Key-recovery attacks on NORX v2.0 and Kravatte

Thomas Fuhr

Agence nationale de la sécurité des systèmes d'information, France

October 10, 2018

Outline of the presentation

Outline

- Why is cryptanalysis of symmetric algorithms necessary ?
- Forgery and key-recovery attacks against NORX **[CFGJR17]**
- Two key-recovery attacks against Kravatte **[CFGGJRS18]**
- Concluding remarks

Outline of the presentation

Outline

- **Why is cryptanalysis of symmetric algorithms necessary ?**
- Forgery and key-recovery attacks against NORX **[CFGJR17]**
- Two key-recovery attacks against Kravatte **[CFGGJSR18]**
- Concluding remarks

Confidence in cryptographic algorithms

Security proofs

- Asymmetric cryptography: link between a difficult problem and the security of a cryptographic algorithm
- Symmetric cryptography: guarantee the robustness of a construction when instanciated with an ideal primitive
- Reduce the perimeter of cryptographic functions to analyse

Cryptanalysis

- Asymmetric cryptography: study of mathematical "difficult" problems (factorisation, DL, lattices...)
- Symmetric cryptography: study of (often ad hoc) cryptographic primitives (AES, Keccak-p,...)
- **Turning properties of cryptographic primitives into attacks on complex constructions**

The need for strong security models - A real-life comparison

Protecting your house against burglars

- Average door: might be OK if
  - You live in a crowded street
  - You have a dog
  - You have an alarm
  - ...

The need for strong security models - A real-life comparison

Protecting your house against burglars

- Average door: might be OK if
  - You live in a crowded street
  - You have a dog
  - You have an alarm
  - ...
- Or you have a **reinforced door**

The need for strong security models - A real-life comparison

Protecting your house against burglars

- Average door: might be OK if
  - You live in a crowded street
  - You have a dog
  - You have an alarm
  - ...
- Or you have a **reinforced door**

Using cryptographic algorithms

- Security models can seem unrealistic at first sight
  - Collision resistance for hash functions
  - Chosen ciphertext scenarii
  - Distinguishing an algorithm from a random primitive

The need for strong security models - A real-life comparison

Protecting your house against burglars

- Average door: might be OK if
  - You live in a crowded street
  - You have a dog
  - You have an alarm
  - ...
- Or you have a **reinforced door**

Using cryptographic algorithms

- Security models can seem unrealistic at first sight
  - Collision resistance for hash functions
  - Chosen ciphertext scenarii
  - Distinguishing an algorithm from a random primitive
- SHA-1 collision, padding oracle attacks, ...
- Algorithms secure in strong models avoid discussions

Conception of symmetric mechanisms

A conservative approach

- Model: SHA-3 competition
- Use a mode with a security proof (Chop-MD, Sponge functions,...)
- Instanciate it with a primitive without known weaknesses
- Discard candidates with unexpected structural properties

Faster algorithms

- Use of faster but weaker primitives
- Seems sound for keyed functions (i.e., no hash functions)
- **Requires more cryptanalysis**
- Examples: NORX, Kravatte

Our results

Cryptanalysis of NORX v2.0

- Forgery attack, can be exploited to recover the key
- FSE 2017, ToSC 2017 and JoC 2018
- Joint work with C. Chaigneau, H. Gilbert, J. Jean, J.-R. Reinhard

Cryptanalysis of Kravatte

- Several key-recovery attacks on early versions
- FSE 2018, ToSC 2018 and JoC 2018
- Joint work with C. Chaigneau, H. Gilbert, J. Guo, J. Jean, J.-R. Reinhard, L. Song

Outline of the presentation

Outline

- Why is cryptanalysis of symmetric algorithms necessary ?
- **Forgery and key-recovery attacks against NORX [CFGJR17]**
- Two key-recovery attacks against Kravatte **[CFGGJSR18]**
- Concluding remarks

A CAESAR Candidate

- **CAESAR competition**: Authenticated Encryption with Associated Data (AEAD)
- **Timeline**
  - March 2014: 56 initial submissions
  - July 2015: 28 candidates selected for 2nd round
  - August 2016: 15 candidates selected for 3rd round
  - March 2018: Announcement of the 7 finalists (without NORX)
- **The NORX family of ciphers** (Aumasson, Jovanovic, Neves)
  - Initial submission: NORX v1
  - August 2015: NORX v2.0 (CAESAR round 2)
  - September 2016: NORX v3.0 (CAESAR round 3)
  - Lightweight variants NORX8 and NORX16

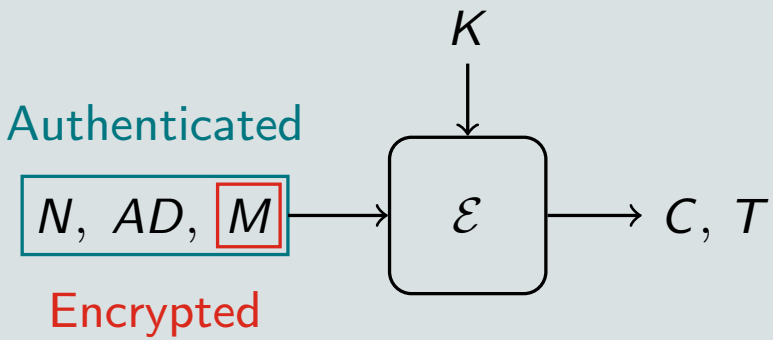
Our results

- Ciphertext-only **forgery attack on full NORX v2.0**
  - Probability $\geq 50\%$ for 2^{66} **forgery attempts**
  - Time complexity 2^{66}
  - Claim: no forgery attempt with proba. $\gg 2^{n-128}$ for n attempts
- Trivial **known-plaintext key-recovery** once a forgery is achieved

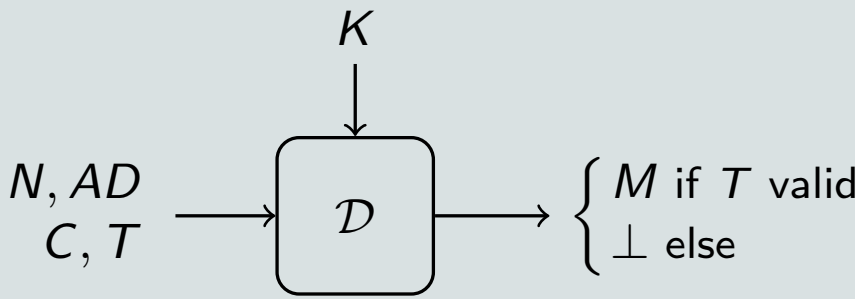
- Related work
  - IND-CCA proof of the mode **[JLM14]**
  - Analyses of the permutation **[AJN14]**, **[AJN15]**, **[BUV17]**
  - Attacks on reduced variants **[BHJMS16]**, **[DMM15]**

AEAD framework

Encryption



Decryption



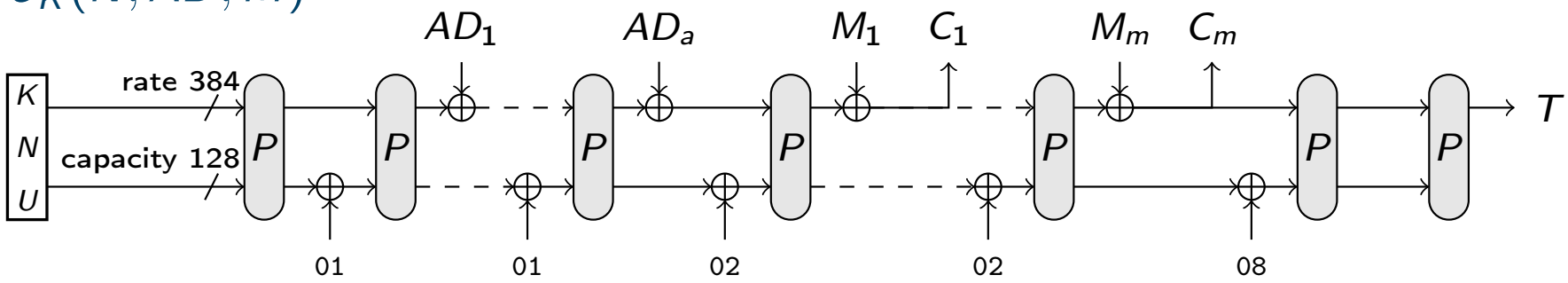
Notations

- M: Plaintext
- AD: Associated data
- N: Nonce
- K: AEAD Key
- C: Ciphertext
- T: Authentication Tag

