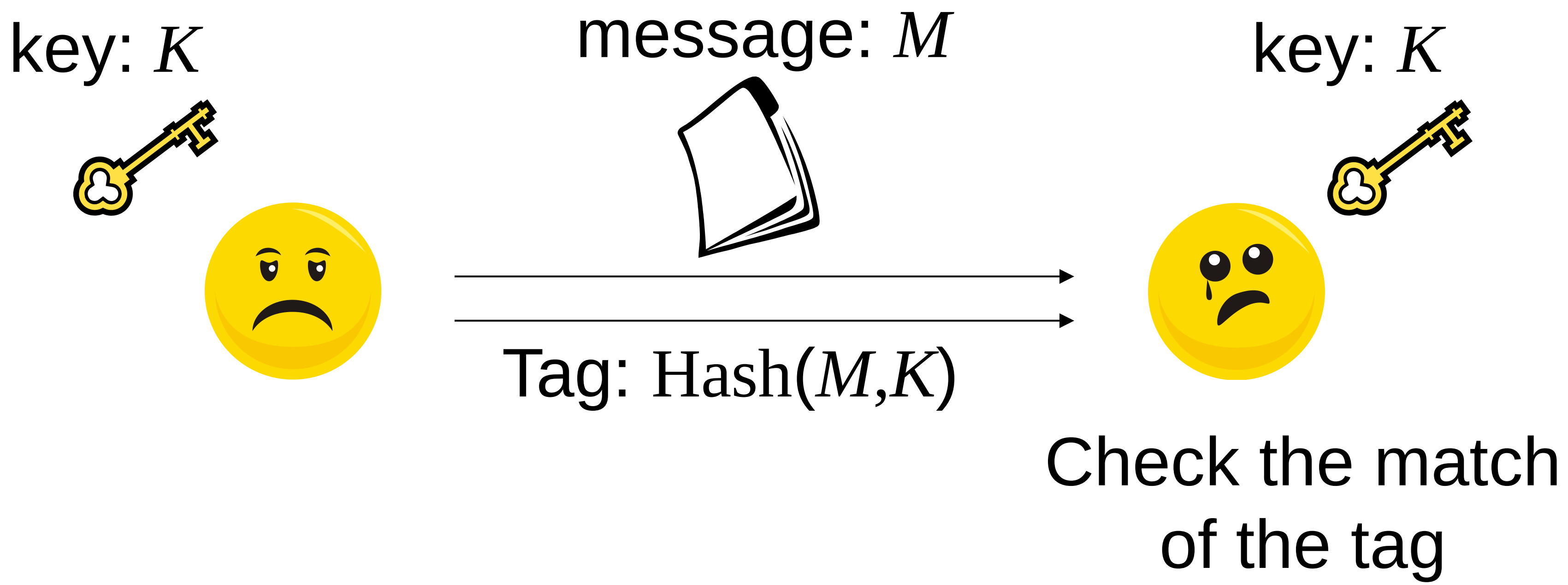
Updates on Generic Attacks against HMAC and NMAC

Jian Guo Nanyang Technological University, Singapore
Thomas Peyrin Nanyang Technological University, Singapore
Yu Sasaki NTT Secure Platform Laboratories, Japan
Lei Wang Nanyang Technological University Singapore
18/August/2014 @ CRYPTO 2014

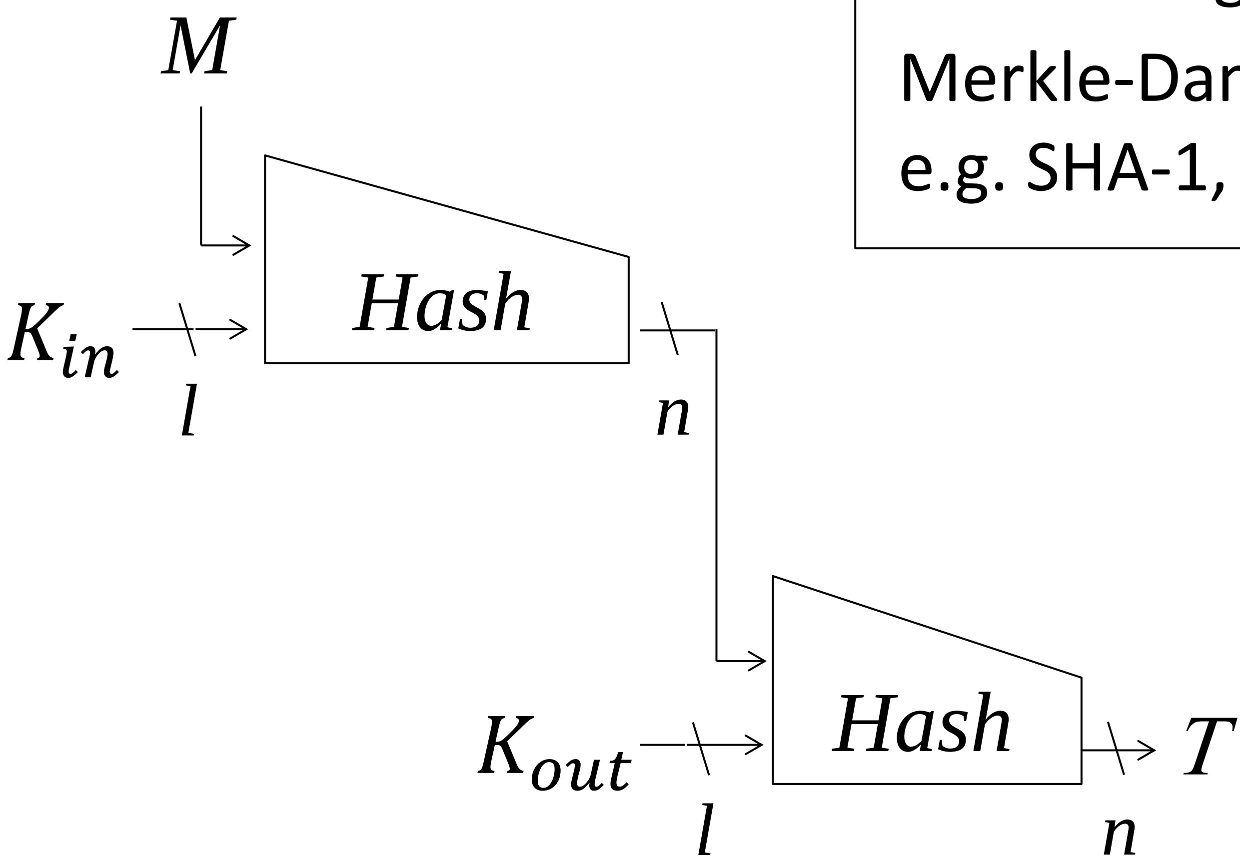
Message Authentication Codes (MAC)

- MAC provides integrity of message.
- often constructed with a hash function.



