Ascon	320	192	128	96	**96**
	320	256	64	128	**128**
CBEAM	256	190	66	128	**128**
ICEPOLE	1280	254	1026	128	**128**
	1280	318	962	256	**256**
Keyak	800	252	548	128	**128**
	1600	252	1348	128	**128**
NORX	512	192	320	128	**128**
	1024	384	640	256	**256**
GIBBON/ HANUMAN	200	159	41	80	**80**
	280	239	41	120	**120**
STRIBOB	512	254	258	192	**192**

NORX



- Submission by Aumasson, Jovanovic, and Neves
- Initialization with K and unique N
- Header – message – trailer
- Parallelism $D \in \{0, \dots, 255\}$ (here, $D = 2$)

NORX: Mode Security

Privacy

$\min\{2^{b/2}, 2^c, 2^\kappa\}$ security

Integrity

$\min\{2^{b/2}, 2^c, 2^\kappa, 2^\tau\}$ security

NORX: Mode Security

Privacy

$\min\{2^{b/2}, 2^c, 2^\kappa\}$ security

Integrity

$\min\{2^{b/2}, 2^c, 2^\kappa, 2^\tau\}$ security

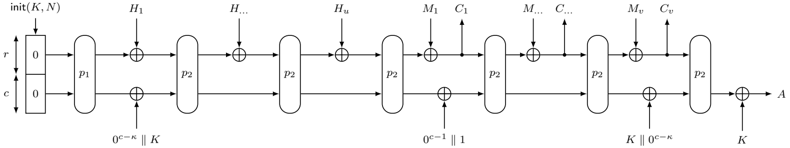
Main Implication

putting $c = \kappa$ does not decrease mode security level

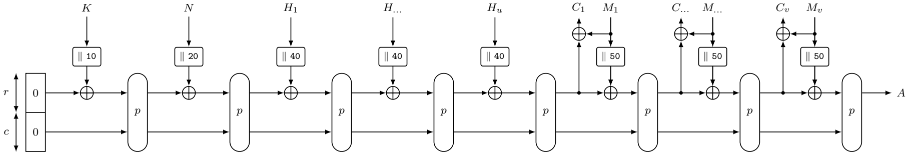
Generalization

- Generalizes to SpongeWrap and DuplexWrap
- Generalizes to CAESAR submission modes
    - Ascon
    - BLNK (used in CBEAM and STRIBOB)
    - ICEPOLE
    - Keyak
    - GIBBON and HANUMAN (two PRIMATEs)

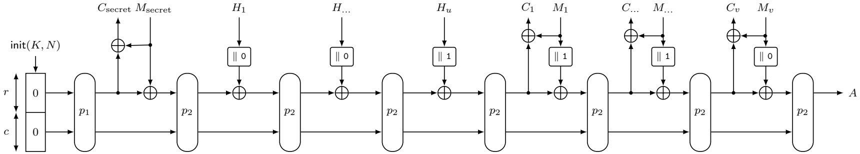
Generalization



Ascon

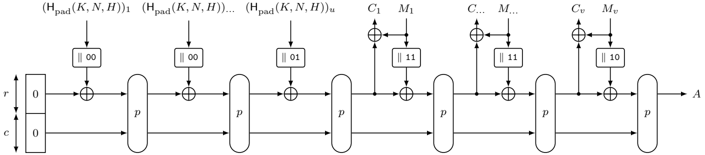


BLNK (used in CBEAM and STRIBOB)

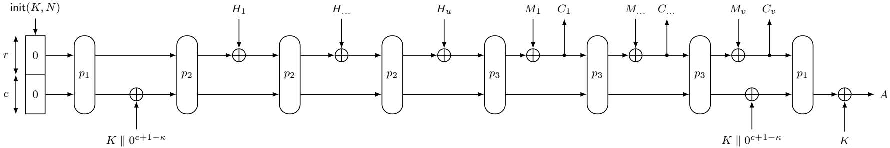


ICEPOLE

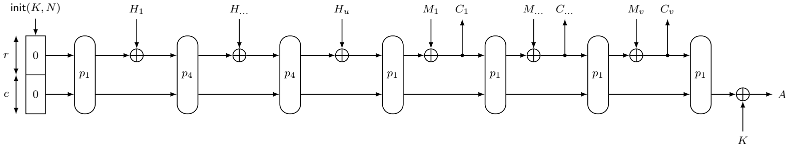
Generalization



Keyak



GIBBON (PRIMATEs)



