PMRF Symposium 2022
Cryptanalysis of KECCAK & Algorithms for Lattice problems

Mahesh Sreekumar Rajasree
Guided by: Prof. Manindra Agrawal & Prof. Shashank K Mehta
Department of Computer Science
Indian Institute of Technology Kanpur

Outline

KECCAK

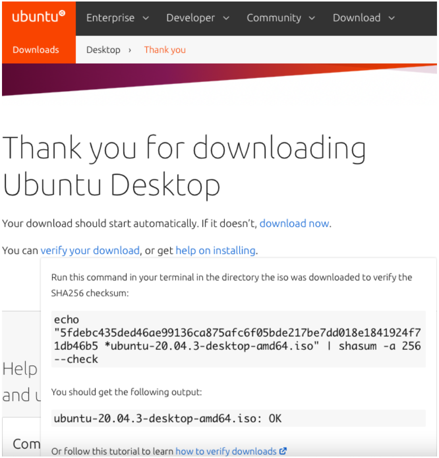


Figure: Snapshot of Ubuntu download page

Run this command in your terminal in the directory the iso was downloaded to verify the SHA256 checksum:

```
echo
"5fdebc435ded46ae99136ca875afc6f05bde217be7dd018e1841924f7
1db46b5 *ubuntu-20.04.3-desktop-amd64.iso" | shasum -a 256
--check
```

You should get the following output:

```
ubuntu-20.04.3-desktop-amd64.iso: OK
```

Figure: Snapshot of Ubuntu download page

- **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.

Let H be a cryptographic hash function.

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

Let H be a cryptographic hash function.

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

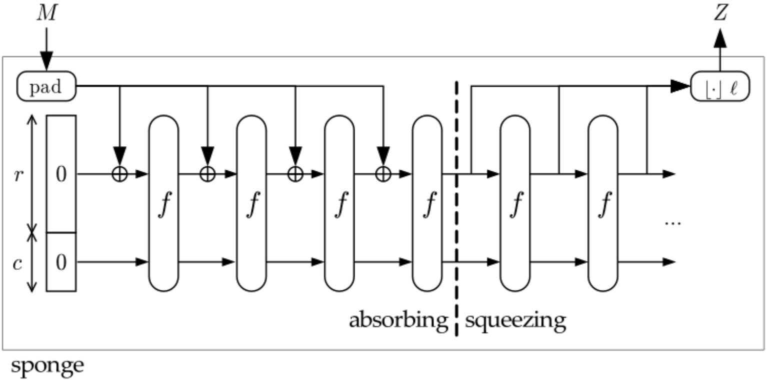
Let H be a cryptographic hash function.

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

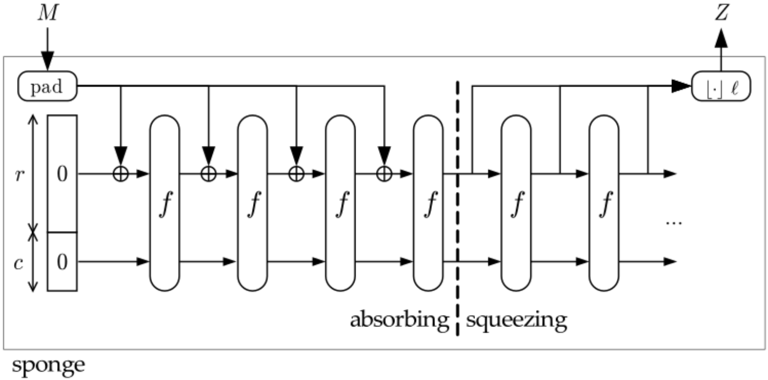
Let H be a cryptographic hash function.