NMAC [BCK96]

- Compute T with 2 hash function calls. $|K| = 2l$.

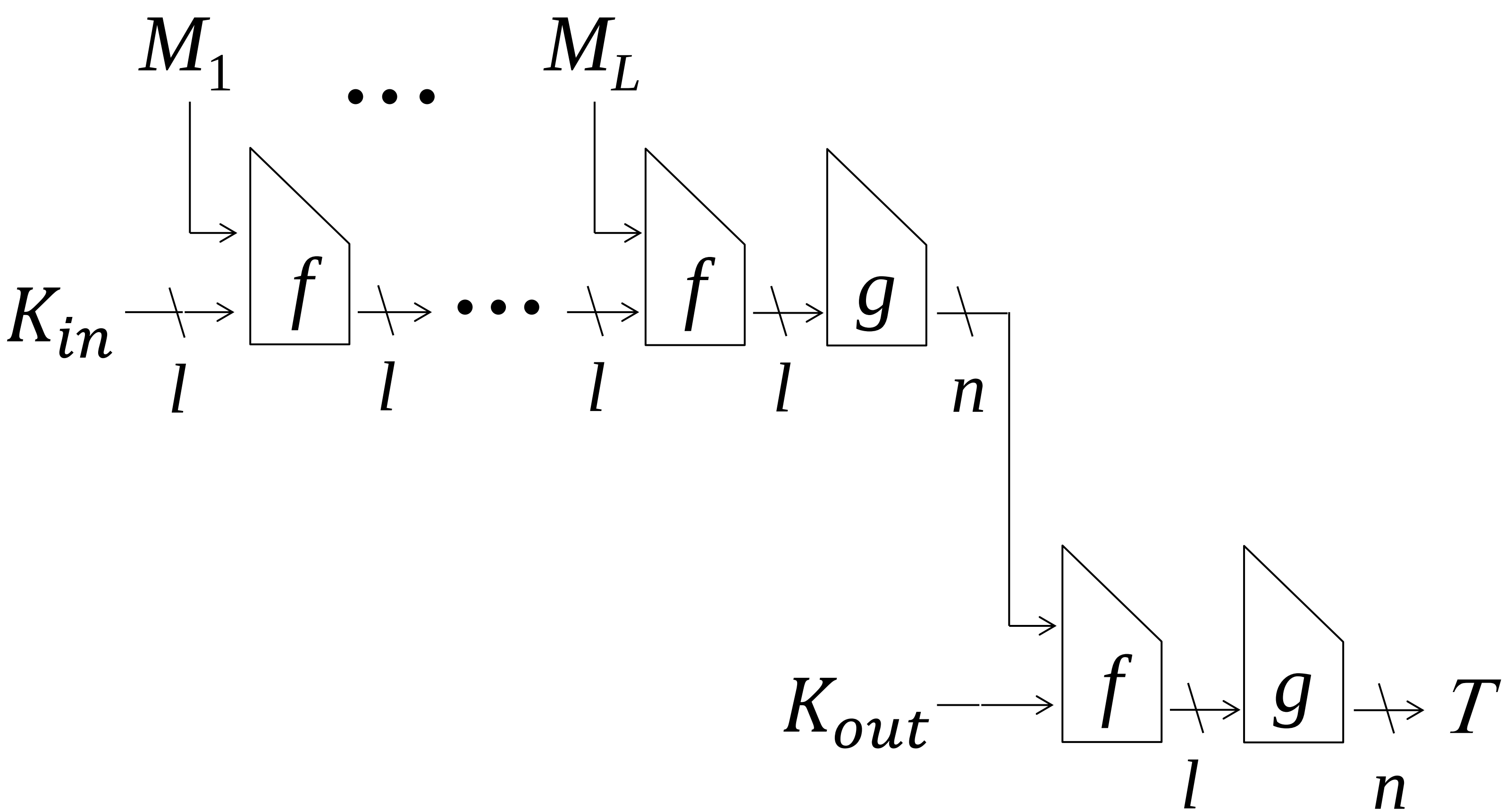


Our target class

Merkle-Damgård hash
e.g. SHA-1, SHA-2

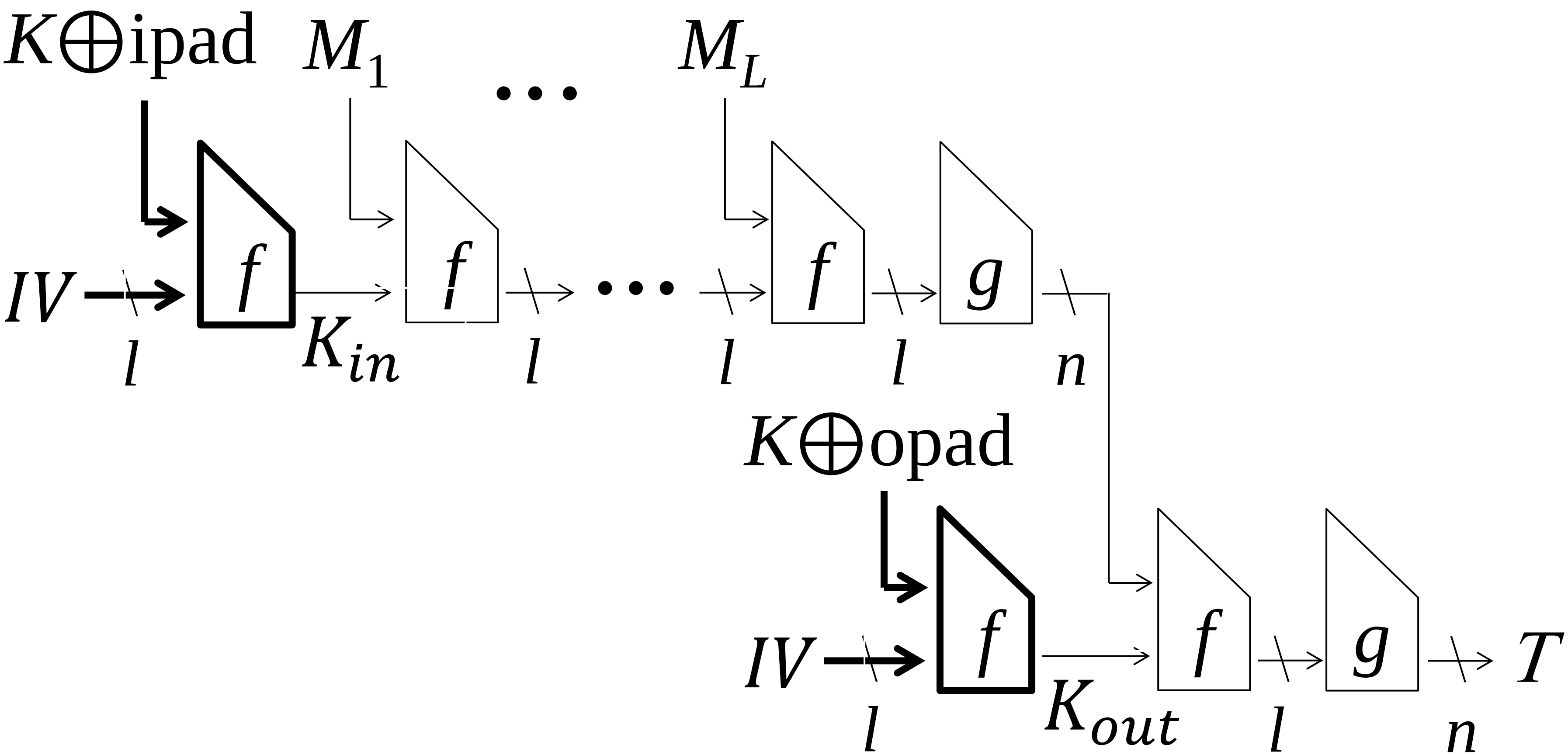
NMAC (compression function level)