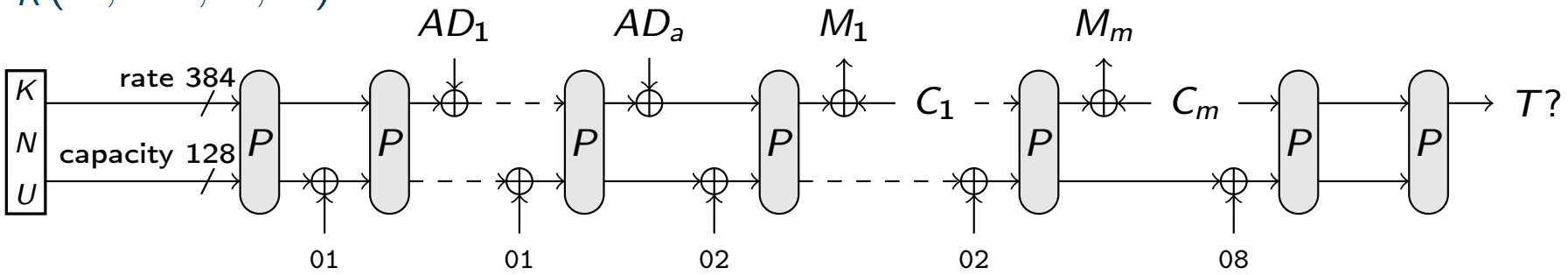
NORX mode of operation

- MonkeyDuplex mode **[BDPV12]**
- This presentation: 128-bit keys, 64-bit nonces

$\mathcal{E}_K(N, AD, M)$



$\mathcal{D}_K(N, AD, C, T)$



The permutation P

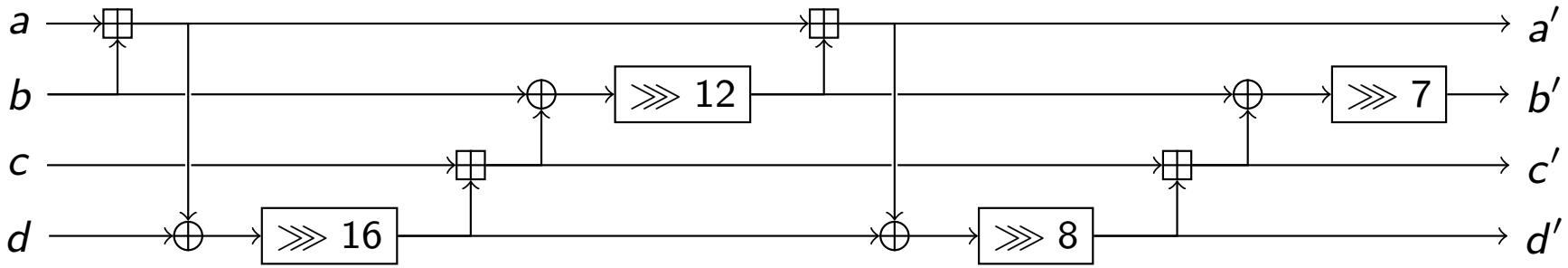
- Inspired by stream cipher ChaCha **[B08]**
- Operates on a 512-bit state S
- Represented as a 4×4 **matrix** of 32-bit words

$$S = \begin{pmatrix} s_0 & s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 & s_7 \\ s_8 & s_9 & s_{10} & s_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix} \begin{matrix} \} \\ \} \end{matrix} \begin{matrix} \text{Rate part} \\ \\ \text{Capacity part} \end{matrix}$$

- P relies on a 128-**bit permutation** G

The permutation P

- G: 4-branch generalised Feistel



$$a', b', c', d' = G'(a, b, c, d)$$
$$x \boxplus y = (x \oplus y) \oplus (x \wedge y) \ll 1$$

- $P = (G_{\text{diag}} \circ G_{\text{col}})^4$ (*i*.e. G on columns then on diagonals)

$$G_{\text{col}} : \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{pmatrix} \quad G_{\text{diag}} : \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_3 & b_0 & b_1 & b_2 \\ c_2 & c_3 & c_0 & c_1 \\ d_1 & d_2 & d_3 & d_0 \end{pmatrix}$$

Properties of P

- Preservation of symmetries **[AJN15]**

$$\begin{pmatrix} a & a & a & a \\ b & b & b & b \\ c & c & c & c \\ d & d & d & d \end{pmatrix} \xrightarrow{P} \begin{pmatrix} a' & a' & a' & a' \\ b' & b' & b' & b' \\ c' & c' & c' & c' \\ d' & d' & d' & d' \end{pmatrix}$$

- More generally, P **commutes with rotations** on columns

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

State S State $S \lll 1$

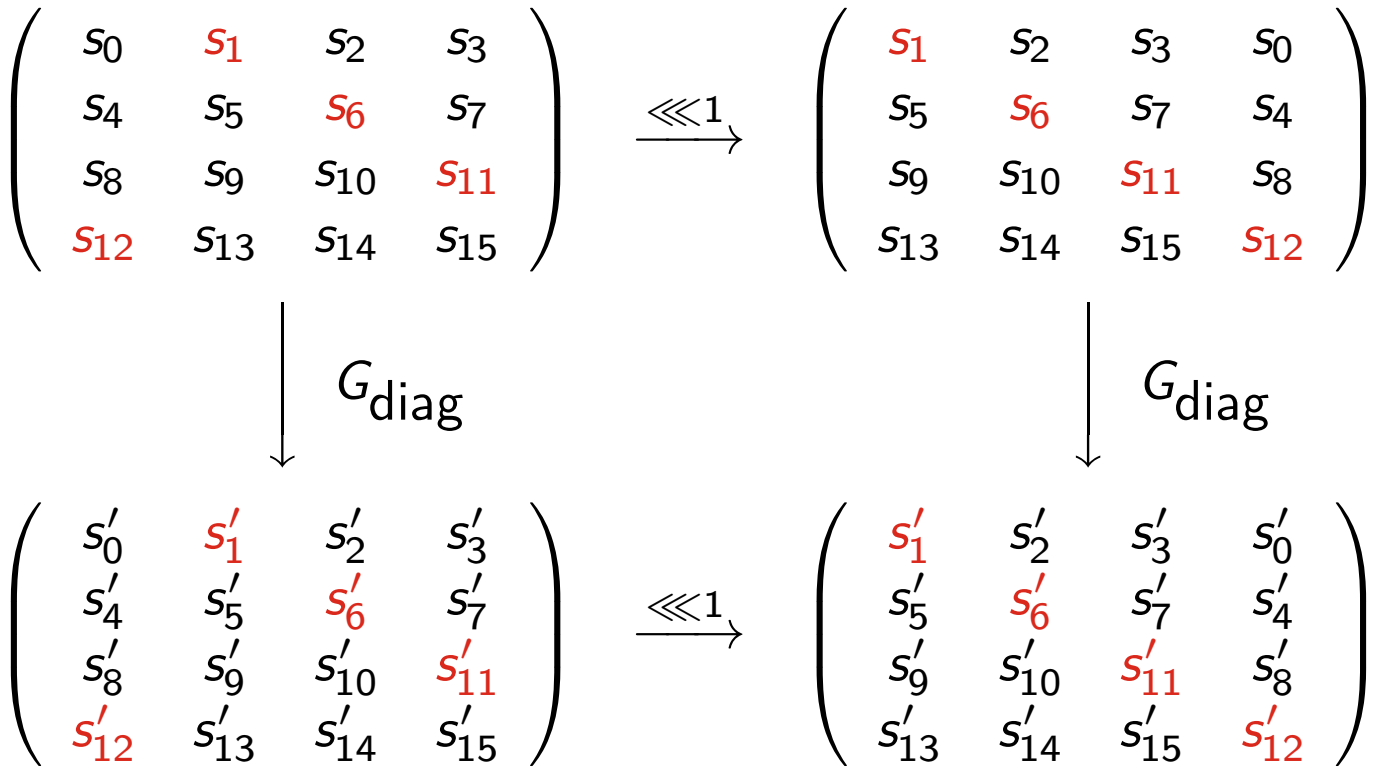
$$\forall i \in \{1, 2, 3\}, P(S \lll i) = P(S) \lll i$$

Sketch of proof

$$\begin{array}{ccc}
\begin{pmatrix} s_0 & s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 & s_7 \\ s_8 & s_9 & s_{10} & s_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix} & \xrightarrow{\lll 1} & \begin{pmatrix} s_1 & s_2 & s_3 & s_0 \\ s_5 & s_6 & s_7 & s_4 \\ s_9 & s_{10} & s_{11} & s_8 \\ s_{13} & s_{14} & s_{15} & s_{12} \end{pmatrix} \\
\downarrow G_{\mathsf{col}} & & \downarrow G_{\mathsf{col}} \\
\begin{pmatrix} s'_0 & s'_1 & s'_2 & s'_3 \\ s'_4 & s'_5 & s'_6 & s'_7 \\ s'_8 & s'_9 & s'_{10} & s'_{11} \\ s'_{12} & s'_{13} & s'_{14} & s'_{15} \end{pmatrix} & \xrightarrow{\lll 1} & \begin{pmatrix} s'_1 & s'_2 & s'_3 & s'_0 \\ s'_5 & s'_6 & s'_7 & s'_4 \\ s'_9 & s'_{10} & s'_{11} & s'_8 \\ s'_{13} & s'_{14} & s'_{15} & s'_{12} \end{pmatrix}
\end{array}$$

Rotation commutes with column layers...

