# Collision Resistant Hash Functions

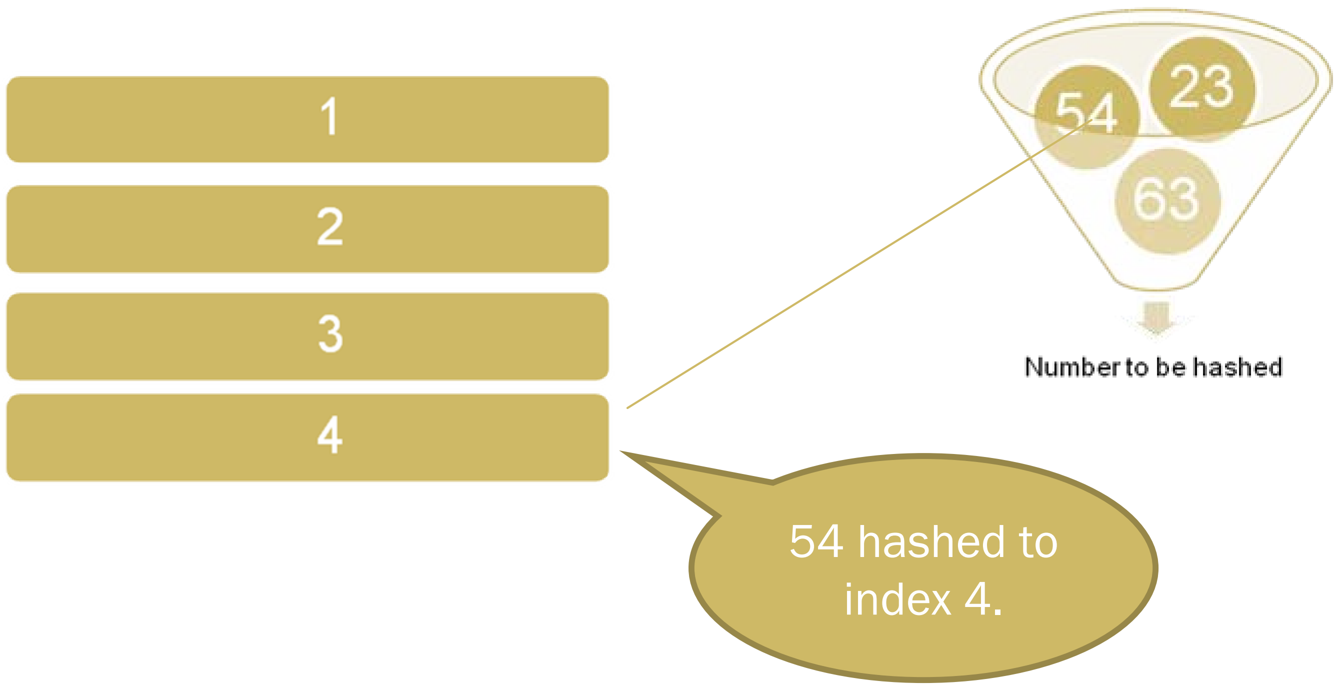
# Hash Functions

- Traditionally, hash functions take arbitrary length strings and compress them into shorter strings

