Blockchain Technology and Applications

CS 731

Cryptographic Techniques for Blockchain

Hash functions

Dr. Ir. Angshuman Karmakar

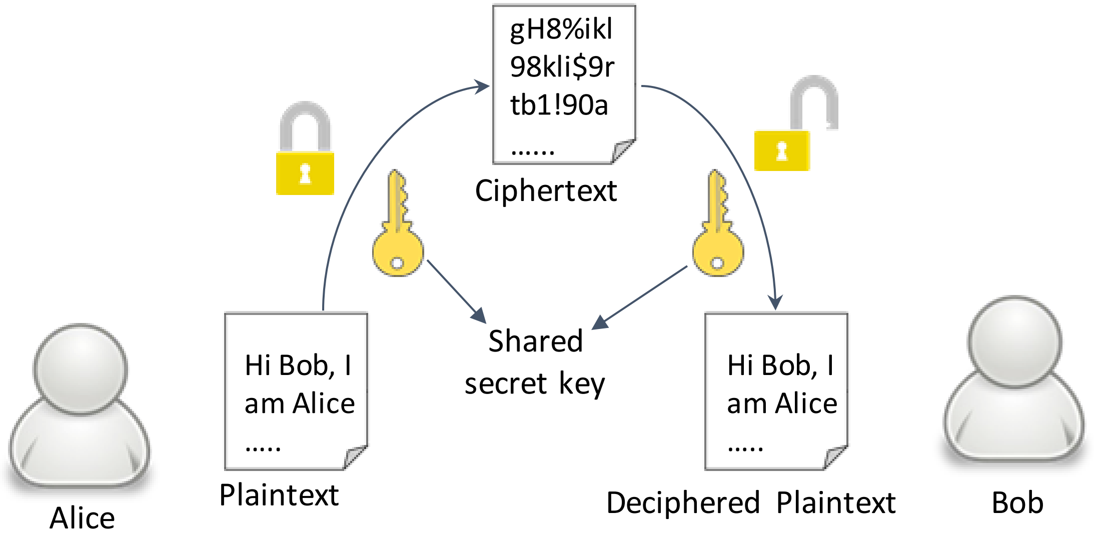
IIT Kanpur

Teaching assistants

- **Sumit Lahiri** (sumitl@cse.iitk.ac.in)
- **Chavan Sujeet** (sujeetc@cse.iitk.ac.in)
- **Indranil Thakur** (indra@cse.iitk.ac.in)

Brief introduction

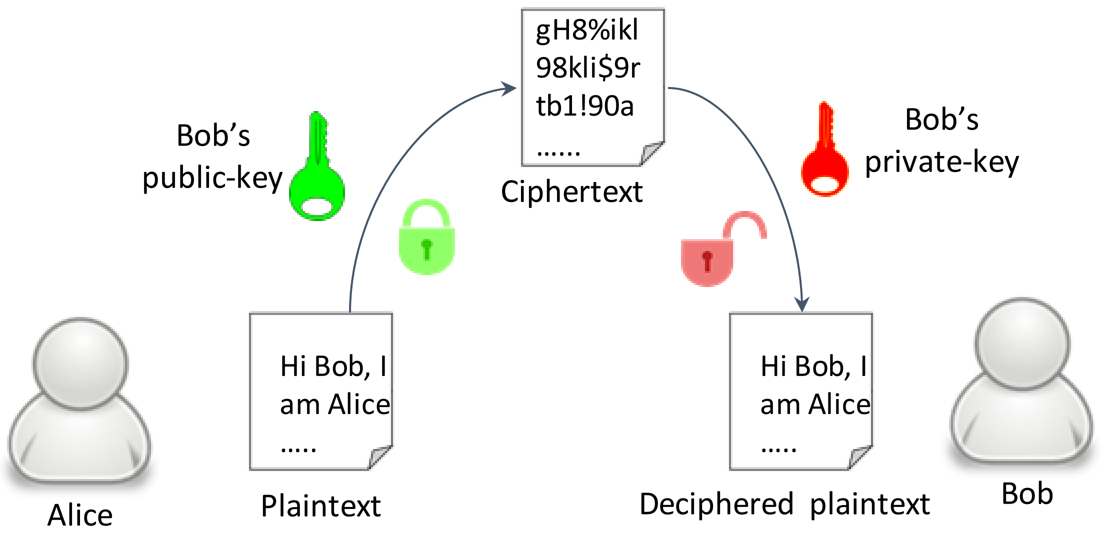
Symmetric-key cryptography



- Fast and lightweight
- Large encryption payload
- Example : Block cipher (AES), Stream cipher (chacha), Hash functions (SHA-1-3)
- Problem?
- Key-establishment