Sketch of proof

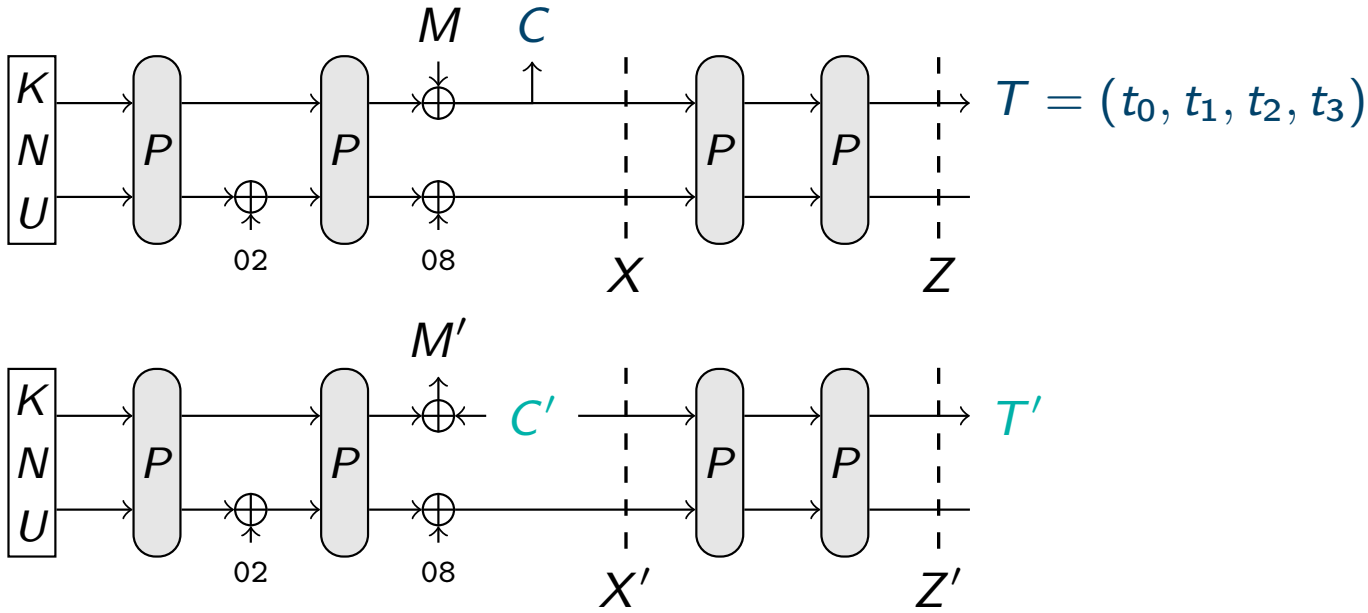


... and with diagonal layers (and therefore with P)

A ciphertext-only forgery attack

(N, C, T): Known Ciphertext

(N, C', T'): Forgery Attempt



- Forgery strategy:
  - If $X = X' \lll 2$
  - Then $Z' = P^2(X') = P^2(X \lll 2) = P^2(X) \lll 2 = Z \lll 2$
  - And $T' = (t_2, t_3, t_0, t_1)$ is a valid tag

Choice of C' and success probability

State X during encryption

$$\begin{pmatrix} C_0 & C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 & C_7 \\ C_8 & C_9 & C_{10} & C_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix} \xrightarrow{\lll 2} \begin{pmatrix} & & & \\ & & & \\ & & & \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix}$$

State X' during forgery attempt

Known, Unknown

Chosen, Fixed

- Conditions: $s_{12} = s_{14}, s_{13} = s_{15}$

Choice of C' and success probability

State X during encryption

$$\begin{pmatrix} C_0 & C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 & C_7 \\ C_8 & C_9 & C_{10} & C_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix} \xrightarrow{\lll 2} \begin{pmatrix} & & C_0 & \\ & & C_4 & \\ & & C_8 & \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix}$$

State X' during forgery attempt

Known, Unknown

Chosen, Fixed

- Conditions: $s_{12} = s_{14}, s_{13} = s_{15}$

Choice of C' and success probability

State X during encryption

$$\begin{pmatrix} C_0 & C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 & C_7 \\ C_8 & C_9 & C_{10} & C_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix} \xrightarrow{\lll 2} \begin{pmatrix} & & C_0 & C_1 \\ & & C_4 & C_5 \\ & & C_8 & C_9 \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix}$$

State X' during forgery attempt

Known, Unknown

Chosen, Fixed

- Conditions: $s_{12} = s_{14}, s_{13} = s_{15}$

Choice of C' and success probability

State X during encryption

$$\begin{pmatrix} C_0 & C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 & C_7 \\ C_8 & C_9 & C_{10} & C_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix} \xrightarrow{\lll 2} \begin{pmatrix} C_2 & & C_0 & C_1 \\ C_6 & & C_4 & C_5 \\ C_{10} & & C_8 & C_9 \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix}$$

State X' during forgery attempt

Known, Unknown

Chosen, Fixed

- Conditions: $s_{12} = s_{14}, s_{13} = s_{15}$

Choice of C' and success probability

State X during encryption

$$\begin{pmatrix} C_0 & C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 & C_7 \\ C_8 & C_9 & C_{10} & C_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix} \xrightarrow{\lll 2} \begin{pmatrix} C_2 & C_3 & C_0 & C_1 \\ C_6 & C_7 & C_4 & C_5 \\ C_{10} & C_{11} & C_8 & C_9 \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix}$$

State X' during forgery attempt

Known, Unknown

Chosen, Fixed

- Conditions: $s_{12} = s_{14}, s_{13} = s_{15}$
- Probability 2^{-64}

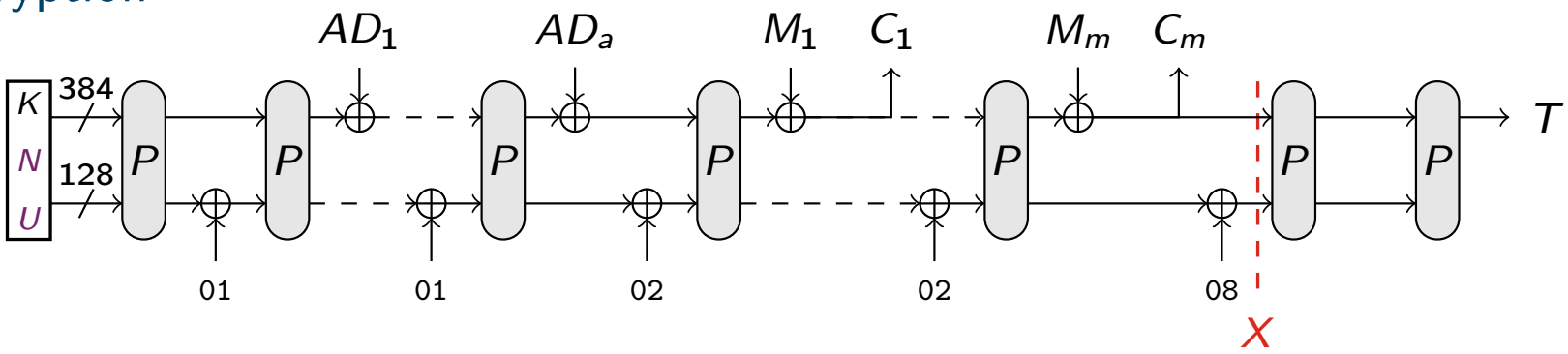
Taking account of the padding

- Padding: $M_{\text{pad}} = M||10^*1$
- Ciphertext: only $|C| = |M|$ bits of the state returned
- ℓ more conditions for $\ell \leq 64$ padding bits
- General case: $\Pr[\text{forgery}] = 2^{-64-\ell}$ for ℓ padding bits
- Best case: 2 padding bits $\Rightarrow \Pr[\text{forgery}] = 2^{-66}$
- **$\Pr[\text{forgery}] \geq 1/2$ for 2^{66} forgery attempts**

Extension of the attack

- General case: attack with any number of blocks of M and AD
- Key recovery: guess s_{12} and s_{13} and compute backwards
  - Verify the initial value of the state
  - Requires the knowledge of the plaintext
  - Known-Plaintext, or Chosen-Ciphertext attack
  - **Complexity** 2^{64}

Encryption



Other versions of NORX

- NORX v2.0, **256-bit keys**: attacks works with **probability** 2^{-130}

- NORX v1: capacity $c = 192$, forgery with probability 2^{-128}
- NORX v3.0: extra key additions thwart the attack
- NORX16: Forgeries with probability 2^{-98} for 96-bit tags
  - Up to 2^{-66} if tag length increased to 128 bits
- NORX8: Different tag extraction thwarts the attack

Discussion on birthday(-bound) attacks

Birthday attacks

Very common in cryptography, attacks relying on the birthday paradox with success probability

$$p = D^2/2^N$$

N: security parameter, D: amount of data handled with a single key

Possible mitigation

Frequent change of keys: use k keys to handle a fraction $D' = D/k$ of your data

Total success probability of an attack at most

$$p = kD'^2/2^N = D^2/(k2^N)$$

Attacks matching the birthday bound

In specific cases (as for NORX), success probability

$$p = D/2^{N/2}$$

Regardless of the frequency of key change $\rightarrow$ **Mitigation inefficient here!**

