Cube Attack against Full KRAVATTE

Jian Guo and Ling Song

Nanyang Technological University, Singapore.
{guojian,songling}@ntu.edu.sg

Abstract. This note analyzes the security of KRAVATTE against the cube attack. We provide an analysis result which recovers the master key of the current version of full KRAVATTE with data and time complexities $2^{136.01}$, and negligible memory. The same could be applied to the first version of KRAVATTE with complexities of $2^{38.04}$, which could be carried out in practice. These results are possible thanks to a clever way of constructing affine spaces bypassing the first permutation layer of KRAVATTE proposed by the designers and a simple yet efficient way to invert the last layer of Sbox in KRAVATTE.

Keywords: KRAVATTE, Cube Attack, KECCAK

1 Introduction

1.1 The KECCAK-p

The KECCAK-p permutations are specified with two parameters: the width of the permutation in bits $\boldsymbol{b}$ and the number of rounds $\boldsymbol{n_r}$. The KECCAK-p permutation with $\boldsymbol{n_r}$ rounds and width $\boldsymbol{b}$ is denoted by KECCAK-$p[\boldsymbol{b}, \boldsymbol{n_r}]$, where $\boldsymbol{n_r}$ is any positive integer and $\boldsymbol{b}$ can be any value of the form $25 \cdot 2^l$ for $l = 0, \ldots, 6$. The $\boldsymbol{b}$-bit state a for the KECCAK-$p[\boldsymbol{b}, \boldsymbol{n_r}]$ permutation is seen as a three-dimensional array of bits, namely $a[5][5][w]$ with $w = 2^l$. The expression $a[x][y][z]$ with $0 \leq x, y < 5$, $0 \leq z < w$, denotes the bit with (x, y, z) coordinate. The coordinates are always considered within modulo 5 for x and y and modulo w for z. The one-dimensional portion $a[*][y][z]$ is called a *row*, $a[x][*][z]$ a *column* and $a[x][y][*]$ a *lane*.

The KECCAK-$p[\boldsymbol{b}, \boldsymbol{n_r}]$ permutation iterates an identical round function (up to a difference of round-dependent constant addition) $\boldsymbol{n_r}$ times, each of which consists of five bijective mappings $\mathtt{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta$, with details as follows.

θ : $A[x][y][z] \leftarrow A[x][y][z] + \Sigma_{y=0}^{4} A[x-1][y][z] + \Sigma_{y=0}^{4} A[x+1][y][z-1]$,

ρ : $A[x][y][z] \leftarrow A[x][y][(z + T(x, y))]$, where $T(x, y)s$ are pre-defined rotation constants,

π : $A[y][2x + 3y][z] \leftarrow A[x][y][z]$,

χ : $A[x][y][z] \leftarrow A[x][y][z] + ((A[x+1][y][z] + 1) \cdot A[x+2][y][z])$,

ι : $A[0][0] \leftarrow A[0][0] + RC_{i_r}$, where RC_{i_r} is the round constant for the i_r-th round.

Here, '+' denotes XOR and '·' denotes logic AND. Expressions in the x and y coordinates should be taken in modulo 5 and expressions in the z coordinate modulo w.

1.2 The Farfalle construction and KRAVATTE

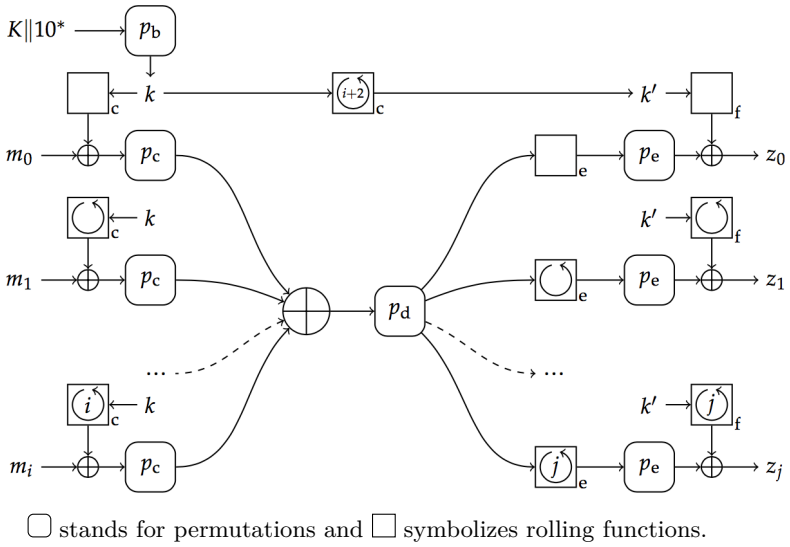

Figure 1: The Farfalle construction [1].

Farfalle [1] is a permutation-based construction for building pseudorandom functions, as shown in Figure 1. It takes a key of variable-length and a message sequence as input, and outputs a bit stream of desired length. Farfalle has three parts: a key derivation, a compression layer and an expansion layer, which makes use of four permutations p_b, p_c, p_d, and p_e, and three rolling functions $roll_c, roll_e$, and $roll_f$. Firstly, the key derivation generates $\boldsymbol{b}$-bit masks from the key using p_b, $roll_c$, and $roll_f$. These masks derived from the key are used for pre/post-whitening. Then, the compression layer computes a $\boldsymbol{b}$-bit accumulator from the message sequence by the parallel application of p_c. Finally, the expansion layer computes rolling states from the accumulator using $roll_e$, and passes the rolling states to p_e to generate the output. Due to the inherent parallelism of the Farfalle construction, Farfalle instances can be very efficient.

