Cryptanalysis of Round-Reduced KECCAK using Non-Linear Structures

Mahesh Sreekumar Rajasree
Center for Cybersecurity, Indian Institute of Technology Kanpur

INDOCRYPT 2019, Hyderabad

Outline

- **Cryptographic hash functions** are hash functions which are resistant to preimage, collision attacks and other attacks.

- **Cryptographic hash functions** are hash functions which are resistant to preimage, collision attacks and other attacks.
- Practical applications include message integrity checks, digital signatures, authentication, etc.

- **Cryptographic hash functions** are hash functions which are resistant to preimage, collision attacks and other attacks.
- Practical applications include message integrity checks, digital signatures, authentication, etc.
- **SHA-3 (Secure Hash Algorithm 3)** is the latest member of the Secure Hash Algorithm family of standards, released by NIST which is based on **KECCAK**.

Let H be a cryptographic hash function.

Attacks

Let H be a cryptographic hash function.

- **Preimage attack:** Given $H(m)$

Attacks

Let H be a cryptographic hash function.

- **Preimage attack:** Given $H(m)$, find any m' such that $H(m') = H(m)$.

Attacks

Let H be a cryptographic hash function.

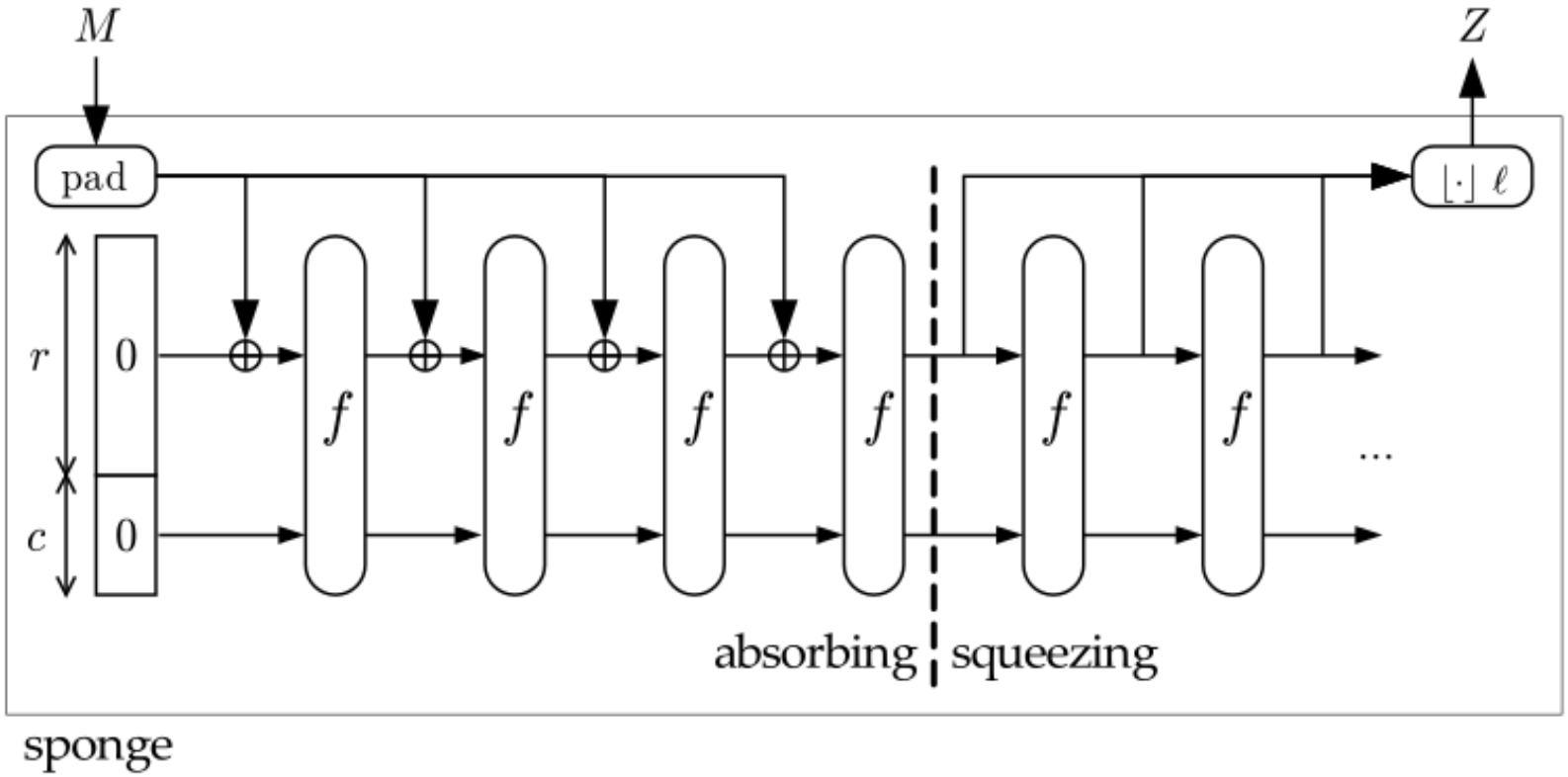
- **Preimage attack:** Given $H(m)$, find any m' such that $H(m') = H(m)$.
- **Collision attack:** Find any $m \neq m'$

Attacks

Let H be a cryptographic hash function.