Outline of the presentation

Outline

- Why is cryptanalysis of symmetric algorithms necessary ?
- Forgery and key-recovery attacks against NORX **[CFGJR17]**
- **Two key-recovery attacks against Kravatte [CFGGJSR18]**
- Concluding remarks

Farfalle and Kravatte

Farfalle construction [BDH+]

- Parallelizable permutation-based PRF of variable input and output length
- Can be used
  - Directly as a MAC, a stream cipher, a KDF
  - Through a mode: as a AEAD, a block cipher of variable block length

Kravatte: Keccak based instantiation

- Several versions
  - **ePrint**, published on IACR ePrint in July 2017 [2016/1188]
  - **ECC**, outlined at ECC 2017 in November 2017
  - ToSC 2018, presented at FSE 2018

Security claim

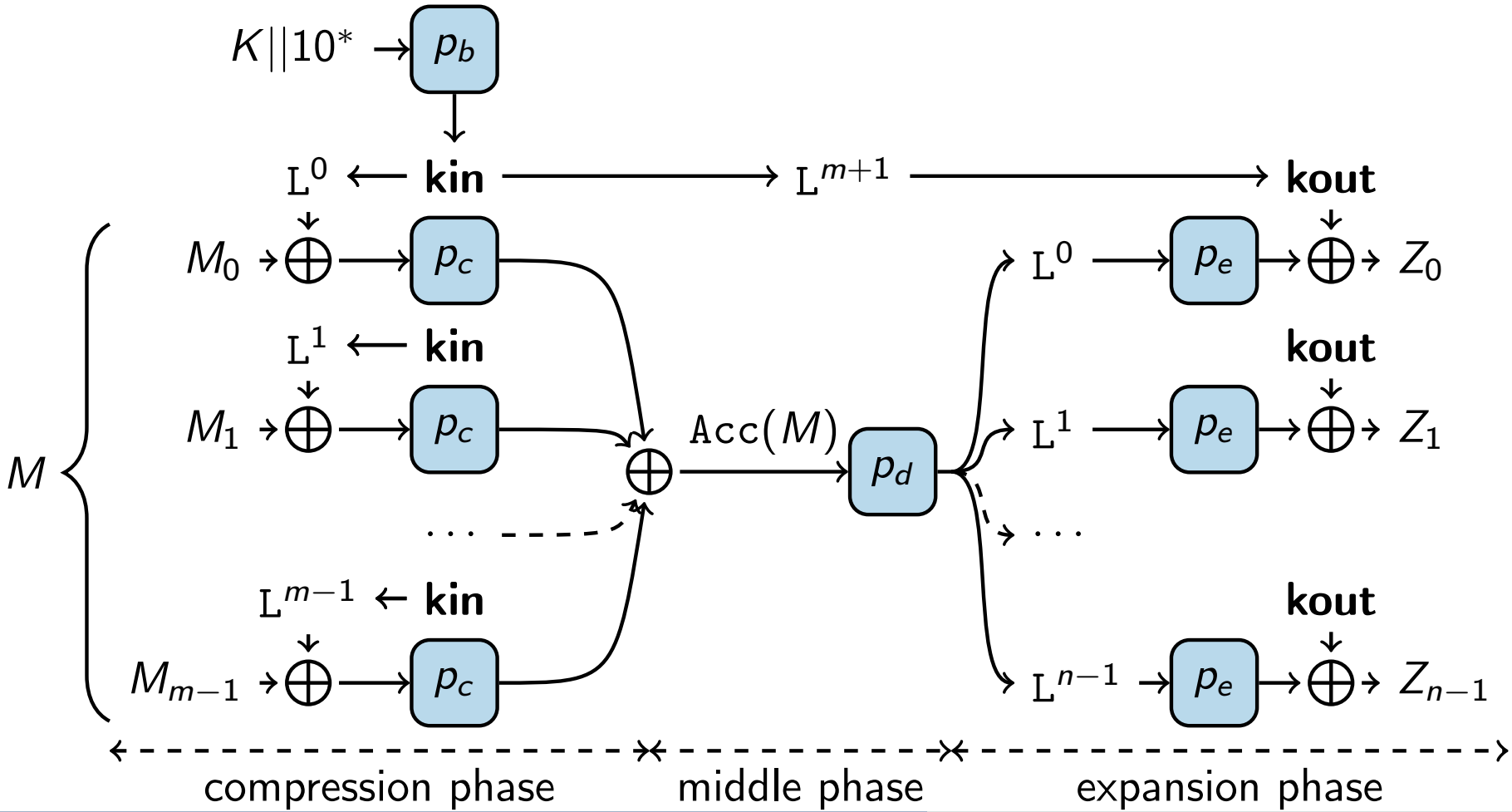
256 bits when $|\text{input} + \text{output blocks}| < 2^{137}$

Kravatte in a nutshell

Targeted Kravatte properties

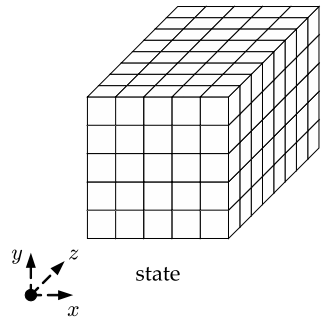
- p_x: r_x rounds of Keccak-p
- Rolling function L: linear permutation

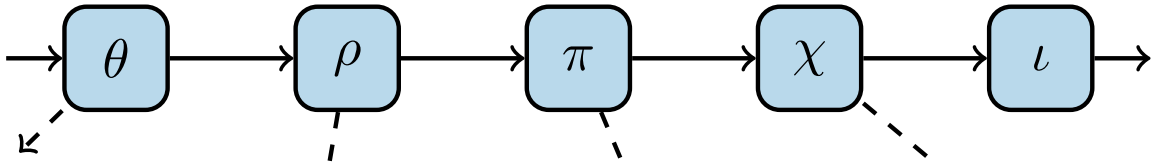
version	r_b, r_c, r_d, r_e
ePrint	6, 6, 4, 4
ECC	6, 6, 6, 6

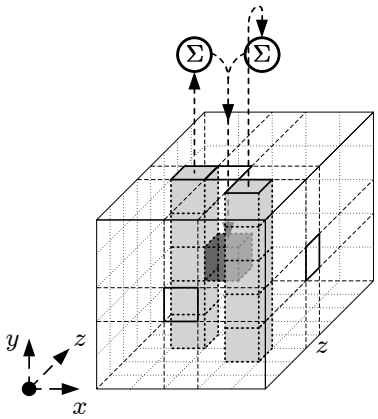


Reminder: the Keccak-p round function

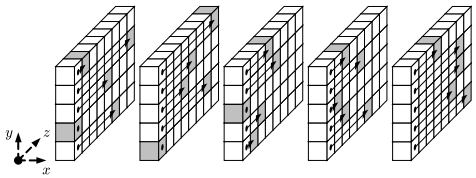
- Keccak-p state: $5 \times 5 \times 64 = 1600$ bits



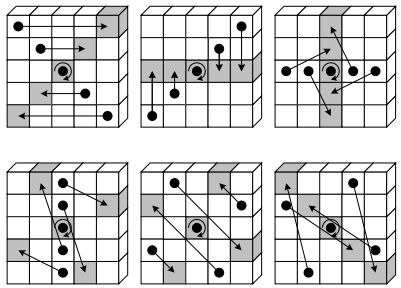




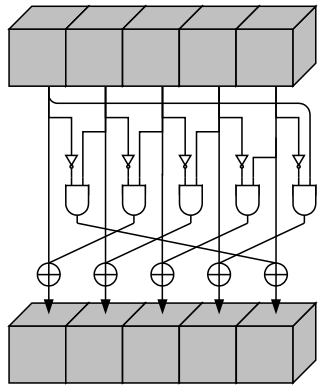
linear



linear



linear



non-linear
$\deg(\chi) = 2$
$\deg(\chi^{-1}) = 3$

source of the state, θ, ρ, π, χ figures: `https://keccak.team/figures.html`

Outline of our key-recovery attacks

One attack abusing a property of the compression phase

- Higher order differential attack **(HO)**

Two attacks on the expansion phase of Kravatte

- Meet-in-the-middle algebraic attack **(MITM)**
- Linear recurrence attack **(LR)**

Attack	version	T	D	M
HO	ePrint	2^{112}	2^{74}	2^{62}
MITM	ePrint	2^{115}	2^{28}	2^{76}
LR	**ePrint**	$\mathbf{2^{65}}$	$\mathbf{2^{51}}$	$\mathbf{2^{51}}$
LR	**ECC**	$\mathbf{2^{134}}$	$\mathbf{2^{88}}$	$\mathbf{2^{88}}$

Linearization techniques

- Solving system of (binary) linear equation: easy with gaussian elimination

$$\sum_{i=0}^{n-1} \alpha_{i,j} x_i = \beta_j, 0 \leq j < m$$

- Solving systems of polynomial equations: generally hard

$$\sum_{I \subset \{0,\ldots,n-1\}} \alpha_{I,j} \left(\prod_{i \in I} x_i \right) = \beta_j, 0 \leq j < m$$

