Cryptanalysis of NORX v2.0

C. Chaigneau\textsuperscript{1}, T. Fuhr\textsuperscript{2}, H. Gilbert\textsuperscript{1,2}, J. Jean\textsuperscript{2}, J.-R. Reinhard\textsuperscript{2}

\textsuperscript{1} University of Versailles \quad \textsuperscript{2} ANSSI, France

ESC Luxembourg, January 16, 2017
NORX family of AEAD algorithms

- submitted in 2014 to the CAESAR competition [AJN14]

- main features
  - two word-size options: $w = 32$ or $64$ bits $\rightarrow$ NORX32 and NORX64 instances
  - key and tag size: $|K| = |T| = 4w = 128$ or $256$ bits $=$ claimed security level
  - sponge-based construction (MonkeyDuplex)
  - state permutation $P$: not ARX, but inspired from ARX primitives, e.g. ChaCha

- 3 main versions
  - NORX v1.0: initial version submitted to CAESAR (March 2014)
  - NORX v2.0: one of the 15 round 2 candidates (August 2015) selected in August 2016 for Round 3
  - NORX v3.0: tweaked version (September 2016) the basis of the Round 3 evaluation
NORX security analysis: published results

- [JLM14]: beyond $2^{c/2}$ security proofs in sponge-based AE modes
  - provides arguments for decreasing the v1.0 capacity by $2w$ in NORX v2.0
- [AJN15]: differential and rotational properties of NORX
- [DMM15]: higher-order differential properties of NORX state/key
- [BHJMS16]: key-recovery for reduced version of NORX v2.0
  - the number of rounds of the underlying permutation is halved
outline of this talk

0 main features of NORX v2.0
1 a strong structural distinguisher for the NORX permutation \( P \)
2 resulting existential forgery attack on full NORX v2.0
   • uses \( D \) accesses to a KP encryption oracle + \( D \) forgery attempts
   • complexity: \( T \approx D \approx 2^{2w+2} \) for \( p^{\text{success}} > 1/2 \)
     \[ T \approx D \approx 2^{66} \] for NORX32 instance \([2^{72} \text{ if inputs are byte strings}]\)
     \[ T \approx D \approx 2^{130} \] for NORX64 instance \([2^{136} \text{ if inputs are byte strings}]\)
resulting key-recovery attack on full NORX v2.0
   • same time and data complexity
3 impact on other variants, extra NORX security issues
NORX v2.0 / NORX32 instance

- **word size:** \( w = 32 \) bits
- **key** \( K \): 128 bits, **tag** \( T \): 128 bits, **nonce** \( N \): 64 bits
- **state size** \( s \): 512 bits = **rate** \( r \): 384 bits + **capacity** \( c \): 128 bits

init\((K, N, w, l, t)\)

- **data limitation:** at most \( 2^{64} \) messages can be processed with one key
permutation $P$

- $P$: state permutation over $\{0,1\}^{16 \times 32}$
- $P = F^4$ [4 rounds]
- $F$: uses a permutation $G$ of $\{0,1\}^{4 \times 32}$

followed by

\[ \equiv G_{\text{col}} \]

[same overall structure as ChaCha permutation]
\[ \{0, 1\}^4 \times 32 \text{ permutation } G \]

where

\[ x \boxdot y \overset{\text{def}}{=} (x \oplus y) \oplus ((x \land y) \ll 1) \]
non-random properties of $P$

- designers’ observation [AJN15b]: existence of weak states (obs$_1$)
  
  if all columns of $S \in \{0,1\}^{16 \times 32}$ are equal i.e. $S = \begin{pmatrix} a & a & a & a \\ b & b & b & b \\ c & c & c & c \\ d & d & d & d \end{pmatrix}$
  
  then: all columns of $\mathbb{G}_{\text{col}}(S)$, of $\mathbb{G}_{\text{diag}}(S)$ and therefore of $P(S)$ are equal

- a stronger distinguishing property observed independently in [BUV17]
  
  $\mathbb{G}_{\text{col}}, \mathbb{G}_{\text{diag}}$ and $P$ commute with the left columns rotation $S \mapsto S \ll 1$

  $\forall S \quad P(S\ll 1) = P(S)\ll 1$ (obs$_2$)

  this implies obs$_1$: if $S\ll 1 = S$ then $P(S)\ll 1 = P(S)$

  more generally: $\forall j \in \{1,2,3\} \quad \forall S \quad P(S\ll j) = P(S)\ll j$ (obs$_3$)