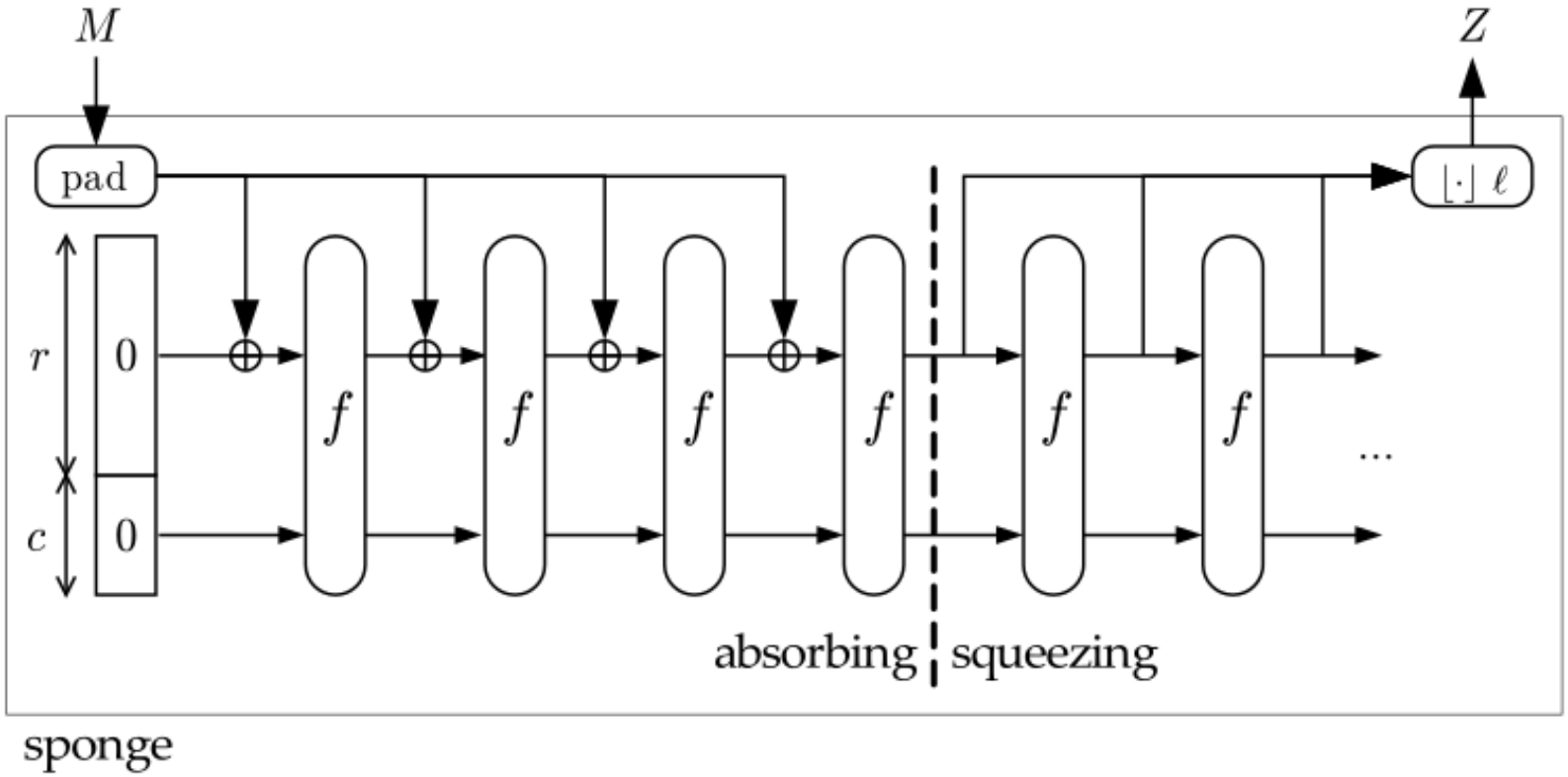
- **Preimage attack:** Given $H(m)$, find any m' such that $H(m') = H(m)$.
- **Collision attack:** Find any $m \neq m'$, such that $H(m) = H(m')$.