- In practice, message is processed block by block.



HMAC [BCK96]

- 2 hash function calls with 1 key of arbitrary key length (K is first padded to block size.)



Security of NMAC and HMAC

- When $n = l$, security is proven up to $O(2^{\frac{n}{2}})$.
(The bound comes from an internal collision)

- Expecting up to $O(2^{\frac{l}{2}})$ is natural for $l > n$.

- The tight attack is known [BO96]. With $O(2^{\frac{l}{2}})$ queries, NMAC/HMAC cannot be PRF.

Security Notions for MAC

- **Existential Forgery**

  find (M, T) where M is not queried yet

- **Selective Forgery**

  find (M, T) where M is selected before attack

- **Universal Forgery**

  find (M, T) for any M

- **Distinguishing-R**

  distinguish MAC oracle and PRF

- **Distinguishing-H**

  distinguish underlying comp. func. from RF

- **Key Recovery**

  Recover (K_{in}, K_{out}) or recovery original K

Known Results

Attack	Prev. Comp.	Ours	Tight?
Existential Forgery	$O(2^{l/2})$		Yes
Selective Forgery	$O(2^{5l/6})$		
Universal Forgery	$O(2^{5l/6})$		
Distinguishing-R	$O(2^{l/2})$		Yes
Distinguishing-H	$O(2^{l/2})$		Yes
Key Recovery	?		

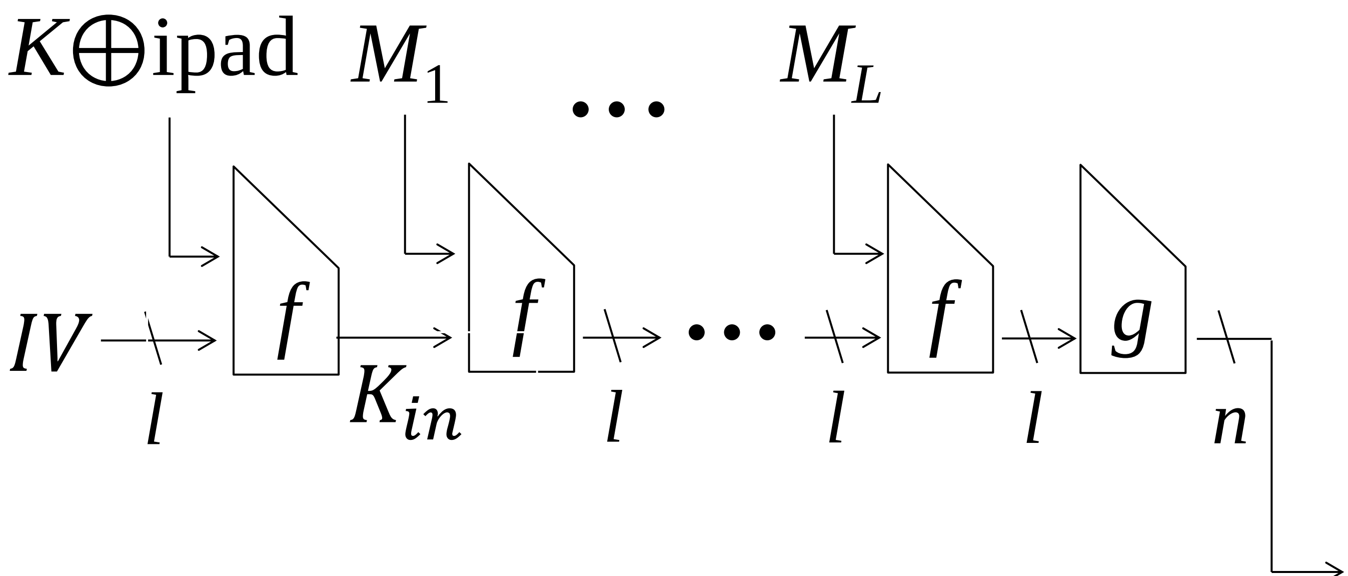
Our Results

Attack	Prev. Comp.	Ours	Tight?
Existential Forgery	$O(2^{l/2})$		Yes
Selective Forgery	$O(2^{5l/6})$	$\boldsymbol{O(2^{l/2})}$	**Yes**
Universal Forgery	$O(2^{5l/6})$	$\boldsymbol{O(2^{3l/4})}$	
Distinguishing-R	$O(2^{l/2})$		Yes
Distinguishing-H	$O(2^{l/2})$		Yes
Key Recovery	$\boldsymbol{O(2^{l})}$	**Off: $\boldsymbol{O(2^{l})}$** **On: $\boldsymbol{O(2^{3l/4})}$**	

