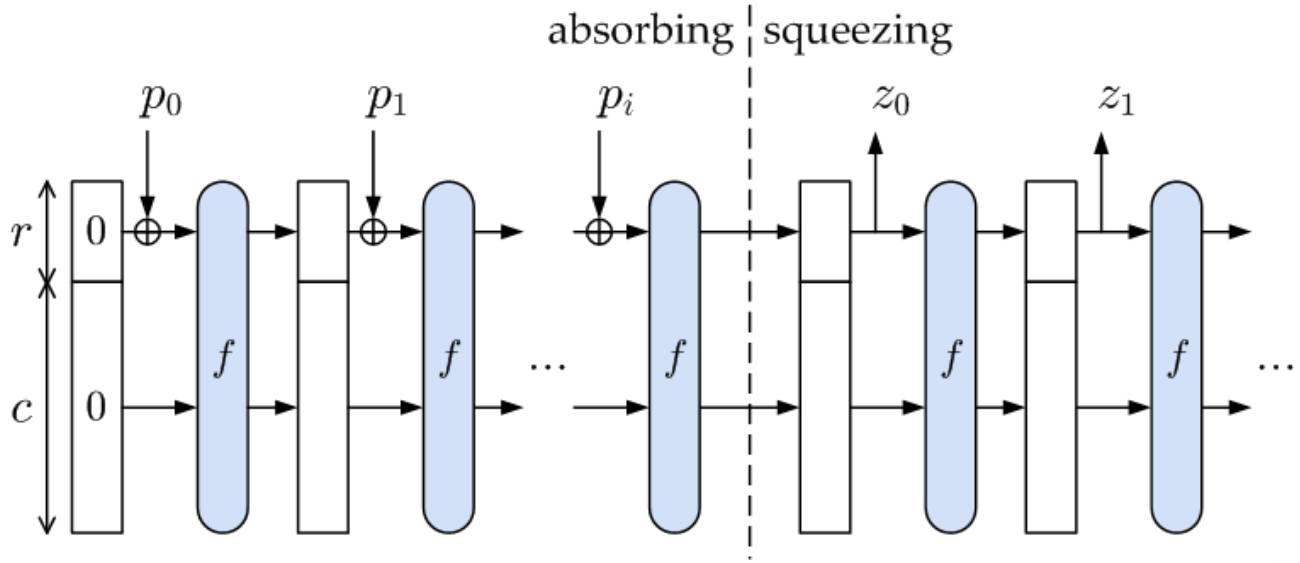
Cube Attacks and Cube-attack-like Cryptanalysis on the Round-reduced Keccak Sponge Function

Itai Dinur[1], Paweł Morawiecki[2,3], Josef Pieprzyk[4], Marian Srebrny[2,3], and Michał Straus[3]

[1]Computer Science Department, École Normale Supérieure, France
[2]Institute of Computer Science, Polish Academy of Sciences,Poland
[3]Section of Informatics, University of Commerce, Kielce, Poland
[4]Queensland University of Technology, Brisbane, Australia

Keccak [Bertoni, Daemen, Peeters and Van Assche]