  $\rightarrow$ in the sequel we will use $j=2$
proof: commuting permutation property of $\mathcal{G}_i^{\text{col}}$

$$
\begin{array}{cccc}
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}
\end{array}
\quad \mapsto \quad
\begin{array}{cccc}
s_1 & s_2 & s_3 & s_0 \\
s_5 & s_6 & s_7 & s_4 \\
s_9 & s_{10} & s_{11} & s_8 \\
s_{13} & s_{14} & s_{15} & s_{12}
\end{array}

\mapsto

\begin{array}{cccc}
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}
\end{array}
\quad \mapsto \quad
\begin{array}{cccc}
s'_{3} & s'_0 & s'_1 & s'_2 \\
s'_7 & s'_4 & s'_5 & s'_6 \\
s'_11 & s'_8 & s'_9 & s'_{10} \\
s'_{15} & s'_{12} & s'_{13} & s'_{14}
\end{array}

\quad \lll
\quad \lll
\quad \lll
\quad \lll

proof: commuting permutation property of $\mathbf{G}_{\text{diag}}$

\[
\begin{array}{cccc}
s_0 & s_1 & s_2 & s_3 \\
s_4 & s_5 & s_6 & s_7 \\
s_8 & s_9 & s_{10} & s_{11} \\
s_{12} & s_{13} & s_{14} & s_{15}
\end{array}
\]

\[
\begin{array}{cccc}
s_1 & s_2 & s_3 & s_0 \\
s_5 & s_6 & s_7 & s_4 \\
s_9 & s_{10} & s_{11} & s_8 \\
s_{13} & s_{14} & s_{15} & s_{12}
\end{array}
\]

\[
\begin{array}{cccc}
s'_0 & s'_1 & s'_2 & s'_3 \\
s'_4 & s'_5 & s'_6 & s'_7 \\
s'_8 & s'_9 & s'_{10} & s'_{11} \\
s'_{12} & s'_{13} & s'_{14} & s'_{15}
\end{array}
\]

\[
\begin{array}{cccc}
s'_1 & s'_2 & s'_3 & s'_0 \\
s'_5 & s'_6 & s'_7 & s'_4 \\
s'_9 & s'_{10} & s'_{11} & s'_8 \\
s'_{13} & s'_{14} & s'_{15} & s'_{12}
\end{array}
\]
forgery attack: conducting idea (1/2)

- how to rotate some state $S$ by 2 columns in order to apply $\text{obs}_3$?

- if $S_c$ is symmetric i.e. $s_{12}$ $s_{13}$ = $s_{14}$ $s_{15}$ i.e. $S_c$

then $S' = S\ll\ll2$ as desired and $\text{obs}_3$ can be applied

this occurs with proba $p = 2^{-2w} = 2^{-64}$ since the capacity is only 128 bits
for forgery attack: conducting idea (2)

if

\[ S_{c} = \text{symmetric} \]

\[ = \text{init}(K, N, w, l, t) \]

\[ M \]

\[ 384 \]

\[ 128 \]

\[ 01 \]

\[ 02 \]

\[ 128 \]

\[ T \]

then:

\[ S_{c} = \text{symmetric} \]

\[ = \text{init}(K, N, w, l, t) \]

\[ C<<<2 \]

\[ 384 \]

\[ 128 \]

\[ 01 \]

\[ 02 \]

\[ 128 \]

\[ T<<<2 \]

due to obs\[3\]

\[ s_{0} \quad s_{1} \quad s_{2} \quad s_{3} \]

\[ s_{4} \quad s_{5} \quad s_{6} \quad s_{7} \]

\[ s_{8} \quad s_{9} \quad s_{10} \quad s_{11} \]

\[ s_{12} \quad s_{13} \quad s_{14} \quad s_{15} \]

\[ s_{12} \quad s_{13} = s_{14} \quad s_{15} \]
Padding affects attack details

- \( S_r = C || s \)

where \( s = 10*1 \oplus x \)

(padding \( \oplus \) unknown mask)

\( \exists C' \) s.t. \( C'||s = (C||s) \ll 2 \iff s \) is a suffix of \( c_9 \)

Proba: \( 2^{-|s|} = 2^{-2} \) or \( 2^{-8} \) depending if NORX operates on bits or bytes

- \( S_c \ll 2 = S_c \) with proba \( 2^{-2w} \)

Complexity: \( T = D = 2^{2w+|s|} = 2^{66} \) or \( 2^{72} \)
forgery attack: summary

