On the Security of Concatenating Hash Functions: Classical and New Results

Itai Dinur

Ben-Gurion University

Hash Functions

- A hash function $H: \{0,1\}^* \rightarrow \{0,1\}^n$ maps inputs of **arbitrary length** into outputs of **fixed** length n

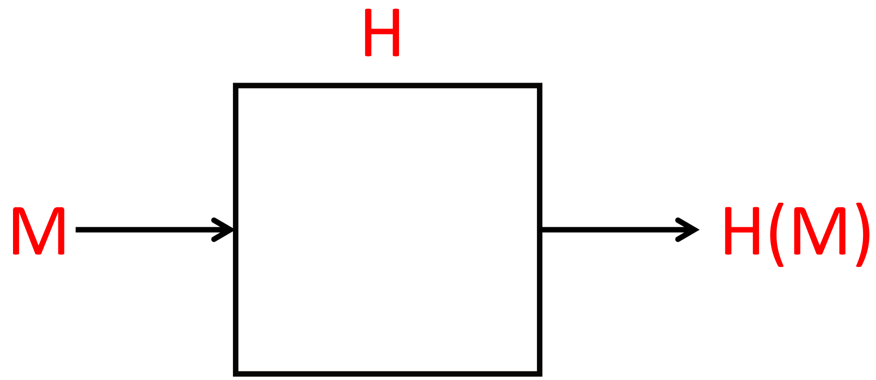
- Have many applications in data structures and algorithms

Cryptographic Hash Functions

- A **cryptographic hash function** is hash function with **stronger requirements**
- Have **many applications** (e.g., protocols, file integrity…)
- Requirements:
  - **Collision resistance**: It is hard to find M and M’ such that M≠M’ and H(M)=H(M’)
  - **Preimage resistance**: Given an arbitrary n-bit string Y, it is hard to find any M such that H(M)=Y
  - **Second preimage resistance**: Given an arbitrary input M, it is hard to find M≠M’ such that H(M)=H(M’)

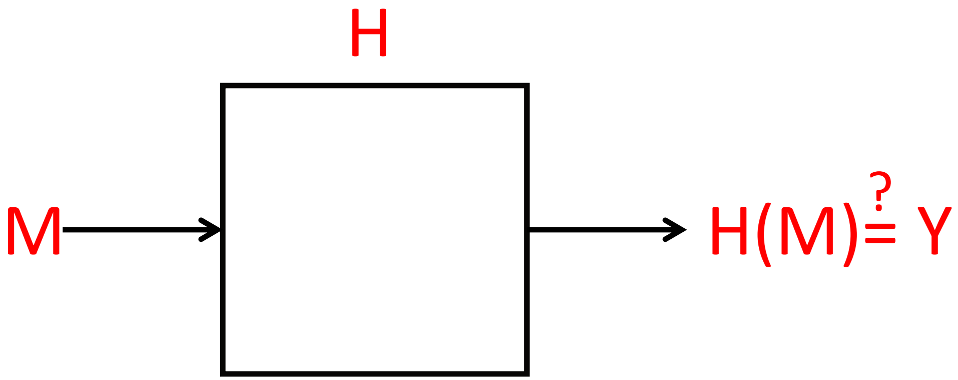
Ideal Hash Functions

- An **ideal hash function:** “random oracle” that **randomly** picks the output for a new query M (and **records** the answer for **consistency**)



Ideal Hash Functions

- **Preimage resistance**
  - **Brute force**: Given n-bit string Y, evaluate H(M) for arbitrary M until H(M)=Y
  - Complexity: 2^n



- **Second preimage resistance**: 2^n (brute force)

Ideal Hash Functions

- **Collision resistance:**
  - Evaluate H on arbitrary inputs $M_1, M_2, \ldots$ until $H(M_i) = H(M_j)$
  - Complexity: $2^{n/2}$ due to the **birthday paradox**

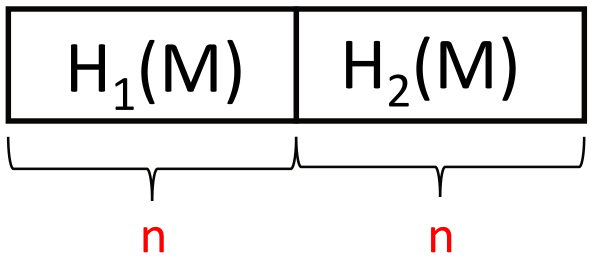
Hash Functions

	Collision Resistance	Preimage Resistance	Second Preimage Resistance
Ideal H	$2^{n/2}$	2^n	2^n

Concatenating Hash Functions

- Assume we have 2 hash function H_1 and H_2 of n bits
- We want a **stronger** construction
- Define a new hash function $H_1 \| H_2$

$$(H_1 \| H_2)(M) =$$



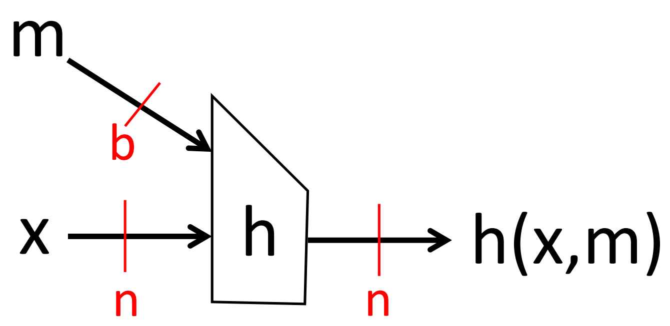
Hash Functions

	Collision Resistance	**Preimage Resistance**	**Second Preimage Resistance**
Ideal H	$2^{n/2}$	2^n	2^n
Ideal $H_1 \| H_2$	2^n	2^{2n}	2^{2n}

Hash Functions in Practice

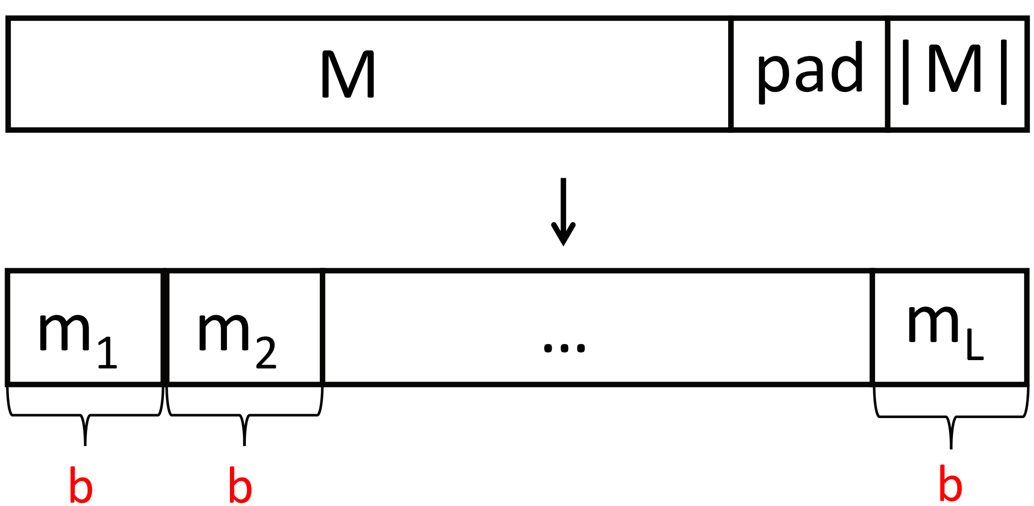
- Apply a **compression function** $h: \{0,1\}^n \times \{0,1\}^b \rightarrow \{0,1\}^n$ in an **iterated** way

- A **standard way** of building a hash function is the **Merkle-Damgård construction**
  - Used in SHA-1, SHA-2,…



