Soumodipta Bose| 2021201086

Using MD transform to obtain a (provably secure) Collision Resistant Hash Function(Code)

Hash Function using MD Transform:

- **merkle_damgard(iv, message):**Merkle Damgard Transform

  Working:

  - Message length = L len(iv)=l
  - Makes message a multiple of length l and appends message size
  - iterates and applies Fixed length hashing using x2 as message and x1 as hash of previous iter

  Args:

  iv (binary string nbit): initialization vector

  message (binary string): message in binary

  Returns:

  binary string nbit: Hashed value

- **Hash(iv, message):**Wrapper for merkle damgard

Usage:

1. The Gen over here is **get_group_parameters()**
2. Fix a binary string as Initialization Vector for Merkle damgard transform, here we are using getrandbits to generate a random initialization vector.
3. Take an input string like "Hello world" which we will hash.
4. Convert the string into binary using utility function **str_to_bin(s)**
5. Use **Hash()** to perform MD Transform and get hash value in binary.
6. Since prime numbers are fixed the limitation is that it can only handle upto 16bits, will be expanded later in future revisions.

Crypto Library:

- **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

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
- **str_to_bin(s):** converts string to binary