HANUMAN (PRIMATEs)

New Security Levels

	b	c	r	κ	**security**
Ascon	320	192	128	96	**96**
	320	256	64	128	**128**
CBEAM	256	190	66	128	**128**
ICEPOLE	1280	254	1026	128	**128**
	1280	318	962	256	**256**
Keyak	800	252	548	128	**128**
	1600	252	1348	128	**128**
NORX	512	192	320	128	**128**
	1024	384	640	256	**256**
GIBBON/ HANUMAN	200	159	41	80	**80**
	280	239	41	120	**120**
STRIBOB	512	254	258	192	**192**

New Security Levels

	b	c	r	$\frac{r}{r_{\text{old}}}$	κ	**security**
Ascon	320	96	224	1.75	96	**96**
	320	128	192	3	128	**128**
CBEAM	256	190	66		128	**128**
ICEPOLE	1280	254	1026		128	**128**
	1280	318	962		256	**256**
Keyak	800	252	548		128	**128**
	1600	252	1348		128	**128**
NORX	512	192	320		128	**128**
	1024	384	640		256	**256**
GIBBON/ HANUMAN	200	159	41		80	**80**
	280	239	41		120	**120**
STRIBOB	512	254	258		192	**192**

New Security Levels

	b	c	r	$\frac{r}{r_{\text{old}}}$	κ	**security**
Ascon	320	96	224	1.75	96	**96**
	320	128	192	3	128	**128**
CBEAM	256	128	128	1.94	128	**128**
ICEPOLE	1280	128	1152	1.12	128	**128**
	1280	256	1024	1.06	256	**256**
Keyak	800	128	672	1.23	128	**128**
	1600	128	1472	1.09	128	**128**
NORX	512	128	384	1.2	128	**128**
	1024	256	768	1.2	256	**256**
GIBBON/ HANUMAN	200	80	120	2.93	80	**80**
	280	120	160	3.90	120	**120**
STRIBOB	512	192	320	1.24	192	**192**

Conclusions

From $\min\{2^{c/2}, 2^{\kappa}\}$ to $\min\{2^{b/2}, 2^{c}, 2^{\kappa}\}$

- Applies to
  - SpongeWrap and DuplexWrap
  - Modes of Ascon, CBEAM, ICEPOLE, Keyak, NORX, PRIMATEs, and STRIBOB

Conclusions

From $\min\{2^{c/2}, 2^{\kappa}\}$ to $\min\{2^{b/2}, 2^{c}, 2^{\kappa}\}$

- Applies to
  - SpongeWrap and DuplexWrap
  - Modes of Ascon, CBEAM, ICEPOLE, Keyak, NORX, PRIMATEs, and STRIBOB
- Current parameter choices overly conservative
- Schemes can operate up to $4\times$ as fast
  without **mode security** degradation

Conclusions

From $\min\{2^{c/2}, 2^{\kappa}\}$ to $\min\{2^{b/2}, 2^{c}, 2^{\kappa}\}$

- Applies to
  - SpongeWrap and DuplexWrap
  - Modes of Ascon, CBEAM, ICEPOLE, Keyak, NORX, PRIMATEs, and STRIBOB
- Current parameter choices overly conservative
- Schemes can operate up to $4\times$ as fast
  without **mode security** degradation

Thank you for your attention!

http://eprint.iacr.org/2014/373

SUPPORTING SLIDES

NORX: Privacy

$\min\{2^{b/2}, 2^c, 2^\kappa\}$ security

NORX: Privacy

$$\min\{2^{b/2}, 2^c, 2^\kappa\} \text{ security}$$

Security Model

- Adversary tries to distinguish $(p, \mathcal{E}_K^p)$ from $(p, \$)$
  - Random permutation p, key K, and AE $\$$
  - Define m = total complexity = $q + \sigma_\mathcal{E}$

NORX: Privacy

$$\min\{2^{b/2}, 2^c, 2^\kappa\} \text{ security}$$

Security Model

- Adversary tries to distinguish $(p, \mathcal{E}_K^p)$ from $(p, \$)$
  - Random permutation p, key K, and AE $\$$
  - Define m = total complexity = $q + \sigma_{\mathcal{E}}$