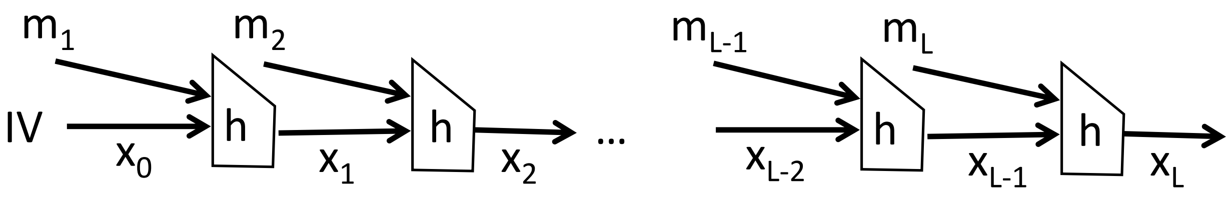
Iterated Hash Functions

- The Merkle-Damgård Construction:
  - 1) Pad the message M to a multiple of b (with 1, and as many 0's as needed and the **length of the message**)
  - 2) Divide the padded message into **blocks** $m_1 m_2 \ldots m_L$



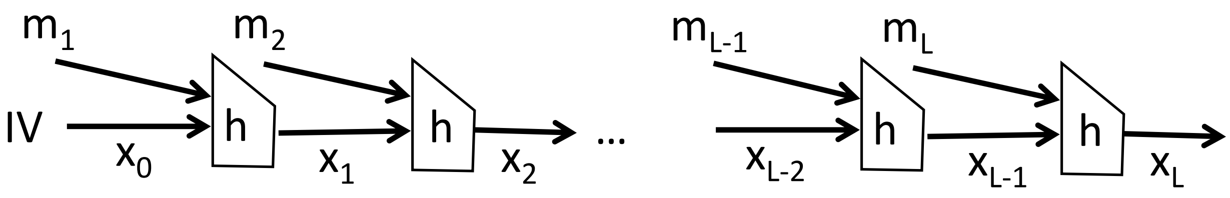
Iterated Hash Functions

- The Merkle-Damgård Construction:
  - 1) Pad the message M to a multiple of b (with 1, and as many 0's as needed and the **length of the message**)
  - 2) Divide the padded message into **blocks** $m_1 m_2 \ldots m_L$
  - 3) Set $x_0 = IV$. For $i=1$ to L, compute $x_i = h(x_{i-1}, m_i)$
  - 4) Output x_L



In This Talk

- Analyze the **security** of **Merkle-Damgård**
  - We assume that the compression function is **ideal** (acts as a **random oracle**)
- Focus on the **concatenation** of two Merkle-Damgård hash functions MD $H_1 \| H_2$
- Present some **classical** and **new** results on the security of this construction



Hash Functions (2003)

	Collision Resistance	Preimage Resistance	Second Preimage Resistance
Ideal H	$2^{n/2}$	2^n	2^n
MD H	$2^{n/2}$	2^n	2^n

Ideal $H_1 \| H_2$	2^n	2^{2n}	2^{2n}
MD $H_1 \| H_2$	2^n	2^{2n}	2^{2n}

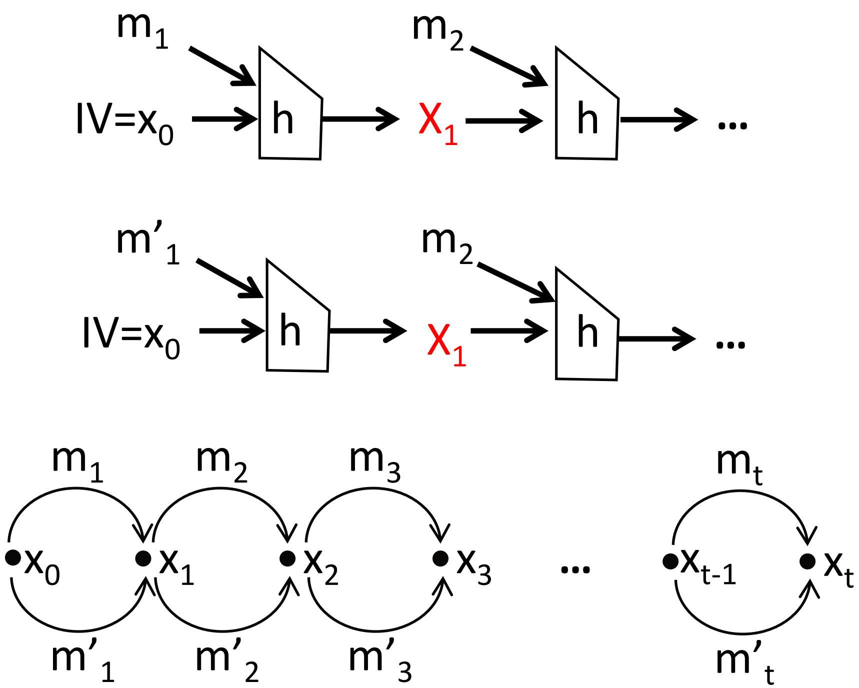
Hash Functions (Joux, 2004)

	Collision Resistance	Preimage Resistance	Second Preimage Resistance
Ideal H	$2^{n/2}$	2^n	2^n
MD H	$2^{n/2}$	2^n	2^n

Ideal $H_1 \| H_2$	2^n	2^{2n}	2^{2n}
MD $H_1 \| H_2$	~~2^n~~	~~2^{2n}~~	~~2^{2n}~~

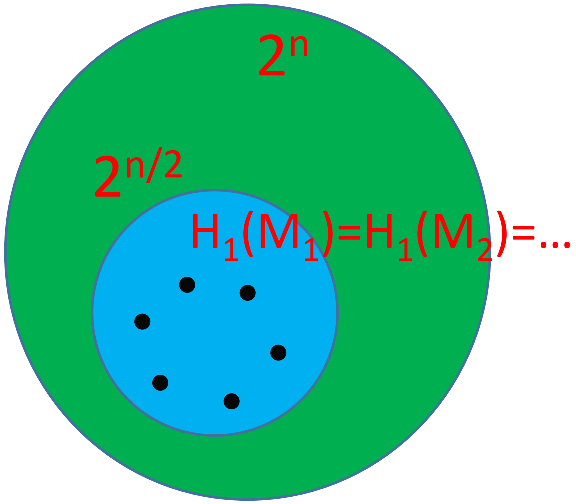
Joux Multicollisions

- Finding a collision in H requires $2^{n/2}$ work
- What about 2^t-multicollision $H(M_1)=H(M_2)= \ldots =H(M_{2^t})$?
- Can be computed in $t \cdot 2^{n/2}$ work
  - Much more efficiently than in a random oracle



Application to the Concatenated Hash

- A collision in $H_1 \| H_2$:
- Messages M and M' such that $H_1(M)=H_1(M')$ and $H_2(M)=H_2(M')$
- Assume that H_1 is iterated



- 1) Find $2^{n/2}$ multicollision in H_1
  - $H_1(M_1)=H_1(M_2)=\ldots$
- 2) Evaluate $H_2(M_1)$, $H_2(M_2)$,… and find $H_2(M_i)=H_2(M_j)$
  - Succeeds with high probability (**birthday paradox**)

- Complexity: about $n \cdot 2^{n/2}$

Hash Functions (2004)

	Collision Resistance	Preimage Resistance	Second Preimage Resistance
Ideal H	$2^{n/2}$	2^n	2^n
MD H	$2^{n/2}$	2^n	2^n

Ideal $H_1 \| H_2$	2^n	2^{2n}	2^{2n}
MD $H_1 \| H_2$	~~2^n~~ $\approx 2^{n/2}$	~~2^{2n}~~ $\approx 2^n$	~~2^{2n}~~ $\approx 2^n$