Sponge Construction



Source: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

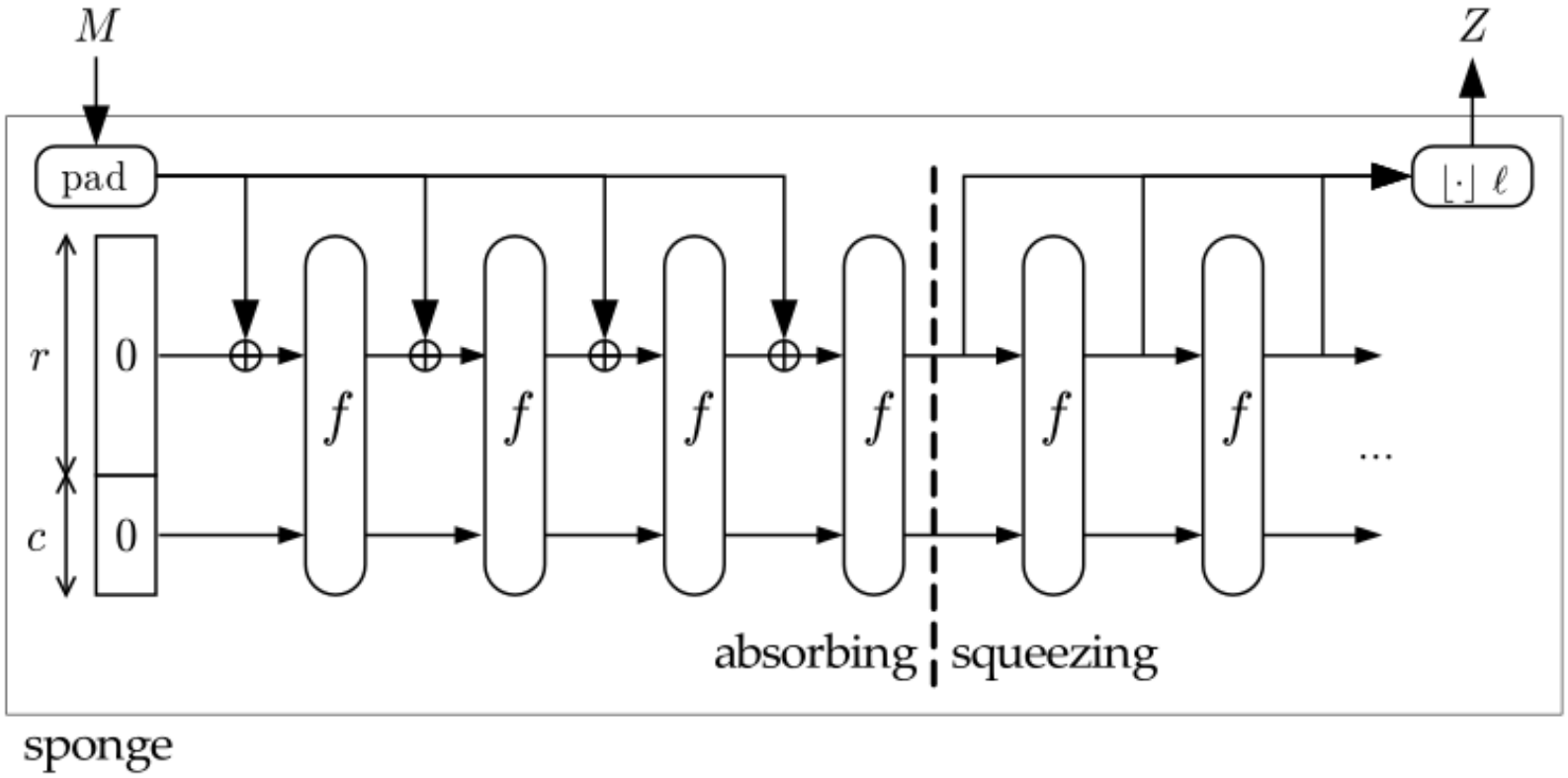
Sponge Construction



Source: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

pad: padding function (10*1)

Sponge Construction



Source: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

pad: padding function (10*1)

f: KECCAK-f permutation

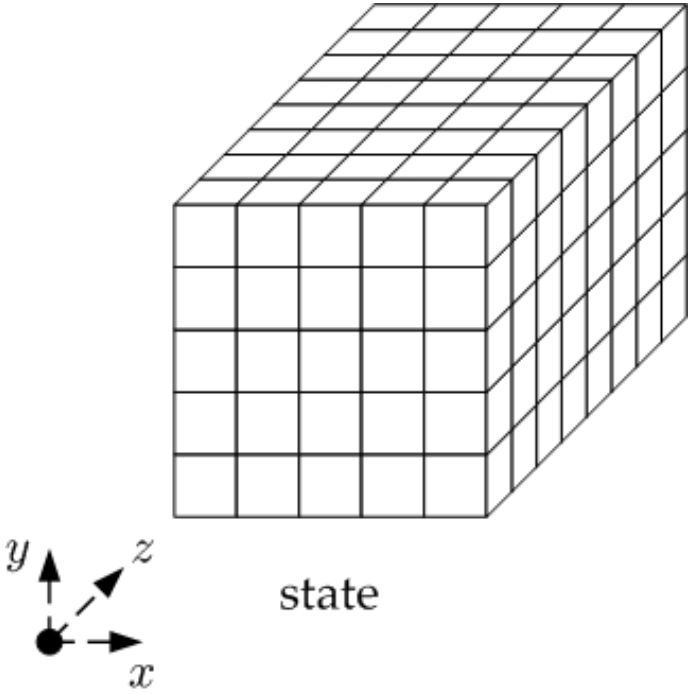


Figure: State

Source: https://keccak.team/figures.html

KECCAK-p permutation

- **Block size:** $5 \times 5 \times 64 = 1600$.

KECCAK-p permutation

- **Block size:** $5 \times 5 \times 64 = 1600$.
- $c = 2\ell, r = 1600 - c$ where $\ell \in \{224, 256, 384, 512\}$.

KECCAK-p permutation

- **Block size:** $5 \times 5 \times 64 = 1600$.
- $c = 2\ell, r = 1600 - c$ where $\ell \in \{224, 256, 384, 512\}$.
- **Number of rounds:** In each round there are five Step mappings $(\theta, \rho, \pi, \chi, \iota)$.

Description of θ

$S'[x,y,z] = S[x,y,z] \oplus P[(x+1)\ mod\ 5][(z-1)\ mod\ 64] \oplus P[(x-1)\ mod\ 5][z]$
where $P[x][z] = \bigoplus_{i=0}^{4} S[x,i,z]$