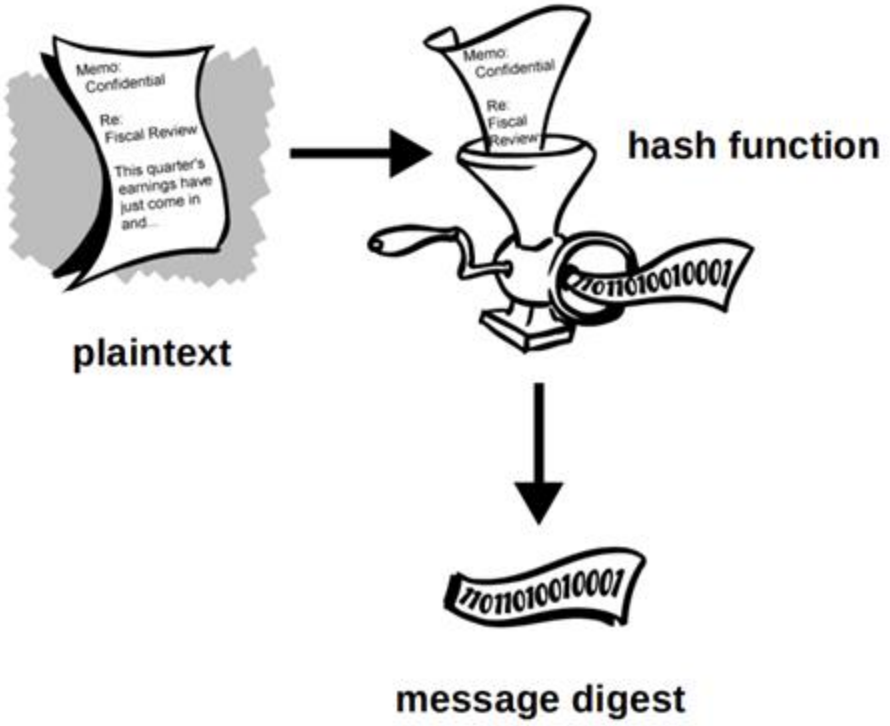
Brief introduction

Asymmetric-key cryptography



- Slower
- Heavyweight--> Complex operations
- No need for previously agreed keys
- Small encrypted payload
- Key-encapsulation mechanism (RSA, Diffie-Hellman)
- Digital Signature (ECDSA, RSA signature), etc.

Hash Functions



This Photo is taken from An Introduction to Cryptography, version 8.0.

$$h : M \rightarrow \{0,1\}^n$$

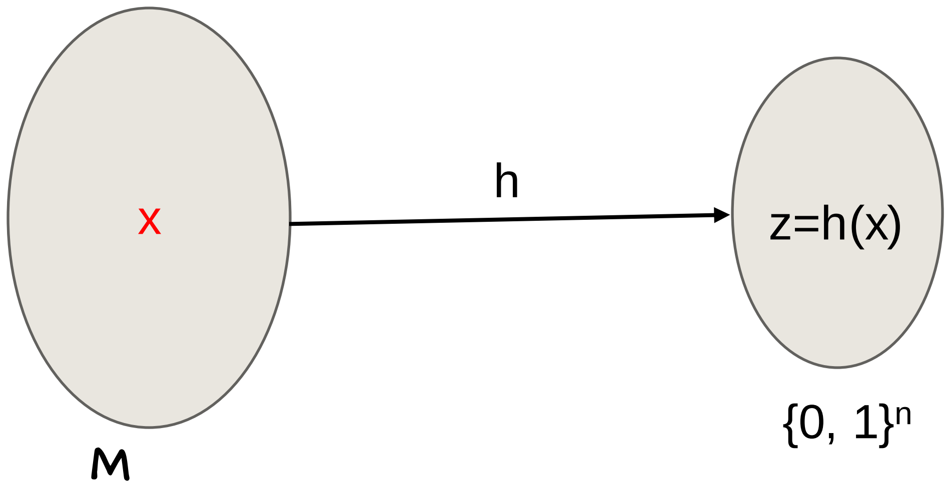
- Easy to compute
- Hard to invert

Properties of a *cryptographically secure* hash function

- Pre-Image Resistance
- Second Pre-Image Resistance
- Collision Resistance
- Typical value of n is 256 or 512

PRE-IMAGE Resistance

Let $h : \mathcal{M} \to \{0,1\}^n$ be a hash function.