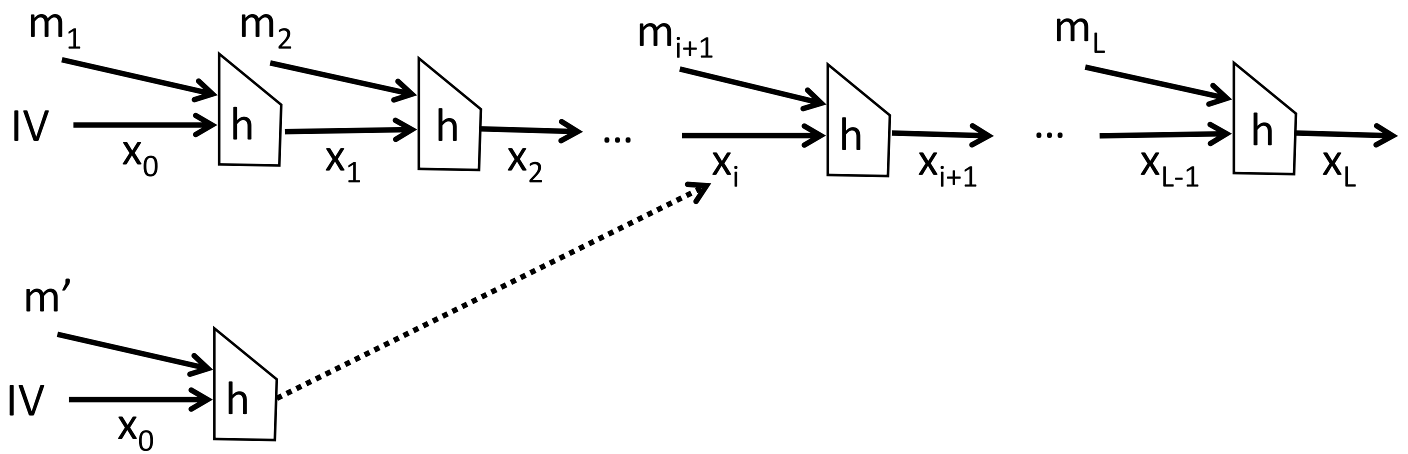
Hash Functions (Kelsey and Schneier, 2005)

	Collision Resistance	**Preimage Resistance**	**Second Preimage Resistance**
Ideal H	$2^{n/2}$	2^n	2^n
MD H	$2^{n/2}$	2^n	~~2^n~~

Ideal $H_1 \| H_2$	2^n	2^{2n}	2^{2n}
MD $H_1 \| H_2$	~~2^n~~ $\approx 2^{n/2}$	~~2^{2n}~~ $\approx 2^n$	~~2^{2n}~~ $\approx 2^n$

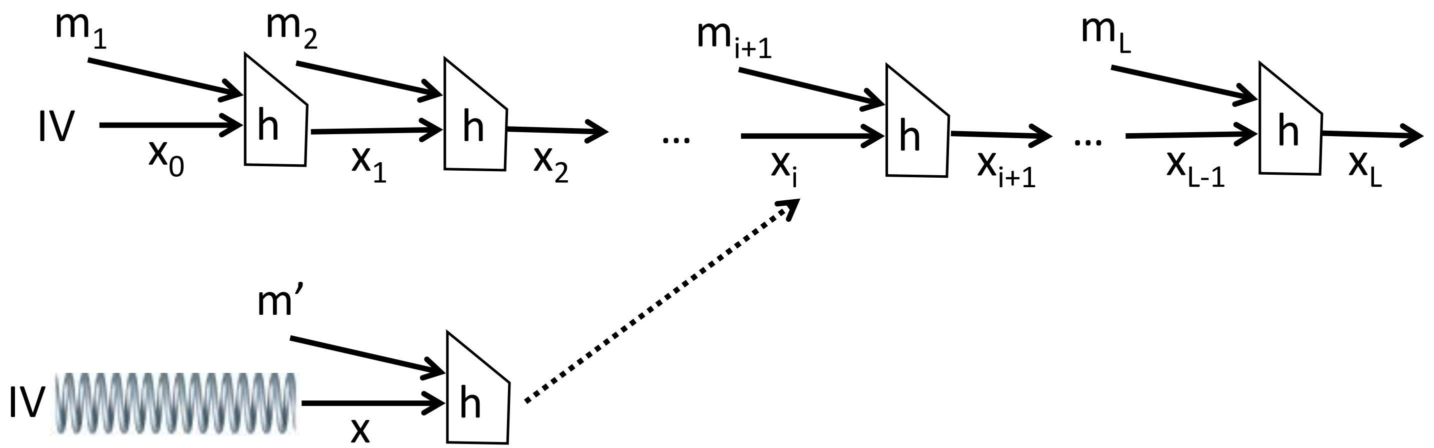
Second Preimage Attack on MD

- Given a (padded) message $M=m_1\|m_2\|\ldots\|m_L$
- We want to find M' such that $H(M')=H(M)$
- Start from IV and try different m' until $h(IV,m')=x_i$
  - Every trial succeeds with probability $L/2^n$
  - Succeeds after $2^n/L$ trials
- Output $m'\|m_{i+2}\|\ldots\|m_L$
- Problem: foiled by MD **message length padding**



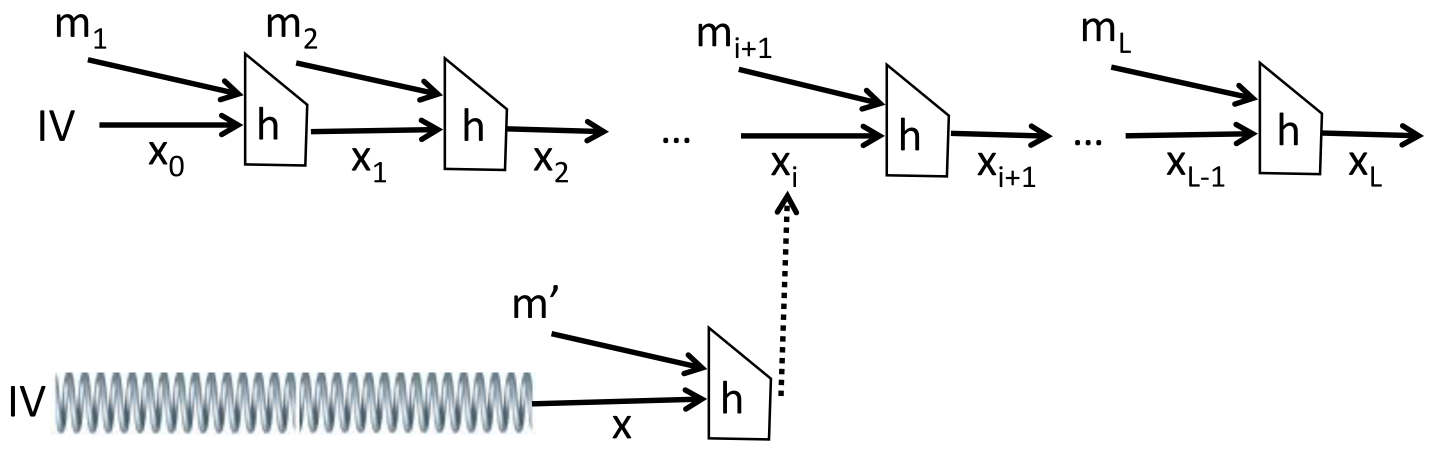
Second Preimage Attack on MD

- Solution of Kelsey and Schneier (2005):
- Build an **expandable message**
- Start from IV and try different m’ until h(x,m’)=x_i