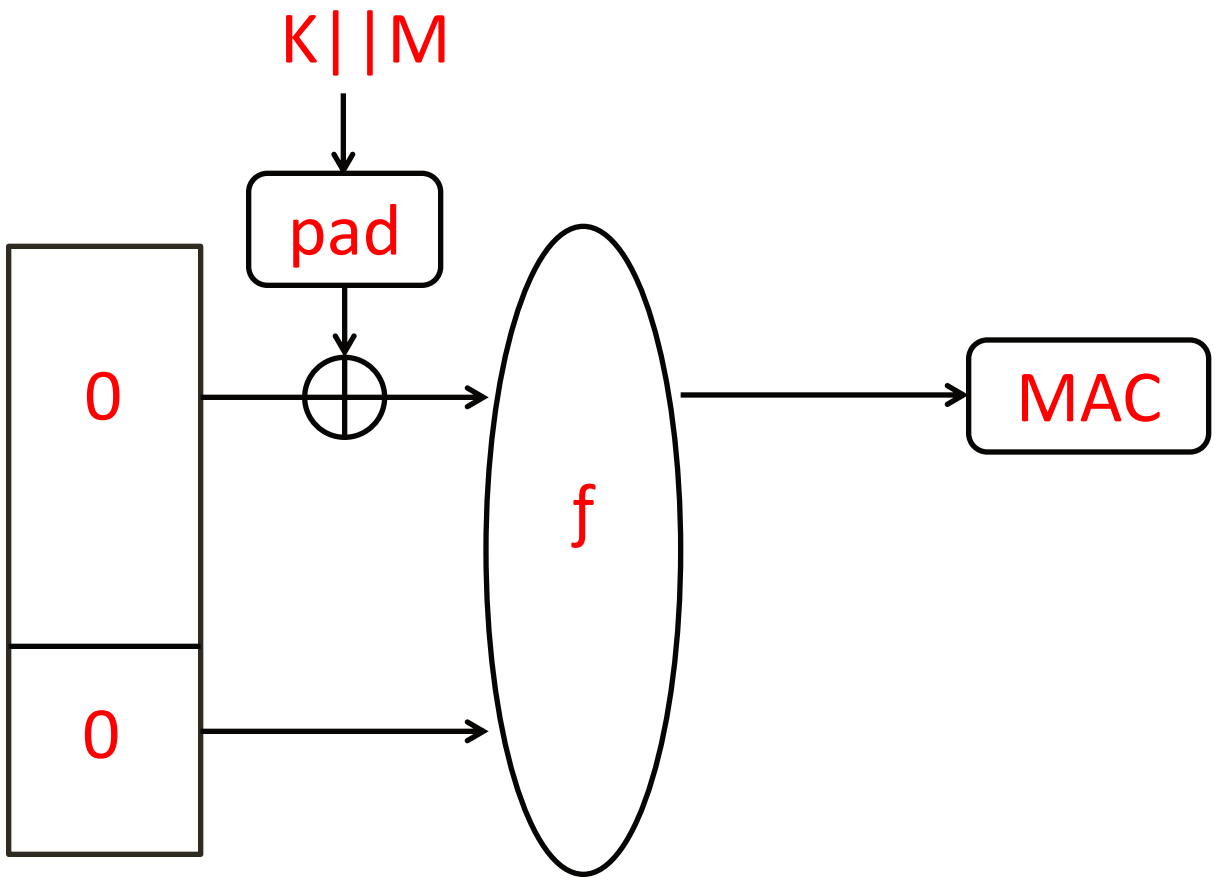
- The new **SHA-3** standard
- uses the **sponge construction**
- ƒ is a permutation that operates on a **1600-bit state**