- **Pre-Image Resistance :** Given h and value z, it is computationally hard to find x such that h(x) = z.

EXAMPLES

Let $h : \{0,1\}^* \to \{0,1\}^n$ be a hash function defined by

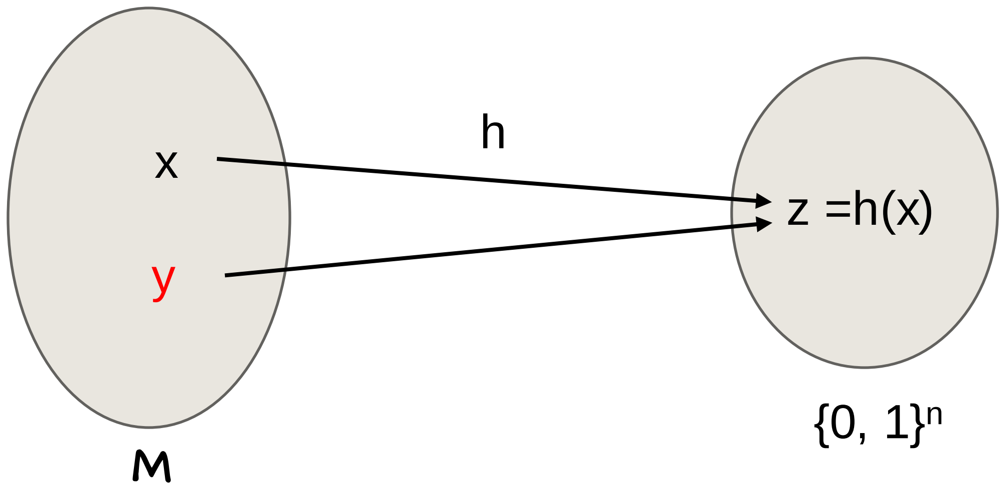
$$h(x) = x \bmod 2^n$$

Is this function h pre-image resistant?

- No, this function is not pre-image resistant.
- For any y in $\{0, 1\}^n$, y is a pre-image of itself.
- Also, the other pre-images are $y + k.2^n$, where k is any integer.

SECOND PRE-IMAGE RESISTANCE

Let $h : \mathcal{M} \to \{0,1\}^n$ be a hash function.



- **Second Pre-Image Resistance :** Given h and value x, it is computationally hard to find $y \neq x$ such that $h(y) = h(x)$.

EXAMPLES

Let $h : \{0,1\}^* \to \{0,1\}^n$ be a hash function defined by

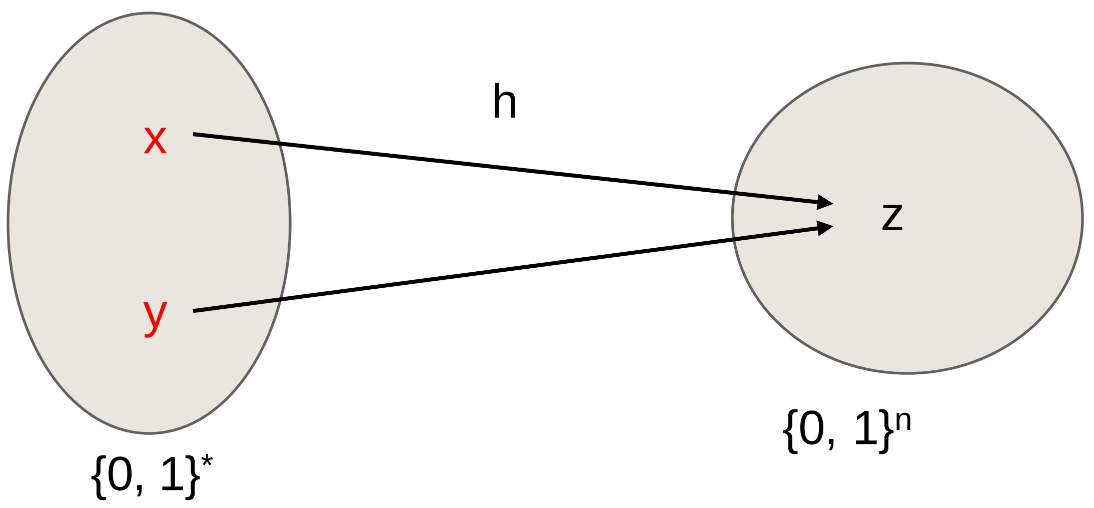
$$h(x) = x \bmod 2^n$$

Is this function h second pre-image resistant?