- Sometimes possible using **linearization**

$$\sum_{I \subset \{0,\ldots,n-1\}} \alpha_{I,j} X_I = \beta_j, 0 \leq j < m$$

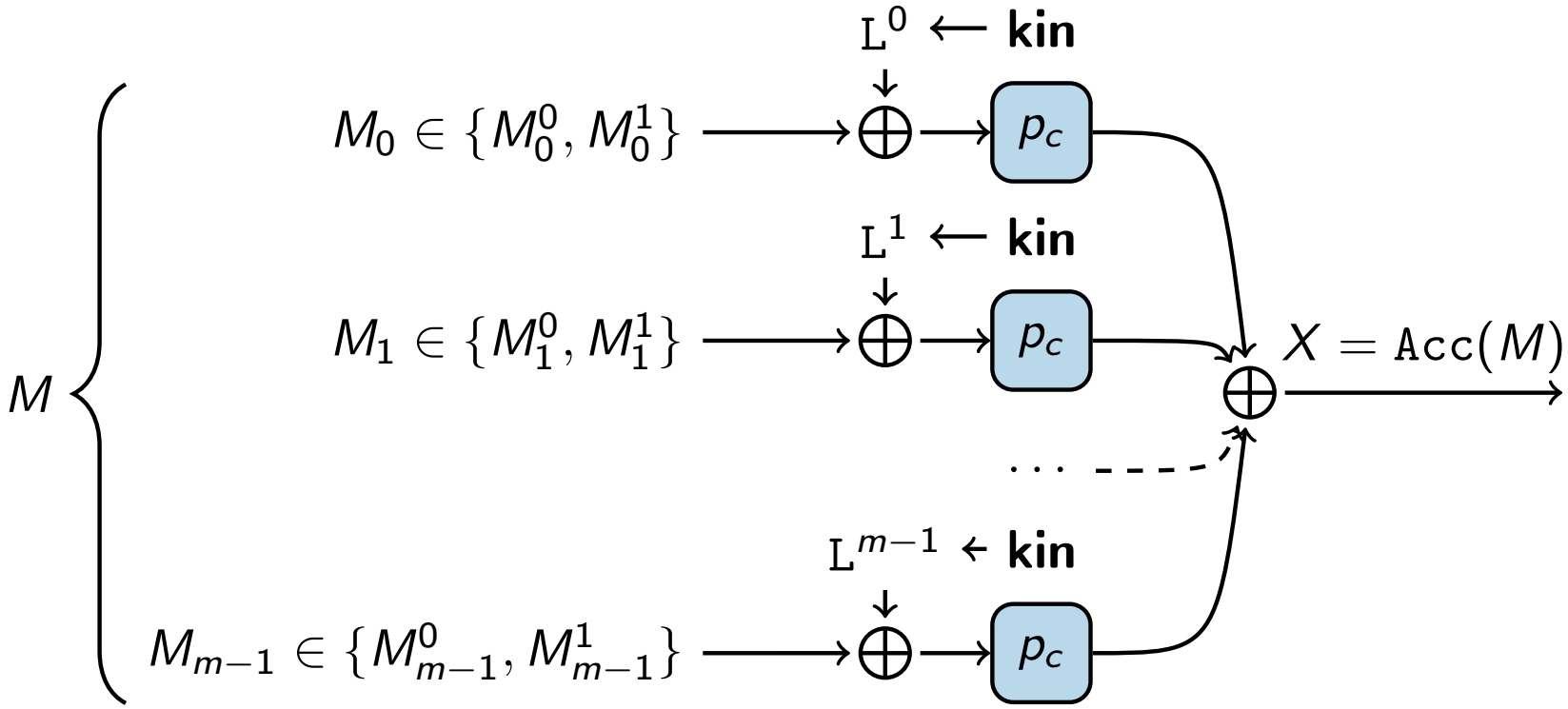
- Requires more equations, computation time might be too high

Higher order differential attack

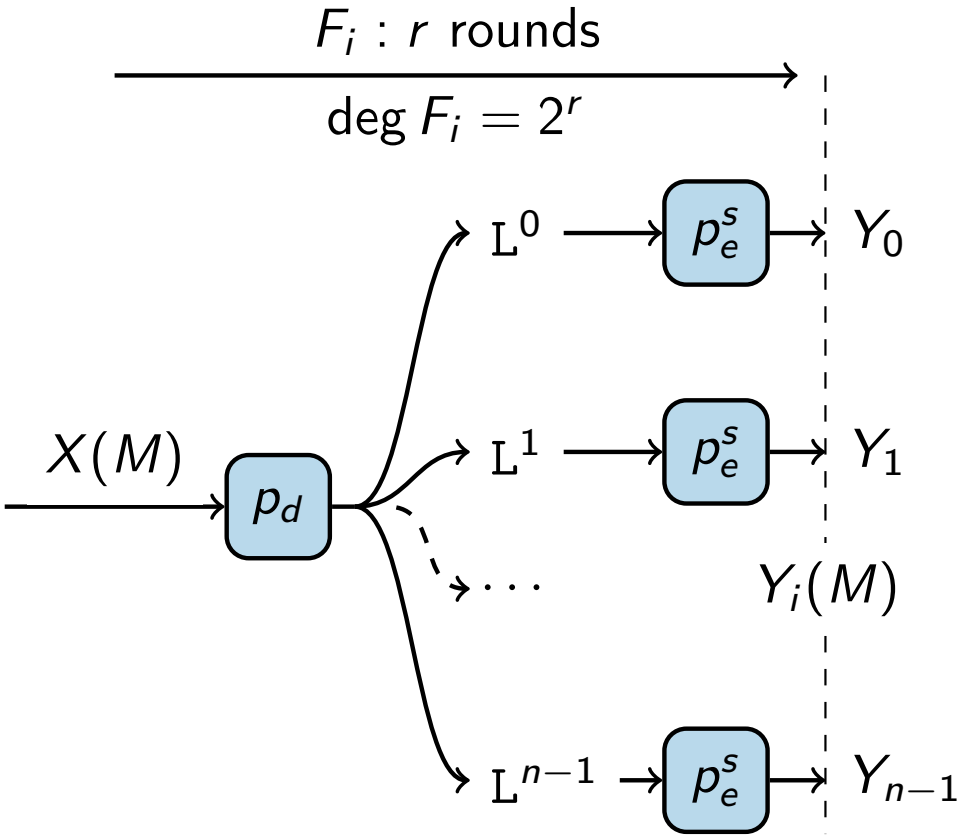
Main observation

Building an affine space of accumulator values

- Denote by $\mathcal{S}$ the following structure of 2^m m-block plaintexts
$$\mathcal{S} = \{M_0^0, M_0^1\} \times \{M_1^0, M_1^1\} \times \ldots \times \{M_{m-1}^0, M_{m-1}^1\}.$$
- $\mathtt{Acc}(\mathcal{S})$ is an affine subspace of $\{0,1\}^b$
- if $m \lll b, \dim(\mathtt{Acc}(\mathcal{S})) = m$ with overwhelming probability



HO distinguisher



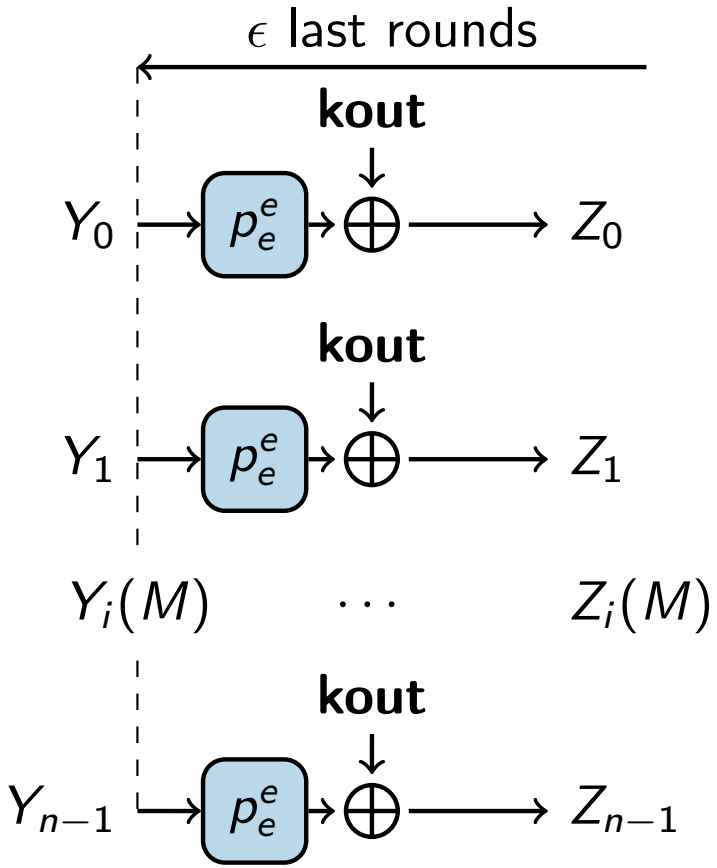
- [Lai94] Summing a function over an affine space of dim. $m \approx$ differentiating m times
- If $m > \deg F_i$, the derivative is 0
- Equation satisfied by $(Y_i(M))_{M \in \mathcal{S}}$ **independently of $\mathbf{k^{in}}$**

$$m > 2^r \Rightarrow \sum_{X \in \mathtt{Acc}(\mathcal{S})} F_i(X) = \sum_{M \in \mathcal{S}} Y_i(M) = 0$$