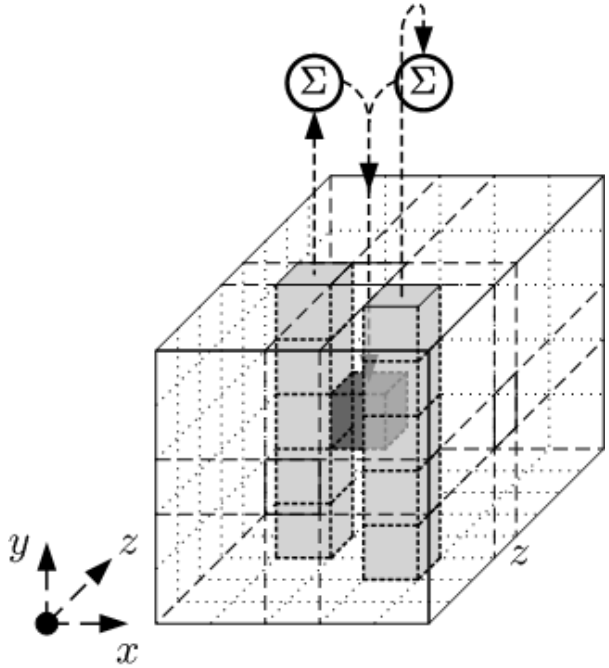


Figure: θ

Source: https://keccak.team/figures.html

Description of ρ

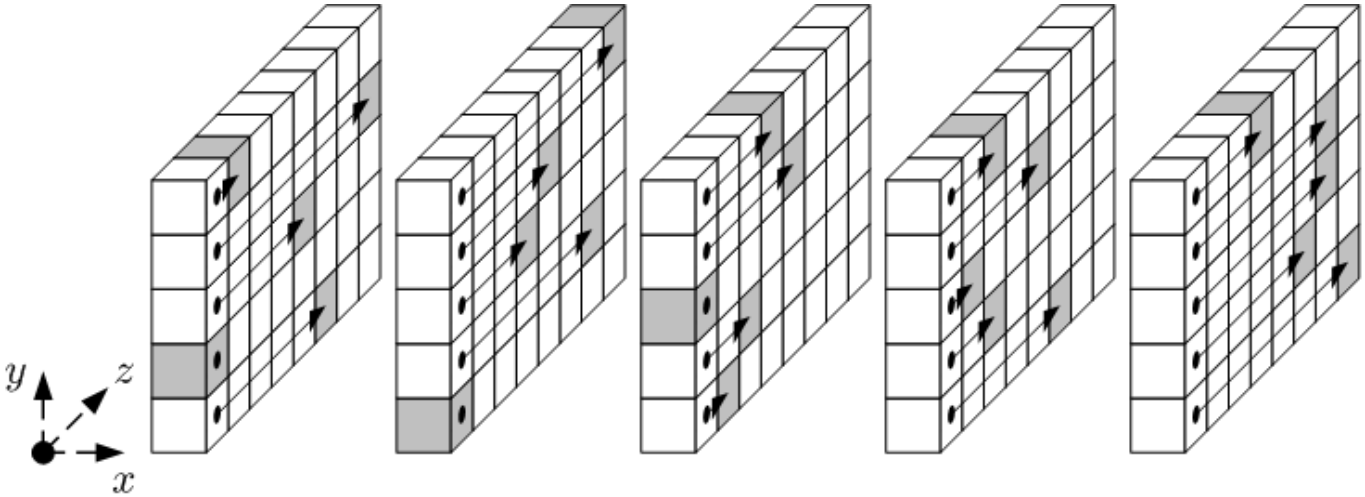


Figure: ρ

Source: https://keccak.team/figures.html

Description of π

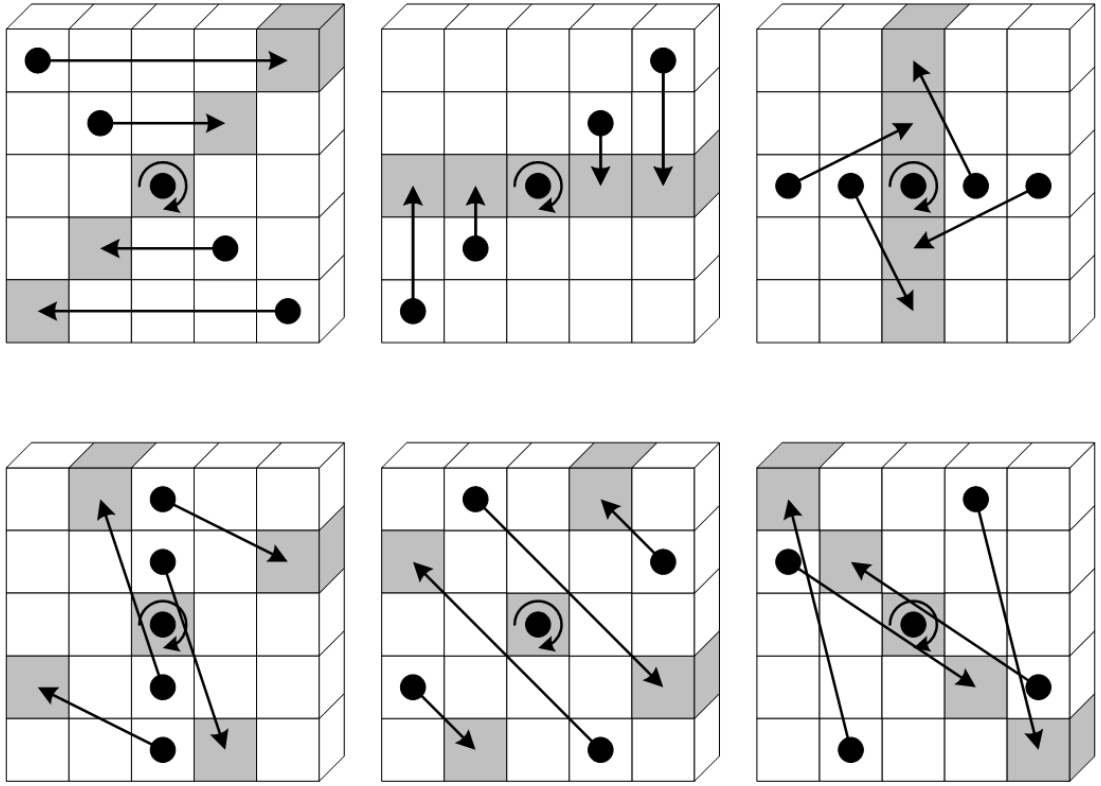


Figure: π

Source: https://keccak.team/figures.html

Description of χ and ι

- χ: Only non-linear function

Description of χ and ι

- χ: Only non-linear function

$$S'[x,y,z] = S[x,y,z] \oplus ((S[(x+1)\ mod\ 5, y, z] \oplus 1) \cdot S[(x+2)\ mod\ 5, y, z])$$