- adversary’s resources: D accesses to an encryption oracle
  - with chosen plaintext length [typically 382 or 376 bits]
  - plaintexts can be known or not [this does not matter here]
    + D forgery attempts

- the attack can be viewed in two ways:
  - \( T = D \approx 1 \rightarrow p_{\text{success}} = 2^{-66} \) \([2^{-72} \text{ if NORX operates on bytes}]\)
  - \( T = D \approx 2^{66} \text{ encryptions} \) \([2^{72} \text{ if NORX operates on bytes}] \rightarrow p_{\text{success}} > 1/2\)

  note: while the NORX key must be changed after \(2^{64}\) encryptions
  this does not affect the marginal success proba. of each forgery attempt
resulting key-recovery attack

\[
\text{init}(K, N, w, l, t) \quad \quad M \quad C \quad 128 \quad T
\]

\[
\text{init}(K, N, w, l, t) \quad \quad M' \quad C' \quad S <<<<< 2
\]

successful forgery

⇒ \( S_c \) is symmetric (and \( S_r \) is known)

⇒ only \( 2^{64} \) possible values for \( S_c \) and thus for \( S \)

exhaustive search of \( S \): test if \( P^2(S) = T||^* \)

⇒ recovery of \( K \) assuming that \( M \) is known [or that \( M' \) is returned]

• backward computation of \( S^{\text{init}} \) starting from \( S \) [or \( S <<<< 2 \)]
  [optionally: test that \( S^{\text{init}} \) has the expected redundancy]

• \( S^{\text{init}} \) provides \( K \)

\[
T = D = 2^{66} \quad (2^{72} \text{ if NORX operates on bytes})
\]
are other NORX variants affected?

- **NORX v1.0:** not affected due to its more conservative capacity 6w

- **NORX v3.0:** while the two properties leveraged in v2.0 remain
  v3.0 appears to be immune due to the introduction of an initial and two final key additions to the capacity

- **NORX 8 and NORX 16:** appear to be immune though their features are quite similar to those of NORX v2.0
some extra NORX security issues

independently from the former attack and the non-random behaviour of P

- the claimed 128-bit security level of NORX32 v2.0 and v3.0
does not hold for the integrity of long messages
  [due to the small capacity $c = 128$ bits]

- the claimed 80-bit security level of the lightweight variant NORX8
does not hold if one can encrypt long plaintexts
  [due to the small state size $s = 128$ bits]
init\( (K, N, w, l, t) \)

\[
\begin{align*}
\text{known long reference ciphertext } \quad &L=2^m \\
M_0 &\quad C_0 \\
M_1 &\quad C_1 \\
M_{L-1} &\quad C_{L-1}
\end{align*}
\]

forgery attempt: only \(C_0\) is modified

init\( (K, N', w, l, t) \)

at each iteration both capacity chains collide with proba \(\approx 2^{-128}\)

\[
\rightarrow \text{ forgery proba } 2^{m-128} \approx 2^{-80} \quad \text{if } L=2^{48}
\]

contradicts the claimed 128-bit integrity, not the security bounds of [JLM14]
NORX-8: too small state size?

- \( w: 8 \text{ bits} \quad r: 40\text{bits} \quad c: 88 \text{ bits} \)
- \(|\text{key}| = |\text{tag}| = 80 \text{ bits} = \text{conjectured security level}\)
- use of the same key is limited to \(2^{24}\) initialization phases
- 80-bit tag \(T\) is generated in two 40-bit halves: \(T = T'||T''\)

\[
\text{init}(K, N, w, l, t)
\]

encryption of L-block sequence of 0 blocks, \(L = 2^m \sim \text{stream cipher}\)

→ classical TD trade-offs can be applied [B95, G97]

→ intermediate state → key value

\[ T = \#\text{states} / L = 2^{128-m} \text{ thus: } T << 2^{80} \text{ if } L >> 2^{48} \]

→ key usage limitation to \(2^{24}\) initializations does not suffice to achieve a 80-bit security
concluding remarks

[on the forgery and key-recovery attack on NORX v2.0]

- yet another invariant permutation attack
  a special type of invariant subspace attack introduced in \[\text{[LMR15]}\]
  - invariant permutation identified here: \(\cdot \ll\ll 2\)
  - associated invariant vector space: \(\{S|S\ll\ll 2=S\}\)

- the attacks on NORX v2.0 leveraged an interaction
  between two non-conservative properties of NORX v2.0
  - small capacity = claimed security level (128 bits for NORX32)
  - a strong structural property of \(P\)
  whether these properties also affect the security of NORX v3.0
  remains an open question