

# Weaknesses of Cuckoo Hashing with a Simple Universal Hash Class: The Case of Large Universes

Martin Dietzfelbinger and Ulf Schellbach

Technische Universität Ilmenau

SofSem 2009

# Outline

- 1 Background
- 2 Main Result
- 3 Relevance
- 4 Experimental Results
- 5 Conclusion

# Outline

- 1 Background
- 2 Main Result
- 3 Relevance
- 4 Experimental Results
- 5 Conclusion

# Dictionary, Hashing, Universal Hashing

## Dictionary:

- dynamic mapping data structure
- supports insertion, deletion, query of entries
- entries: pairs  $(x, d)$  of a key  $x$  and associated data  $d$

# Dictionary, Hashing, Universal Hashing

## Dictionary:

- dynamic mapping data structure
- supports insertion, deletion, query of entries
- entries: pairs  $(x, d)$  of a key  $x$  and associated data  $d$

identify  $(x, d)$  with  $x$ , refer to “query” as lookup

# Dictionary, Hashing, Universal Hashing

## Hashing:

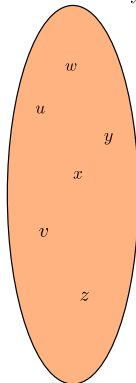
- realizes a dictionary

# Dictionary, Hashing, Universal Hashing

## Hashing:

- realizes a dictionary

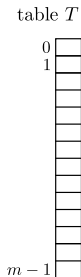
universe  $U$  of keys



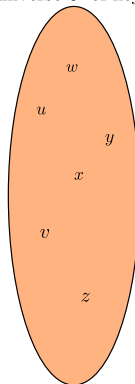
# Dictionary, Hashing, Universal Hashing

## Hashing:

- realizes a dictionary



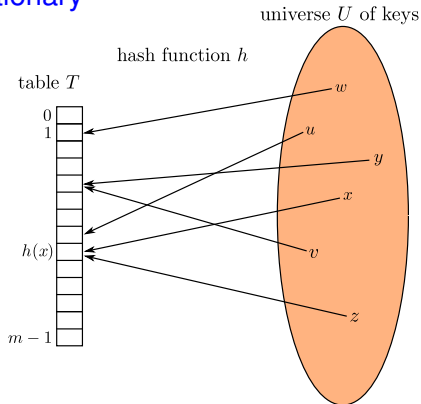
universe  $U$  of keys



# Dictionary, Hashing, Universal Hashing

## Hashing:

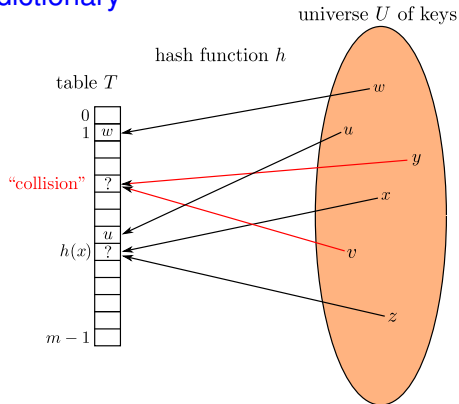
- realizes a dictionary



# Dictionary, Hashing, Universal Hashing

## Hashing:

- realizes a dictionary

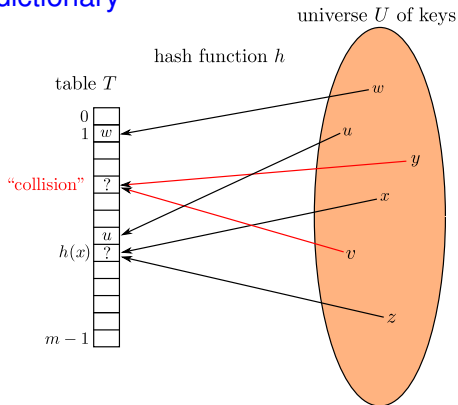


different ways to handle “collisions”  $\rightsquigarrow$  different hashing schemes

# Dictionary, Hashing, Universal Hashing

## Hashing:

- realizes a dictionary

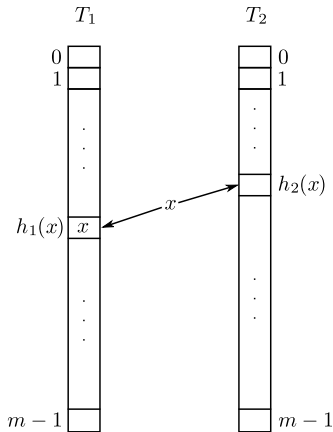


different ways to handle “collisions”  $\rightsquigarrow$  different hashing schemes