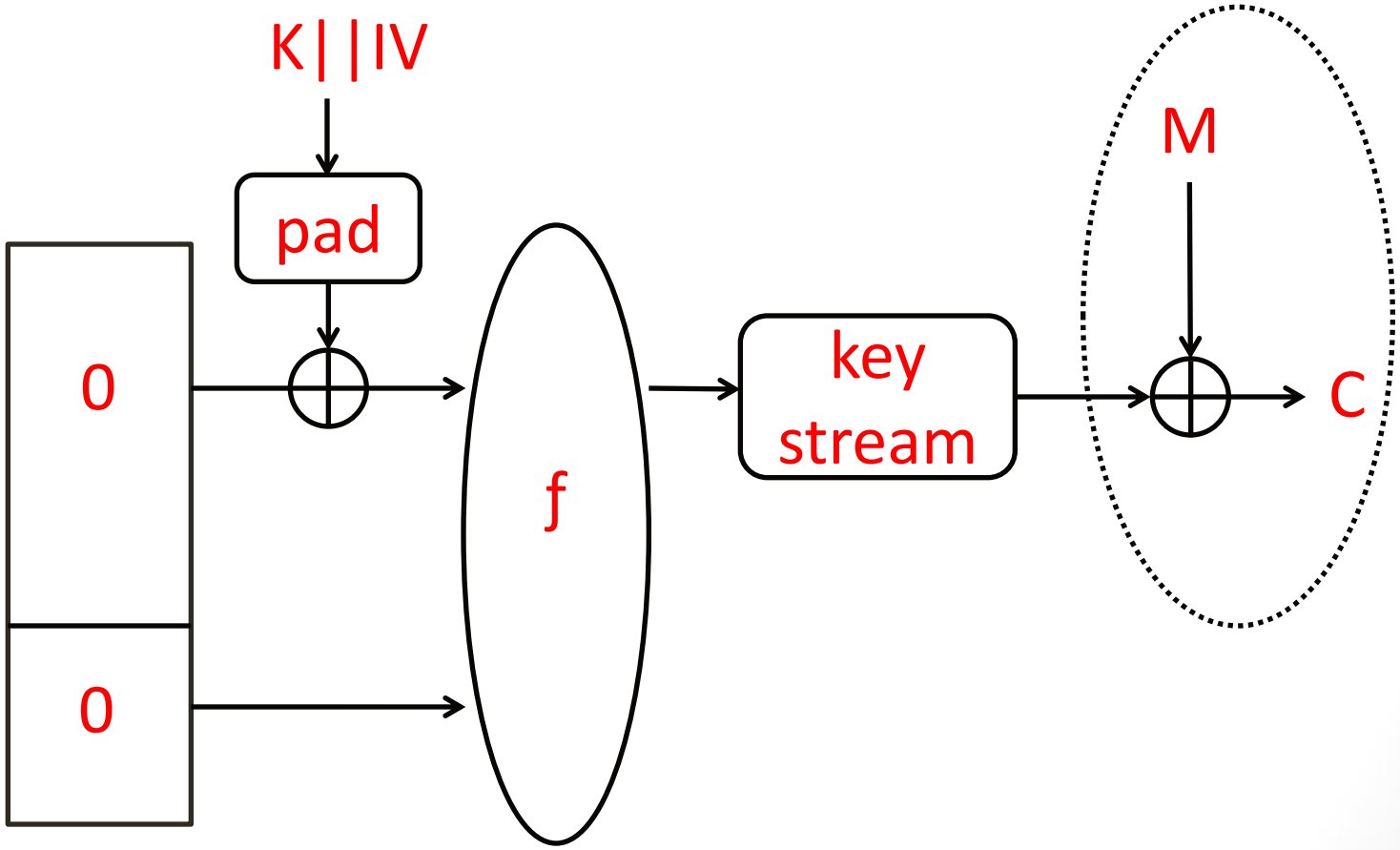
Keccak Sponge Function

- Keccak was extensively analyzed in the key-less mode (as a hash function)
- Due to flexibility of the sponge function, Keccak can be used in the keyed modes (MAC, stream cipher, authenticated cipher)
- Our work focuses on the **keyed modes**.

Keccak as a MAC



Keccak as a Stream Cipher



Our Results

- We analyzed **keyed** modes (MAC, stream cipher, authenticated cipher) of round-reduced Keccak
- mounted **practical cube attacks** on round-reduced Keccak, up to 6 rounds
- proposed **new cube-like-attack** bringing new results for more rounds . Some of the attacks are not practical, yet **much faster** than exhaustive search