Description of χ and ι

- χ: Only non-linear function

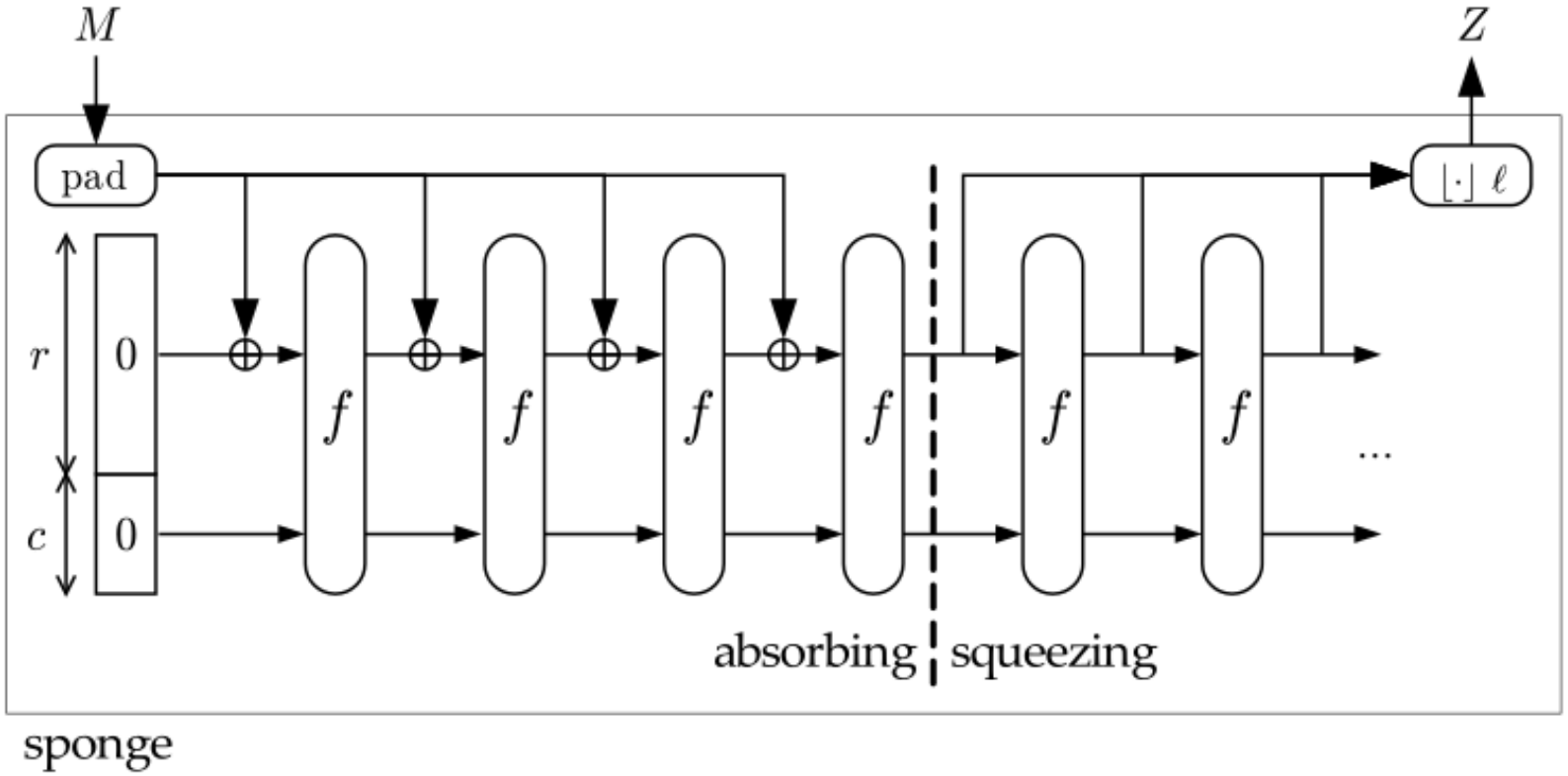
$$S'[x,y,z] = S[x,y,z] \oplus ((S[(x+1)\ mod\ 5, y, z] \oplus 1) \cdot S[(x+2)\ mod\ 5, y, z])$$

- ι:

$$S'[0,0] = S[0,0] \oplus RC_i$$

where RC_i is a constant which depends on i where i is the round number.

Recap



Source: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

Results

Rounds	Instances	Our Results	Previous Results
2	384	$\mathbf{2^{113}}$	2^{129}[Guo et al., 2016]
	512	$\mathbf{2^{321}}$	2^{384}[Guo et al., 2016]
3	384	$\mathbf{2^{321}}$	2^{322}[Guo et al., 2016]
	512	$\mathbf{2^{475}}$	2^{482}[Guo et al., 2016]
4	384	$\mathbf{2^{371}}$	2^{378}[Morawiecki et al., 2013]

Table: Summary of preimage attacks

Preimage attack

1. If all input bits are variables, then the output of KECCAK is a **non-linear polynomial**.

Preimage attack

1. If all input bits are variables, then the output of KECCAK is a **non-linear polynomial**.
2. This is due to χ function.

Preimage attack

1. If all input bits are variables, then the output of KECCAK is a **non-linear polynomial**.
2. This is due to χ function.
3. To avoid this, we will equate one of the terms in the product to some **constant**.

Preimage attack

1. If all input bits are variables, then the output of KECCAK is a **non-linear polynomial**.
2. This is due to χ function.
3. To avoid this, we will equate one of the terms in the product to some **constant**.
4. θ must also be controlled to avoid diffusion.