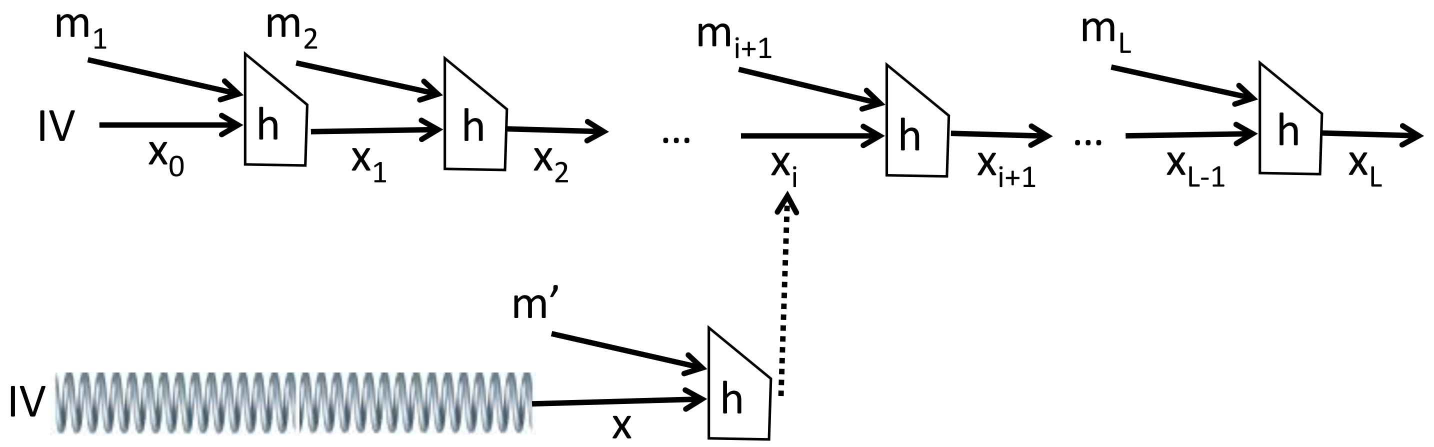
Second Preimage Attack on MD

- Solution of Kelsey and Schneier (2005):
- Build an **expandable message**
- Start from IV and try different m' until $h(x,m')=x_i$
- Select message of **appropriate length**



Second Preimage Attack on MD

- Attack complexity (for $L<2^{n/2}$):
- "Hitting" x_i: $2^n/L$
- Total complexity: $2^n/L$



Hash Functions (2005)

	Collision Resistance	**Preimage Resistance**	**Second Preimage Resistance**
Ideal H	$2^{n/2}$	2^n	2^n
MD H	$2^{n/2}$	2^n	~~2^n~~ $2^n/L$

Ideal $H_1 \| H_2$	2^n	2^{2n}	2^{2n}
MD $H_1 \| H_2$	~~2^n~~ $\approx 2^{n/2}$	~~2^{2n}~~ $\approx 2^n$	~~2^{2n}~~ $\approx 2^n$

Hash Functions (2015)

	Collision Resistance	**Preimage Resistance**	**Second Preimage Resistance**
Ideal H	$2^{n/2}$	2^n	2^n
MD H	$2^{n/2}$	2^n	~~2^n~~ $2^n/L$
Ideal $H_1 \Vert H_2$	2^n	2^{2n}	2^{2n}
MD $H_1 \Vert H_2$	~~2^n~~ $\approx 2^{n/2}$	~~2^{2n}~~ $\approx 2^n$	~~2^{2n}~~ ~~$\approx 2^n$~~ $<<2^n$

- MD $H_1 \Vert H_2$ is **weaker than** ideal H !

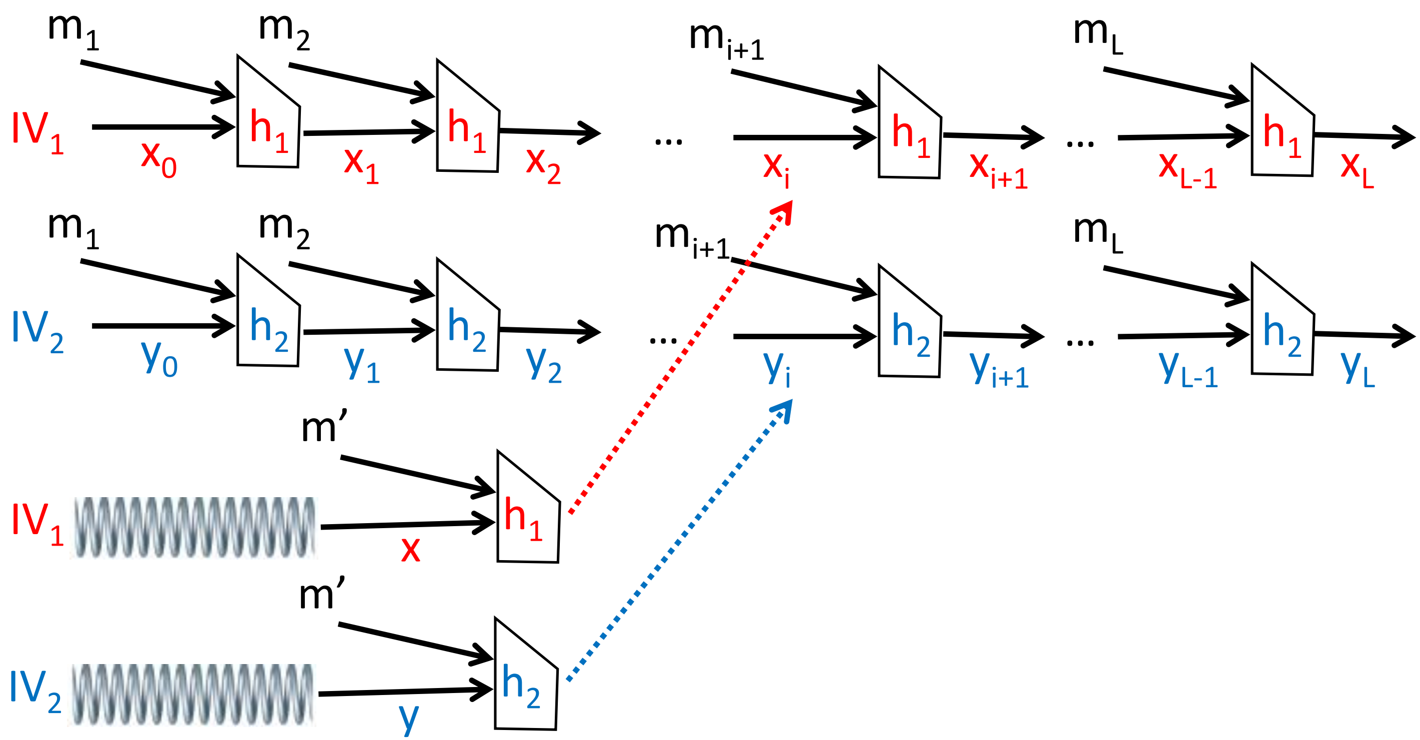
Second Preimage Attack on Concatenated MD

- A **second preimage** for $H_1\|H_2$:
- Given M, find M' such that $H_1(M')=H_1(M)$ and $H_2(M')=H_2(M)$

- We want an algorithm **more efficient** than 2^n

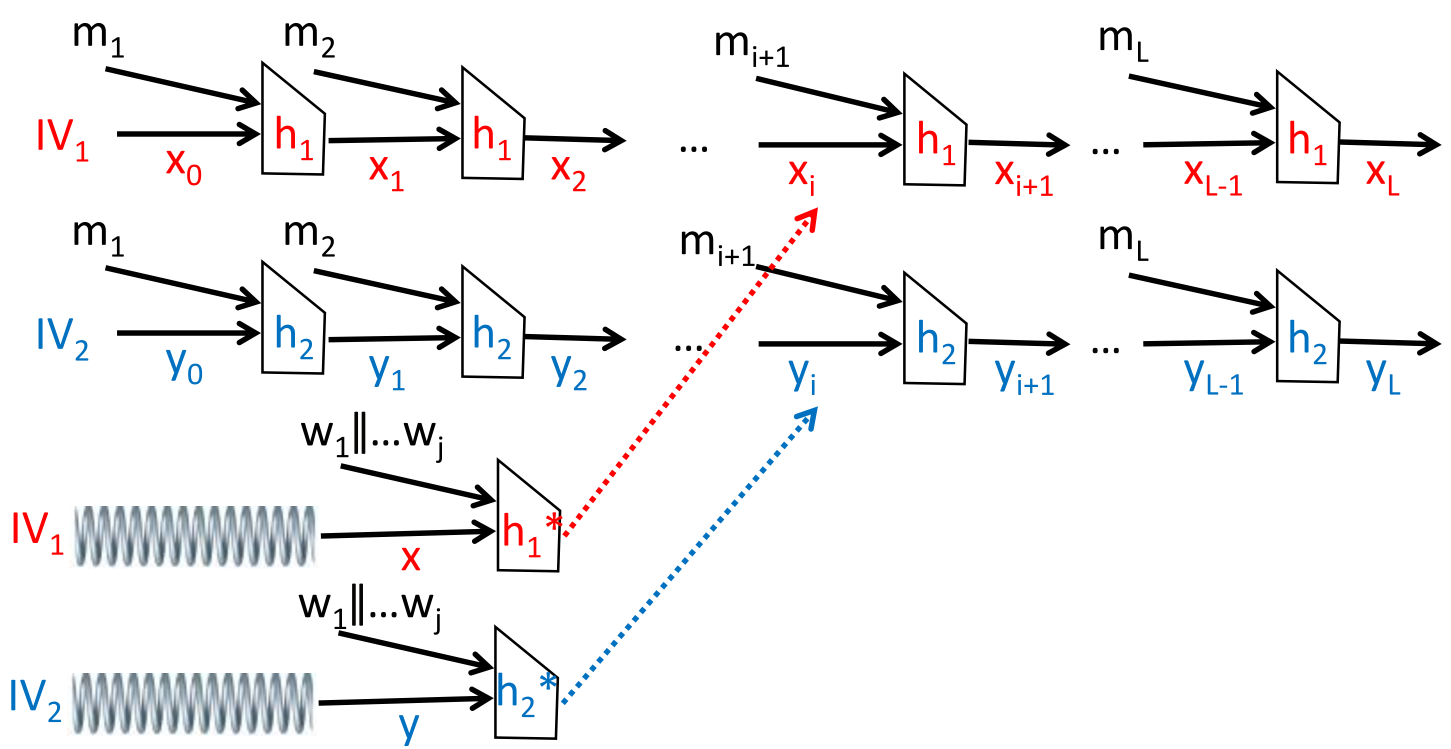
Second Preimage Attack on Concatenated MD