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

KECCAK: Sponge Construction



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



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

pad: padding function (10*1)

f: KECCAK-f permutation

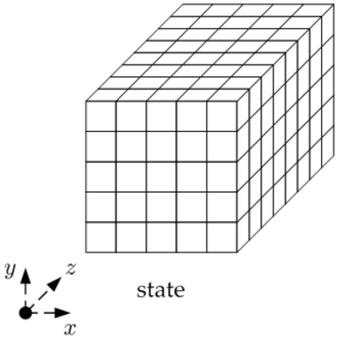


Figure: State

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

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 $P[x][z] = \bigoplus_{i=0}^{4} S[x,i,z]$

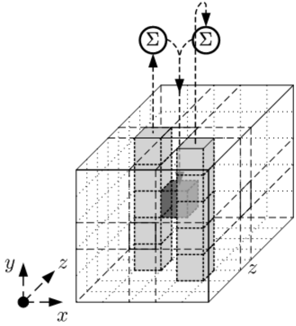


Figure: θ

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

Description of ρ

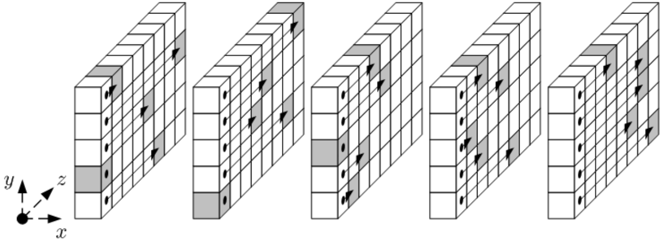


Figure: ρ

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

Description of π

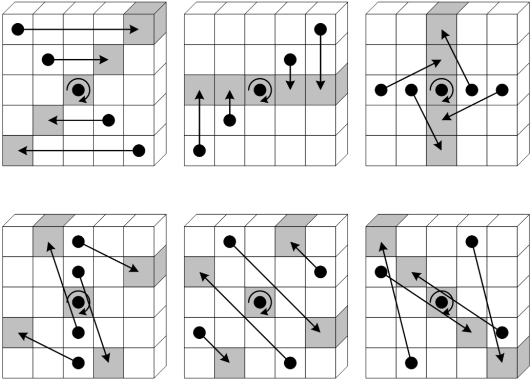


Figure: π

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

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

Description of χ and ι

- χ: Only non-linear function

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

- 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.

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.

$$f = \underbrace{(\iota \circ \chi \circ \pi \circ \rho \circ \theta) \circ (\iota \circ \chi \circ \pi \circ \rho \circ \theta) \circ \cdots}_{r}$$

Studied non-linear structure on KECCAK hash family and gave better preimage attacks.

Studied non-linear structure on KECCAK hash family and gave better preimage attacks.

Rounds	Instances	Our Results	Previous Results
2	384	$\mathbf{2^{113}}$	2^{129}[1]
	512	$\mathbf{2^{321}}$	2^{384}[1]
3	384	$\mathbf{2^{321}}$	2^{322}[1]
	512	$\mathbf{2^{475}}$	2^{482}[1]
4	384	$\mathbf{2^{371}}$	2^{378}[2]

Table: Summary of preimage attacks

$\mathbb{Z}^n$-isomorphism

- Most of the widely used cryptosystems like RSA, Diffie-Hellman and Elliptic Curve cryptosystem are based on integer factorization or discrete logarithm problem

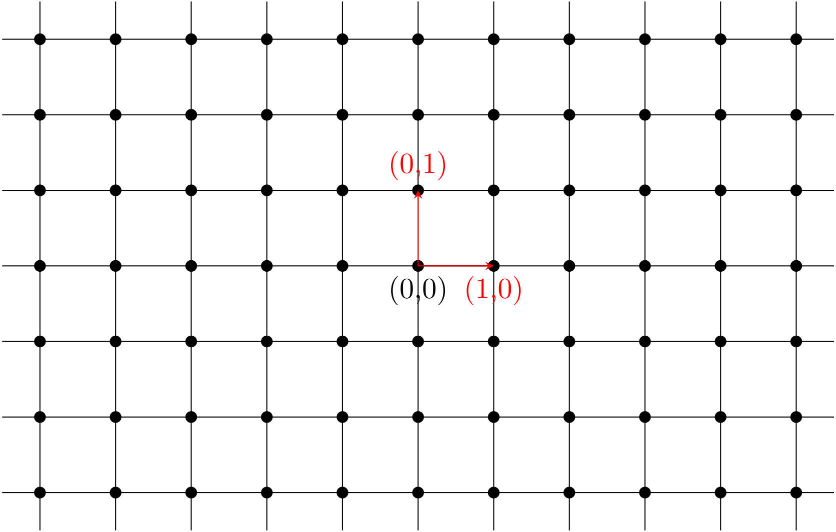
- Most of the widely used cryptosystems like RSA, Diffie-Hellman and Elliptic Curve cryptosystem are based on integer factorization or discrete logarithm problem
- These systems are vulnerable to quantum attacks that use the Shor's algorithm, which efficiently solves the above problems.

