Soumodipta Bose| 2021201086

Using DLP to build fixed-length Collison Resistant Hash function(Code)

Fixed Length Hash Function:

- **hash(x1, x2):** Generates fixed length hash using DLP

  Args:

  x1 (int): input to be compressed

  x2 (int): input to be compressed

  Returns:

  int : integer after 50% compression

- **generator(p, q):**Returns a primitive root of p

  Args:

  p (int): safe prime number

  q (int): safe prime number

  Returns:

  int: primitive root

- **get_group_parameters():**Gets the group parameters

  Working:

  For now prime no. selection is static using a 16 bit Sophie Germain safe prime, will move towards safe prime generation in next update with more time

  Returns:

  p,q,g,h: Returns all the group parameters

- **hash_wrapper(x1, x2):**hash wrapper for binary strings

  Args:

  x1 (binary string): binary number

  x2 (binary string): binary number

  Returns:

  binary string: binary number

Usage:

1. Take two integer numbers as input.
2. The Gen over here is **get_group_parameters()**
3. The **generator(p,q)** returns a random primitive root of p, here p and q are both safe primes.

4. The group parameters are generated globally so once set cannot be changed in that cycle, i.e. value of g and h are random and they are fixed for that cycle of use while in memory.
5. The hashed value is displayed in binary.

Utility functions:

- **dec_to_bin_wo_pad(x):** Converts decimal to binary without padding
- **dec_to_bin(x, size):**Converts decimal to binary with padding
- **bin_to_dec(x):** converts binary to decimal