**Universal Hashing:**  $h$  is chosen at random from a class  $\mathcal{H}$

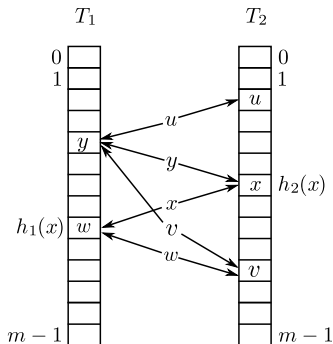
# Cuckoo Hashing (Pagh and Rodler 2001)

## the scheme



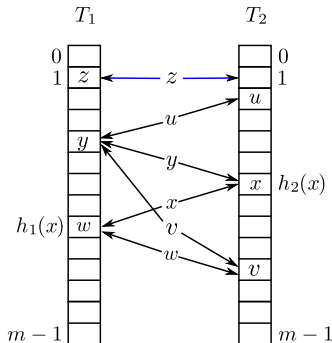
# Cuckoo Hashing (Pagh and Rodler 2001)

## Example: “suitable” functions



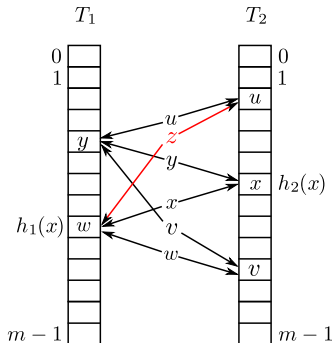
# Cuckoo Hashing (Pagh and Rodler 2001)

## Example: “suitable” functions



# Cuckoo Hashing (Pagh and Rodler 2001)

## Example: “not suitable” functions



# Cuckoo Hashing (Pagh and Rodler 2001)

known, easy to prove:

## Lemma

*keys from  $S \subseteq U$  can be placed according to  $h_1$  and  $h_2$*

$\Leftrightarrow$

$\nexists S' \subseteq S: |S'| > |\{h_1(x) \mid x \in S'\}| + |\{h_2(x) \mid x \in S'\}|$

# Suitability and Failure Probability

## Definition

*$h_1$  and  $h_2$  are suitable for  $S \subseteq U$*

*$:\Leftrightarrow$*

*keys from  $S$  can be placed according to  $h_1$  and  $h_2$*

# Suitability and Failure Probability

## Definition

*$h_1$  and  $h_2$  are suitable for  $S \subseteq U$*

$:\Leftrightarrow$

*keys from  $S$  can be placed according to  $h_1$  and  $h_2$*

## Definition

*failure probability  $p_{\text{failure}} := \Pr_{S, h_1, h_2}(h_1, h_2 \text{ not suitable for } S)$*

# The Hash Function Family

## multiplicative class

### Definition

Let  $k, l \in \mathbb{N}$  with  $l \leq k$ .

*multiplicative class*  $\mathcal{H}_{k,l}^{mult}$ : functions

$$h_a(x) := (a \cdot x \bmod 2^k) \operatorname{div} 2^{k-l},$$

$0 < a < 2^k$  odd, where  $U = [2^k] := \{0, 1, \dots, 2^k - 1\}$ ,  $m = 2^l$ .

# The Hash Function Family

## multiplicative class

### Definition

Let  $k, l \in \mathbb{N}$  with  $l \leq k$ .

*multiplicative class*  $\mathcal{H}_{k,l}^{mult}$ : functions

$$h_a(x) := (a \cdot x \bmod 2^k) \operatorname{div} 2^{k-l},$$

$0 < a < 2^k$  odd, where  $U = [2^k] := \{0, 1, \dots, 2^k - 1\}$ ,  $m = 2^l$ .

- very simple, efficient, “universal” class

# The Hash Function Family

## multiplicative class

### Definition

Let  $k, l \in \mathbb{N}$  with  $l \leq k$ .

*multiplicative class*  $\mathcal{H}_{k,l}^{\text{mult}}$ : functions

$$h_a(x) := (a \cdot x \bmod 2^k) \operatorname{div} 2^{k-l},$$

$0 < a < 2^k$  odd, where  $U = [2^k] := \{0, 1, \dots, 2^k - 1\}$ ,  $m = 2^l$ .