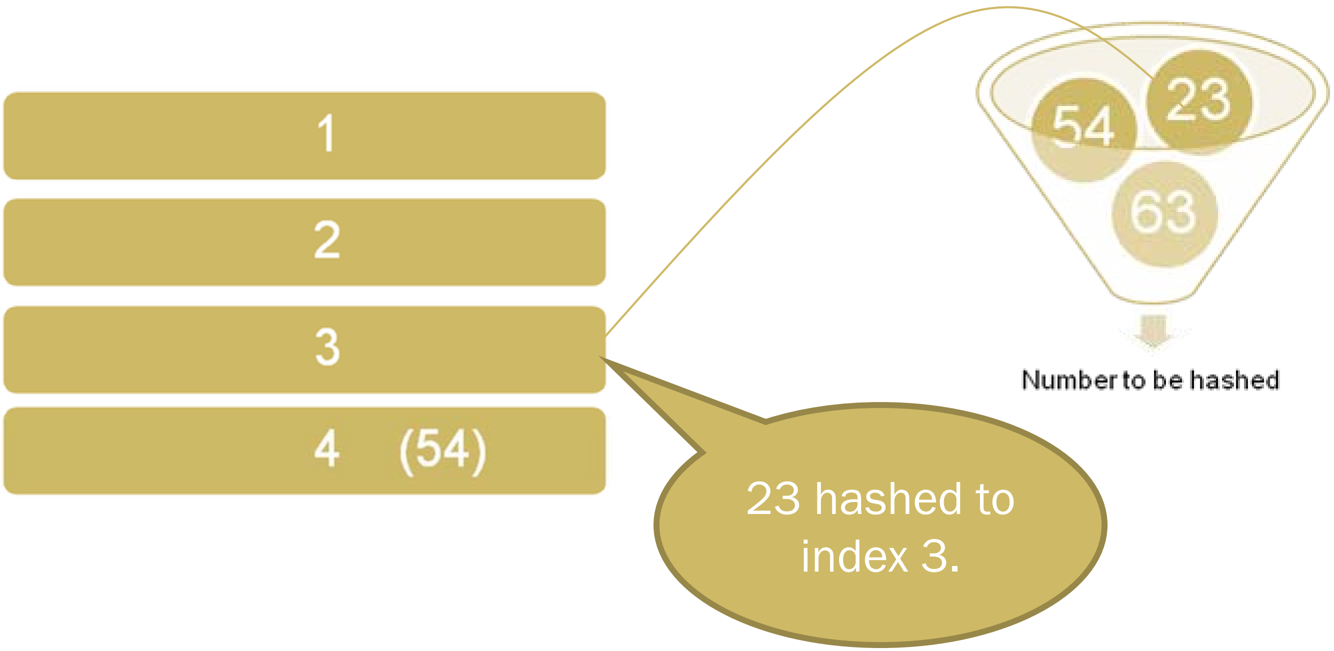
- Classically used in data structures for improved look-up times in storage/retrieval

- Collisions for the hash function H are distinct inputs x and y such that H(x) = H(y)