HO attacks

Last ϵ-round attacks

- Express $Y_i(M)$ as a function of **kout**, and $Z_i(M)$
- For one structure, combine using the HO distinguisher to get equations in **kout**
- Consider outputs long enough to collect enough equations to solve for **kout**



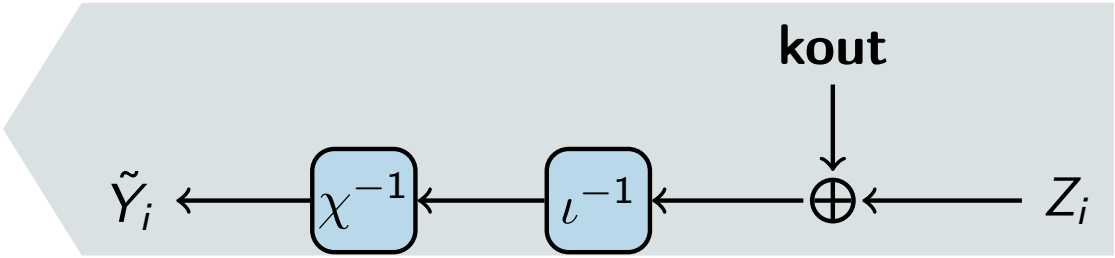
$$\sum_{M \in \mathcal{S}} \texttt{Keccak-}p^{-\epsilon}(\textbf{kout} \oplus Z_i(M)) = 0$$

HO attack with one final round ($\epsilon = 1$)

Attacking Kravatte-ePrint by local exaustive search

- F has $4 + 4 - 1 = 7$ Keccak-p rounds, $\deg F = 128$
- Using a 129-block structure, we can set up an HO distinguisher
- Note: the linear part of the last round can be ignored
- The system can be solved row-by-row, by exhaustive search
- Each block of equation provides a 5-bit condition on each row of **kout**
- With $n = 2$, most **kout** rows are determined

$$T = D = 2^{129}(129 + 2) \approx 2^{136}$$



Experimental verification

- Attack tested on a round reduced version of Kravatte

HO attack with two final rounds ($\epsilon = 2$)

Attacking Kravatte-ePrint by linearization

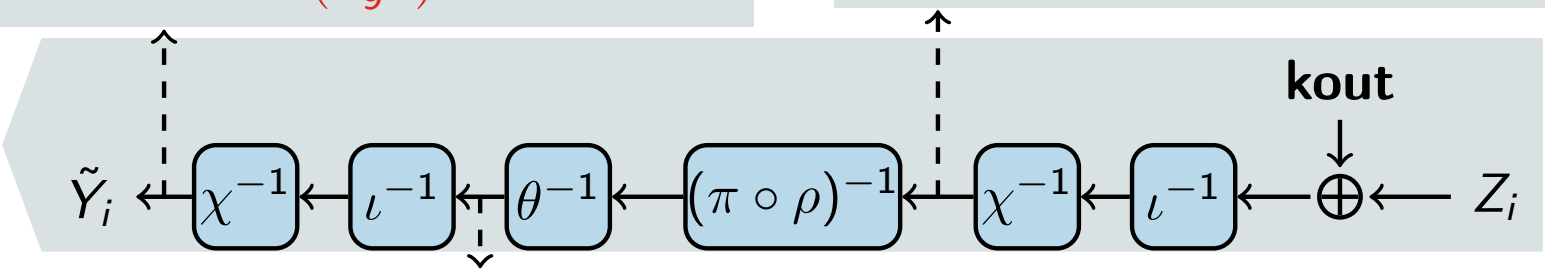
- F has $4 + 4 - 2 = 6$ Keccak-p rounds, $\deg F = 64$
- Using a 65-block structure, we can set up an HO distinguisher

$$\sum_{M \in \mathcal{S}} \text{Keccak-}p^{-2}(\mathbf{kout} \oplus Z_i(M)) = 0$$

- **Linearization** by considering every monomial in **kout** as a fresh variable

III. By combination through a degree 3 function, every bit is a LC of $N_2 = \binom{N_1}{1} + \binom{N_1}{2} + \binom{N_1}{3} \approx 2^{36.5}$ monomials $\lll \binom{1600}{9} \approx 2^{77}$

I. For each row, only 20 monomials of degree ≤ 3 in 5 **kout** variables can be formed by χ^{-1}
Total number of **kout** monomials: $N_1 = 320 \times 20 = 6400 \approx 2^{13}$



II. Linear diffusion breaks locality: every bit is a LC of up to N_1 monomials

HO attack with two final rounds ($\epsilon = 2$)

Attacking Kravatte-ePrint by linearization

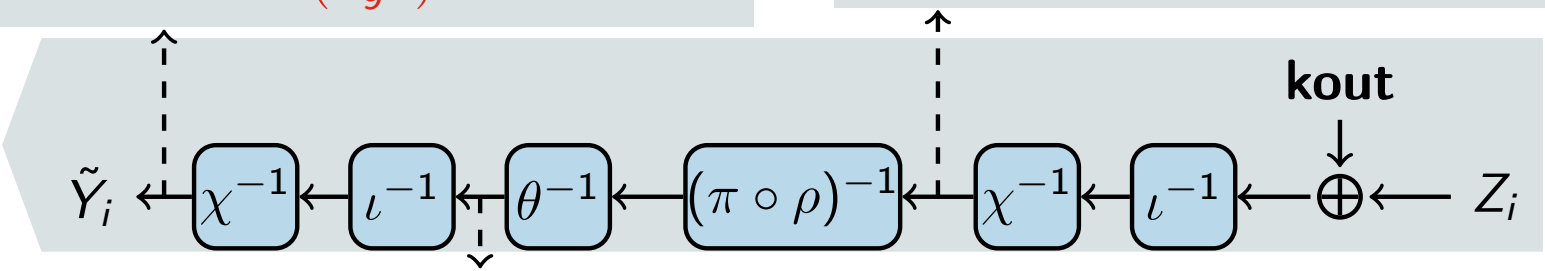
- F has $4 + 4 - 2 = 6$ Keccak-p rounds, $\deg F = 64$
- Using a 65-block structure, we can set up an HO distinguisher

$$\sum_{M \in \mathcal{S}} \text{Keccak-}p^{-2}(\mathbf{kout} \oplus Z_i(M)) = 0$$

- **Linearization** by considering every monomial in **kout** as a fresh variable
- Complexity: $T = 2^{138}, D = 2^{90}$

III. By combination through a degree 3 function, every bit is a LC of $N_2 = \binom{N_1}{1} + \binom{N_1}{2} + \binom{N_1}{3} \approx 2^{36.5}$ monomials $\lll \binom{1600}{9} \approx 2^{77}$

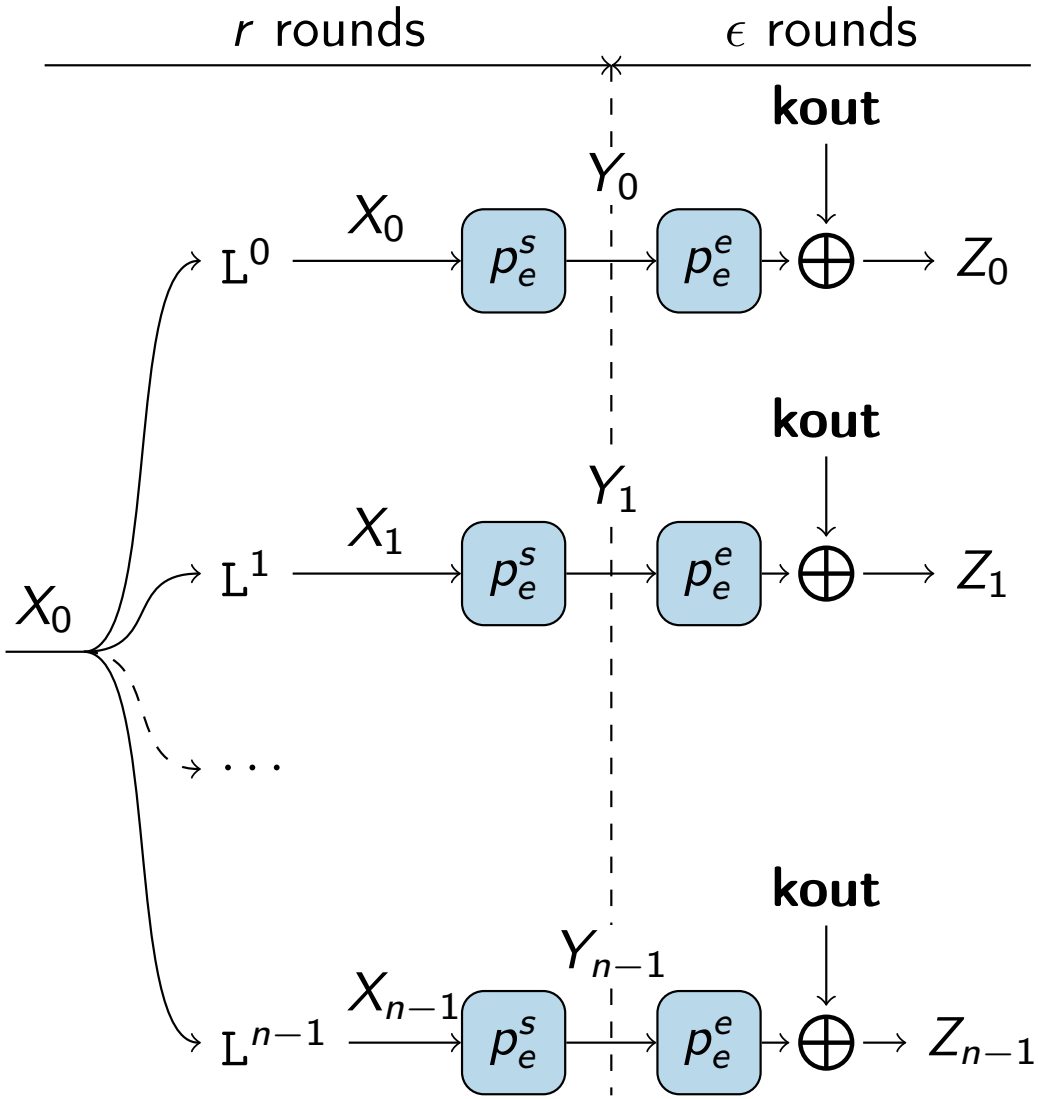
I. For each row, only 20 monomials of degree ≤ 3 in 5 **kout** variables can be formed by χ^{-1}
Total number of **kout** monomials: $N_1 = 320 \times 20 = 6400 \approx 2^{13}$