Recent Techniques for Generic Attacks against HMAC

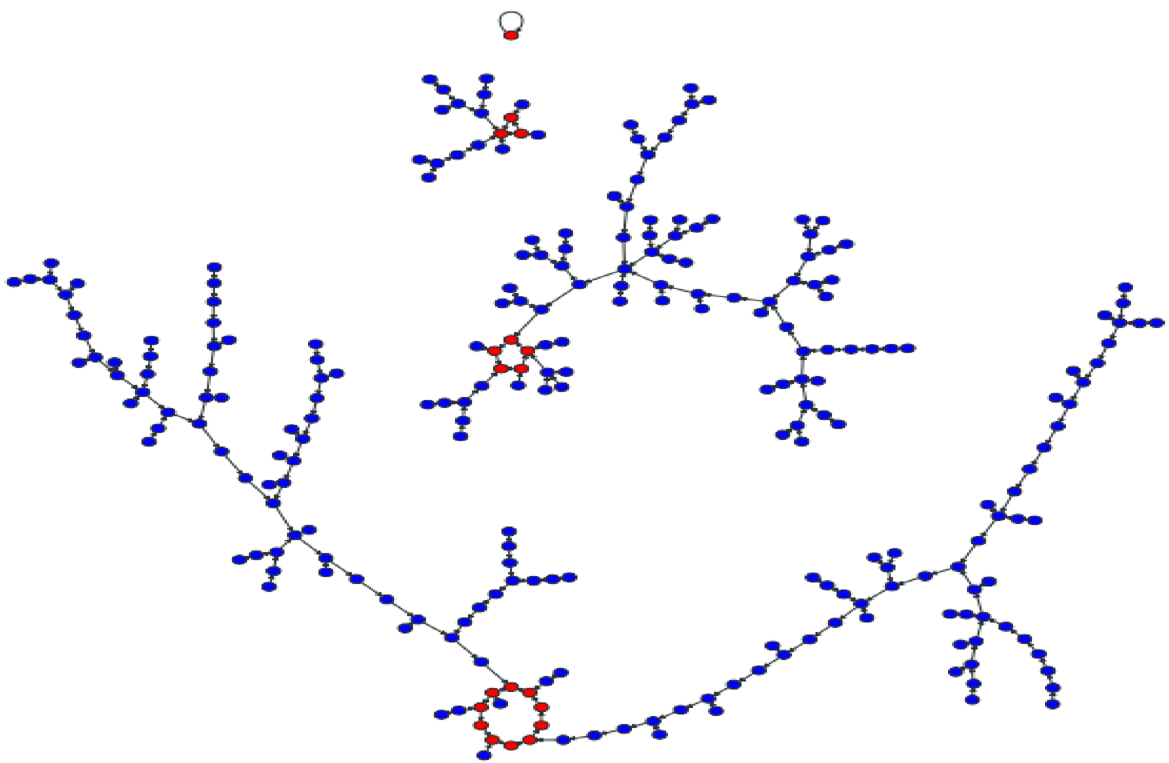
Overall Idea

- Inner function accepts a long message.
- Detect properties of f offline in order to reduce the online cost.
- Draw a functional graph f.



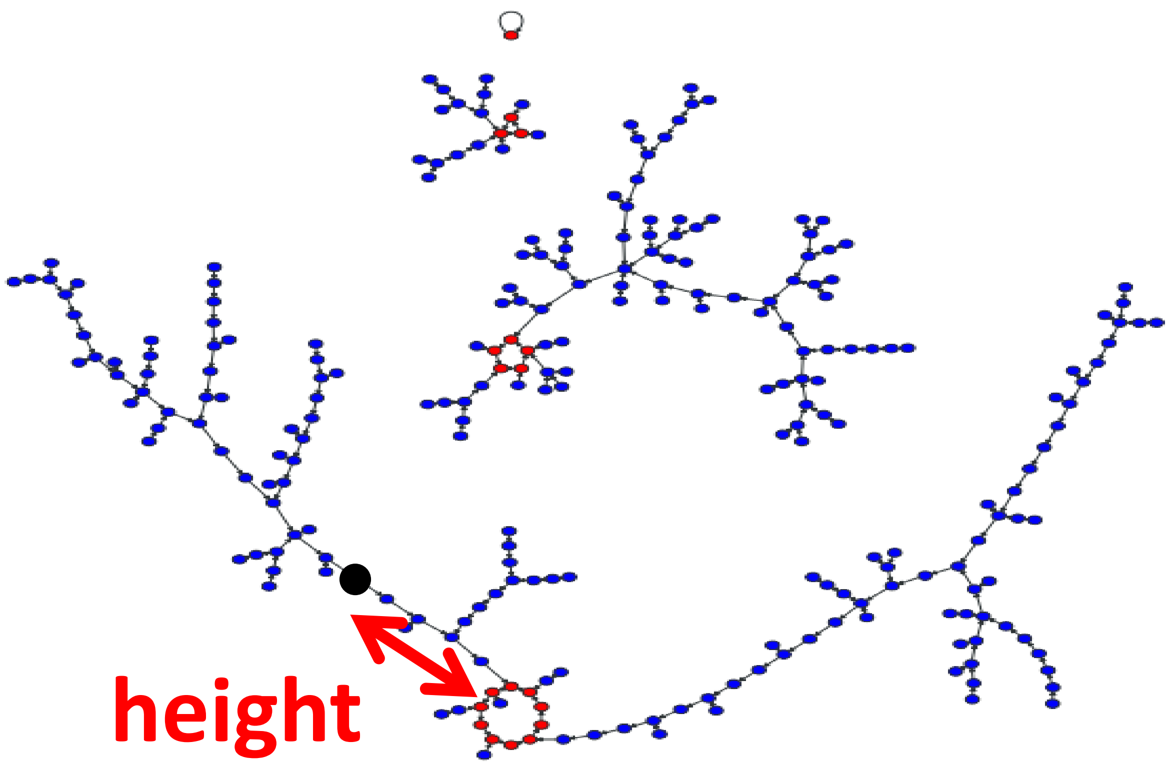
Functional Graph

- Fix message value for all blocks to const, e.g. 0.
- $f_0 : \{0,1\}^l \rightarrow \{0,1\}^l$
- f_0 can be represented as a graph



Functional Graph

- The largest cycle size: $O(2^{l/2})$
- The longest tail size: $O(2^{l/2})$
- **Height** of node (λ): distance to reach the cycle