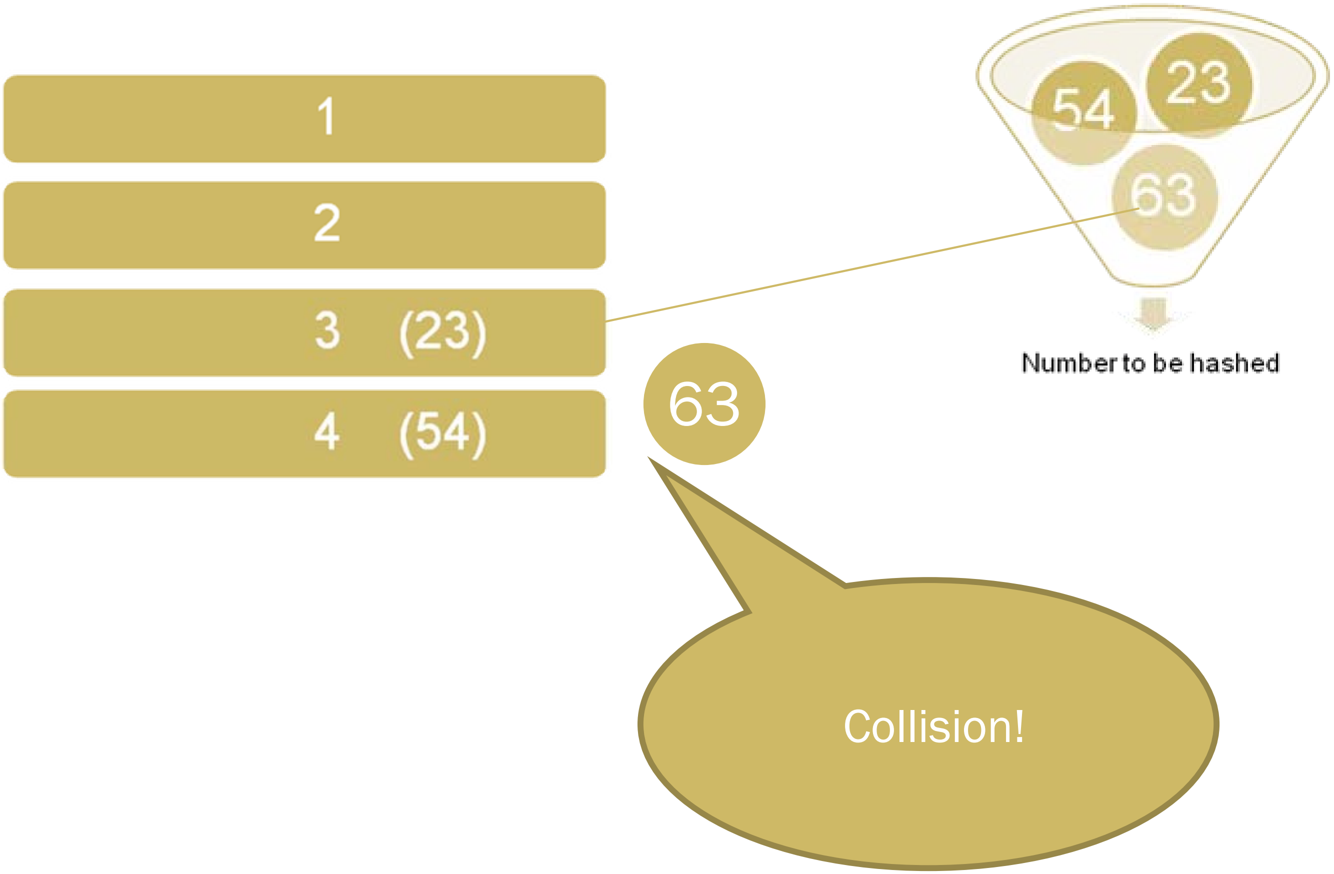
# An Example



# An Example



# An Example

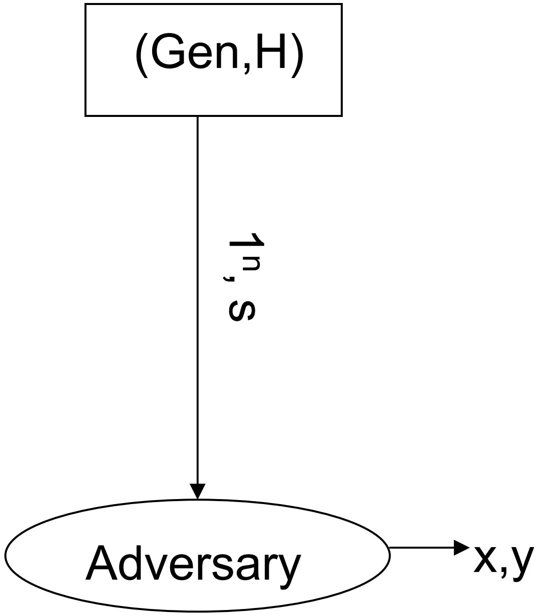


# Collision Resistance

- A function H is *collision resistant* if it is infeasible for any probabilistic polynomial-time algorithm to find a collision in H
- We deal with a family of functions indexed by s, $H^s(x) = H(s,x)$
- A **hash function** is a pair of algorithms (Gen, H) where $Gen(1^n)$ outputs the index s (for choosing $H^s$)
- If $H^s$ is defined only for inputs x of a certain length, we say it is a **fixed length** hash function

# Defining Collision Resistant Hash Function

Hash-game




The output of Hash-game is 1 if and only if $x \neq y$ and $H^s(x) = H^s(y)$

A hash function (Gen,H) is collision resistant if for all probabilistic polynomial time adversaries A there exists a negligible function negl such that

$$\Pr[\text{Output of Hash-game} = 1] \leq \text{negl}(n)$$

# Birthday Attack

We can find a collision on a hash function by hashing random numbers until these hashed values have a collision