- Given a (padded) message $M=m_1\|m_2\|\ldots\|m_L$
- Require: $h_1(x,m')=x_i$ and $h_2(y,m')=y_i$
- Every trial succeeds with probability $L/2^{2n}$
- Attack succeeds after $2^{2n}/L > 2^n$ trials ($L<2^n$)
- Standard approach is **inefficient**



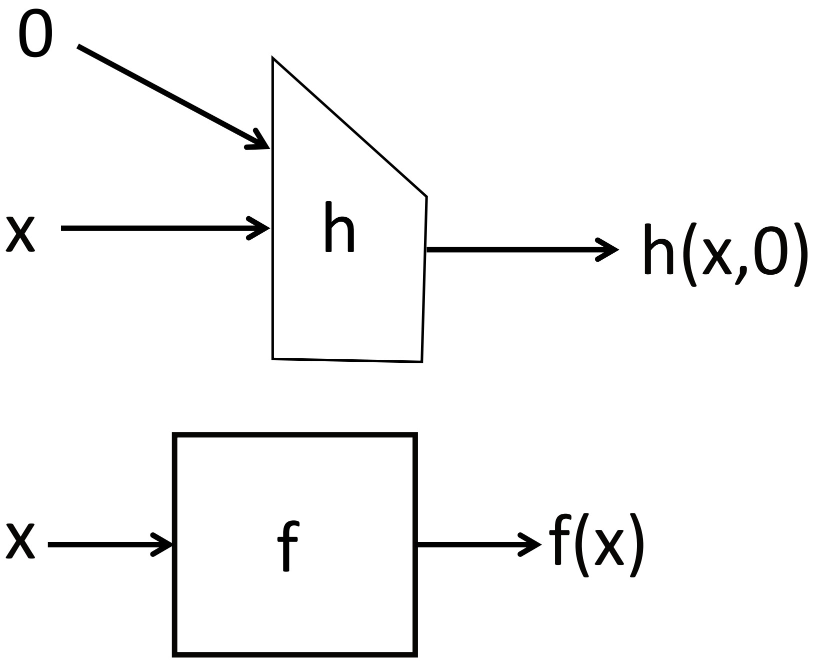
A Different Approach

- We will select a **single target** (x_i,y_i) that is **much easier to hit** with a **specially crafted message** $w_1\|\ldots\|w_j$
- Define: $h^*(x,w_1\|\ldots\|w_j)= h(\ldots h(h(x,w_1),w_2)\ldots)$
- Require: $h_1^*(x,w_1\|\ldots\|w_j)=x_i$ **and** $h_2^*(y,w_1\|\ldots\|w_j)=y_i$



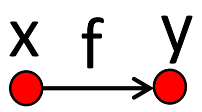
A Different Approach

- **Fix** to 0 the message block input to h
- Define f(x)=h(x,0)

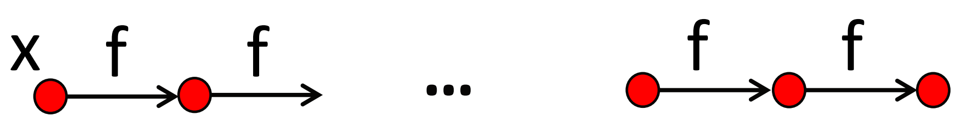


A Different Approach

- f(x) is a mapping from n bits to n bits
- Define a **graph**:
  - **Nodes** are the **states**
  - There is an **edge** from x to y if f(x)=y

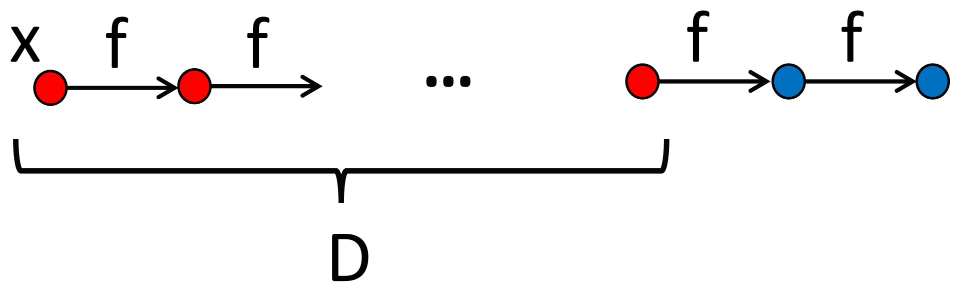


- f can be **iterated** f(…f(f(x))…)
- Interested in states obtained after **applying f many times**