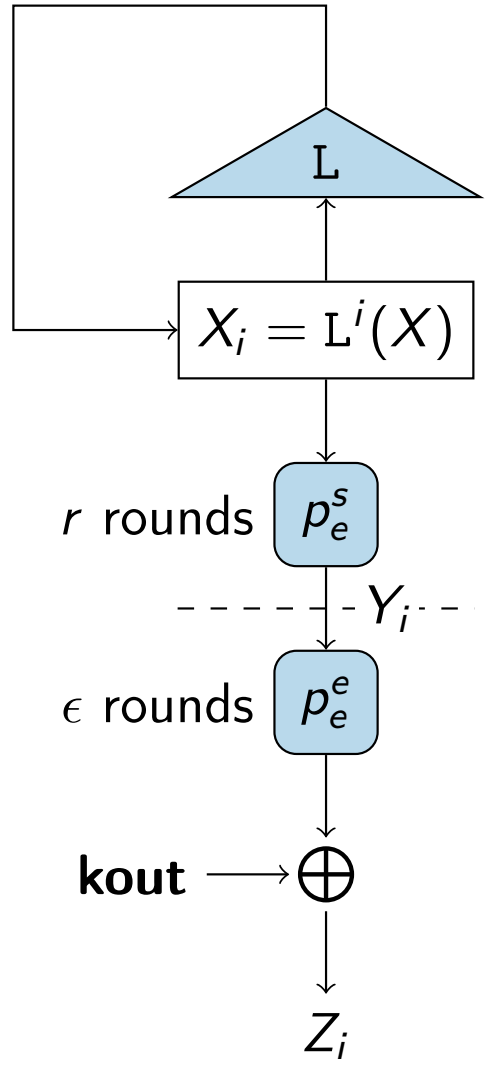
II. Linear diffusion breaks locality: every bit is a LC of up to N_1 monomials

Expansion phase attacks

Expansion phase seen as a stream cipher



≡



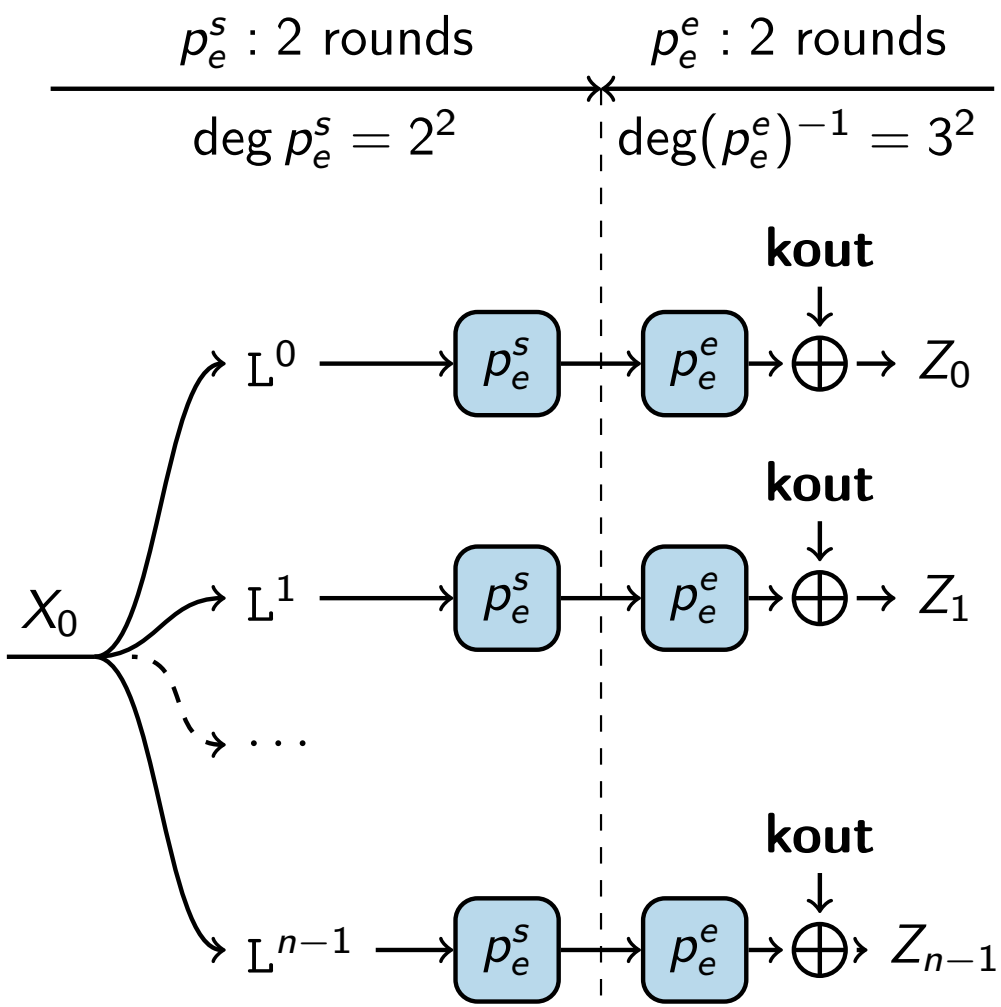
MITM attack on Kravatte-ePrint

Linearization

- **Unknowns**: the initial rolling state X_0 and **kout**
- Form equations by equating the expressions of Y_i as a function of X_0 and as a function of **kout**
- Collect equations and solve

Complexity

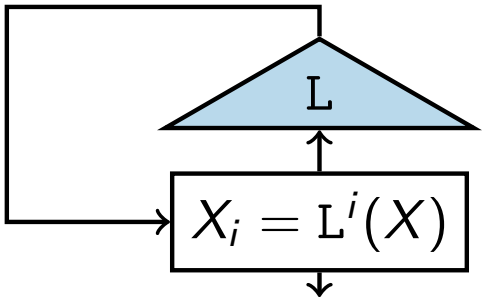
- # of monomials $N \approx 2^{38}$
- $D = N/1600 \approx 2^{28}$
- $T \approx T_{\text{solve}} = N^3 \approx 2^{115}$



$$\texttt{Keccak-}p^2(\mathrm{L}^i(X_0)) = \texttt{Keccak-}p^{-2}(\textbf{kout} + Z_i)$$

Linear recurrence distinguisher

- Use filtered LFSR cryptanalysis techniques from [Key76, RH07, RGH07]



Linear recurrence of the rolling state

- The rolling state is updated linearly

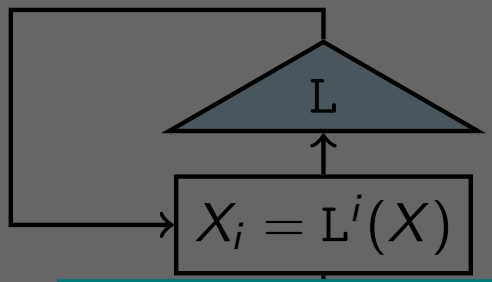
$$X_{i+1} = LX_i$$

- It is a linear recurrence sequence

$$\begin{aligned}(P.X)_i &= \sum_j a_j X_{i+j} \\ &= \sum_j a_j L^{i+j} X_0 \\ &= (L^i P(L)) X_0 \\ &= 0 \text{ if } P(L) = 0\end{aligned}$$

Linear recurrence distinguisher

- Use filtered LFSR cryptanalysis techniques from [Key76, RH07, RGH07]