- No, this function is not second pre-image resistant.
- For any x in {0, 1}*, we have to find y (≠ x) in {0, 1}* such that h(y) = h(x).
- Take $y = x+k*2^n$ ($\neq x$), $k \neq 0$ but $h(x+2^n) = x+2^n \pmod{2^n} = x \pmod{2^n} = h(x)$
- $x+k*2^n$ is the second pre-image of h(x).

COLLISION RESISTANCE

Let $h : \{0,1\}^* \to \{0,1\}^n$ be a hash function.



- **Collision Resistance :** Given h, it is computationally hard to find any pair (x, y) with $y \neq x$ and satisfies h(y) = h(x) = z.

EXAMPLES

Let $g : \{0,1\}^{n+1} \to \{0,1\}^n$ be a hash function defined by

g(x) = [x/2]

Is this function g collision resistant?

- Take any even integer x from $\{0, 1\}^{n+1}$.
- Since x is even, therefore x = 2k, where k is an integer.
- g(x) = [x/2] = [2k/2] = k and g(x+1) = [(2k+1)/2] = [k+1/2] = k
- (x,x+1) is a collision pair.

Collision-resistance Implies SEC-PI-RESISTANCE

- Let h be a hash function which is collision resistance but not second pre-image resistant.

- There exists algorithm $\mathbb{A}$ for the hash function h such that any input value x, the algorithm $\mathbb{A}$ efficiently compute y such that $y \neq x$, and $h(y) = h(x)$ and returns this y as second pre-image of h(x).

- We take an input x' and call the algorithm $\mathbb{A}$ with this input value x'. Then the algorithm $\mathbb{A}$ returns y' such that $y' \neq x'$, and $h(y') = h(x')$

- So, here (x', y') is the collision pair for the hash function h.

- The hash function h is not collision resistant.
    - Therefore, not second pre-image resistance implies not collision resistance
- Contrapositively, we can say that collision resistance implies second pre-image resistance.

Collision Resistance Implies Pre-image Resistance

- It can also be proven that collision resistance implies pre-image resistance[1]
- It shows that collision resistance is much harder to achieve than the other two properties.
- Almost all attacks on hash functions try to break their collision resistance property.
  - E.g. : SHA2, SHA1, MD5, SNEFRU

[1]https://crypto.stackexchange.com/questions/71966/collision-resistant-hash-function-implies-one-way-function#:~:text=1%20Answer&text=Collision%2Dresistance%20implies%20one%2Dwayness,means%20%7B0%2C1%7D.

Game definitions

- An attacker is provided information and must find other information that meets certain criteria
- Preimage Resistance
  - Given: H(M)
  - Find: M
- Second Preimage Resistance
  - Given: H(M) and M
  - Find: M’ such that H(M) = H(M’) and M ≠ M’
- Collision Resistance
  - Given: nothing
  - Find: M and M´ such that H(M) = H(M’) and M ≠ M’

BIRTHDAY PARADOX PROBLEM

✓ Suppose there are some students in a class. The problem is to find a pair of students having the same birthday. How many students are required to ensure the existence of that pair?

- **Pigeonhole Principle :** Suppose 20 many pigeons are put in 19 pigeonholes. Then there exists one pigeonhole which contains two or more pigeons.

- Similarly, if n many pigeons are put in n-1 pigeonholes. Then there exists one pigeonhole which contains two or more pigeons.

- From **Pigeonhole principle** we can say that if there are 366 many students, then there exists at least one pair of students having the same birthday.

BIRTHDAY PARADOX PROBLEM

✓ Suppose there are some students in a class. What is the minimum number of students such that there exist at least one pair of students having the same birthday with probability at least 1/2.

Assumption:

- Number of students : N
- One year = 365 days
- Everyone has the equal chance of being born on any day

BIRTHDAY PARADOX PROBLEM

✓ Suppose N = 2, then what is the probability of these students having the same birthday?