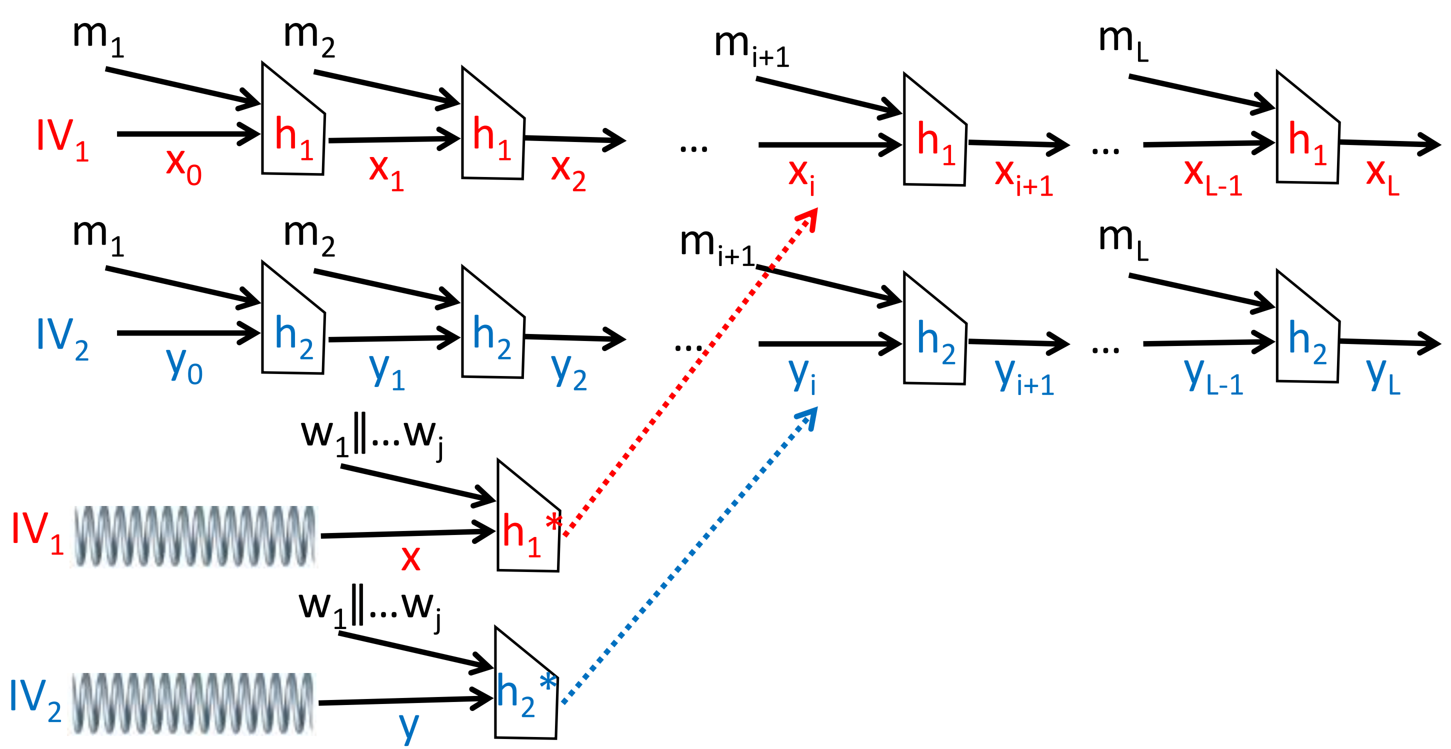
Deep Iterates

- Let $D \leq 2^{n/2}$ be a parameter
- Definition: A **deep iterate** is a node of **depth** (at least) D in the graph



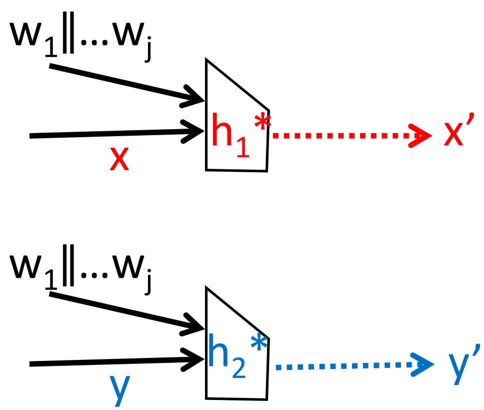
Second Preimage Attack on Concatenated MD

- Define $f_1(x)=h_1(x,0)$ and $f_2(y)=h_2(y,0)$
- Target: x_i **deep iterate** in f_1 and y_i **deep iterate** in f_2
- Require: $h_1^*(x,w_1\|\ldots\|w_j)=x_i$ **and** $h_2^*(y,w_1\|\ldots\|w_j)=y_i$



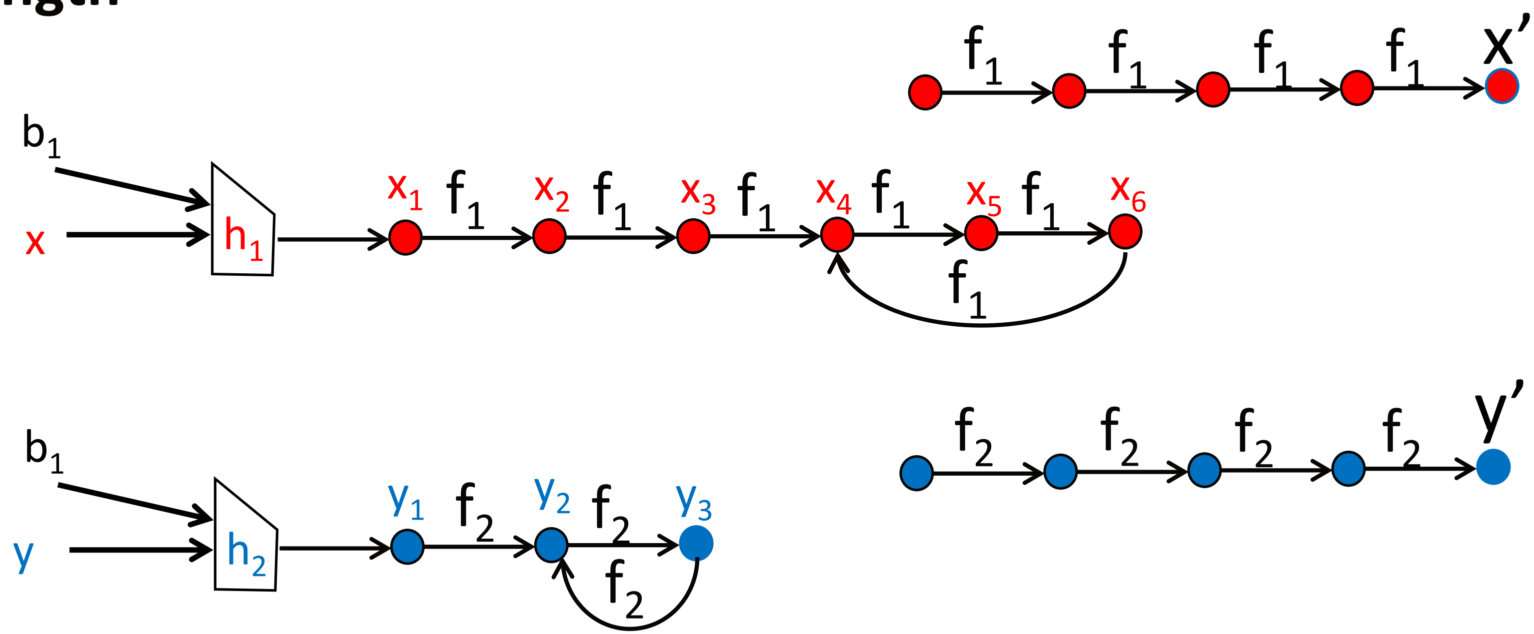
Deep Iterates

- Develop an **algorithm** that given **arbitrary states** x, y and a **deep iterates** x', y', finds $\mathbf{w_1,\ldots,w_j}$ such that $h_1^*(x, \mathbf{w_1\|\ldots\|w_j})=x'$ and $h^*(y, \mathbf{w_1\|\ldots\|w_j})=y'$ with less than 2^n work
  - For an **arbitrary** nodes x', y' this requires 2^{2n} work !

