NORX

A Parallel and Scalable Authenticated Encryption Scheme

Jean-Philippe Aumasson[1] (@veorq)

Philipp Jovanovic[2] (@daeinar)

Samuel Neves[3] (@sevenps)

[1]Kudelski Security, Switzerland

[2]University of Passau, Germany

[3]University of Coimbra, Portugal

ESORICS 2014

Wrocław, September 08, 2014

Outline

1. Motivation
2. Specification
3. Performance
4. Security Analysis
5. Conclusion

“Nearly all of the symmetric encryption modes you learned about in school, textbooks, and Wikipedia are (potentially) insecure.”

—Matthew Green

Authenticated Encryption

Authenticated Encryption

Types

- AE: ensure *confidentiality*, *integrity*, and *authenticity* of a message
- AEAD: AE + ensure *integrity* and *authenticity* of associated data (e.g. routing information in IP packets)

Generic Composition

- Symmetric encryption algorithm (confidentiality)
- Message Authentication Code (MAC) (authenticity, integrity)

Applications

- Standard technology to protect in-transit data
- IPSec, SSH, TLS, ...

Authenticated Encryption

Problems with Existing AE(AD) Schemes

- Interaction flaws between enc. and auth. in generic composition
- Weak primitives (e.g. RC4)
- Broken modes (e.g. EAXprime)
- Misuse resistant solutions barely used
- No reliable standards
- More examples: `http://competitions.cr.yp.to/disasters.html`

⇒ Lots of room for improvements

Authenticated Encryption

Problems with Existing AE(AD) Schemes

- Interaction flaws between enc. and auth. in generic composition
- Weak primitives (e.g. RC4)
- Broken modes (e.g. EAXprime)
- Misuse resistant solutions barely used
- No reliable standards
- More examples: `http://competitions.cr.yp.to/disasters.html`

$\Rightarrow$ Lots of room for improvements

CAESAR

- **C**ompetition for **A**uthenticated **E**ncryption: **S**ecurity, **A**pplicability, and **R**obustness.
- **Goals**: Identify a portfolio of *authenticated ciphers* that
  - offer advantages over AES-GCM (the current de-facto standard) and
  - are suitable for widespread adoption.
- **Overview**:
  - March 15 2014 – End of 2017
  - 1st round: 57 submissions
  - `http://competitions.cr.yp.to/caesar.html`
- **Further Information**:
  - AEZoo: `https://aezoo.compute.dtu.dk`
  - Speed comparison: `http://www1.spms.ntu.edu.sg/~syllab/speed`

NO(T A)RX

Overview of NORX

Main Design Goals

- High security
- Efficiency
- Simplicity
- Scalability
- Online
- Side-channel robustness
  (e.g. constant-time operations)
- High key agility

Overview of NORX

Parameters

- *Word size:* $W \in \{32, 64\}$ bits
- *Number of rounds:* $1 \leq R \leq 63$
- *Parallelism degree:* $0 \leq D \leq 255$
- *Tag size:* $|A| \leq 10W$

Instances

Rank	NORXW-R-D	Nonce size ($2W$)	Key size ($4W$)	Tag size ($4W$)	Classification
1	NORX64-4-1	128	256	256	Standard
2	NORX32-4-1	64	128	128	Standard
3	NORX64-6-1	128	256	256	High security
4	NORX32-6-1	64	128	128	High security
5	NORX64-4-4	128	256	256	High throughput

Overview of NORX

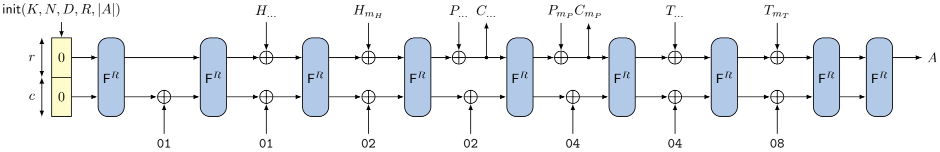
Parameters

- *Word size:* $W \in \{32, 64\}$ bits
- *Number of rounds:* $1 \leq R \leq 63$
- *Parallelism degree:* $0 \leq D \leq 255$
- *Tag size:* $|A| \leq 10W$

Instances

Rank	NORXW-R-D	Nonce size ($2W$)	Key size ($4W$)	Tag size ($4W$)	Classification
1	NORX64-4-1	128	256	256	Standard
2	NORX32-4-1	64	128	128	Standard
3	NORX64-6-1	128	256	256	High security
4	NORX32-6-1	64	128	128	High security
5	NORX64-4-4	128	256	256	High throughput