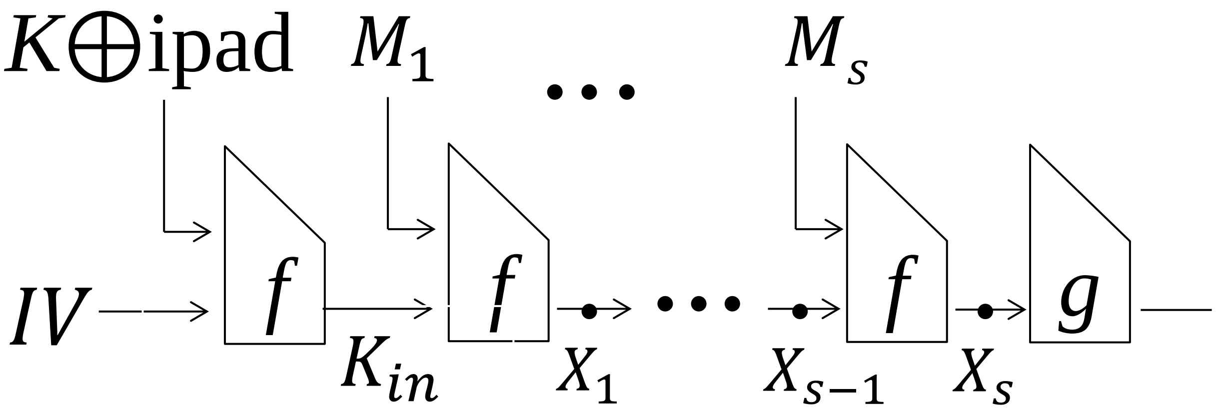


Improved Universal Forgery

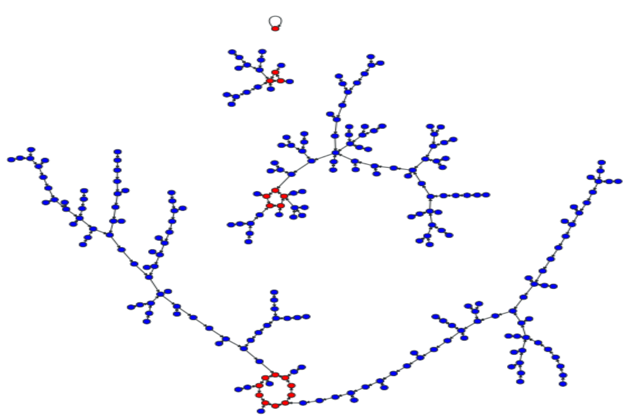
Previous Attack Idea [PW14] (1/3)

Target Message: $M = M_1||M_2||\cdots||M_s$

Online: 2^s unknown internal state values



Offline: generate 2^{l-s} nodes in the random graph



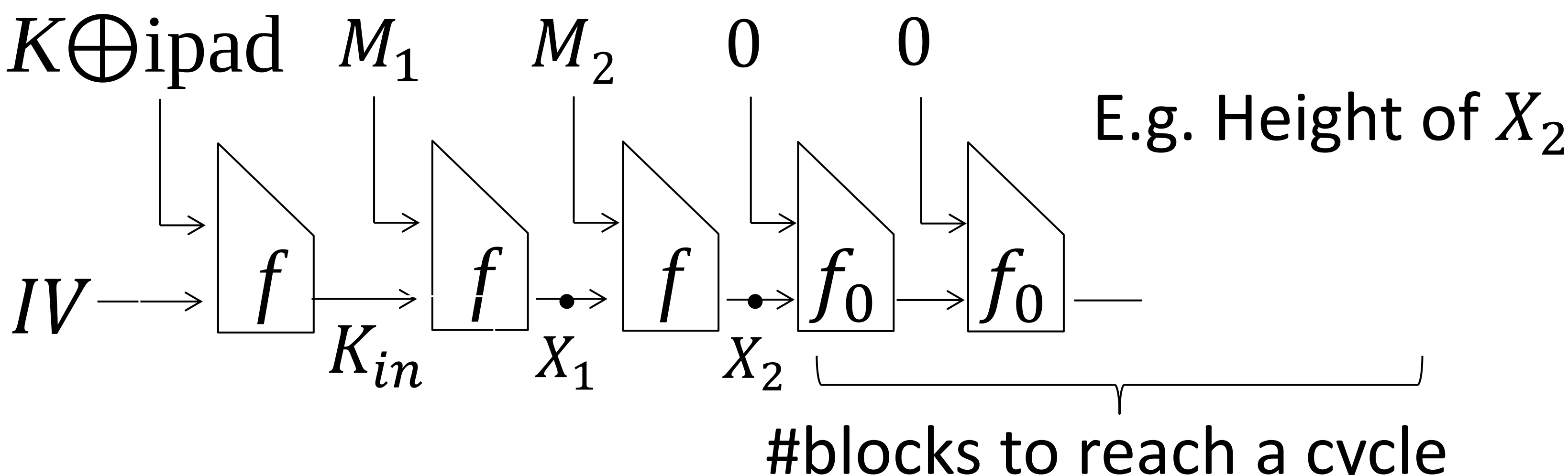
2^{l-s} nodes:

$n_1, n_2, \ldots, n_{l-s}$

Previous Attack Idea [PW14] (2/3)

- Internal state values $(X_1, \ldots, X_s)$ are unknown.
- Need to test all pairs of (X_i, n_j): $O(2^l)$ cost.

- Height of $(X_1, \ldots, X_s)$ can be recovered.
  - [LPW13] detects the height of each node with $O(2^{l/2})$.



Previous Attack Idea [PW14] (3/3)

Online

$(X_1, \lambda(X_1))$

$(X_2, \lambda(X_2))$

$(X_3, \lambda(X_3))$

$(X_4, \lambda(X_4))$

$\vdots$

$(X_s, \lambda(X_s))$

Offline

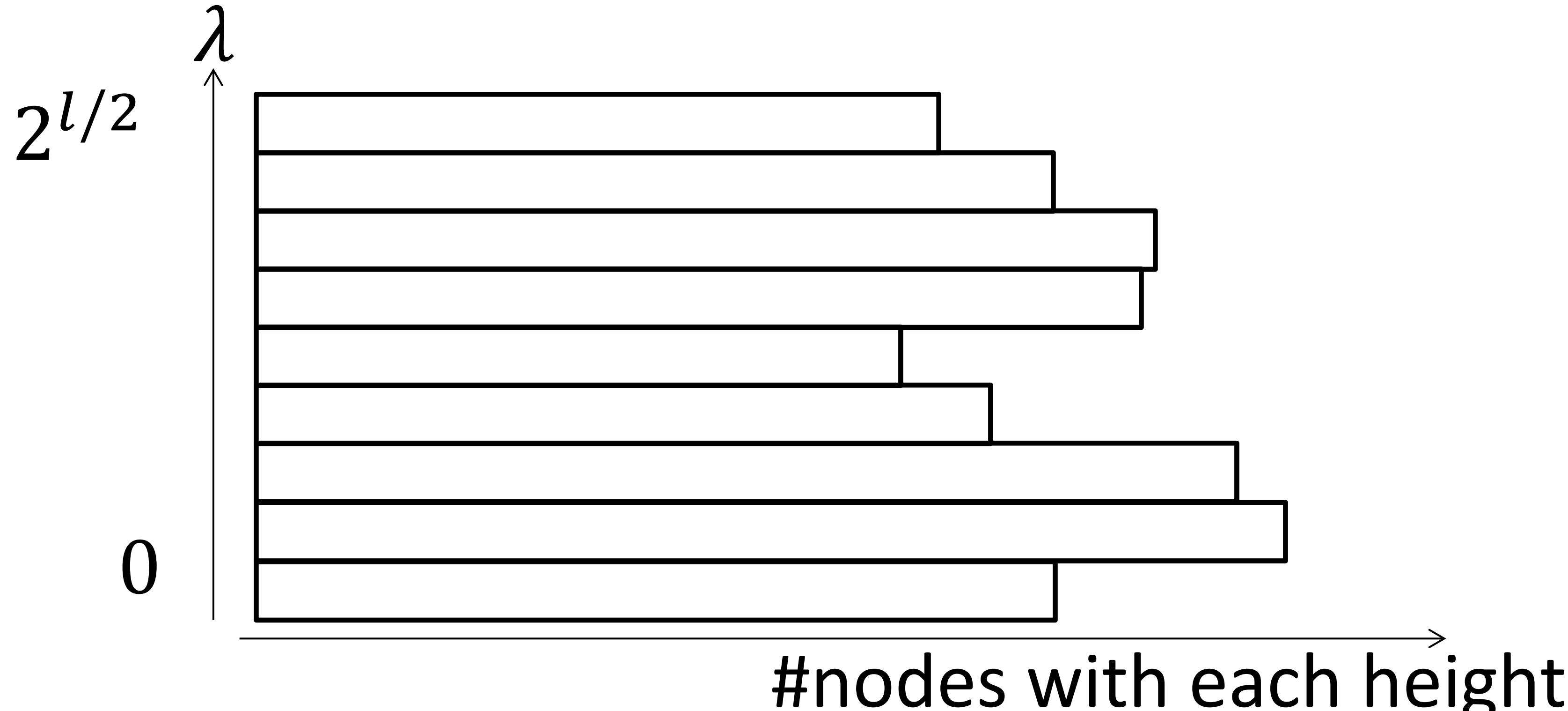
$\lambda(X_1)$	$n_1^{X_1}, n_2^{X_1}, n_3^{X_1}, n_4^{X_1}$
$\lambda(X_2)$	$n_1^{X_2}, n_2^{X_2}$
$\lambda(X_3)$	$n_1^{X_3}, n_2^{X_3}, n_3^{X_3}, n_4^{X_3}$ …
$\lambda(X_4)$	$n_1^{X_4}$
$\vdots$	$\vdots$
$\lambda(X_s)$	$n_1^{X_s}, n_2^{X_s}, n_3^{X_s}, n_4^{X_s}$