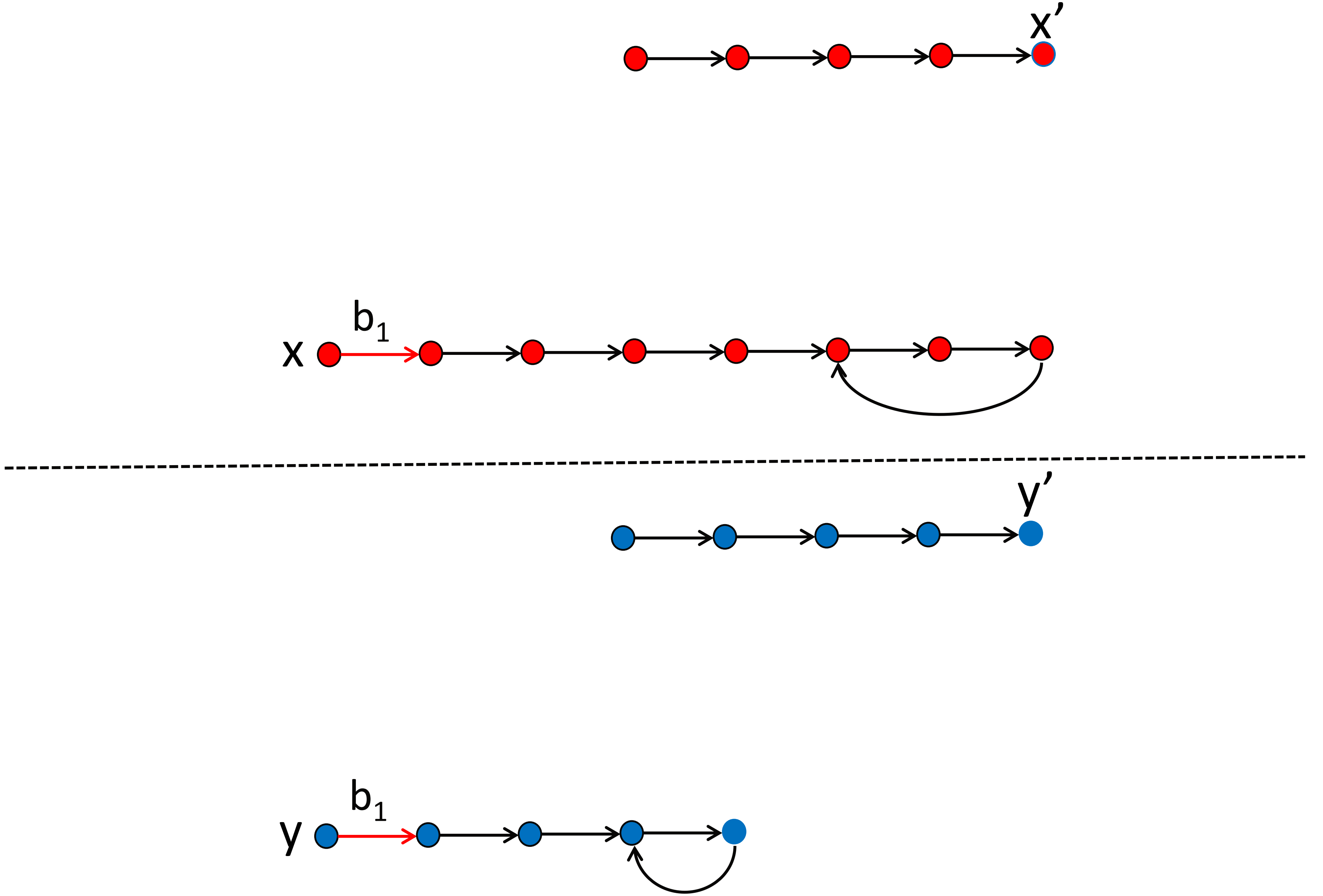


The Algorithm

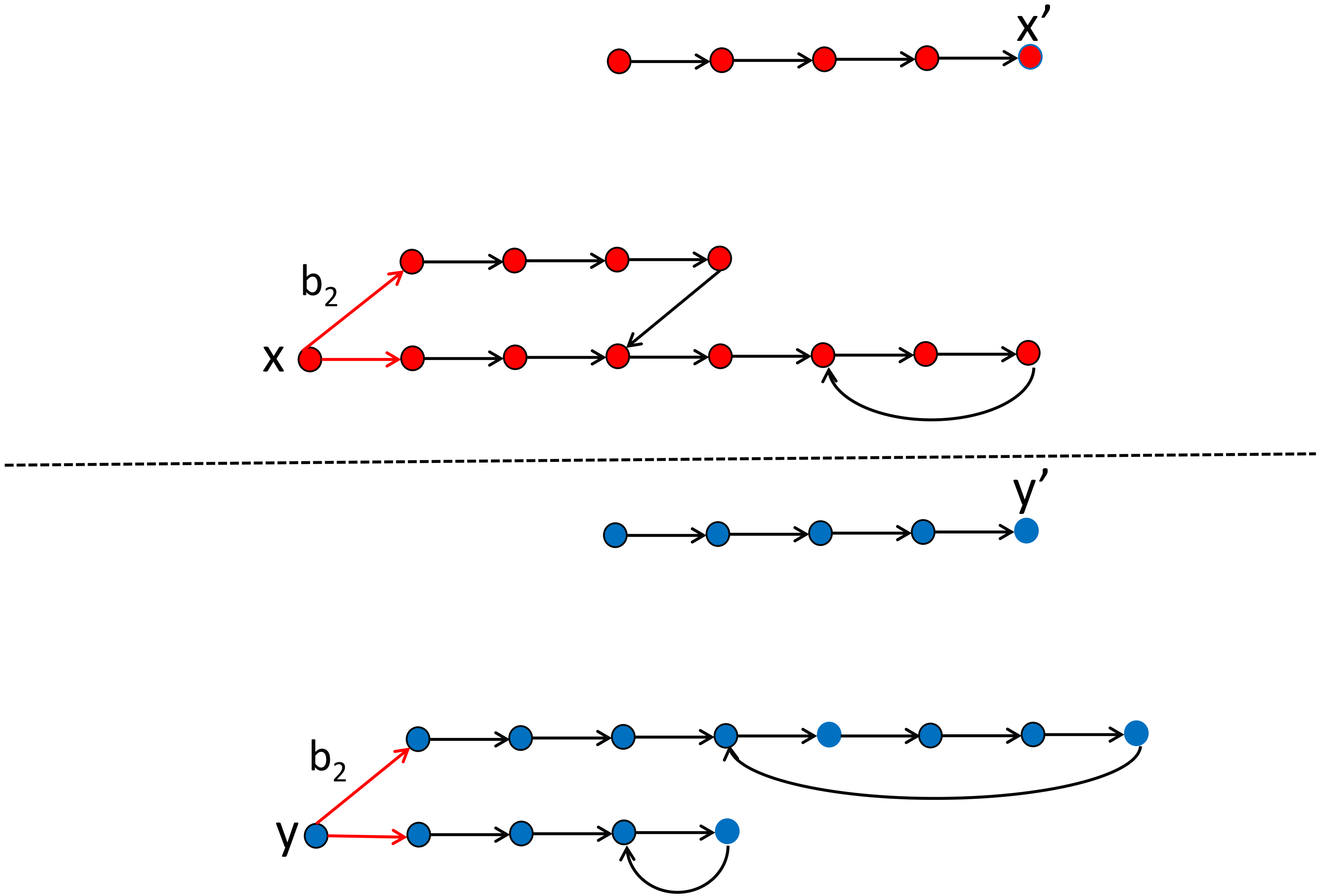
- Algorithm: for different $\mathbf{w_1}$ values, evaluate **messages** of the form $\mathbf{w_1||0...||0}$ from x and y
  - **Store** all encountered states
  - **Stop** on a **collision** with a **previous evaluated state** (look ahead)
- Repeat until success:
  - $h_1^*(x, \mathbf{w_1||0...||0})=x'$ and $h^*(y, \mathbf{w_1||0...||0})=y'$ with **same message length**