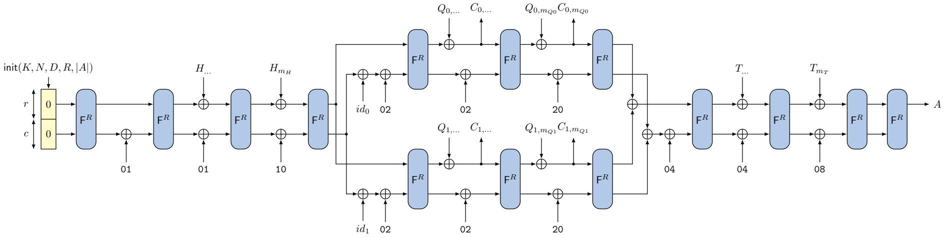
NORX Mode



NORX in Sequential Mode ($D = 1$)

Features

- (Parallel) monkeyDuplex construction (derived from Keccak/SHA-3)
- Processes header, payload and trailer data in one-pass
- Data expansion via multi-rate padding: 10^*1
- Extensible (e.g. sessions, secret message numbers)
- Parallelisable



NORX in Parallel Mode ($D = 2$)

Features

- (Parallel) monkeyDuplex construction (derived from Keccak/SHA-3)
- Processes header, payload and trailer data in one-pass
- Data expansion via multi-rate padding: 10^*1
- Extensible (e.g. sessions, secret message numbers)
- Parallelisable

The State

- NORX operates on a state of *16 W*-bit sized words

W	Size	Rate	Capacity
32	512	320	192
64	1024	640	384

- Arrangement of rate (data processing) and capacity (security) words:

s_0	s_1	s_2	s_3
s_4	s_5	s_6	s_7
s_8	s_9	s_{10}	s_{11}
s_{12}	s_{13}	s_{14}	s_{15}

Initialisation

- Load nonce, key and constants into state S:

u_0	n_0	n_1	u_1
k_0	k_1	k_2	k_3
u_2	u_3	u_4	u_5
u_6	u_7	u_8	u_9

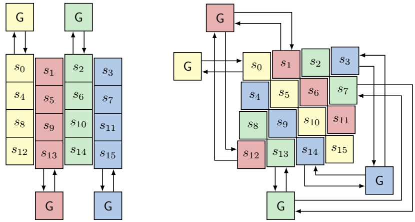
- Parameter integration:

$$s_{14} \leftarrow s_{14} \oplus (R \lll 26) \oplus (D \lll 18) \oplus (W \lll 10) \oplus |A|$$

- Apply round permutation F^R to S

The Permutation F^R

The Permutation F



The Permutation G

1: $a \longleftarrow \mathrm{H}(a, b)$
2: $d \longleftarrow (a \oplus d) \ggg r_0$
3: $c \longleftarrow \mathrm{H}(c, d)$
4: $b \longleftarrow (b \oplus c) \ggg r_1$
5: $a \longleftarrow \mathrm{H}(a, b)$
6: $d \longleftarrow (a \oplus d) \ggg r_2$
7: $c \longleftarrow \mathrm{H}(c, d)$
8: $b \longleftarrow (b \oplus c) \ggg r_3$

The Non-linear Operation H

$$\mathrm{H} : \mathbb{F}_2^{2n} \rightarrow \mathbb{F}_2^n, (x, y) \mapsto (x \oplus y) \oplus \big((x \wedge y) \ll 1\big)$$

Rotation Offsets (r_0, r_1, r_2, r_3)

32-bit: $(8, 11, 16, 31)$ 64-bit: $(8, 19, 40, 63)$

The Permutation F^R

Features

- F and G derived from ARX-primitives ChaCha/BLAKE2
- H is an "approximation" of integer addition:

$$a + b = (a \oplus b) + \big((a \wedge b) \ll 1\big)$$

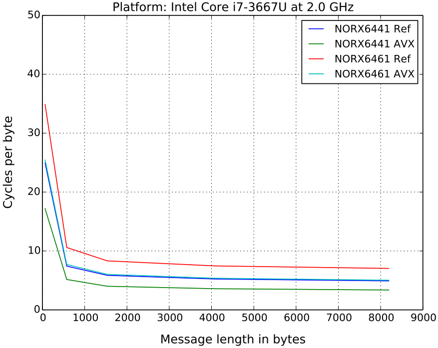
- LRX permutation
- No SBoxes or integer additions
- SIMD friendly
- HW friendly
- High diffusion
- Constant-time