- The match of nodes is checked only if the height matches. The cost is reduced from $O(2^l)$.
- Previous attack cost: $O(2^{5l/6})$.

Our Idea for Improvement

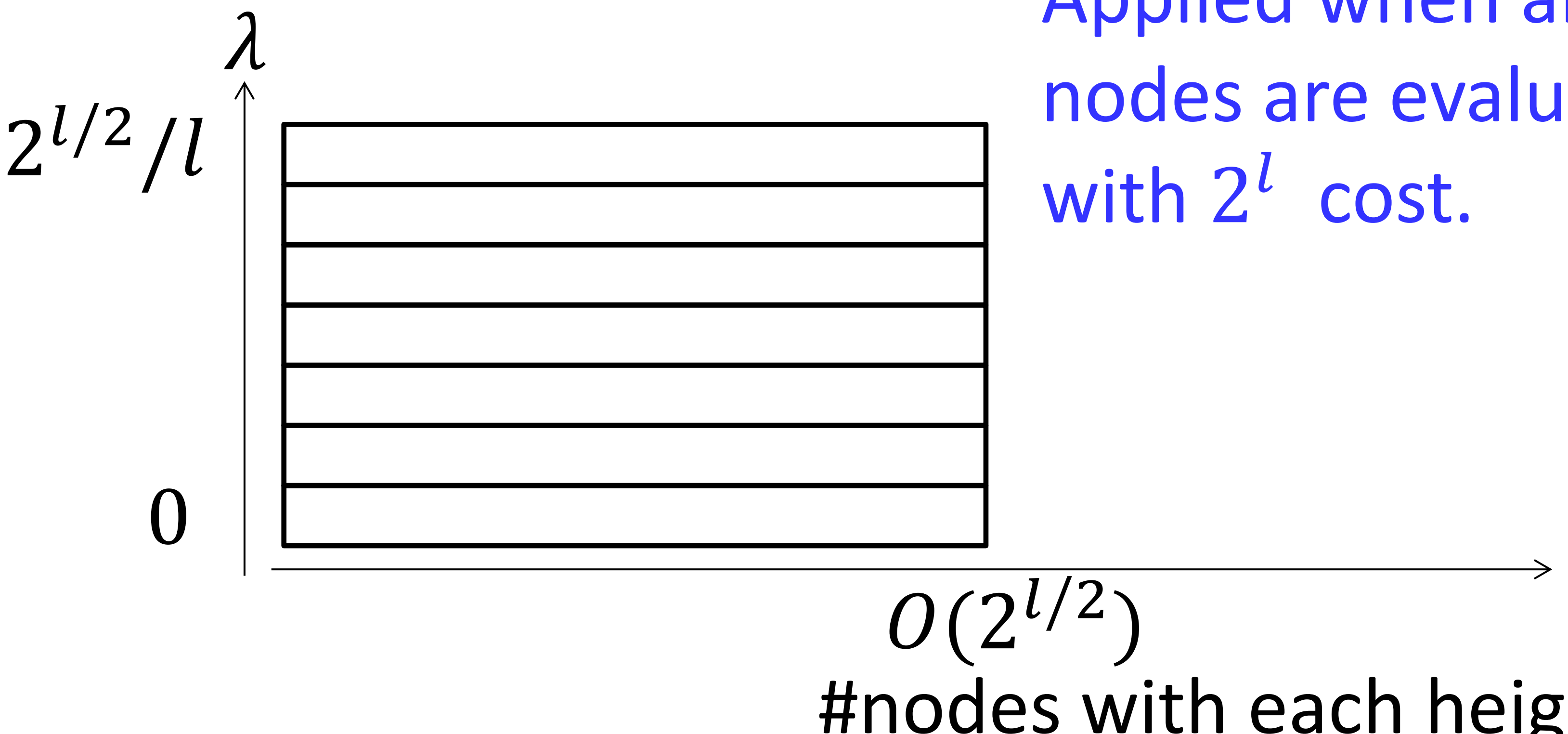
Use more information on the height distribution

- Which height is the most popular?
- Reducing the attack complexity only by collecting nodes with the popular height



[Mutafchiev88, Lemma 2]

Theorem 4 (**[13, Lemma 2]**). *If $l \to \infty$ and $\lambda = o(2^{l/2})$, the mean value of the λ-th stratum S_λ is $\sqrt{\pi/2} * 2^{l/2}$.*



Applied when all nodes are evaluated with 2^l cost.

Difficulties for Applying the Lemma

- [Mutafchiev88, Lemma 2] shows the property of the entire functional graph, which requires $O(2^l)$ cost to draw.