Simplified Proof Idea

- Everything "fine" as long as no collision or key guess

NORX: Privacy

$$\min\{2^{b/2}, 2^c, 2^\kappa\} \text{ security}$$

Security Model

- Adversary tries to distinguish $(p, \mathcal{E}_K^p)$ from $(p, \$)$
    - Random permutation p, key K, and AE $\$$
    - Define m = total complexity = $q + \sigma_\mathcal{E}$

Simplified Proof Idea

- Everything "fine" as long as no collision or key guess
- Colliding $\mathcal{E}$-state with $\mathcal{E}$-state $\longrightarrow \sigma_\mathcal{E}^2/2^b$ (unique nonce)

NORX: Privacy

$$\min\{2^{b/2}, 2^c, 2^\kappa\} \text{ security}$$

Security Model

- Adversary tries to distinguish $(p, \mathcal{E}^p_K)$ from $(p, \$)$
    - Random permutation p, key K, and AE $\$$
    - Define m = total complexity = $q + \sigma_\mathcal{E}$

Simplified Proof Idea

- Everything “fine” as long as no collision or key guess
- Colliding $\mathcal{E}$-state with $\mathcal{E}$-state $\longrightarrow \sigma_\mathcal{E}^2/2^b$ (unique nonce)
- Colliding $\mathcal{E}$-state with p-query $\longrightarrow \sigma_\mathcal{E} q/2^c$ (naive)

NORX: Privacy

$$\min\{2^{b/2}, 2^c, 2^\kappa\} \text{ security}$$

Security Model

- Adversary tries to distinguish $(p, \mathcal{E}_K^p)$ from $(p, \$)$
  - Random permutation p, key K, and AE $\$$
  - Define m = total complexity = $q + \sigma_\mathcal{E}$

Simplified Proof Idea

- Everything "fine" as long as no collision or key guess
- Colliding $\mathcal{E}$-state with $\mathcal{E}$-state $\longrightarrow \sigma_\mathcal{E}^2/2^b$ (unique nonce)
- Colliding $\mathcal{E}$-state with p-query $\longrightarrow \sigma_\mathcal{E} q/2^c$ (naive)

  - p-query fixes rate part of $\mathcal{E}$-state

NORX: Privacy

$\min\{2^{b/2}, 2^c, 2^\kappa\}$ security

Security Model

- Adversary tries to distinguish $(p, \mathcal{E}_K^p)$ from $(p, \$)$
  - Random permutation p, key K, and AE $\$$
  - Define m = total complexity = $q + \sigma_{\mathcal{E}}$

Simplified Proof Idea

- Everything "fine" as long as no collision or key guess
- Colliding $\mathcal{E}$-state with $\mathcal{E}$-state $\longrightarrow \sigma_{\mathcal{E}}^2/2^b$ (unique nonce)
- Colliding $\mathcal{E}$-state with p-query $\longrightarrow \sigma_{\mathcal{E}} q/2^c$ (naive)
  - p-query fixes rate part of $\mathcal{E}$-state
  - $\#\{$relevant $\mathcal{E}$-states$\} =: \rho$

NORX: Privacy

$$\min\{2^{b/2}, 2^c, 2^\kappa\} \text{ security}$$

Security Model

- Adversary tries to distinguish $(p, \mathcal{E}_K^p)$ from $(p, \$)$
  - Random permutation p, key K, and AE $\$$
  - Define m = total complexity = $q + \sigma_\mathcal{E}$

Simplified Proof Idea

- Everything "fine" as long as no collision or key guess
- Colliding $\mathcal{E}$-state with $\mathcal{E}$-state $\longrightarrow \sigma_\mathcal{E}^2/2^b$ (unique nonce)
- Colliding $\mathcal{E}$-state with p-query $\longrightarrow \cancel{\sigma_\mathcal{E} q/2^c}$ (naive)
  $\longrightarrow \rho q/2^c$ (multiplicity)
  - p-query fixes rate part of $\mathcal{E}$-state
  - $\#\{\text{relevant } \mathcal{E}\text{-states}\} =: \rho$

NORX: Privacy

$$\min\{2^{b/2}, 2^c, 2^\kappa\} \text{ security}$$

Security Model

- Adversary tries to distinguish $(p, \mathcal{E}_K^p)$ from $(p, \$)$
  - Random permutation p, key K, and AE $\$$
  - Define m = total complexity = $q + \sigma_\mathcal{E}$

