SHA-3 Hash Function

Compiled by - Nazmus Sakib Akash

Introducing SHA-3

In 2004, collision-finding attacks against MD5 and SHA-0 were announced by Xiaoyun Wang.

- One year later it was claimed that the attack could be extended to SHA-1 and it was claimed that a collision search would take 263 steps, which is considerably less than the 280 achieved by the birthday attack

The Wang attack should be taken serious and NIST held two public workshops to assess the status of SHA and to solicit public input on its cryptographic hash function policy and standard.

Subsequently, NIST decided to develop an additional hash function, to be named SHA-3, through a public competition

Design

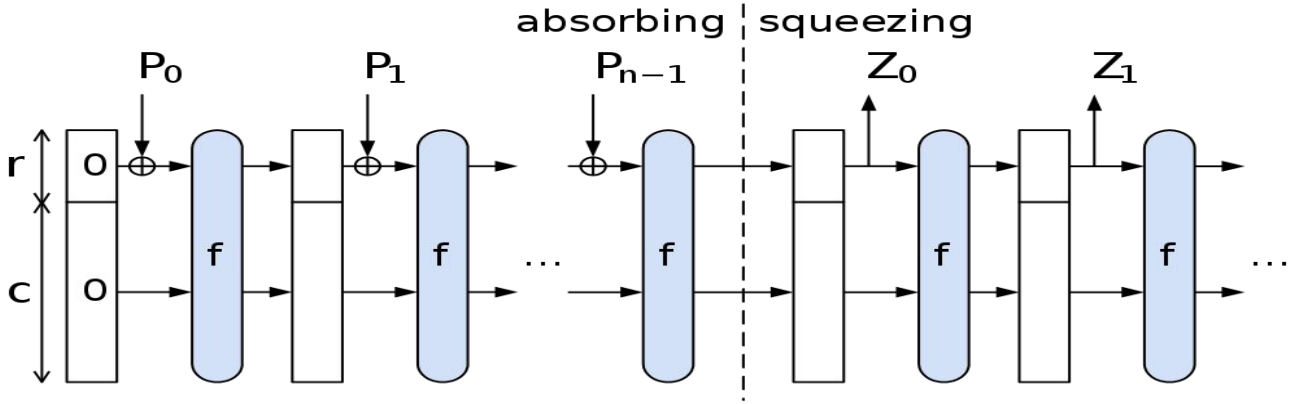
SHA-3 uses the “Sponge Construction”

- Data is “absorbed” into the sponge, then the result is “squeezed” out.

Absorbing phase: Message blocks are X-ORed into a subset of the state, which is then transformed as a whole using a permutation function f.

Squeeze phase: Output blocks are read from the same subset of the state, alternated with the state transformation function f.

Design (Contd.)



The sponge construction for hash functions. Pi are input, Zi are hashed output.

How it works

Given an **input bit string N**, **a padding function pad**, a **permutation function f** that operates on bit blocks of **width b**, a **rate r** and an **output length d**, we have **capacity c = b − r** and the sponge construction **Z = sponge[f,pad,r](N,d)**, yielding a bit **string Z** of length d, works as follows:

1. Pad the input *N* using the pad function, yielding a padded bit string *P* with a length divisible by *r* (such that *n* = len(*P*)/*r* is integer)

2. Break P into n consecutive r-bit pieces P0, ..., Pn−1

How it works (Contd.)

3. Initialize the state S to a string of b zero bits.

4. Absorb the input into the state: for each block Pi:

   - extend Pi at the end by a string of c zero bits, yielding one of length b
   - XOR that with S
   - apply the block permutation f to the result, yielding a new state S

5. Initialize Z to be the empty string

How it works (Contd.)

6. While the length of Z is less than d:

   - append the first r bits of S to Z
   - if Z is still less than d bits long, apply f to S, yielding a new state S

7. Truncate Z to d bits.

The fact that the internal state S contains c additional bits of information in addition to what is output to Z prevents the length extension attacks that SHA-2, SHA-1, MD5 and other hashes based on the Merkle–Damgård construction are susceptible to.