# Relies on 'Birthday Paradox'

Theorem: If we pick independent random numbers in [1,2,...,N] with uniform distribution $\theta\sqrt{N}$ times, we get at least one number twice with probability tending to

$$(1-e^{-(\theta/2)*\theta})$$

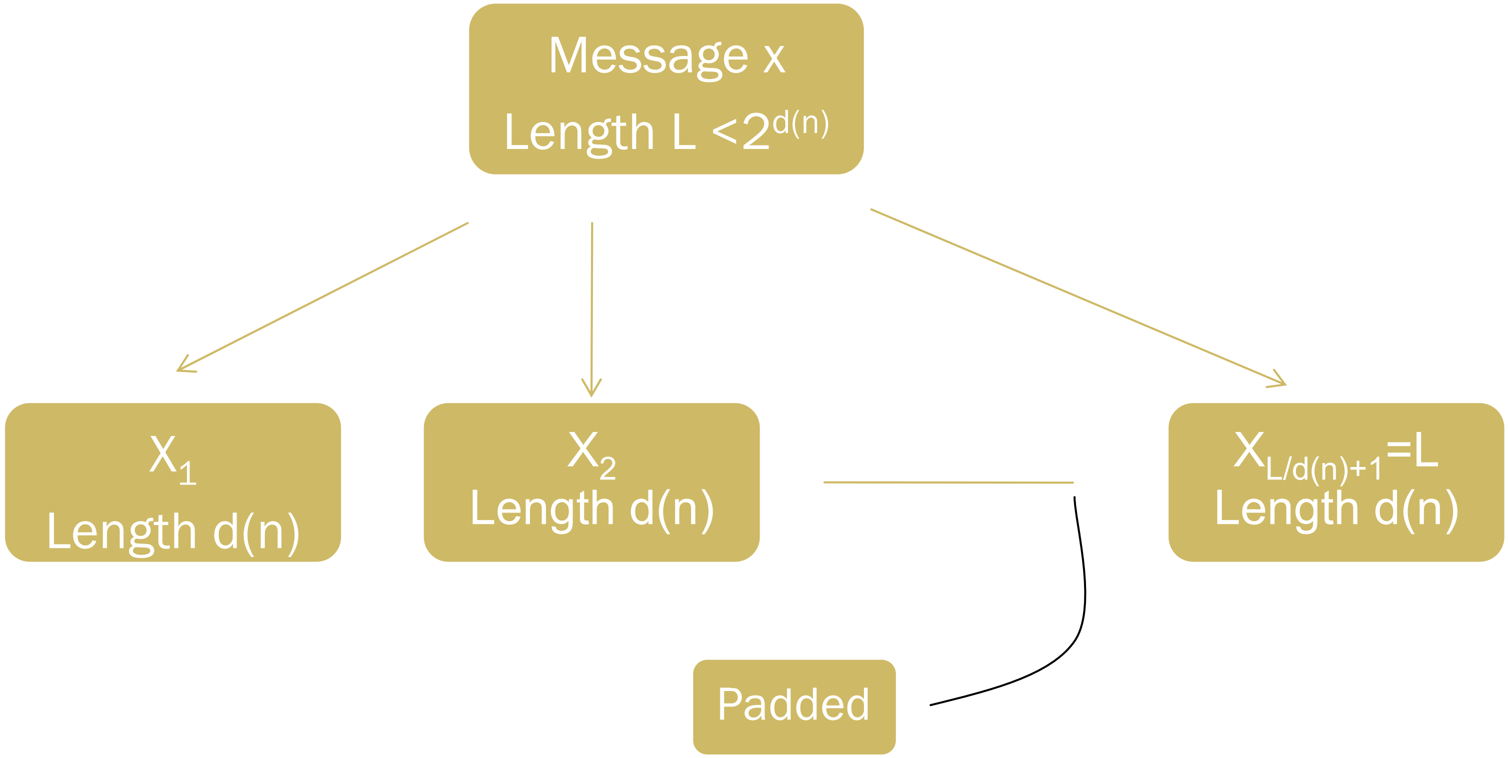
For $N = 365$ and $\theta = 1.2$ we get prob ~ 50%

Birthday Paradox: If there are 23 people,with probability at least ½ at least two of them have the same birthday