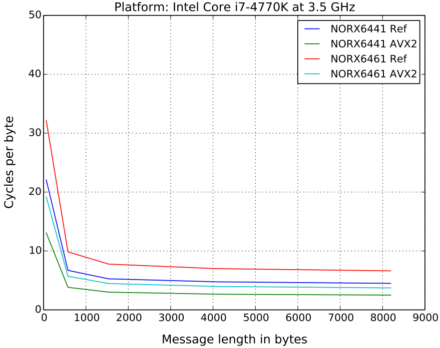
Requirements for Secure Usage of NORX

1. **Unique nonces**
2. **Abort on tag verification failure**

Performance

SW Performance (x86)

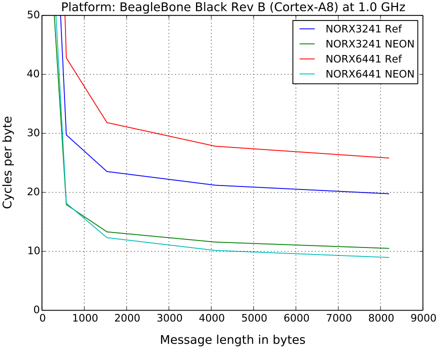


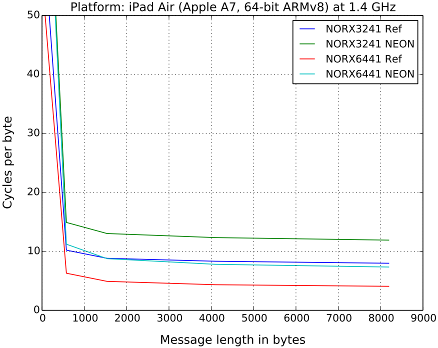


Platform	Implementation	cpb	MiBps
Ivy Bridge: i7 3667U @ 2.0 GHz	AVX	3.37	593
Haswell: i7 4770K @ 3.5 GHz	AVX2	2.51	1390

Table: NORX64-4-1 performance

SW Performance (ARM)





Platform	Implementation	cpb	MiBps
BBB: Cortex-A8 @ 1.0 GHz	NEON	8.96	111
iPad Air: Apple A7 @ 1.4 GHz	Ref	4.07	343

Table: NORX64-4-1 performance

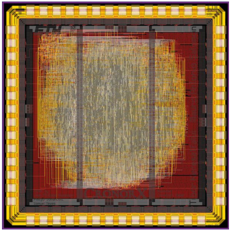
SW Performance (SUPERCOP)