- Most of the widely used cryptosystems like RSA, Diffie-Hellman and Elliptic Curve cryptosystem are based on integer factorization or discrete logarithm problem
- These systems are vulnerable to quantum attacks that use the Shor's algorithm, which efficiently solves the above problems.
- Lattice based cryptosystems are one of the candidates for post-quantum cryptosystem. The security of such systems are based on the hardness of Shortest Vector Problem (SVP) and Closest Vector Problem (CVP).

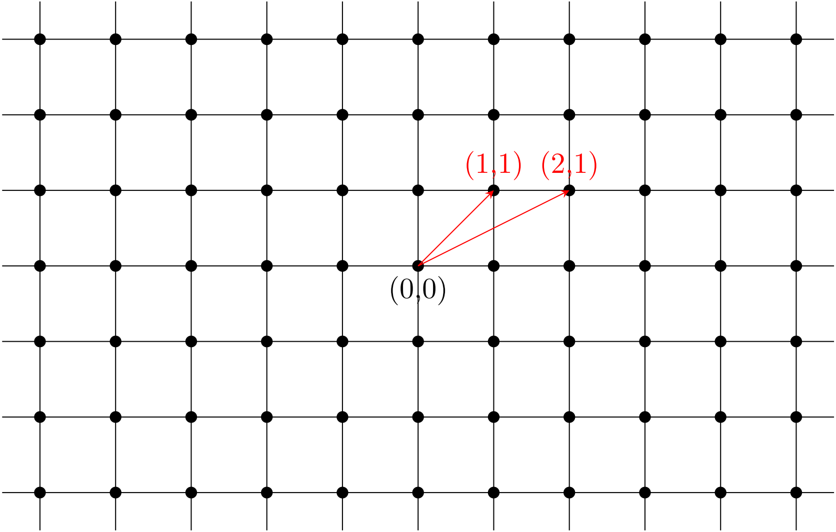
Let $B = [b_1, \ldots, b_n]$ is a set of linearly independent vectors. A **lattice** $\mathcal{L}(B)$ is the set of all integer linear combinations of the vectors in B, i.e.,

$$\mathcal{L}(B) = \{Bz | \forall z \in \mathbb{Z}^n\}$$

Lattice



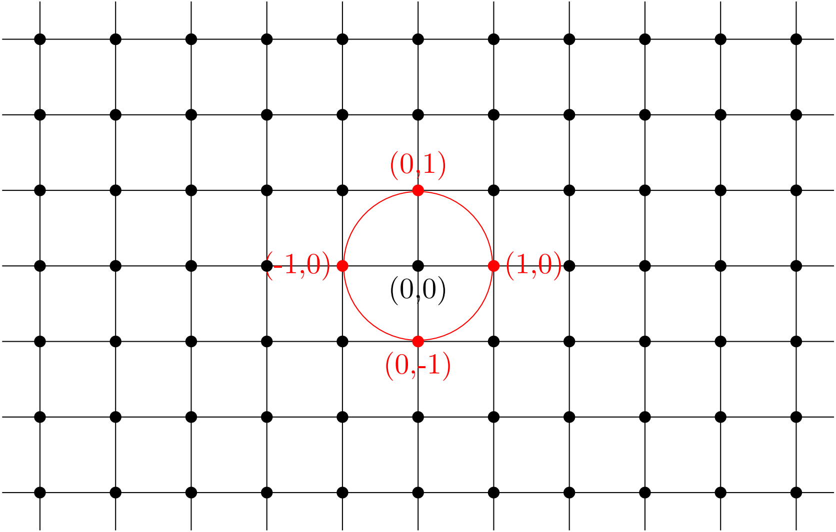
$\mathbb{Z}^2$ lattice



Let $B = [b_1, \ldots, b_n]$ is a set of linearly independent vectors. A **lattice** $\mathcal{L}(B)$ is the set of all integer linear combinations of the vectors in B, i.e.,

$$\mathcal{L}(B) = \{Bz | \forall z \in \mathbb{Z}^n\}$$

Shortest Vector Problem (SVP):- Given a lattice $\mathcal{L}(B)$, find a non-zero shortest vector v in $\mathcal{L}(B)$, i.e., $||v|| \leq ||w||, \forall w \in \mathcal{L}(B) \setminus \{0\}$.