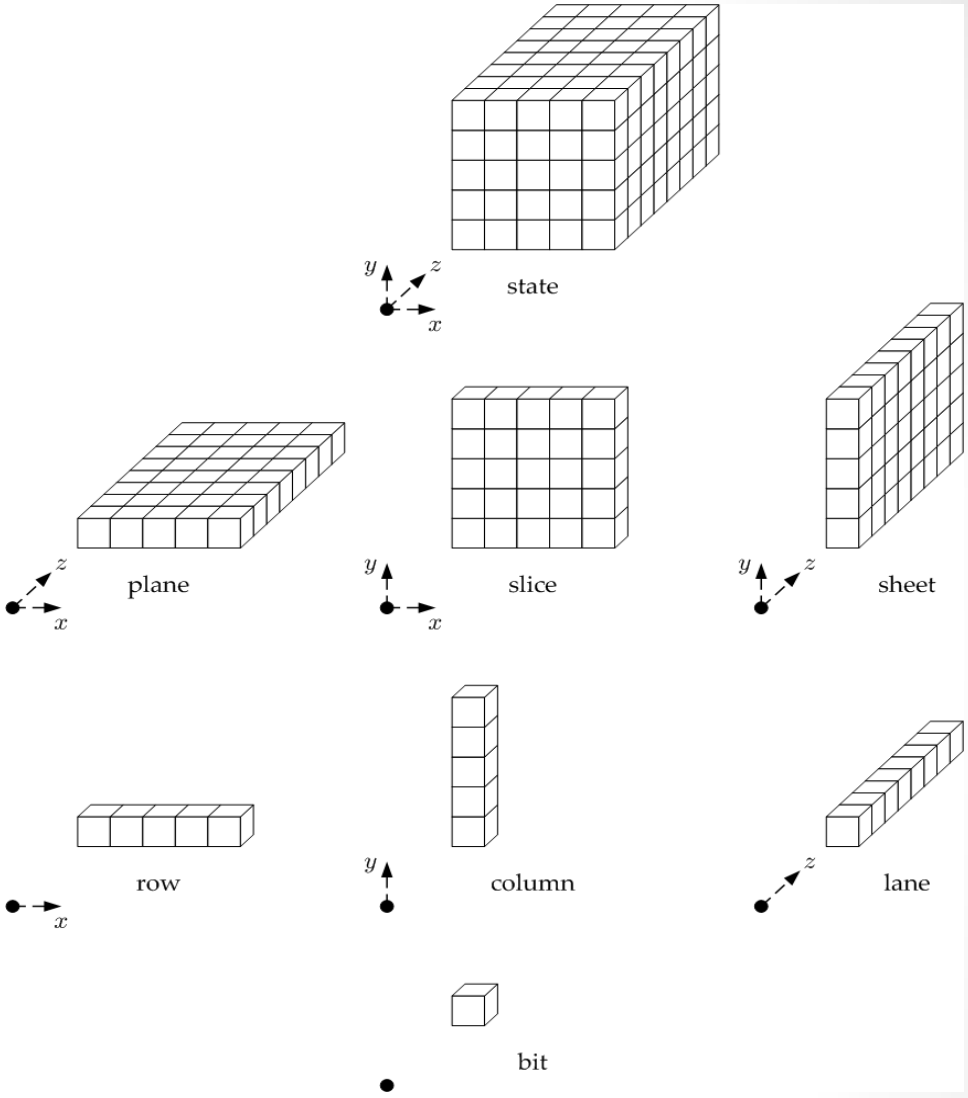
Our Attacks

Rounds	Mode	Type of attack	Attack complexity	Generic
5	MAC	key recovery	2^{36}	2^{128}
6	stream cipher	key recovery	2^{36}	2^{128}
7	AE (Keyak)	key recovery	2^{65}	2^{128}
8	MAC	forgery	2^{129}	2^{256}
9	stream cipher	keystream prediction	2^{256}	2^{512}

Keccak

The Inner State

- Can be viewed as a 5x5x64-bit cube
- Or as a 5x5 matrix, where each cell is a 64-bit lane in the direction of the z axis



Source : http://keccak.noekeon.org

Keccak Permutation

- f is a **24**-round permutation on the 1600-bit state
- Each round consists of a linear and a non-linear mapping.
- The algebraic degree of a round is 2.
- The non-linear mapping (Sbox layer) multiplies bits (bitwise AND) from two consecutive lanes.