Proposed in [1], KRAVATTE is a Farfalle instance based on KECCAK-$p[\boldsymbol{b}, \boldsymbol{n_r}]$. Specifically,

$$p_b = p_c = \text{KECCAK-}p[1600, 6],$$
$$p_d = p_e = \text{KECCAK-}p[1600, 4].$$

$roll_f$ is the identity, and $roll_e = roll_c$. We refer to [1] for details and they are omitted here as this analysis does not depends on them as long as all the rolling functions are linear and their inverses are easily computable.

2 Cube Attack against KRAVATTE

2.1 The Cube Attack

The cube attack was introduced by Dinur and Shamir [3] in 2009, to analyze the security of primitives with low algebraic degrees. The idea is similar to high order differential attacks, when the algebraic degree of a polynomial is d, applying $d+1$-th derivative will result in 0. Similarly, for a primitive f with the degree of the algebraic expression at most d, the sum $\sum_{x \in S} f(x) = 0$ for any affine subspace S of dimension at least $d+1$.

As observed by the Keccak designers, the algebraic degree of the KECCAK Sbox (a.k.a., the χ operation) is 2.

2.2 Cube Attack against KRAVATTE

As already described in [1, Section 5.4], the cube attack (or high order differential) works for round-reduced variant. The idea is to bypass the p_c layer, and find a big enough affine subspace of the output of p_c. It can be constructed as follows [1].

> *Each string has the form* $m(\lambda) = m_0^{\lambda_0} \| m_1^{\lambda_1} \| \cdots \| m_d^{\lambda_d}$*, where* $\lambda_i \in \{0,1\}$ *and* $m_i^0 \neq m_i^1$ *for all* i*. If we denote* $r_i = p_c(m_i^0 + roll^i(k))$ *and* $r_i^* = p_c(m_i^1 + roll^i(k))$ *and* $r_i' = r_i + r_i^* \neq 0$*, the value of the accumulator for the input string with label* λ *is* $x_\lambda = \sum_i r_i + \sum_i \lambda_i r_i'$*. Over the space of input strings, this is an affine space.*

Note the rest parts of KRAVATTE are mainly p_d and p_e which have in total 8 rounds. The algebraic degree of each round is 2. Without any further details of the round functions, the cube attack can already analyze up to 7 rounds, since its degree is $2^7 = 128$. This requires a data complexity of $129 \cdot 2^{129} = 2^{136.01}$ message blocks and the same time complexity.

Below, we show how we can invert the last round, so to extend this attack to the full KRAVATTE without increasing the attack complexities.

2.3 Inverting the last round of KRAVATTE and Improved Cube Attack

The core idea is that, the first three operations θ, ρ, π of the KECCAK-p round function are linear, which do not increase the algebraic degrees. So the zero-sum property still holds at the point of the state just before the last χ. This zero-sum could be checked against for each Sbox of the last χ individually, i.e., we can take out the 5 bits of the PRF output at the position of the first row of the first slice, i.e., the first Sbox, recover the 5 corresponding bits of the post-whitening key k'

by simply bruteforcing all the 2^5 choices, compute χ^{-1} and check against the zero-sum property. This can be repeated to all other Sboxes until we recover the full post-whitening key k'. Note the knowledge of k' give us the master key K by computing $p_b^{-1}(roll_c^{-(i+2)}(k'))$. This attack has the same data complexity as before, but increases the computation complexity by a factor of 2^5 for bruteforcing, and this factor can be removed by filtering repeated occurrence of 5-bit values at the output of each Sbox, since we are checking for zero-sum properties. This dramatically reduces the data involved in the bruteforce stage from 2^{129} to no more than 2^5, and now the bruteforce takes less than 2^{10} to complete for each Sbox and thus less than $320 \cdot 2^{10}$ for all Sboxes. Therefore, recovering the master key of the full KRAVATTE costs data and time complexities of $2^{136.01}$.

We note the same applies to a previous version of KRAVATTE [2] with complexities $33 \cdot 2^{33} = 2^{38.04}$, as the KECCAK-$p$ permutation after the accumulator had 6 rounds only.

3 Conclusion

There is a tradeoff between the attack complexities and number of round attacked. For instance, we can attack one round less with complexities $65 \cdot 2^{65} = 2^{71.02}$. This analysis leaves no security margin for the current version of KRAVATTE. However, inverting more than one rounds seems infeasible now, so increasing the number of rounds of p_d and/or p_e will likely resist this attack.

References

1. Bertoni, G., Daemen, J., Hoffert, S., Peeters, M., Van Assche, G., Van Keer, R.: Farfalle: Parallel Permutation-Based Cryptography. Cryptology ePrint Archive, Report 2016/1188 (2016), https://eprint.iacr.org/2016/1188
2. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: Farfalle: Parallel Permutation-Based Cryptography, version 20170101:153600. Cryptology ePrint Archive, Report 2016/1188 (2016), https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2016/1188&version=20170101:153600&file=1188.pdf
3. Dinur, I., Shamir, A.: Cube Attacks on Tweakable Black Box Polynomials. In: Joux, A. (ed.) Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings. LNCS, vol. 5479, pp. 278–299. Springer (2009), https://doi.org/10.1007/978-3-642-01001-9_16