(0,1)
(-1,0)
(1,0)
(0,0)
(0,-1)

Algorithm	**Time**	**Space**	**Deterministic/Randomized**
Enumeration [3]	$n^{O(n)}$	$poly(n)$	Deterministic
AKS [2]	$2^{O(n)}$	$2^{O(n)}$	Randomized
Voronoi based [1]	$\tilde{O}(2^{2n})$	$\tilde{O}(2^n)$	Deterministic
Gaussian Sampling [3]	$2^{n+o(n)}$	$2^{n+o(n)}$	Randomized

Table: Summary of SVP algorithms

$\mathbb{Z}^n$-isomorphism

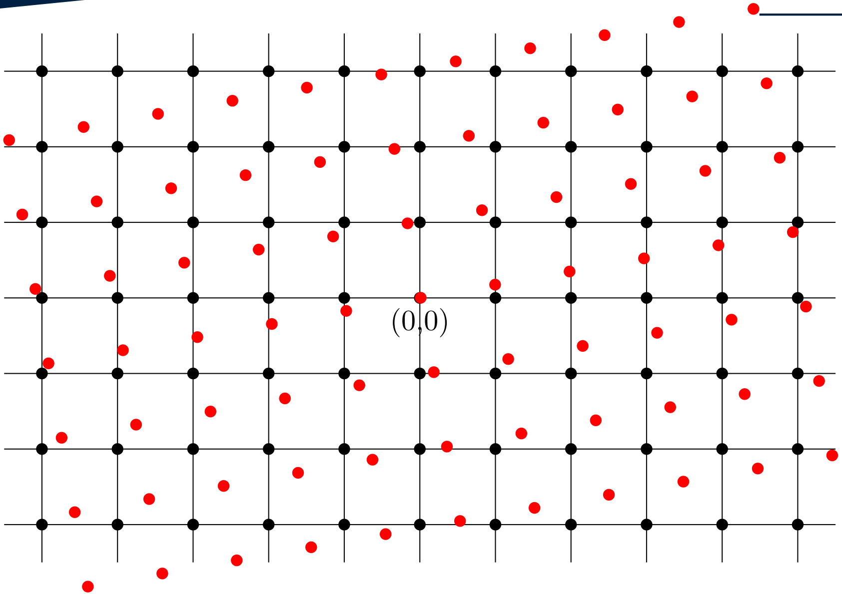
Given a lattice $\mathcal{L}$, decide whether $\mathcal{L}$ is isomorphic to $\mathbb{Z}^n$.

Given a lattice $\mathcal{L}$, decide whether $\mathcal{L}$ is isomorphic to $\mathbb{Z}^n$.
In other words, given a basis B, decide whether there exists an orthonormal matrix O and a unimodular matrix U such that

$$OB = U$$

Note: Any basis of $\mathbb{Z}^n$ is a unimodular matrix and vice-versa.

$\mathbb{Z}^2$ isomorphism



$\mathbb{Z}^n$-isomorphism

- $\mathbb{Z}^n$-isomorphism is known to be in NP$\cap$Co-NP.

$\mathbb{Z}^n$-isomorphism

- $\mathbb{Z}^n$-isomorphism is known to be in NP$\cap$Co-NP.
- $\mathbb{Z}^n$-isomorphism can be solved using an SVP algorithm but it takes exponential time.

Given a primitive vector $v \in \mathbb{Z}^n$ such that $||v|| > 1$,

Given a primitive vector $v \in \mathbb{Z}^n$ such that $||v|| > 1$, there exists $b_2, \ldots, b_n$ such that

Given a primitive vector $v \in \mathbb{Z}^n$ such that $||v|| > 1$, there exists $b_2, \ldots, b_n$ such that

1. $[v, b_2, \ldots, b_n]$ is a basis of $\mathbb{Z}^n$.

Given a primitive vector $v \in \mathbb{Z}^n$ such that $||v|| > 1$, there exists $b_2, \ldots, b_n$ such that

1. $[v, b_2, \ldots, b_n]$ is a basis of $\mathbb{Z}^n$.
2. $||b_i|| < ||v||, \forall i \in \{2, \ldots, n\}$.