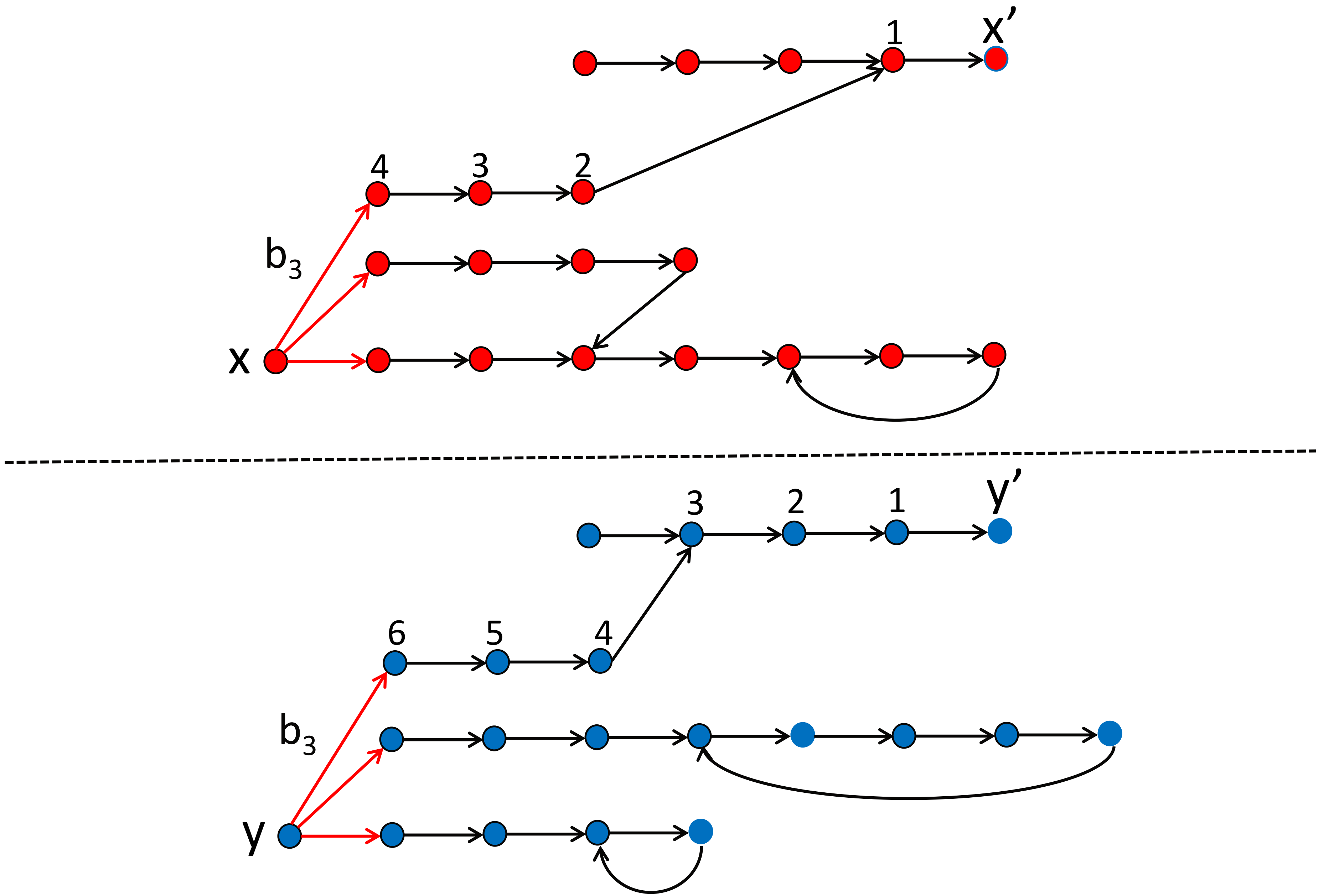
The Algorithm



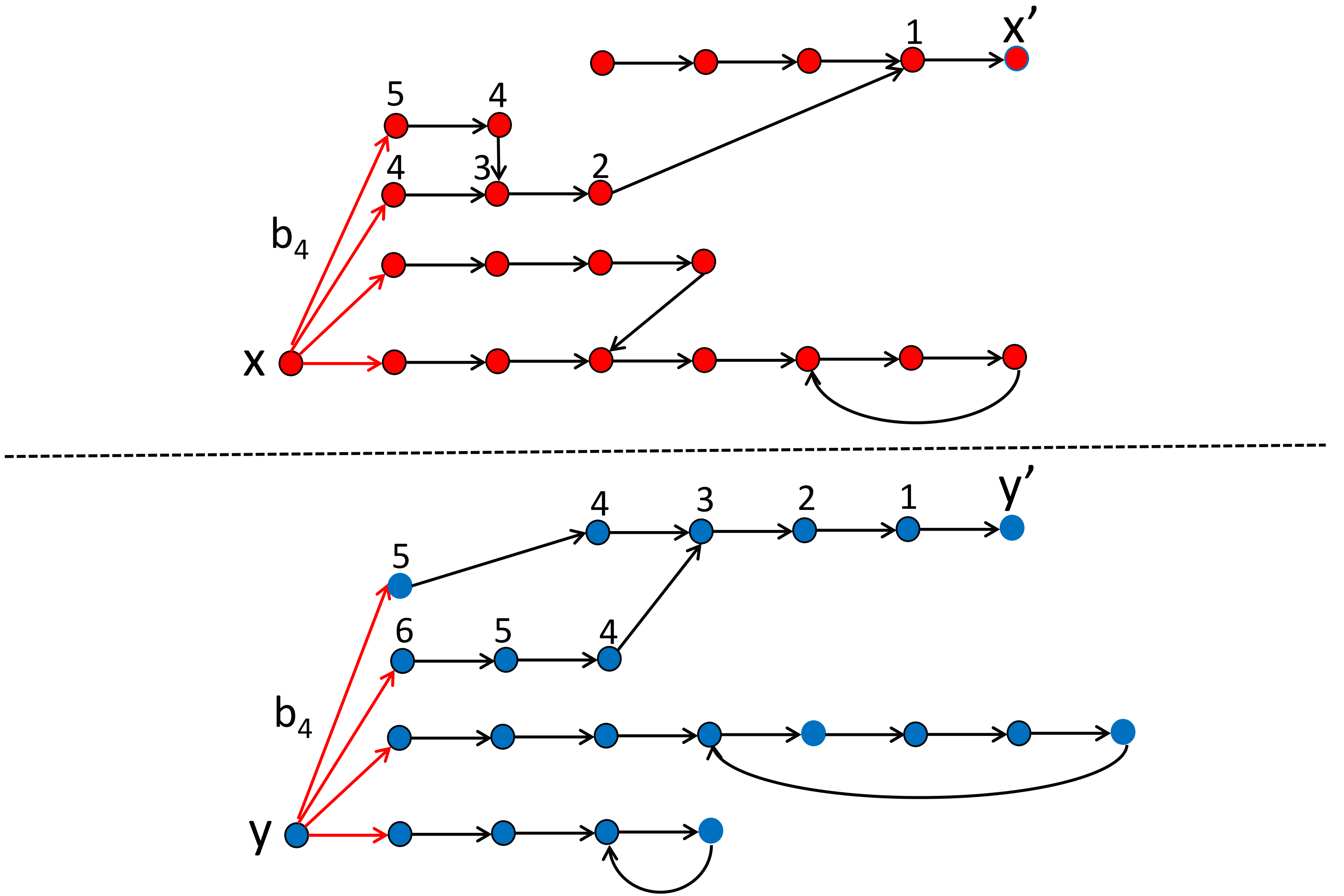
The Algorithm



The Algorithm

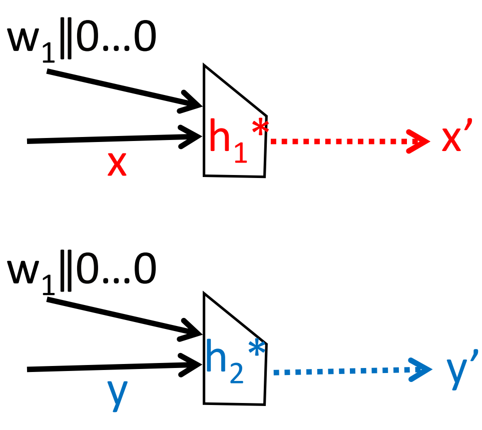


The Algorithm



The Algorithm

- Algorithm: Evaluate **messages** of the form $\mathbf{w_1||0\ldots||0}$ from x and y until a **collision** with a **previous evaluated state**
- Reason for **efficiency**: "**look ahead**"



Hash Functions (2015)

	Collision Resistance	**Preimage Resistance**	**Second Preimage Resistance**
Ideal H	$2^{n/2}$	2^n	2^n
MD H	$2^{n/2}$	2^n	~~2^n~~ $2^n/L$

Ideal $H_1 \| H_2$	2^n	2^{2n}	2^{2n}
MD $H_1 \| H_2$	~~2^n~~ $\approx 2^{n/2}$	~~2^{2n}~~ $\approx 2^n$	~~2^{2n}~~ ~~$\approx 2^n$~~ $2^{3n/4}$ (opt)

- MD $H_1 \| H_2$ is **weaker than** ideal H !

Conclusions

- We showed that **concatenation** of two Merkle-Damgård hash functions is **weaker** than a **single ideal hash function**
    - **HAIFA mode** is stronger than the **concatenation** of two Merkle-Damgård hash functions
- Attacks are **not practical** (for hash functions used in practice $n \geq 160$)
- **New insight** into the security of hash functions
- New application of **random mappings** to cryptanalysis of concatenated hash functions

Thanks for your attention!