titan0	wintermute	sachr	hydra9	h6dragon	flops	h6sandy	gcc16	floodyberry	h1mips
aegis128l	aes128gcmv1	morus640128v1	aegis128l	norx6441v1	morus640128v1	morus640128v1	morus640128v1	norx6441v1	morus1280128v1
tiaoxinv1	aes256gcmv1	morus1280128v1	tiaoxinv1	morus640128v1	norx3241v1	morus1280128v1	morus1280256v1	norx6461v1	morus1280256v1
aegis128	norx6441v1	morus1280256v1	aegis128	norx3241v1	hs1sivlov1	morus1280256v1	morus1280128v1	norx3241v1	hs1sivlov1
aegis256	norx6461v1	norx6441v1	aegis256	morus1280128v1	norx6441v1	norx6441v1	hs1sivlov1	norx6444v1	wheeshtv1mr3fr1t256
kiasuneq128v1	norx3241v1	aegis128l	kiasuneq128v1	morus1280256v1	morus1280128v1	norx3241v1	norx6441v1	ascon96v1	wheeshtv1mr3fr1t128
morus1280128v1	norx6444v1	norx6461v1	silverv1	norx6461v1	morus1280256v1	norx6461v1	hs1sivv1	morus640128v1	norx3241v1
morus1280256v1	morus640128v1	aegis128	morus1280128v1	hs1sivlov1	norx3261v1	hs1sivlov1	ascon96v1	norx3261v1	aegis128l
silverv1	norx3261v1	norx3241v1	deoxysneq128128v1	norx3261v1	aegis128l	aes128gcmv1	norx6444v1	aes128gcmv1	hs1sivv1
morus640128v1	ascon96v1	aegis256	deoxysneq256128v1	aegis128l	wheeshtv1mr3fr1t128	tiaoxinv1	norx3241v1	aes256gcmv1	wheeshtv1mr3fr3t256
kiasueq128v1	ascon128v1	norx3261v1	kiasueq128v1	wheeshtv1mr3fr1t256	wheeshtv1mr3fr1t256	wheeshtv1mr3fr1t128	norx6461v1	ascon128v1	aegis128
deoxysneq128128v1	aes128otrsv1	aes128gcmv1	morus1280256v1	wheeshtv1mr3fr1t128	hs1sivv1	wheeshtv1mr3fr1t256	aegis128l	pi64cipher128v1	norx6441v1
deoxysneq256128v1	aes128otrpv1	norx6444v1	deoxyseq128128v1	tiaoxinv1	norx6461v1	aegis128l	aegis128	pi64cipher256v1	norx3261v1
aes128gcmv1	aes256otrsv1	aes256gcmv1	morus640128v1	aegis128	aegis128	aes256gcmv1	ascon128v1	aes128otrsv1	aegis256
aes256gcmv1	aesjambuv1	aes128otrsv1	deoxyseq256128v1	wheeshtv1mr3fr3t256	wheeshtv1mr3fr3t256	hs1sivv1	pi64cipher256v1	aes128otrpv1	tiaoxinv1
deoxyseq128128v1	aes256otrpv1	aes128otrpv1	aes128gcmv1	hs1sivv1	aegis256	wheeshtv1mr3fr3t256	pi64cipher128v1	omdsha512k256n256tau256v1	norx6444v1
aes128cpfbv1	omdsha512k512n256tau256v1	pi32cipher256v1	aes128cpfbv1	aegis256	aes128gcmv1	norx3261v1	aegis256	omdsha512k512n256tau256v1	hs1sivhiv1
deoxyseq256128v1	omdsha512k256n256tau256v1	pi32cipher128v1	aes256gcmv1	aes128gcmv1	aes256gcmv1	aegis128	silverv1	omdsha512k128n128tau128v1	norx6461v1
norx6441v1	omdsha512k128n128tau128v1	aes256otrsv1	aes256cpfbv1	hs1sivhiv1	hs1sivhiv1	norx6444v1	icepole256av1	pi32cipher256v1	silverv1
aes256cpfbv1	stribob192r1	aesjambuv1	norx6441v1	aes256gcmv1	wheeshtv1mr5fr7t256	aegis256	icepole128av1	pi32cipher128v1	wheeshtv1mr5fr7t256
norx6461v1	omdsha256k128n96tau64v1	aes256otrpv1	hs1sivlov1	norx6444v1	cba1	hs1sivhiv1	icepole128v1	aesjambuv1	yaes128v2
hs1sivlov1	omdsha256k128n96tau96v1	kiasuneq128v1	wheeshtv1mr3fr1t128	wheeshtv1mr5fr7t256	cba3	kiasuneq128v1	norx3261v1	aes256otrsv1	aes128otrsv1
norx3241v1	omdsha256k192n104tau128v1	ascon96v1	wheeshtv1mr3fr1t256	cba8	cba2	scream10v2	aes128cpfbv1	aes256otrpv1	aes128otrpv1
wheeshtv1mr3fr1t128	omdsha256k128n96tau128v1	pi64cipher256v1	norx3241v1	cba1	cba4	scream10v1	pi32cipher128v1	omdsha256k192n104tau128v1	icepole256av1
wheeshtv1mr3fr1t256	omdsha256k256n104tau160v1	pi64cipher128v1	norx6461v1	cba4	cba6	ascon96v1	pi32cipher256v1	omdsha256k256n104tau160v1	icepole128v1
hs1sivv1	omdsha256k256n248tau256v1	ascon128v1	wheeshtv1mr3fr3t256	cba5	cba5	scream12v2	aes128otrsv1	omdsha256k128n96tau64v1	icepole128av1
norx6444v1	lakekeyakv1	kiasueq128v1	hs1sivv1	aes128cpfbv1	cba8	wheeshtv1mr5fr7t256	yaes128v2	omdsha256k128n96tau128v1	cba1
wheeshtv1mr3fr3t256	seakeyakv1	stribob192r1	norx6444v1	cba3	aes128cpfbv1	scream12v1	aes128otrpv1	omdsha256k128n96tau96v1	cba3
ascon96v1	oceankeyakv1	trivia0v1	ascon96v1	silverv1	cba7	iscream12v2	cba2	omdsha256k256n248tau256v1	cba2
norx3261v1	ketjesrv1	ifeedaes128n104v1	aesjambuv1	cba2	aeadaes128ocbtaglen96v1	iscream12v1	cba3	lakekeyakv1	cba7
hs1sivhiv1	riverkeyakv1	ifeedaes128n96v1	norx3261v1	cba7	aeadaes128ocbtaglen128v1	pi64cipher256v1	cba1	seakeyakv1	cba4
pi64cipher256v1	ketjejrv1	lakekeyakv1	wheeshtv1mr5fr7t256	cba6	silverv1	icepole128av1	cba4	trivia0v1	cba5
aesjambuv1	trivia0v1	riverkeyakv1	hs1sivhiv1	aeadaes128ocbtaglen128v1	aeadaes128ocbtaglen64v1	silverv1	cba8	stribob192r1	cba8
scream10v2	aes128poetv1aes4	seakeyakv1	scream10v1	aeadaes128ocbtaglen64v1	cba9	icepole256av1	cba5	oceankeyakv1	cba6
scream10v1	aes128poetv1aes128	pi16cipher096v1	scream10v2	aeadaes128ocbtaglen96v1	aeadaes192ocbtaglen96v1	icepole128v1	cba7	pi16cipher096v1	aeadaes128ocbtaglen96v1
icepole256av1	aes128poetv1gf128mul	pi16cipher128v1	ascon128v1	cba9	aeadaes192ocbtaglen64v1	iscream14v2	cba6	pi16cipher128v1	aeadaes128ocbtaglen64v1
icepole128av1		lacv11	icepole128av1	yaes128v2	aeadaes192ocbtaglen128v1	iscream14v1	aes128n8clocv1	riverkeyakv1	aeadaes128ocbtaglen128v1
icepole128v1		ketjesrv1	icepole128v1	aeadaes192ocbtaglen128v1	cba10	aes128cpfbv1	aes128n12clocv1	ketjesrv1	aes128n8clocv1
ascon128v1		oceankeyakv1	icepole256av1	aeadaes192ocbtaglen96v1	yaes128v2	kiasueq128v1	aeadaes128ocbtaglen128v1	ketjejrv1	aes128n12clocv1
scream12v2		deoxysneq128128v1	pi64cipher256v1	aeadaes192ocbtaglen64v1	norx6444v1	pi64cipher128v1	aeadaes128ocbtaglen96v1	aes128poetv1aes4	aes256otrsv1
scream12v1		ketjejrv1	scream12v2	cba10	aeadaes256ocbtaglen128v1	ascon128v1	aeadaes128ocbtaglen64v1	aes128poetv1aes128	aes128cpfbv1
iscream12v2		deoxysneq256128v1	scream12v1	aeadaes256ocbtaglen128v1	aeadaes256ocbtaglen64v1	aes128otrsv1	aes128n12silcv1	aes128poetv1gf128mul	ascon96v1