Given a primitive vector $v \in \mathbb{Z}^n$ such that $||v|| > 1$, there exists $b_2, \dots, b_n$ such that

1. $[v, b_2, \dots, b_n]$ is a basis of $\mathbb{Z}^n$.
2. $||b_i|| < ||v||, \forall i \in \{2, \dots, n\}$.

The proof of this theorem uses concepts from number theory.

Conjecture

Given a primitive vector $v_1, \ldots, v_k \in \mathbb{Z}^n$ such that

1. $||v_1|| \geq ||v_2|| \geq \cdots \geq ||v_{k+1}|| > 1$

Conjecture

Given a primitive vector $v_1, \ldots, v_k \in \mathbb{Z}^n$ such that

1. $||v_1|| \geq ||v_2|| \geq \cdots \geq ||v_{k+1}|| > 1$
2. there exists $v_{k+1}, \ldots, v_n$ such that $[v_1, v_2, \ldots, v_n]$ is a basis of $\mathbb{Z}^n$

Given a primitive vector $v_1, \ldots, v_k \in \mathbb{Z}^n$ such that

1. $||v_1|| \geq ||v_2|| \geq \cdots \geq ||v_{k+1}|| > 1$
2. there exists $v_{k+1}, \ldots, v_n$ such that $[v_1, v_2, \ldots, v_n]$ is a basis of $\mathbb{Z}^n$

there exists $b_{k+1}, \ldots, b_n$ such that

Given a primitive vector $v_1, \ldots, v_k \in \mathbb{Z}^n$ such that

1. $||v_1|| \geq ||v_2|| \geq \cdots \geq ||v_{k+1}|| > 1$
2. there exists $v_{k+1}, \ldots, v_n$ such that $[v_1, v_2, \ldots, v_n]$ is a basis of $\mathbb{Z}^n$

there exists $b_{k+1}, \ldots, b_n$ such that

1. $[v_1, \ldots, v_k, b_{k+1}, \ldots, b_n]$ is a basis of $\mathbb{Z}^n$.

Given a primitive vector $v_1, \dots, v_k \in \mathbb{Z}^n$ such that

1. $||v_1|| \geq ||v_2|| \geq \cdots \geq ||v_{k+1}|| > 1$
2. there exists $v_{k+1}, \dots, v_n$ such that $[v_1, v_2, \dots, v_n]$ is a basis of $\mathbb{Z}^n$

there exists $b_{k+1}, \dots, b_n$ such that

1. $[v_1, \dots, v_k, b_{k+1}, \dots, b_n]$ is a basis of $\mathbb{Z}^n$.
2. $||b_i|| < ||v_1||, \forall i \in \{k+1, \dots, n\}$.

THANK YOU!

Rajasree M.S. (2019) Cryptanalysis of Round-Reduced KECCAK Using Non-linear Structures. In: Hao F., Ruj S., Sen Gupta S. (eds) Progress in Cryptology – INDOCRYPT 2019. INDOCRYPT 2019. Lecture Notes in Computer Science, vol 11898. Springer, Cham.
`https://doi.org/10.1007/978-3-030-35423-7_9`

Guo, J., Liu, M. and Song, L., 2016, December. Linear structures: Applications to cryptanalysis of round-reduced Keccak. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 249-274). Springer, Berlin, Heidelberg.

Morawiecki, P., Pieprzyk, J. and Srebrny, M., 2013, March. Rotational cryptanalysis of round-reduced Keccak. In International Workshop on Fast Software Encryption (pp. 241-262). Springer, Berlin, Heidelberg.

Kannan, R., 1987. Minkowski's convex body theorem and integer programming. Mathematics of operations research, 12(3), pp.415-440.

- Micciancio, D. and Voulgaris, P., 2013. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. SIAM Journal on Computing, 42(3), pp.1364-1391.
- Ajtai, M., Kumar, R. and Sivakumar, D., 2001, July. A sieve algorithm for the shortest lattice vector problem. In Proceedings of the thirty-third annual ACM symposium on Theory of computing (pp. 601-610).
- Aggarwal, D., Dadush, D., Regev, O. and Stephens-Davidowitz, N., 2015, June. Solving the shortest vector problem in 2n time using discrete Gaussian sampling. In Proceedings of the forty-seventh annual ACM symposium on Theory of computing (pp. 733-742).