- very simple, efficient, “universal” class
- Pagh/Rodler (2004) report on experimental observations:  
 $\mathcal{H}_{k,l}^{\text{mult}}$  does not work well for cuckoo hashing

# Outline

- 1 Background
- 2 Main Result**
- 3 Relevance
- 4 Experimental Results
- 5 Conclusion

# Constant Failure Probability for “Bad” Sets

## Theorem (informal)

$m \ll |U|$  &  $S \subseteq U$  a constructed “bad” set of size  $\approx (7/8)m$

$\Rightarrow$

$$p_{\text{failure}} = \Omega(1)$$

for randomly chosen multiplicative hash functions

# Constant Failure Probability for “Bad” Sets

## Theorem (informal)

$m \ll |U|$  &  $S \subseteq U$  a constructed “bad” set of size  $\approx (7/8)m$

$\Rightarrow$

$$p_{\text{failure}} = \Omega(1)$$

for randomly chosen multiplicative hash functions

Contrast:

Hash values independent and uniformly random

$\Rightarrow$

$$p_{\text{failure}} = O(1/m) \text{ for each set } S \subseteq U, |S| = (1 - \delta)m$$

# Constant Failure Probability for “Bad” Sets

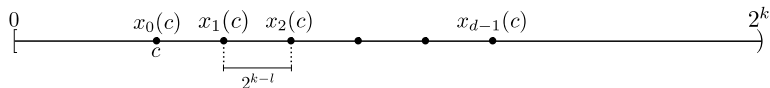
## structure of “bad” sets $S$

- Let  $d := \lceil (7/8)m/3 \rceil$ .
- Define  $x_i(c) := (c + i \cdot 2^{k-l}) \bmod 2^k$ , for  $c \in [2^k]$  and  $i \in [2^l]$ .

# Constant Failure Probability for “Bad” Sets

## structure of “bad” sets $S$

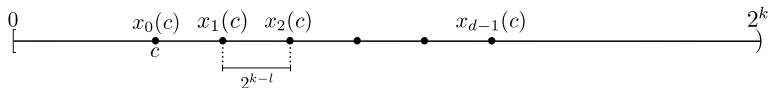
- Let  $d := \lceil (7/8)m/3 \rceil$ .
- Define  $x_i(c) := (c + i \cdot 2^{k-l}) \bmod 2^k$ , for  $c \in [2^k]$  and  $i \in [d]$ .
- “Grid set”  $G_c := \{x_i(c) \mid i \in [d]\}$ , for  $c \in [2^k]$ .



# Constant Failure Probability for “Bad” Sets

## structure of “bad” sets $S$

- Let  $d := \lceil (7/8)m/3 \rceil$ .
- Define  $x_i(c) := (c + i \cdot 2^{k-l}) \bmod 2^k$ , for  $c \in [2^k]$  and  $i \in [d]$ .
- “Grid set”  $G_c := \{x_i(c) \mid i \in [d]\}$ , for  $c \in [2^k]$ .

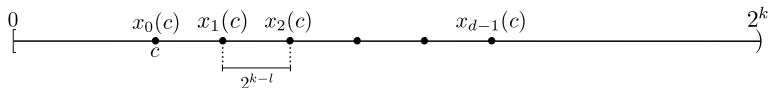


- Bad set  $S = S( , ) =$

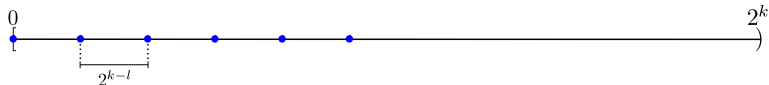
# Constant Failure Probability for “Bad” Sets

## structure of “bad” sets $S$

- Let  $d := \lceil (7/8)m/3 \rceil$ .
- Define  $x_i(c) := (c + i \cdot 2^{k-l}) \bmod 2^k$ , for  $c \in [2^k]$  and  $i \in [d]$ .
- “Grid set”  $G_c := \{x_i(c) \mid i \in [d]\}$ , for  $c \in [2^k]$ .