Preimage attack

1. If all input bits are variables, then the output of KECCAK is a **non-linear polynomial**.
2. This is due to χ function.
3. To avoid this, we will equate one of the terms in the product to some **constant**.
4. θ must also be controlled to avoid diffusion.
5. Make sure that the **number of equations** are not more than the **number of variables**.

Preimage attack on 2 rounds KECCAK-512

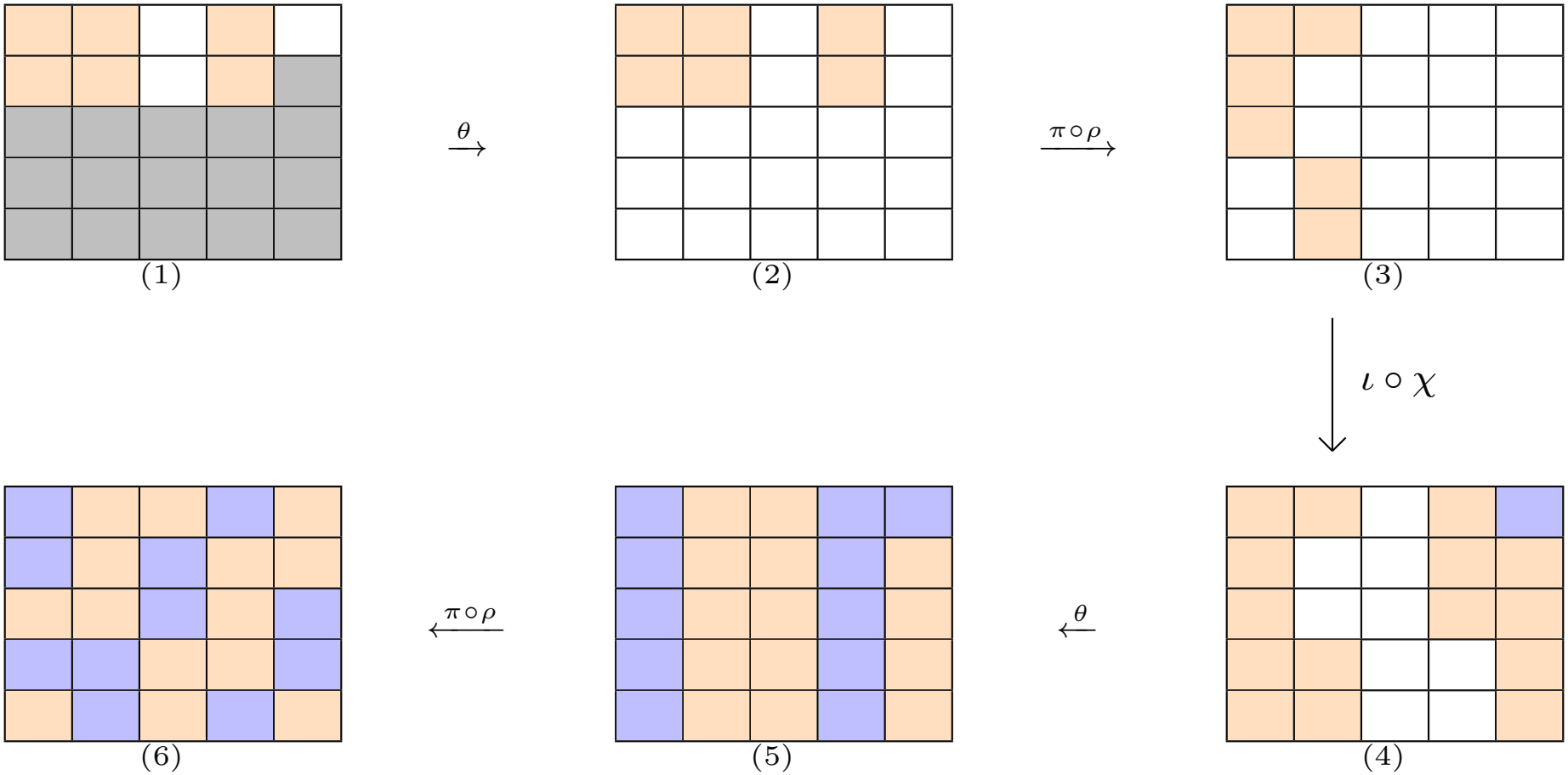


Figure: Preimage attack on 2-rounds KECCAK-512

Preimage attack on 2 rounds KECCAK-512

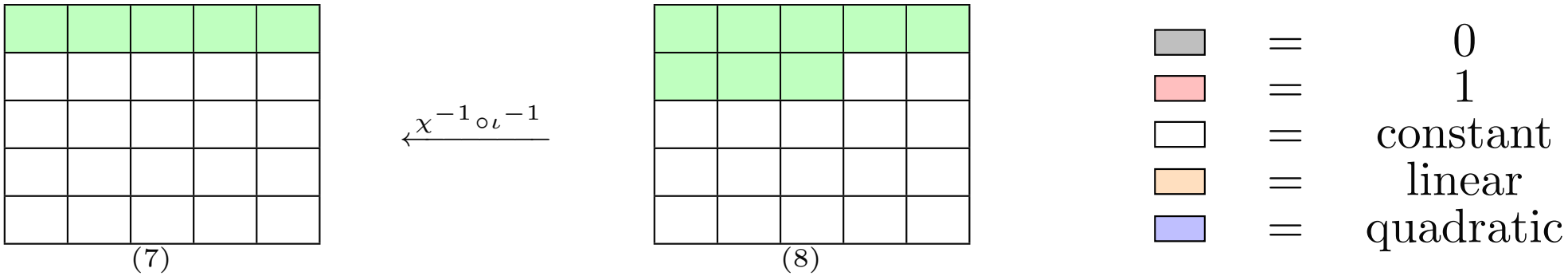


Figure: Preimage attack on 2-rounds KECCAK-512

Preimage attack on 2 rounds KECCAK-512

- Number of variables $= 6 \times 64 = 384$.