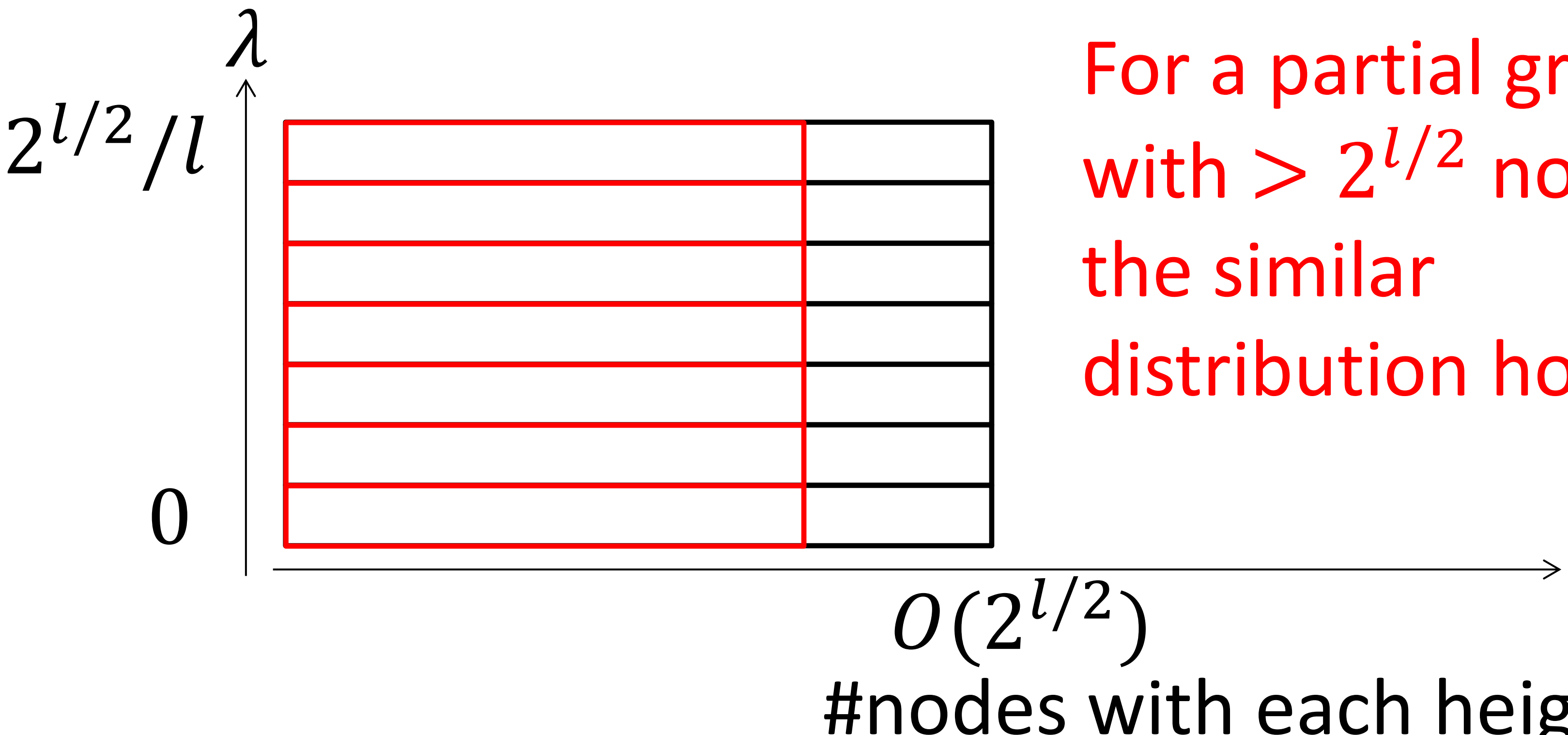
- No advantage compared to brute force attack.

- Need to detect the distribution for a part of the functional graph.

Reducing Nodes in the Graph

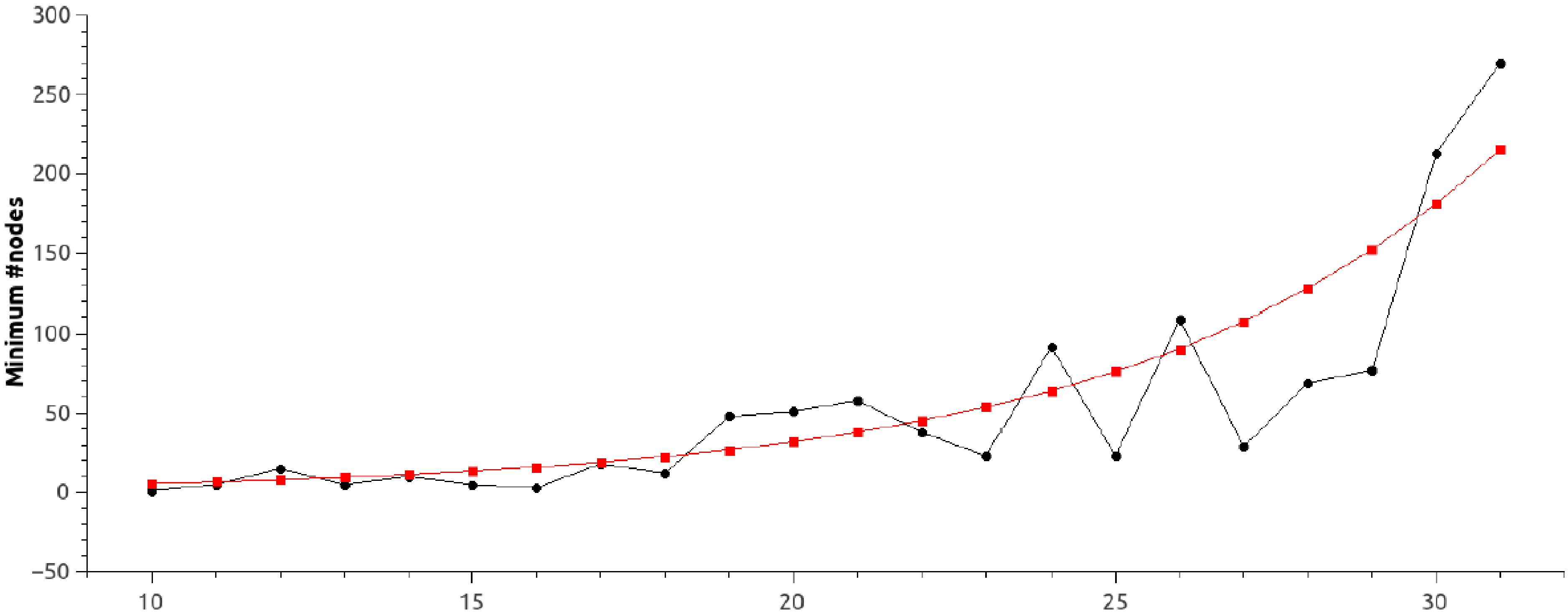
No proven result is known --> **Our Conjecture**

Conjecture 1. If in total 2^t distinct nodes, where $l/2 \leq t \leq l$ holds, are collected following the procedure in Section 5.1, then for any integer λ satisfying $1 \leq \lambda \leq 2^{l/2}/l$, there are $\Theta(2^{t-l/2})$ nodes collected with the height value λ.



For a partial graph with $> 2^{l/2}$ nodes, the similar distribution holds.