Linear recurrence of the rolling state

Reminder: Linear recurrence sequence

- Consider a polynomial $P(x) = \sum_j a_j x^j$, and a sequence $u = (u_i)$
- $(P.u)$ is a sequence obtained by the action of P on u

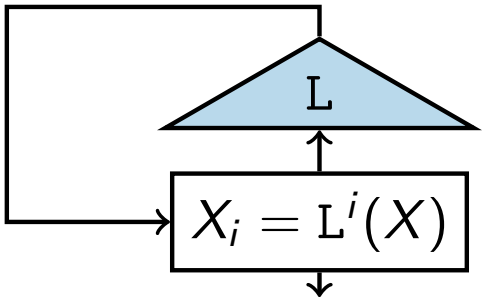
$$(P.u)_i = \sum_j a_j u_{i+j}.$$

- P annihilates u if $P.u = 0$, u is a linear recurrence sequence

$= 0$ if $P(L) = 0$

Linear recurrence distinguisher

- Use filtered LFSR cryptanalysis techniques from [Key76, RH07, RGH07]



Linear recurrence of the rolling state

- The rolling state is updated linearly

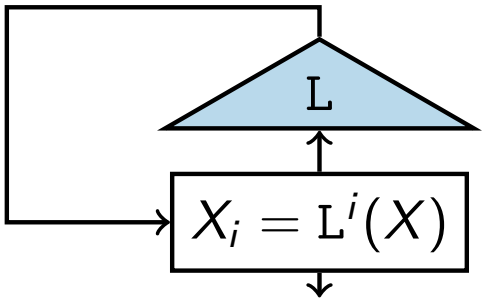
$$X_{i+1} = LX_i$$

- It is a linear recurrence sequence

$$\begin{aligned}(P.X)_i &= \sum_j a_j X_{i+j} \\ &= \sum_j a_j L^{i+j} X_0 \\ &= (L^i P(L)) X_0 \\ &= 0 \text{ if } P(L) = 0\end{aligned}$$

Linear recurrence distinguisher

- Use filtered LFSR cryptanalysis techniques from [Key76, RH07, RGH07]

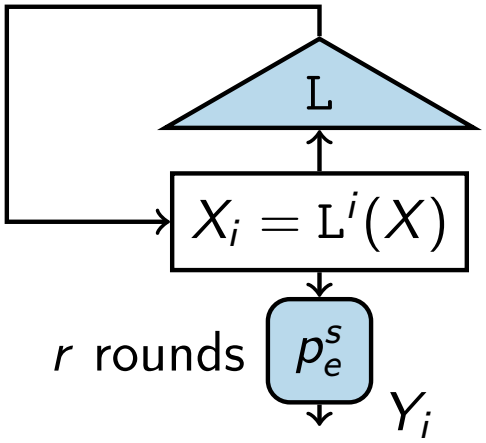


Linear recurrence of the rolling state

- The rolling state is updated linearly
$$X_{i+1} = LX_i$$
- It is a linear recurrence sequence, **as is any LC w of its components**
$$\begin{aligned}(P.w^T X)_i &= \sum_j a_j w^T X_{i+j} \\ &= w^T \sum_j a_j L^{i+j} X_0 \\ &= \dots \\ &= 0 \text{ if } P(L) = 0\end{aligned}$$

Linear recurrence distinguisher

- Use filtered LFSR cryptanalysis techniques from [Key76, RH07, RGH07]



Case of the filtered linear register

- The **monomial state** $X^{\leq d}$: vector of monomials of deg $\leq d = 2^r$
- The monomial state is updated linearly

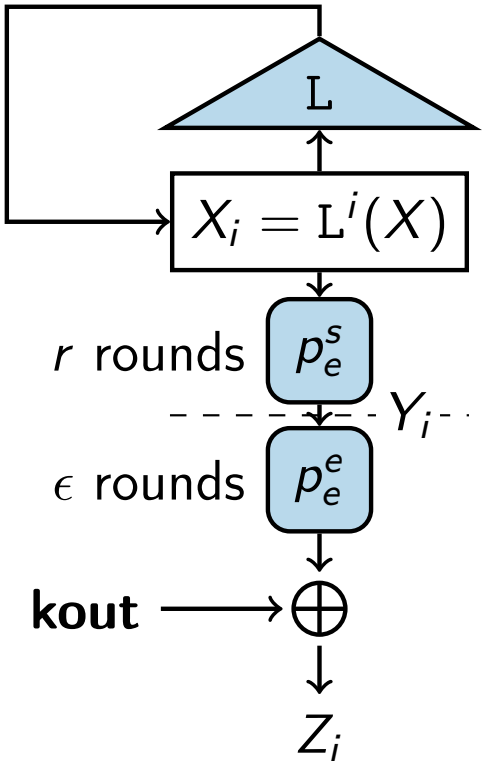
$$X_{i+1}^{\leq d} = L_{\leq d} X_i^{\leq d}$$

- It is a linear recurrence sequence, as is any LC w of its components, **thus** Y

$$\begin{aligned} (P.Y)_i &= \sum_j a_j w_Y^T X_{i+j}^{\leq d} \\ &= \dots \\ &= 0 \text{ if } P(L_{\leq d}) = 0 \end{aligned}$$

Linear recurrence distinguisher

- Use filtered LFSR cryptanalysis techniques from [Key76, RH07, RGH07]



Case of the filtered linear register

- The **monomial state** $X^{\leq d}$: vector of monomials of deg $\leq d = 2^r$
- The monomial state is updated linearly

$$X_{i+1}^{\leq d} = L_{\leq d} X_i^{\leq d}$$

- It is a linear recurrence sequence, as is any LC w of its components, **thus** Y

$$\begin{aligned}(P.Y)_i &= \sum_j a_j w_Y^T X_{i+j}^{\leq d} \\ &= \dots \\ &= 0 \text{ if } P(L_{\leq d}) = 0\end{aligned}$$

LR attacks

- Reuse last round attack framework from HO attacks, replacing HO distinguiser by LR distinguisher

$$(P^*.\texttt{Keccak-}p^{-\epsilon}(\mathbf{kout} \oplus Z))_i = 0$$

Linear recurrence polynomial for Kravatte

Determination of P^*

- Kravatte rolling function **only affects 320 bits of the state**
- Restricted update matrix M, corresponding monomial update matrix $M_{\leq d}$
- $P_{M_{\leq d}}$ cancels the sequences of all monomials involving the last plane
- The other monomials are constant and cancelled by $x + 1$
- $P^* = (x+1)P_{M_{\leq d}}$, $\deg P^*$: # monomials of deg $\leq d$ in 320 variables

Computation of P^*

- Considering $\alpha = x \mod P_M \in GF(2^{320})$,
$$P^* = \prod_{k:HW(k)\leq d} (X + \alpha^k)$$
- Can be computed in time T_P quasilinear in $\deg P^*$, using fast polynom multiplication [Sch77]

r	$\deg P^*$	T_P
2	2^{28}	2^{40}
3	2^{51}	2^{65}
4	2^{88}	2^{104}

Verified for $r = 2$

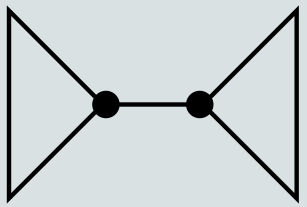
version	$r + \epsilon$	T	$D \approx \deg P^*$	$M \approx \deg P^*$
ePrint	3+1	2^{65}	2^{51}	2^{51}
ECC	4+2	2^{134}	2^{88}	2^{88}

Concluding remarks

Attack	version	T	D	M
HO	ePrint	2^{112}	2^{74}	2^{62}
MITM	ePrint	2^{115}	2^{28}	2^{76}
LR	**ePrint**	$\mathbf{2^{65}}$	$\mathbf{2^{51}}$	$\mathbf{2^{51}}$
LR	**ECC**	$\mathbf{2^{134}}$	$\mathbf{2^{88}}$	$\mathbf{2^{88}}$

Properties leveraged by the attacks

- Low algebraic degree of χ and χ^{-1}
- Ability to bypass a part of the construction to focus on a reduced number of rounds
- Special points in the `Farfalle` construction
  - The convergence point [HO]
  - The divergence point [MITM, LR]



Tweaked version of Kravatte (FSE 2018)

- Resisting HO: increase the number of rounds to 6 (versus ePrint version)
- Resisting LR: make the rolling function in expansion phase **non-linear**

Outline of the presentation

Outline

- Why is cryptanalysis of symmetric algorithms necessary ?
- Forgery and key-recovery attacks against NORX **[CFGJR17]**
- Two key-recovery attacks against Kravatte **[CFGGJSR18]**
- **Concluding remarks**

Designing cryptographic algorithms

Reducing a too large security margin

- Block ciphers: reducing number of rounds might be OK
- Obvious option considered by cryptanalysts
- Modifying other parameters: **doubtful**

Complex constructions with non ideal primitives

- Lose the benefit of an eventual security proof
- High risk of early broken versions (AEZ, Kravatte)
- **Require a large effort of cryptanalysis to obtain confidence**

Thank you for your attention !