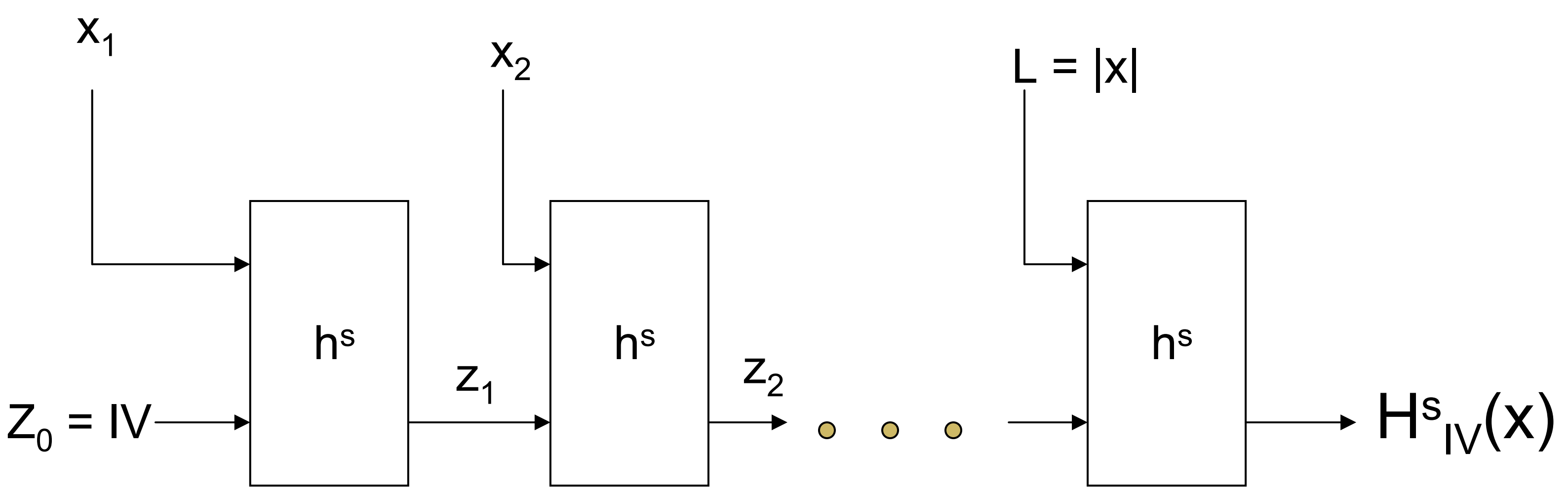
And for $\theta = 2.1$ we get prob ~ 90% !

# Merkle Damgard Transform

Constructing hash functions $H^s(x)$ from fixed length hash functions ($h^s$) with inputs of length 2d(n) and output length d(n)



# Merkle Damgard Transform




Theorem: If (Gen,h) is a fixed length collision resistant hash function, then (Gen, H) is a collision resistant hash function

# HMAC: A Message Authentication Code

HMAC is the current industry standard as CBC-MAC is deemed to be slow

(Gen,h): A fixed length hash function

(Gen,H): Hash function after applying MD transform to (Gen,h)