Source: `http://www1.spms.ntu.edu.sg/~syllab/speed`

- NORX among the fastest CAESAR ciphers
- Fastest Sponge-based scheme
- Reference implementation has competitive speed, too

HW Performance (ASIC)



ASIC implementation and hardware evaluation by ETHZ students (under supervision of Frank K. Gürkaynak):

- Parameters: $W \in \{32, 64\}$, $R \in \{2, \ldots, 16\}$ and $D = 1$
- Technology: 180 nm UMC
- Frequency: 125 MHz
- Area requirements: 59 kGE
- NORX64-4-1 performance: 10 Gbps ≈ 1200 MiBps

Security Analysis

Security

Sponge Security Bounds

- Classic: $\min\{2^{c/2}, 2^{|K|}\}$
  - NORX designed towards this bound
  - Expected security levels ($c - e - 1$, $e = 2W$): 127 and 255 bits
- Improved*: $\min\{2^{b/2}, 2^{c}, 2^{|K|}\}$
  - Nonce-based sponges in the ideal perm. model
  - Includes NORX with $D \neq 1$
  - Effects: rate $+2W$ bits ($\approx +16\%$ performance)

* P. Jovanovic, A. Luykx, and B. Mennink, Beyond $2^{c/2}$ Security in Sponge-Based Authenticated Encryption Modes, Advances in Cryptology - ASIACRYPT 2014. To appear.