The Cube Attack [Dinur and Shamir '09]

- A key recovery related to **high order differential cryptanalysis** (Lai 1994)
- Based on the **algebraic representation** of an output bit of a cryptosystem as a multivariate polynomial over GF(2): $P(v_1,\ldots,v_m,x_1,\ldots,x_n)$
  - $x_1,\ldots,x_n$ are secret variables (key bits)
  - $v_1,\ldots,v_m$ are public variables (**plaintext** bits in block ciphers and **MACs**, **IV** bits in stream ciphers)

The Cube Attack – Brief Intro

- For any polynomial P and term t of variables multiplied together, we can express P as $P=tP_t+Q$ where:
  - all the variables in P_t are disjoint from the variables in t
  - each term in the multivariate polynomial Q misses at least one variable from t
- P_t is called the **superpoly** of t in P
- In the cube attack we exploit **linear** superpolys

The Cube Attack – Brief Intro

- **Preprocessing**: Find a cube t of $\{v_1,\ldots,v_m\}$ such that P_t is **linear** in $\{x_1,\ldots,x_n\}$ (finding cubes with linear superpolys is a heuristic task which can be **very difficult** in practice)

The Cube Attack – Brief Intro

- **Preprocessing**: Find a cube t of $\{v_1,\ldots,v_m\}$ such that P_t is **linear** in $\{x_1,\ldots,x_n\}$ (finding cubes with linear superpolys is a heuristic task which can be **very difficult** in practice)

- **Online**: Sum over the cube t. The result of summation (0 or 1) is equal to the linear superpolys. This way we can form a linear equation.
  - Solve a set of linear equations and recover the key $\{x_1,\ldots,x_n\}$

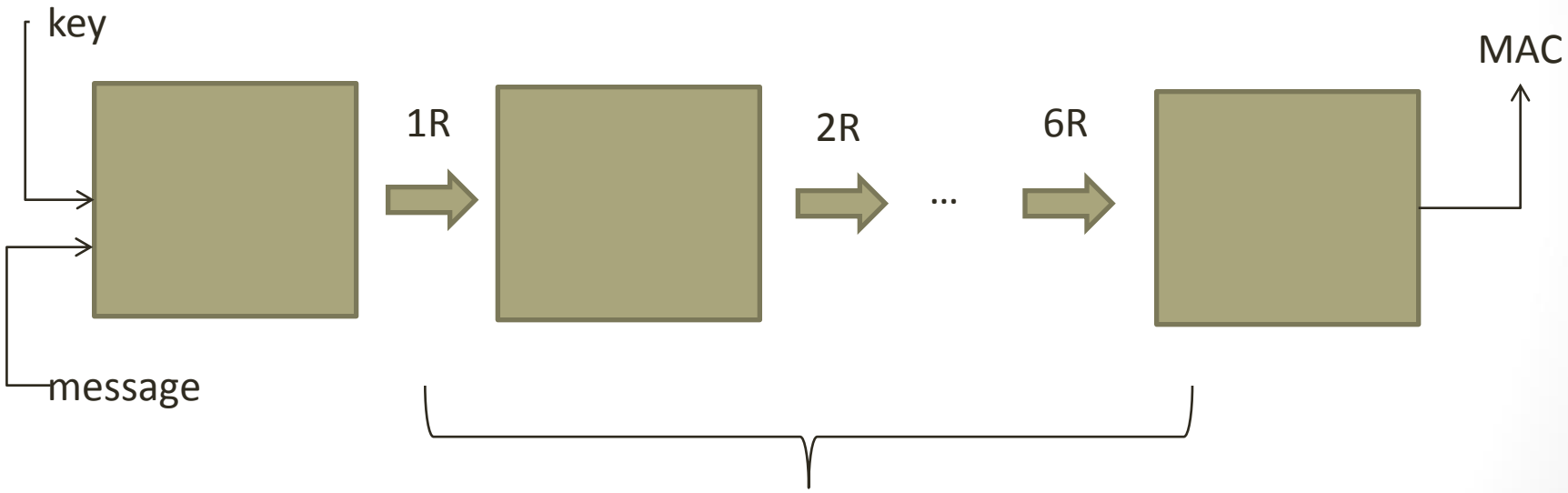
Limitations of Cube Attack

- An inherent limitation for the classic cube attack is an algebraic degree of the algorithm. Too high degree (e.g., 128) makes finding linear superpolys infeasible.

- Can we design a variant of the attack with a better-understood preprocessing phase?

New Approach – High Level View

- Let's consider Keccak-MAC , the algebraic degree of a single round is 2.