Preimage attack on 2 rounds KECCAK-512

- Number of variables $= 6 \times 64 = 384$.
- Number of equations for first $\theta = 3 \times 64 = 192$.

Preimage attack on 2 rounds KECCAK-512

- Number of variables $= 6 \times 64 = 384$.
- Number of equations for first $\theta = 3 \times 64 = 192$.
- One equation for padding.

Preimage attack on 2 rounds KECCAK-512

- Number of variables $= 6 \times 64 = 384$.
- Number of equations for first $\theta = 3 \times 64 = 192$.
- One equation for padding.
- Number of equations between message variable and hash bits $= 3 * 64 - 1 = 191$.

Preimage attack on 2 rounds KECCAK-512

- Number of variables $= 6 \times 64 = 384$.
- Number of equations for first $\theta = 3 \times 64 = 192$.
- One equation for padding.
- Number of equations between message variable and hash bits $= 3 * 64 - 1 = 191$.
- **Complexity $\mathbf{2^{512-191} = 2^{321}}$.**

Preimage attack on 3 rounds KECCAK-384

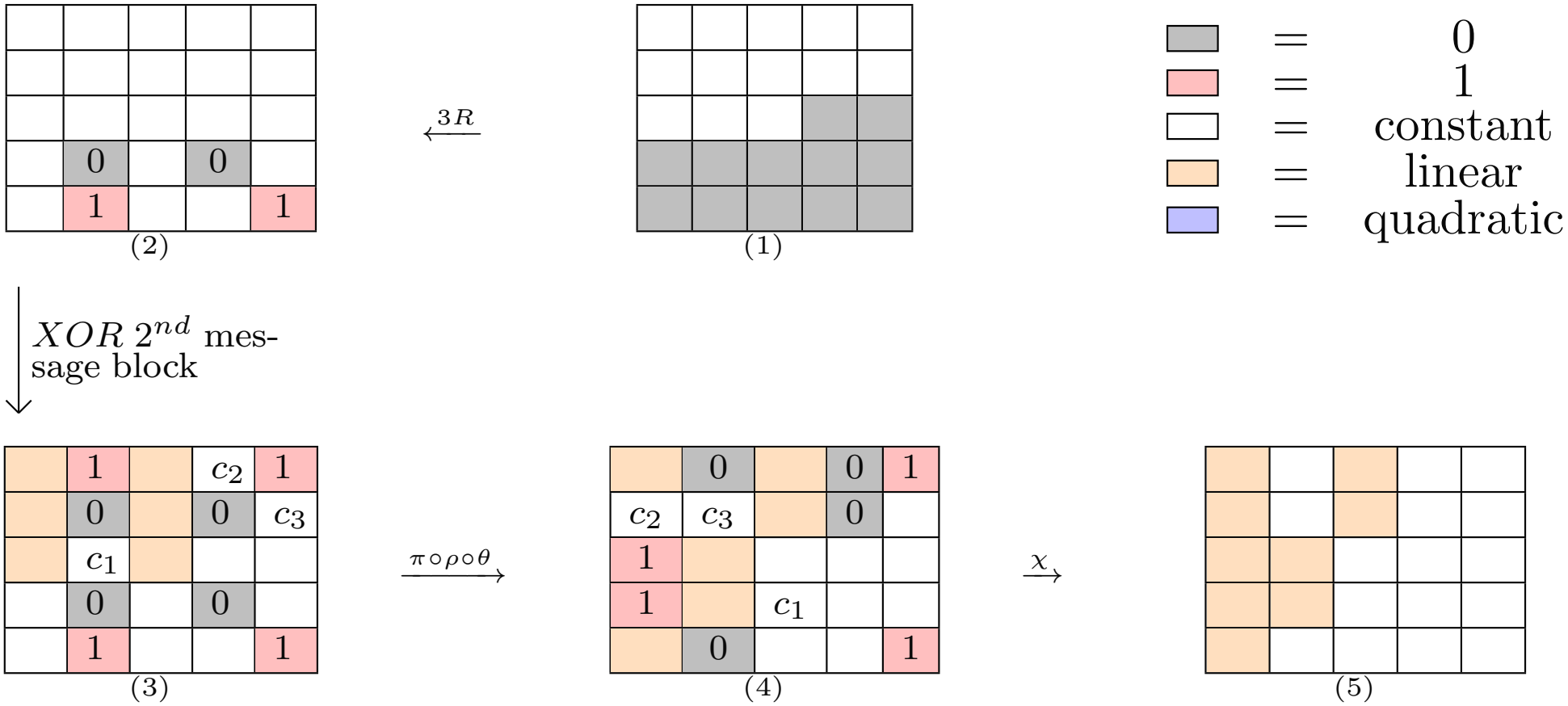


Figure: Preimage attack on 3-rounds KECCAK-384

Preimage attack on 3 rounds KECCAK-384

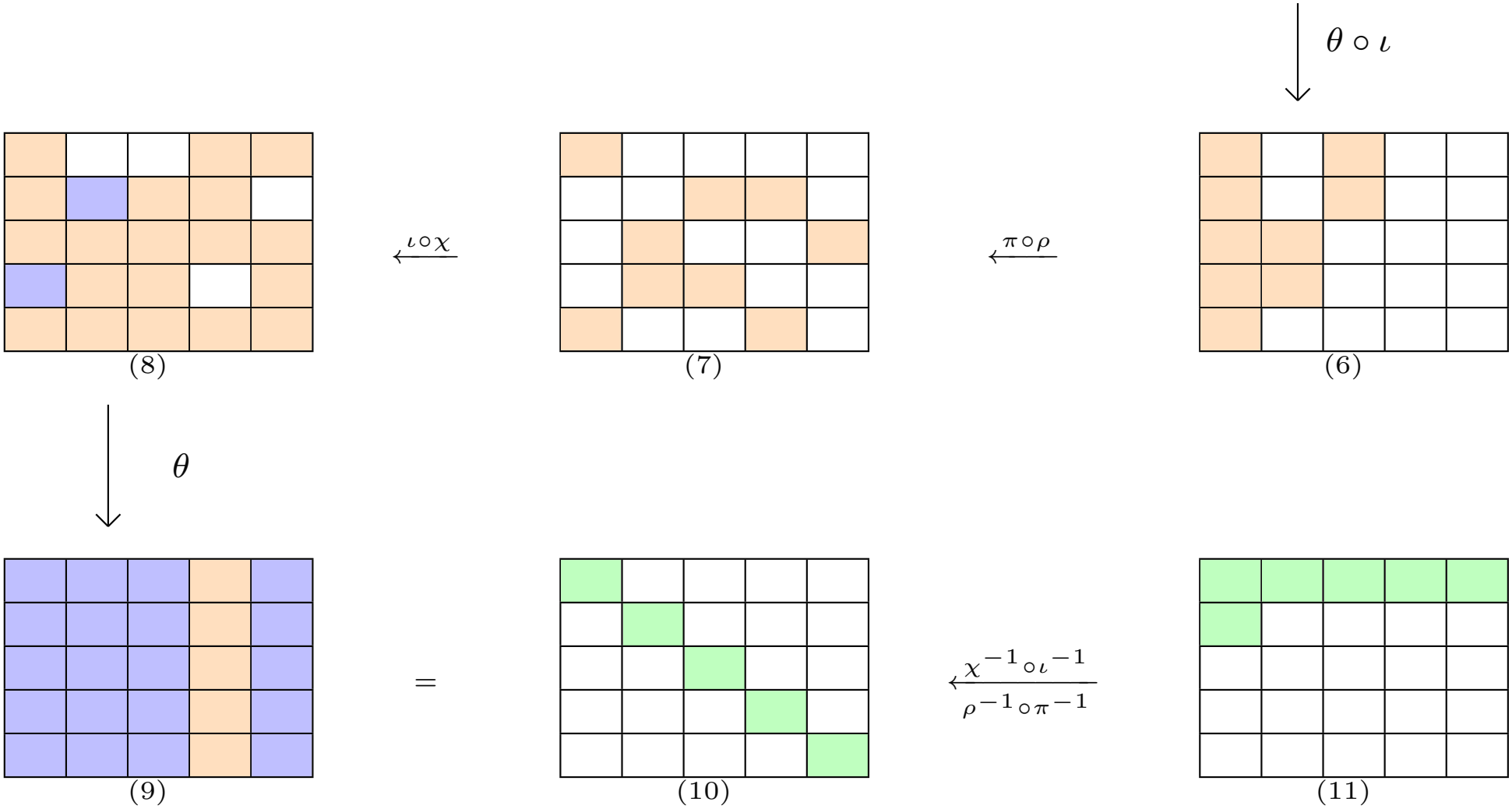


Figure: Preimage attack on 3-rounds KECCAK-384

Preimage attack on 3 rounds KECCAK-384

1. Number of variables $= 6 \times 64 = 384$.

Preimage attack on 3 rounds KECCAK-384

1. Number of variables $= 6 \times 64 = 384$.
2. Number of equations for first $\theta = 2 \times 64 = 128$.

Preimage attack on 3 rounds KECCAK-384

1. Number of variables $= 6 \times 64 = 384$.
2. Number of equations for first $\theta = 2 \times 64 = 128$.
3. Number of equations for second $\theta = 3 \times 64 = 192$.

Preimage attack on 3 rounds KECCAK-384