Experimental Results



- Attack was improved with the strict height distribution.
- When $2^{l/4} \leq |M| \leq 2^{3l/4}$, both offline and online costs are balanced with $O(2^{3l/4})$.

Concluding Remarks

Proposed improved generic attack on NMAC, HMAC and similar MACs

- Selective Forgery with $O(2^{l/2})$ Tight !
- Universal Forgery with $O(2^{3l/4})$ Improved !!
- Tradeoff for Key Recovery Attack First trail !!!

Previous lemma was generalized as a conjecture.
The experiment matches the conjecture well.
Its formal proof is an open problem.

Thank you for your attention !!

Two Types of Selective Forgery

this talk

1. Strong constraint on selected message:
$$O(2^{l/2})$$

2. Large amount of freedom degrees:
$$O(2^{2l/3})$$

Distinguishing-H Attack [LPW13]

Offline:

- Draw a functional graph of f_0. Find a largest cycle length L.

Cost: $O(2^{l/2})$

Online:

- $\mathrm{Query}_1 = M_1 || \overbrace{0||0|| \cdots ||0}^{2^{l/2}} || M_2 || \overbrace{0||0|| \cdots ||0}^{2^{l/2}+L}$
- $\mathrm{Query}_2 = M_1 || \underbrace{0||0|| \cdots ||0}_{2^{l/2}+L} || M_2 || \underbrace{0||0|| \cdots ||0}_{2^{l/2}}$

Cost: $O(2^{l/2})$

Simple Application for Selective Forgery

- Offline
  - Draw a functional graph
  - Select $Qery_1$ as a target

Cost: $O(2^{l/2})$

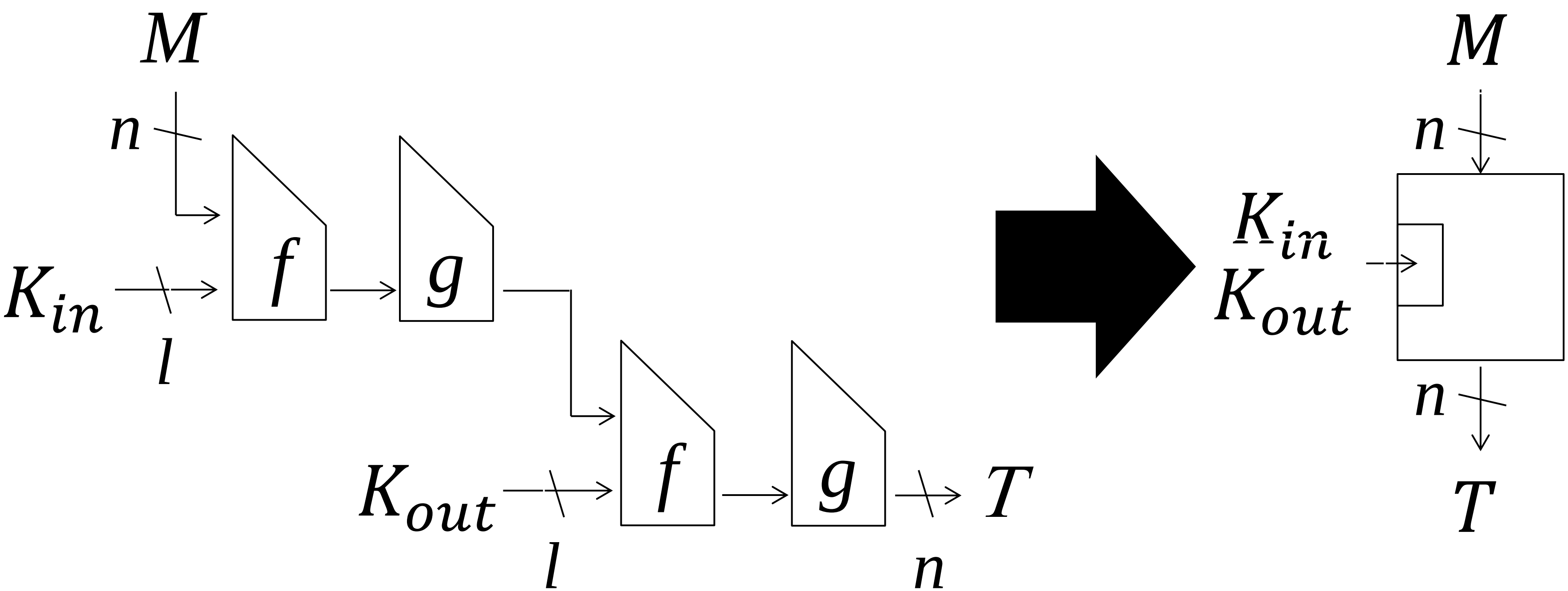
- Online
  - Send $Qery_2$ to the oracle to obtain tag T.
  - $(Query_2, T)$ is a valid tag.

Cost: $O(2^{l/2})$

Hellman’s Tradeoff for Key Recovery

Simple Application for NMAC ($|K| = 2l$)

- Regard NMAC as n-bit to n-bit function
- Simple Hellman's TM-tradeoff:
  - Precomp $= O(2^{2l})$, Online Mem=Time$= O(2^{3l/4})$



Easy Generic Key Recovery with $O(2^l)$

1. Recover K_{in} with $O(2^l)$ cost.
   - Find a collision of the inner function with online queries. (existential forgery attack)
   - Guess K_{in} and check if the collision is obtained.

2. Exhaustive search on K_{out} with $O(2^l)$ cost.

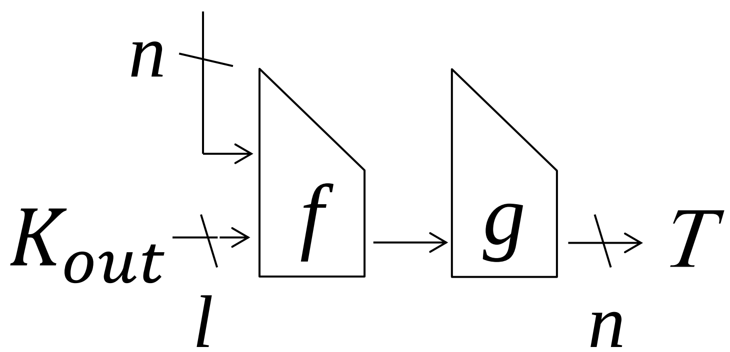
$2n$-bit key is recovered with $O(2^l)$, which is already better than simple tradeoff on $2n$ bits.

This motivated us to find an improved tradeoff for the key recovery attack.

Idea

Firstly recover K_{out}

- Input message is unknown.
- Combine:
  - ➢ Hellman's tradeoff
  - ➢ Inner state recovery



Secondly recover K_{in}.

- Cannot be simple.
  - ➢ Use the height distribution (based on our conjecture)