6 rounds, then degree 2^6=64 (at most)

New Approach – High Level View

- having algebraic degree 64, it would be very hard to find linear superpolys in reasonable time (most likely, need to sum over 64 bits, so 2^{64} calls…)

- with smaller cubes (e.g., 32), we end up, very likely, with a highly non-linear superpoly

New Approach – High Level View

- having algebraic degree 64, it would be very hard to find linear superpolys in reasonable time (most likely, need to sum over 64 bits, so 2^{64} calls…)
- with smaller cubes (e.g., 32), we end up, very likely, with a highly non-linear superpoly
- what if the superpolys of the output bits are highly non-linear, but depend on only a small subset of the key bits. How the attack might look like in such a case?

New Approach - Precomputation

For each of the 2^{16} possible values, we calculate cube sums (each cube sum bit corresponds to the evaluation of a different output bit superpoly at the 16-bit key value) and keep in memory such a table.

Key bits	Cube sums
000…..0000	011010100 …
000…..0001	110100011 …
000…..0010	000011111 …
000…..0011	111010100 …
000…..0100	010110110 …
…..	….
111…..1111	111010101 …

New Approach - Precomputation

- Assuming that we sum over 32-bit cube and all the superpolys depen only on (the same) 16 key bits, the precomputation cost is: $2^{32} * 2^{16} = 2^{48}$

- In memory we keep a table with 2^{16} records, where each record is a distinct key value and the corresponding vector of the calculated cube sums (superpoly evaluations).

New Approach – Online Phase

- In online phase (given black-box access to the oracle), we sum over the 32-bit cube and having the calculated sums, we search the table for the vector of cube sums and recover the corresponding 16-bit key value

- Thus, cost is 2^{32} calls

Balanced Variant of the Attack

- The precomputation is the bottleneck of the attack, as it is much more expensive than the online attack.
- For Keccak, it is possible to balance complexities of the precomputation and online phase. Therefore, a total cost is reduced. More details in the paper.

New Approach – Some Remarks

- To recover the complete, 128-bit key, we would have to use other cubes (which have superpolys that depend on different small subset of key bits).
- Another option is to recover the remaining part of the key via exhaustive search.

New Approach – Some Remarks

- To recover the complete, 128-bit key, we would have to use other cubes (which have superpolys that depend on different small subset of key bits).
- Another option is to recover the remaining part of the key via exhaustive search.
- We assumed that for each key a corresponding vector of cube sums is (almost) unique. The assumption is based on a theoretical model and is supported by our experiments on Keccak.

New Approach - Requirement

- The attack is based our strong precondition that, *"…superpoly consists of only a part of the key bits (e.g., 16 key bits instead of 128)."*

- For Keccak, the presented attack works if:
  - In the 1st round, cube bits $\{v_1,\ldots,v_{32}\}$ are not multiplied together
  - In the 1st round, only a part of key bits are multiplied with cube bits