$X_1 = \text{First student's birthday}$

$$
\begin{aligned}
\Pr[\text{Two students have the different birthday}] &= \Pr[\text{Second student's different birthday} | X_1].\Pr[X_1] \\
&= \frac{364}{365} \times \frac{365}{365} \\
\Pr[\text{Two students having the same birthday}] &= 1 - \Pr[\text{Two students having the different birthday}] \\
&= 1 - \frac{364}{365} \times \frac{365}{365} \\
&= 0.0027
\end{aligned}
$$

BIRTHDAY PARADOX PROBLEM

$X_1 = \text{First student's birthday}$

$X_2 = \text{Second student's birthday which is different from} X_1 | X_1$

$X_3 = \text{Third student's birthday which is different from} X_1, X_2 | X_1, X_2$

$X_i = i^{th} \text{student's birthday which is different from} X_1, X_2, \ldots, X_{i-1} | X_1, X_2, \ldots, X_{i-1}$

$$\Pr[X_1] = \frac{365}{365}$$

$$\Pr[X_2] = \frac{365}{365} \times \frac{(365-1)}{365}$$

$$\Pr[X_i] = \frac{365}{365} \times \frac{(365-1)}{365} \times \frac{(365-2)}{365} \ldots \frac{(365-i+1)}{365}$$

BIRTHDAY PARADOX PROBLEM

$$\begin{aligned}\Pr[\text{All students have the different birthday}] &= \Pr[X_1]\Pr[X_2]\ldots\Pr[X_N] \\ &= \frac{365}{365}\frac{(365-1)}{365}\frac{(365-2)}{365}\cdots\frac{(365-N+1)}{365}\end{aligned}$$

$$\begin{aligned}\Pr[\text{A pair of students have the same birthday}] &= 1 - \Pr[\text{All students have the different birthday}] \\ &= 1 - \frac{365}{365}\frac{(365-1)}{365}\frac{(365-2)}{365}\cdots\frac{(365-N+1)}{365} \\ &= 1 - \frac{365!}{(365-N)! \times 365^N}\end{aligned}$$

$$\text{For N} \sim 23, \Pr[\text{A pair of students have the same birthday}] \sim \frac{1}{2}$$

COLLISION PROBABILITY

Let $h : \mathcal{M} \to \{0,1\}^n$ be a hash function, where $\mathcal{M}$ is message set.

- The total number of possible hash values is 2^n and we assume $|\mathcal{M}| >> 2^n$.
- Atleast, two massage mapped into one hash value, by Pigeonhole principle.

What is the minimum number of chosen messages from $\mathcal{M}$ such that there exists at least one pair of messages having the same hash value with a probability atleast ½ ?

COLLISION PROBABILITY

$X_1 = \text{First message's hash value}$

$X_2 = \text{Second message's hash value which is different from} \, X_1$

$X_3 = \text{Third message's hash value which is different from} \, X_1, X_2$

$X_i = i^{th} \text{ message's hash value which is different from} \, X_1, X_2, X_{i-1}$

$$\Pr[X_1] = \frac{2^n}{2^n}$$

$$\Pr[X_2] = \frac{2^n}{2^n} \times \frac{(2^n - 1)}{2^n}$$

$$\Pr[X_i] = \frac{2^n}{2^n} \times \frac{(2^n - 1)}{2^n} \dots \frac{(2^n - k + 1)}{2^n}$$

COLLISION PROBABILITY

- We randomly sample k many messages from $\mathcal{M}$.

$$\begin{aligned}
\Pr[\text{All k messages have the different hash values}] &= \Pr[X_1]\Pr[X_2]\dots\Pr[X_k] \\
&= \frac{2^n}{2^n}\frac{(2^n-1)}{2^n}\frac{(2^n-2)}{2^n}\cdots\frac{(2^n-k+1)}{2^n}
\end{aligned}$$

$$\begin{aligned}
\Pr[\text{A pair of messages have the same hash values}] &= 1-\Pr[\text{All k messages have the different hash values}] \\
&= 1-\frac{2^n}{2^n}\frac{(2^n-1)}{2^n}\frac{(2^n-2)}{2^n}\cdots\frac{(2^n-k+1)}{2^n} \\
&= 1-\frac{2^n!}{(2^n-k)!\times(2^n)^k}
\end{aligned}$$

- For k ~ $2^{n/2}$, the probability of getting collision is ½.

MERKLE-DAMGARD CONSTRUCTION

- A construction method of a collision resistant hash function from a one-way compression function.

- Currently used hash functions SHA1, SHA2, MD5 follows this construction.