Cryptanalysis

NODE – The (NO)RX (D)ifferential Search (E)ngine*

- Framework for automatic search of differential trails in F^R
- Uses constraint / SAT solvers (STP, Boolector, CryptoMiniSat)
- Some results:

R	type	NORX32	NORX64	
1	nonce	$< 2^{-60}$	$< 2^{-53}$	bound
4	perm.	2^{-584}	2^{-836}	best

- Bonus: Variant of NODE allowed us to break Wheesht and McMambo, two other CAESAR candidates

* J-P. Aumasson, P. Jovanovic, and S. Neves, Analysis of NORX, Latincrypt 2014. To appear.
Software available on GitHub: `https://github.com/norx/NODE`

Cryptanalysis

NODE – The (NO)RX (D)ifferential Search (E)ngine*

- Framework for automatic search of differential trails in F^R
- Uses constraint / SAT solvers (STP, Boolector, CryptoMiniSat)
- Some results:

R	type	NORX32	NORX64	
1	nonce	$< 2^{-60}$	$< 2^{-53}$	bound
4	perm.	2^{-584}	2^{-836}	best

- Bonus: Variant of NODE allowed us to break Wheesht and McMambo, two other CAESAR candidates

* J-P. Aumasson, P. Jovanovic, and S. Neves, Analysis of NORX, Latincrypt 2014. To appear.

Software available on GitHub: `https://github.com/norx/NODE`

Cryptanalysis

NODE – The (NO)RX (D)ifferential Search (E)ngine*

- Framework for automatic search of differential trails in F^R
- Uses constraint / SAT solvers (STP, Boolector, CryptoMiniSat)
- Some results:

R	type	NORX32	NORX64	
1	nonce	$< 2^{-60}$	$< 2^{-53}$	bound
4	perm.	2^{-584}	2^{-836}	best

- Bonus: Variant of NODE allowed us to break Wheesht and McMambo, two other CAESAR candidates

* J-P. Aumasson, P. Jovanovic, and S. Neves, Analysis of NORX, Latincrypt 2014. To appear.

Software available on GitHub: `https://github.com/norx/NODE`

Cryptanalysis

NODE – The (NO)RX (D)ifferential Search (E)ngine*

- Framework for automatic search of differential trails in F^R
- Uses constraint / SAT solvers (STP, Boolector, CryptoMiniSat)
- Some results:

R	type	NORX32	NORX64	
1	nonce	$< 2^{-60}$	$< 2^{-53}$	bound
4	perm.	2^{-584}	2^{-836}	best

- Bonus: Variant of NODE allowed us to break Wheesht and McMambo, two other CAESAR candidates

* J-P. Aumasson, P. Jovanovic, and S. Neves, Analysis of NORX, Latincrypt 2014. To appear.

Software available on GitHub: `https://github.com/norx/NODE`

Conclusion

Open Problems

- Cryptanalysis: linear, algebraic, (adv.) differential, (adv.) rotational
- Side-channel attacks
- Further implementations: e.g. FPGAs, microcontroller

Take Aways

Features of NORX

- Secure, fast, and scalable
- Based on well-analysed primitives: ChaCha/BLAKE(2)/Keccak
- Clean and simple design
- HW and SW friendly
- Parallelisable
- Side-channel robustness considered during design phase
- Straightforward to implement
- No AES dependence

Further Information

https://norx.io

Contact:
jovanovic@fim.uni-passau.de
@Daeinar

Take Aways

Features of NORX

- Secure, fast, and scalable
- Based on well-analysed primitives: ChaCha/BLAKE(2)/Keccak
- Clean and simple design
- HW and SW friendly
- Parallelisable
- Side-channel robustness considered during design phase
- Straightforward to implement
- No AES dependence

Further Information

`https://norx.io`

Contact:
jovanovic@fim.uni-passau.de
@Daeinar

Supplement: NORX vs AES-GCM

	NORX	AES-GCM
High performance	yes (on many platforms)	depends (high with AES-NI)
High key agility	yes	no
Timing resistance	yes	no (bit-slicing, AES-NI required)
Misuse resistance	A+N / LCP+X (exposes $P \oplus P'$)	no (exposes K)
Parallelisation	yes	yes
Extensibility	yes (sessions, secret msg. nr., etc.)	no
Simple implementation	yes	no