New Approach - Details

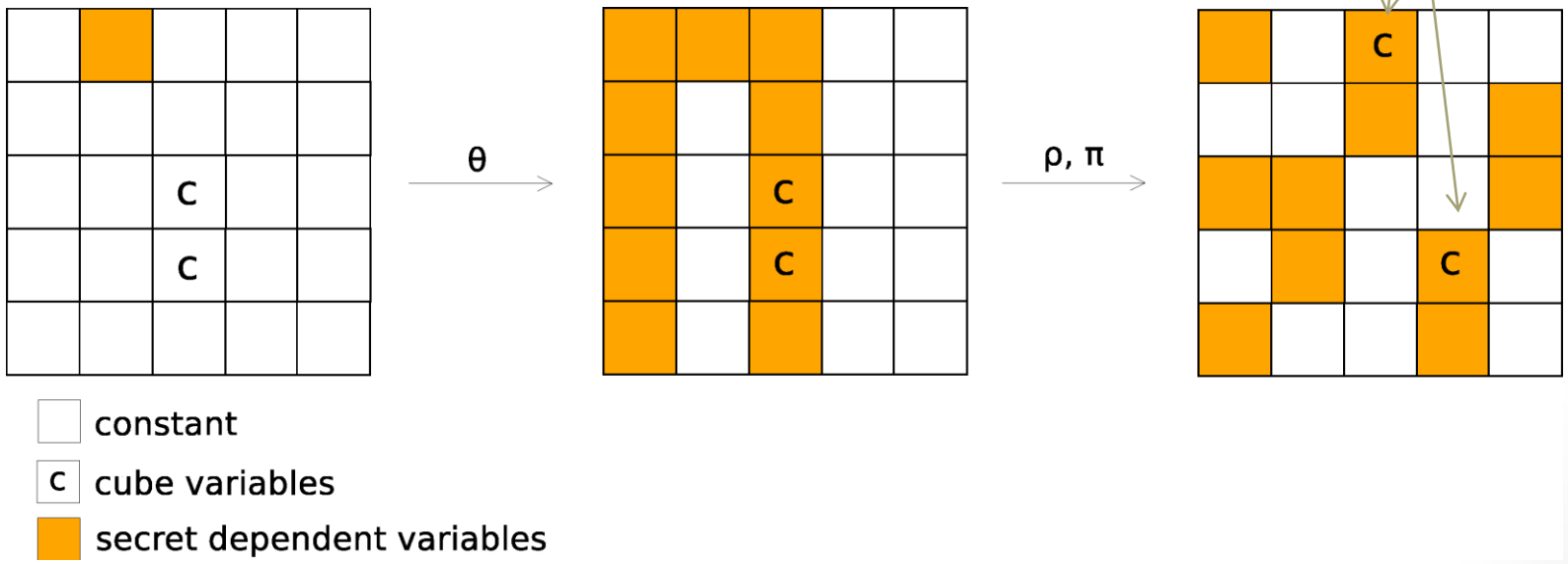
- Let k_i be the (128-16) key bits which are **not** multiplied with the cube bits $\{v_1,\ldots,v_{32}\}$
- We want to know whether a monomial $k_i v_1 v_2 v_3 v_4 \ldots v_{32}$ can be in the ANF?

New Approach - Details

- Let k_i be the (128-16) key bits which are **not** multiplied with the cube bits $\{v_1,\ldots,v_{32}\}$
- We want to know whether a monomial $k_i v_1 v_2 v_3 v_4 \ldots v_{32}$ can be in the ANF?
- The degree of the Keccak state of round 2 in $v_1,\ldots v_{32},k_i$ is 1. Since the degree of round 2-6 is 32, then the output degree in $v_1,\ldots v_{32},k_i$ is **only 32**. Thus the 33-degree monomial **cannot** appear in the ANF.

1st Round – Closer Look

Adjacent lanes to ”cube” lanes
do NOT depend on secret variables. So,
no multiplication between ”c” and ”secret”
in the next Sbox layer.



New Approach - Summary

- better defined and easier to analyze precomputation phase, no ”random walk” for a good cube as in the classic cube attack
- exploits the slow diffusion of the internal mappings to attack more rounds than the cube attack

New Approach - Summary

- better defined and easier to analyze precomputation phase, no ”random walk” for a good cube as in the classic cube attack
- exploits the slow diffusion of the internal mappings to attack more rounds than the cube attack
- technique also applicable to other SPN schemes, e.g., already used for 6-round key-recovery attack on Ascon (authenticated cipher) [*Dobraunig et al. 2015]*
- Limitation: unlike the cube attack, the new techniques do not seem to be efficiently applicable to feedback shift register based stream ciphers

Conclusion and Future Work

- We focused on the keyed modes of Keccak
- A new technique of key-recovery attack was proposed, leading to theoretical results for 7 rounds (much faster than exhaustive search).
- The technique is also applicable to **other SPN** schemes

Conclusion and Future Work

- We focused on the keyed modes of Keccak
- A new technique of key-recovery attack was proposed, leading to theoretical results for 7 rounds (much faster than exhaustive search).
- The technique is also applicable to **other SPN** schemes
- Forgery (MAC mode) and keystream prediction (stream cipher mode), up to 9 rounds (more details in the paper)
- our work in progress: attacks on (round-reduced) PROST and PRIMATE – AE schemes from CAESAR competition

Thank you for your attention!