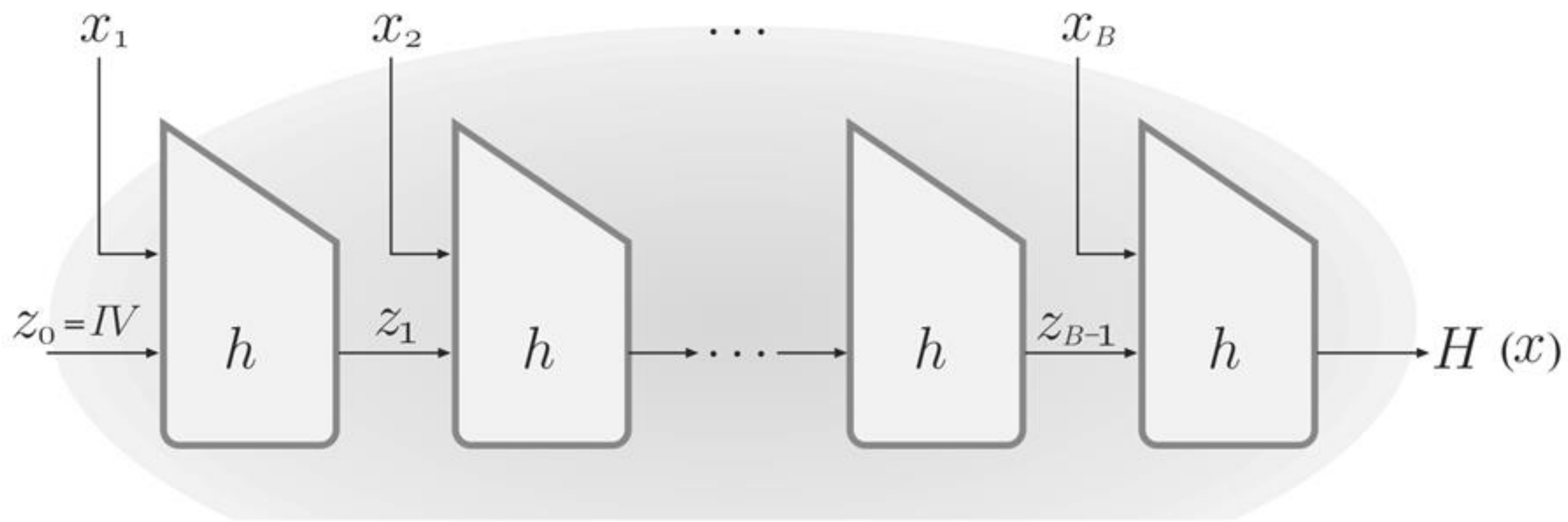
- Let h be a one-way compression function which takes a input string of length n+n' and outputs a string of length n, where n' ≥ n. Take r ≤ n' and a string IV of length n. Construct the collision-resistant hash function H as follows:

- Input a message m of length L where $L \leq 2^r$.

- We padded the string 100.... L with the message m, so that the resultant message m' = m100.... L will be multiple of n'.

- Split m' as a sequence of n' bit blocks $x_1x_2..x_B$.
- We apply many hash functions on this imput sequentially
- The i-th hash function takes as inputs x_i and the output of the previous hash function
  - The 0-th has function takes input a constant initialization vector (IV) instead of the output of previous hash function
- The output of the final hash function is the hash of the message

MERKLE-DAMGARD CONSTRUCTION



This function H is collision resistance

Puzzle friendliness of a Hash function

- Puzzle friendly hash functions
  - For every possible n-bit target output y
  - Value k chosen from a distribution with high min-entropy (well spread out)
  - Find x, such that H(k||x)=y
  - In time *significantly* smaller than 2^n
  - Once it is easy to verify the solution
- SHA-3

Applications of hash function

- Data Integrity
    - Suppose an user is downloading a large file.
    - After downloading user can check the hash or message digest of the file and match with the message digest mentioned in the website
    - Easy to detect the error but not possible to fix
- Proof of ownership
    - Someone wrote a document with a patent idea.
    - The person hashes the document and post it into a public place e.g. a newspaper
    - This will prove that the person had this idea on or before the date of publication of the newspaper
- Pseudo-random number generator
    - Generating true random numbers is very costly
    - Hashes are often used to hash the true random numbers and generate many pseudo-random numbers
- Digital signatures
    - Hash the digest of a message instead of the whole message to save computational time on signing the whole message

Thank you !