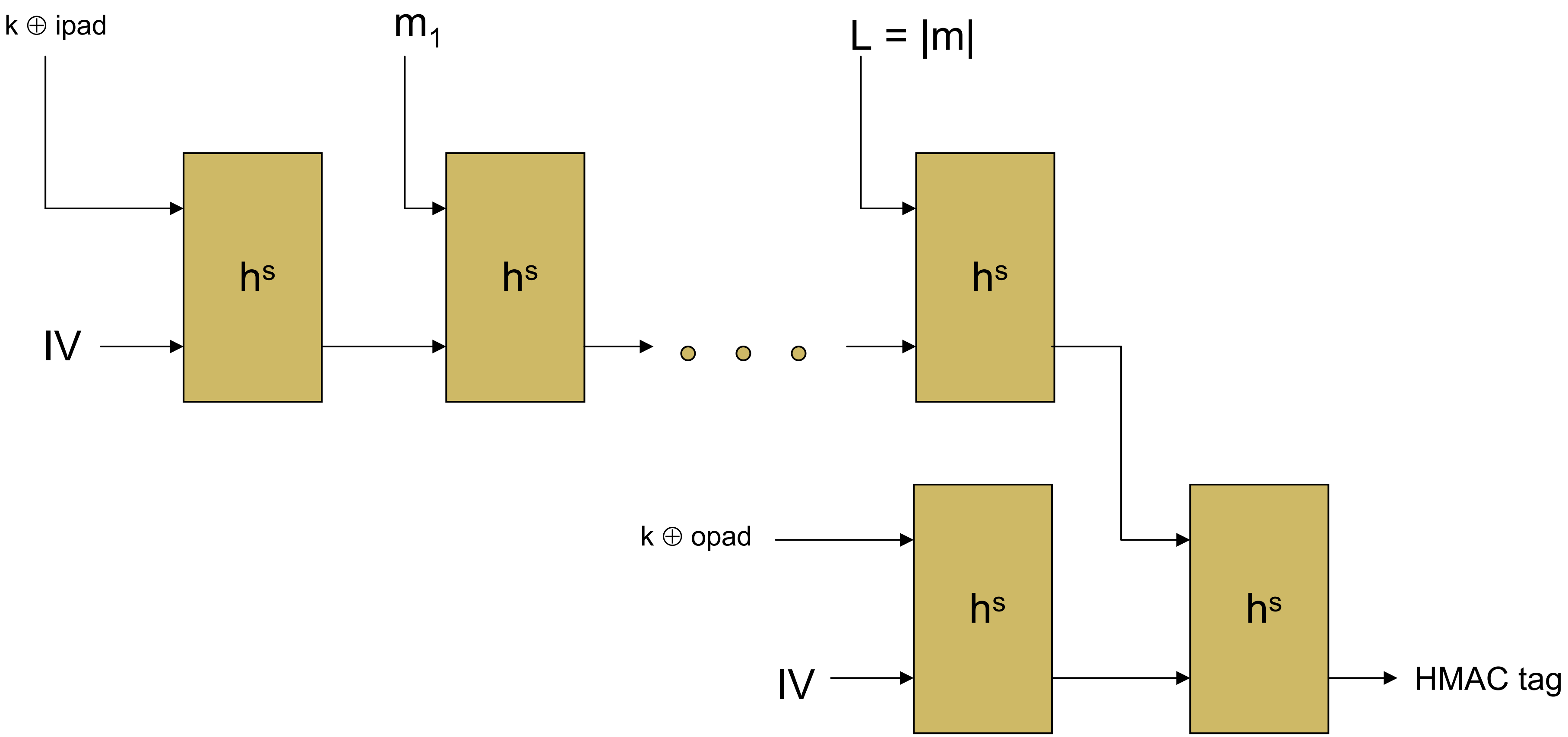
Fixed constants: *IV*, *opad* and *ipad*

$$\text{HMAC tag for } m = H^{s}_{IV}(\,(k \oplus \text{opad}) \,||\, H^{s}_{IV}(\,(k \oplus \text{ipad}) \,||\, m))$$

opad: 0x36 repeated as many times as needed

ipad: 0x5C repeated as many times as needed

# HMAC Construction



# Collision Resistant Hash Functions in Practice

- Popular practical constructions
  - MD5 (broken in 2004, should no longer be used)
  - SHA-1, SHA-2 (uses Merkle-Damgard transform) and others

- Theoretical Constructions
  - Based on hardness of the discrete logarithm problem

# A Fixed Length Hash Function

Let P be a polynomial time algorithm that on input $1^n$ outputs a cyclic group G of order q (length of q is n) and generator g

Gen: Run $P(1^n)$ obtain (G,q,g); select uniformly at random an element h from G; Output **s = (G,q,g,h)**

H: On input $x_1$ and $x_2$ (both in the range 0 to q-1), output

$$H^s(x_1,x_2) = g^{x_1} h^{x_2}$$

Theorem: If discrete logarithm problem is hard relative to P then the above is a fixed length collision resistant hash function