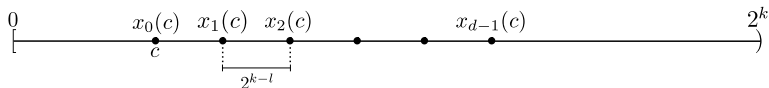
- Bad set  $S = S(, ) = G_0$



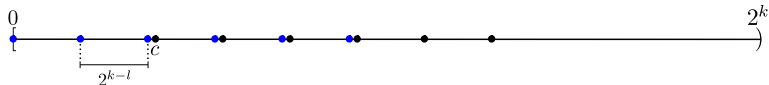
# Constant Failure Probability for “Bad” Sets

## structure of “bad” sets $S$

- Let  $d := \lceil (7/8)m/3 \rceil$ .
- Define  $x_i(c) := (c + i \cdot 2^{k-l}) \bmod 2^k$ , for  $c \in [2^k]$  and  $i \in [d]$ .
- “Grid set”  $G_c := \{x_i(c) \mid i \in [d]\}$ , for  $c \in [2^k]$ .



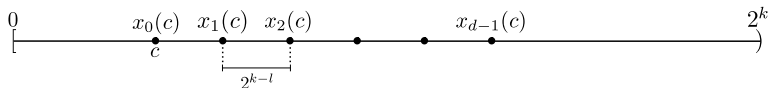
- Bad set  $S = S(c, \quad) = G_0 \cup G_c$



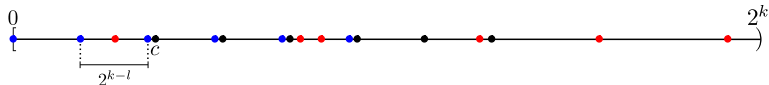
# Constant Failure Probability for “Bad” Sets

## structure of “bad” sets $S$

- Let  $d := \lceil (7/8)m/3 \rceil$ .
- Define  $x_i(c) := (c + i \cdot 2^{k-l}) \bmod 2^k$ , for  $c \in [2^k]$  and  $i \in [d]$ .
- “Grid set”  $G_c := \{x_i(c) \mid i \in [d]\}$ , for  $c \in [2^k]$ .



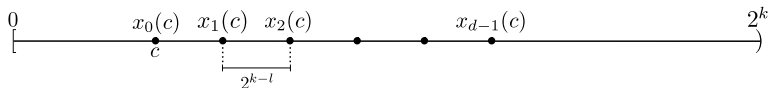
- Bad set  $S = S(c, R_c) = G_0 \cup G_c \cup R_c$



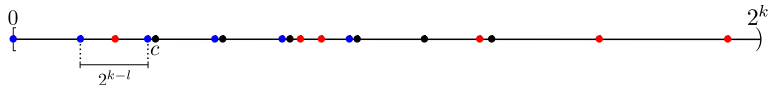
# Constant Failure Probability for “Bad” Sets

## structure of “bad” sets $S$

- Let  $d := \lceil (7/8)m/3 \rceil$ .
- Define  $x_i(c) := (c + i \cdot 2^{k-l}) \bmod 2^k$ , for  $c \in [2^k]$  and  $i \in [2^l]$ .
- “Grid set”  $G_c := \{x_i(c) \mid i \in [d]\}$ , for  $c \in [2^k]$ .



- Bad set  $S = S(c, R_c) = G_0 \cup G_c \cup R_c$



where  $c$ ,  $0 < c < 2^k$  odd, and  $R_c \subseteq U - (G_0 \cup G_c)$ ,  $|R_c| = d$ , are chosen uniformly at random.

# Constant Failure Probability for “Bad” Sets

## Theorem (precise)

### Theorem

$l \geq 14$ ,  $k - \log k \geq 3l + 5$  &  $S = G_0 \cup G_c \cup R_c$  chosen randomly,  
( $|S| = 3d \leq (7/8) \cdot m + 2 \rightsquigarrow$  load factor  $\ll 1/2$ )

$\Rightarrow$

$$p_{\text{failure}} = \Omega(1)$$

for randomly chosen multiplicative hash functions

# Constant Failure Probability for “Bad” Sets

## Theorem (precise)

### Theorem

$l \geq 14, k - \log k \geq 3l + 5$  &  $S = G_0 \cup G_c \cup R_c$  chosen randomly,  
( $|S| = 3d \leq (7/8) \cdot m + 2 \rightsquigarrow$  load factor  $\ll 1/2$ )  
 $\Rightarrow$   
 $p_{\text{failure}} = \Omega(1)$   
for randomly chosen multiplicative hash functions