Simplified Proof Idea

- Everything "fine" as long as no collision or key guess
- Colliding $\mathcal{E}$-state with $\mathcal{E}$-state $\longrightarrow \sigma_\mathcal{E}^2/2^b$ (unique nonce)
- Colliding $\mathcal{E}$-state with p-query $\longrightarrow \cancel{\sigma_\mathcal{E} q/2^c}$ (naive)
  $\longrightarrow \rho q/2^c$ (multiplicity)
  - p-query fixes rate part of $\mathcal{E}$-state
  - $\#\{$relevant $\mathcal{E}$-states$\} =: \rho \leq \max\left\{r, \left(\frac{\sigma_\mathcal{E} 2^c}{q 2^r}\right)^{1/2}\right\}$

NORX: Integrity

$\min\{2^{b/2}, 2^c, 2^\kappa, 2^\tau\}$ security

NORX: Integrity

$$\min\{2^{b/2}, 2^c, 2^\kappa, 2^\tau\} \text{ security}$$

Security Model

- Adversary with access to $(p, \mathcal{E}_K^p, \mathcal{D}_K^p)$ aims to forge
  - Random permutation p and key K
  - Define m = total complexity = $q + \sigma_\mathcal{E} + \sigma_\mathcal{D}$
- Technical issue: adversary can re-use nonce!

NORX: Integrity

$$\min\{2^{b/2}, 2^c, 2^\kappa, 2^\tau\} \text{ security}$$

Security Model

- Adversary with access to $(p, \mathcal{E}_K^p, \mathcal{D}_K^p)$ aims to forge
  - Random permutation p and key K
  - Define m = total complexity = $q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}}$
- Technical issue: adversary can re-use nonce!

Simplified Proof Idea

- Collisions not involving $\mathcal{D}$-state $\longrightarrow \sigma_{\mathcal{E}}^2/2^b + \rho q/2^c$

NORX: Integrity

$$\min\{2^{b/2}, 2^c, 2^\kappa, 2^\tau\} \text{ security}$$

Security Model

- Adversary with access to $(p, \mathcal{E}_K^p, \mathcal{D}_K^p)$ aims to forge
  - Random permutation p and key K
  - Define m = total complexity = $q + \sigma_\mathcal{E} + \sigma_\mathcal{D}$
- Technical issue: adversary can re-use nonce!

Simplified Proof Idea

- Collisions not involving $\mathcal{D}$-state $\longrightarrow \sigma_\mathcal{E}^2/2^b + \rho q/2^c$
- Collisions involving $\mathcal{D}$-state $\longrightarrow m\sigma_D/2^c$ (nonce re-use)

NORX: Integrity

$$\min\{2^{b/2}, 2^c, 2^\kappa, 2^\tau\} \text{ security}$$

Security Model

- Adversary with access to $(p, \mathcal{E}_K^p, \mathcal{D}_K^p)$ aims to forge
  - Random permutation p and key K
  - Define m = total complexity = $q + \sigma_\mathcal{E} + \sigma_\mathcal{D}$
- Technical issue: adversary can re-use nonce!

Simplified Proof Idea

- Collisions not involving $\mathcal{D}$-state $\longrightarrow \sigma_\mathcal{E}^2/2^b + \rho q/2^c$
- Collisions involving $\mathcal{D}$-state $\longrightarrow m\sigma_D/2^c$ (nonce re-use)

$\sigma_\mathcal{D}$ relatively small

NORX: Integrity

$$\min\{2^{b/2}, 2^c, 2^\kappa, 2^\tau\} \text{ security}$$

Security Model

- Adversary with access to $(p, \mathcal{E}_K^p, \mathcal{D}_K^p)$ aims to forge
  - Random permutation p and key K
  - Define m = total complexity = $q + \sigma_\mathcal{E} + \sigma_\mathcal{D}$
- Technical issue: adversary can re-use nonce!

Simplified Proof Idea

- Collisions not involving $\mathcal{D}$-state $\longrightarrow \sigma_\mathcal{E}^2/2^b + \rho q/2^c$
- Collisions involving $\mathcal{D}$-state $\longrightarrow m\sigma_D/2^c$ (nonce re-use)

$\sigma_\mathcal{D}$ relatively small

- As long as no collisions, forgery $\rightarrow \sigma_\mathcal{D}/2^\tau$