1. Number of variables $= 6 \times 64 = 384$.
2. Number of equations for first $\theta = 2 \times 64 = 128$.
3. Number of equations for second $\theta = 3 \times 64 = 192$.
4. One equation for padding.

Preimage attack on 3 rounds KECCAK-384

1. Number of variables $= 6 \times 64 = 384$.
2. Number of equations for first $\theta = 2 \times 64 = 128$.
3. Number of equations for second $\theta = 3 \times 64 = 192$.
4. One equation for padding.
5. Number of equations between message variables and hash bits $= 63$.

Preimage attack on 3 rounds KECCAK-384

1. Number of variables $= 6 \times 64 = 384$.
2. Number of equations for first $\theta = 2 \times 64 = 128$.
3. Number of equations for second $\theta = 3 \times 64 = 192$.
4. One equation for padding.
5. Number of equations between message variables and hash bits $= 63$.
6. **Complexity $\mathbf{2^{384-63} = 2^{321}}$.**

Conclusion

- We have presented the best theoretical preimage attack for round-reduced KECCAK.

Conclusion

- We have presented the best theoretical preimage attack for round-reduced KECCAK.
- Would be interesting to see whether non-linear structures along with other techniques can be used to find better preimage attacks for higher rounds.

Thank You

Questions?

References

Guo, J., Liu, M., and Song, L. (2016).
Linear structures: applications to cryptanalysis of round-reduced keccak.
In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 249–274. Springer.

Morawiecki, P., Pieprzyk, J., and Srebrny, M. (2013).
Rotational cryptanalysis of round-reduced keccak.
In *International Workshop on Fast Software Encryption*, pages 241–262. Springer.