- we establish:  $p_{\text{failure}} > 2^{-24}$

# Constant Failure Probability for “Bad” Sets

## Theorem (precise)

### Theorem

$l \geq 14$ ,  $k - \log k \geq 3l + 5$  &  $S = G_0 \cup G_c \cup R_c$  chosen randomly,  
( $|S| = 3d \leq (7/8) \cdot m + 2 \rightsquigarrow$  load factor  $\ll 1/2$ )

$\Rightarrow$

$$p_{\text{failure}} = \Omega(1)$$

for randomly chosen multiplicative hash functions

- we establish:  $p_{\text{failure}} > 2^{-24}$
- experiments indicate:  $p_{\text{failure}} \approx 0.02 \gg 2^{-24}$

# Basic Structure of the Proof

We show that (for  $m \ll |U|$  large enough)...

- (1) ... a fraction of more than  $1/7$  of all hash function pairs  $(h_1, h_2)$  has an “almost uniform distribution” (aud),

# Basic Structure of the Proof

We show that (for  $m \ll |U|$  large enough)...

- (1) ... a fraction of more than  $1/7$  of all hash function pairs  $(h_1, h_2)$  has an “almost uniform distribution” (aud), i. e.: If  $x$  chosen uniformly at random from  $U$  or  $\{1, 3, \dots, 2^k - 1\}$  then

$$\forall (i, j) \in [2^l]^2: \quad \frac{1}{4} \cdot 2^{-2l} \leq \Pr((h_1(x), h_2(x)) = (i, j)) \leq 4 \cdot 2^{-2l} .$$

# Basic Structure of the Proof

We show that (for  $m \ll |U|$  large enough)...

- (1) ... a fraction of more than  $1/7$  of all hash function pairs  $(h_1, h_2)$  has an “almost uniform distribution” (aud), i. e.: If  $x$  chosen uniformly at random from  $U$  or  $\{1, 3, \dots, 2^k - 1\}$  then

$$\forall (i, j) \in [2^l]^2: \quad \frac{1}{4} \cdot 2^{-2l} \leq \Pr((h_1(x), h_2(x)) = (i, j)) \leq 4 \cdot 2^{-2l} .$$

- (2) ...  $p_{\text{failure}}(S(c, R_c), h_1, h_2 \mid (h_1, h_2) \text{ is a pair with aud}) > 2^{-21}$ .

# Basic Structure of the Proof

We show that (for  $m \ll |U|$  large enough)...

- (1) ... a fraction of more than  $1/7$  of all hash function pairs  $(h_1, h_2)$  has an “almost uniform distribution” (aud), i. e.: If  $x$  chosen uniformly at random from  $U$  or  $\{1, 3, \dots, 2^k - 1\}$  then

$$\forall (i, j) \in [2^l]^2: \quad \frac{1}{4} \cdot 2^{-2l} \leq \Pr((h_1(x), h_2(x)) = (i, j)) \leq 4 \cdot 2^{-2l} .$$

- (2) ...  $p_{\text{failure}}(S(c, R_c), h_1, h_2 \mid (h_1, h_2) \text{ is a pair with aud}) > 2^{-21}$ .  
(3) ... (1), (2)  $\Rightarrow p_{\text{failure}}(S(c, R_c), h_1, h_2) > 2^{-24}$ .



# Outline

- 1 Background
- 2 Main Result
- 3 Relevance**
- 4 Experimental Results
- 5 Conclusion

# Universality

## Definition

$\mathcal{H}$  is  $(c, k)$ -universal if for arbitrary distinct keys  $x_1, \dots, x_k \in U$ , arbitrary values  $y_1, \dots, y_k \in [m]$  and a function  $h \in \mathcal{H}$  chosen uniformly at random,

$$\Pr(h(x_1) = y_1, \dots, h(x_k) = y_k) \leq \frac{c}{m^k}.$$

# Universality

## Definition

$\mathcal{H}$  is *(c, k)-universal* if for arbitrary distinct keys  $x_1, \dots, x_k \in U$ , arbitrary values  $y_1, \dots, y_k \in [m]$  and a function  $h \in \mathcal{H}$  chosen uniformly at random,

$$\Pr(h(x_1) = y_1, \dots, h(x_k) = y_k) \leq \frac{c}{m^k}.$$

$\mathcal{H}$  is *c-universal* if for arbitrary keys  $x \neq y$  and  $h$  chosen uniformly at random,

$$\Pr(h(x) = h(y)) \leq \frac{c}{m}.$$

# Universality

## Definition

$\mathcal{H}$  is *(c, k)-universal* if for arbitrary distinct keys  $x_1, \dots, x_k \in U$ , arbitrary values  $y_1, \dots, y_k \in [m]$  and a function  $h \in \mathcal{H}$  chosen uniformly at random,

$$\Pr(h(x_1) = y_1, \dots, h(x_k) = y_k) \leq \frac{c}{m^k}.$$

$\mathcal{H}$  is *c-universal* if for arbitrary keys  $x \neq y$  and  $h$  chosen uniformly at random,

$$\Pr(h(x) = h(y)) \leq \frac{c}{m}.$$

- $\mathcal{H}_{k,l}^{\text{mult}}$  is 2-universal.

## Related Work

- “ $\mathcal{H} (O(1), O(\log m))$ -universal  
 $\Rightarrow p_{\text{failure}} = O(1/m)$  for each  $S$  of size  $(1 - \delta)m$ .”  
(load factor  $\approx 1/2$ )  
(Pagh, 2001)

## Related Work

- “ $\mathcal{H} (O(1), O(\log m))$ -universal  
 $\Rightarrow p_{\text{failure}} = O(1/m)$  for each  $S$  of size  $(1 - \delta)m$ .”  
(load factor  $\approx 1/2$ )  
(Pagh, 2001)
- “ $\mathcal{H}$  1-universal,  $S$  sufficiently random,  $m < |U|^{1/3}$   
 $\Rightarrow p_{\text{failure}} = O(1/m)$  for  $|S| = (1 - \delta)m$ .”  
(Mitzenmacher and Vadhan, 2008)

## Related Work

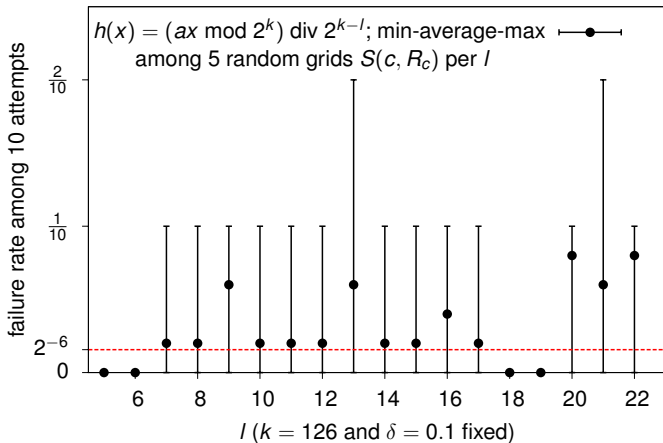
- “ $\mathcal{H} (O(1), O(\log m))$ -universal  
 $\Rightarrow p_{\text{failure}} = O(1/m)$  for each  $S$  of size  $(1 - \delta)m$ .”  
(load factor  $\approx 1/2$ )  
(Pagh, 2001)
- “ $\mathcal{H}$  1-universal,  $S$  sufficiently random,  $m < |U|^{1/3}$   
 $\Rightarrow p_{\text{failure}} = O(1/m)$  for  $|S| = (1 - \delta)m$ .”  
(Mitzenmacher and Vadhan, 2008)
- “ $\mathcal{H}_{k,l}^{\text{mult}}$  (2-universal),  $S$  fully random,  $m > |U|^{11/12}$   
 $\Rightarrow p_{\text{failure}} = 1 - o(1)$  for  $|S| = (1/2)m$ ”  
(load factor  $1/4$  !)  
(Dietzfelbinger and Schellbach, SODA 2009)

# Outline

- 1 Background
- 2 Main Result
- 3 Relevance
- 4 Experimental Results**
- 5 Conclusion

# Experimental Results

Multiplicative class with  $k = 126$  and  $\delta = 0.1$



# Outline

- 1 Background
- 2 Main Result
- 3 Relevance
- 4 Experimental Results
- 5 Conclusion**

# Conclusion

- Cuckoo hashing combined with  $\mathcal{H}_{k,l}^{\text{mult}}$  is unsuitable in any situation where a constant failure probability is not tolerable.
- Care must be taken when interpreting the result by Mitzenmacher and Vadhan—check carefully that the conditions are satisfied.

## Open Problem:

- Prove that cuckoo hashing works with the class of constant degree ( $\geq 2$ ) polynomials over a prime field  $\mathbb{Z}_p$ .

Thank You!