

# Weaknesses of Cuckoo Hashing with a Simple Universal Hash Class: The Case of Large Universes<sup>\*</sup>

Martin Dietzfelbinger and Ulf Schellbach

Technische Universität Ilmenau, Germany

{martin.dietzfelbinger,ulf.schellbach}@tu-ilmenau.de

**Abstract.** Cuckoo hashing was introduced by Pagh and Rodler in 2001 [12]. A set  $S$  of  $n$  keys is stored in two tables  $T_1$  and  $T_2$  each of which has  $m$  cells of capacity 1 such that constant access time is guaranteed. For  $m \geq (1+\varepsilon)n$  and hash functions  $h_1, h_2$  that are  $c \log n$ -wise independent, Pagh [11] showed that the keys of an arbitrary set  $S$  can be stored using  $h_1$  and  $h_2$  with a probability of  $1 - O(1/n)$ .

Here we prove that a family of simple hash functions that can be evaluated fast is *not* sufficient to guarantee this behavior, namely there exists a “bad” set  $S$  of size  $\approx (7/8) \cdot m$  for which the probability that the keys of  $S$  cannot be stored using  $h_1$  and  $h_2$  is  $\Omega(1)$ . Experiments indicate that the bad sets cause the cuckoo scheme to fail with a probability much larger than formally proved in our main theorem.

Our result shows that care must be taken when using cuckoo hashing in combination with very simple hash classes, if a small failure probability is essential since frequent rehashing cannot be tolerated.

**Key words:** data structures, randomized algorithms, cuckoo hashing, universal hash classes, lower bounds

## 1 Introduction

### 1.1 Cuckoo Hashing

Given two hash tables  $T_1, T_2$ , each of size  $m$ , and hash functions  $h_1, h_2$  mapping a *universe*  $U$  of keys to  $[m] = \{0, 1, \dots, m-1\}$ , the two possible positions for a key  $x$  are cell  $h_1(x)$  of  $T_1$  and cell  $h_2(x)$  of  $T_2$ . For a given set  $S \subseteq U$  and hash functions  $h_1, h_2$  we say that  $h_1$  and  $h_2$  are *suitable for*  $S$  if it is possible to store the keys from  $S$  in  $T_1, T_2$  according to  $h_1, h_2$  in such a way that distinct keys are stored in distinct table cells. For a detailed description and the basic analysis of cuckoo hashing, see [14]. In [11] it is shown that if  $m \geq (1+\varepsilon)n$  for some constant  $\varepsilon > 0$  and  $h_1, h_2$  are chosen at random from a  $c \log n$ -wise independent class of hash functions, for a suitable constant  $c$ , then for each  $S$  of size  $n$  the probability that  $h_1$  and  $h_2$  are suitable for  $S$  is  $1 - O(1/n)$ .

---

<sup>\*</sup> Research supported in part by DFG grant DI 412/10-1.

## 1.2 The Hash Function Family

We consider the following hash function family. Let  $1 \leq l \leq k$  and  $U = [2^k]$ . Then  $\mathcal{H}_{k,l}^{\text{mult}} := \{h_a : U \rightarrow [2^l] \mid a \in O_k\}$ , where  $O_k := \{1, 3, 5, \dots, 2^k - 1\}$ , and for  $x \in U$  we let  $h_a(x) := (a \cdot x \bmod 2^k) \operatorname{div} 2^{k-l} \in [2^l] = [m]$ . We refer to this family as the *multiplicative class* [5]. In [5], the multiplicative class is proved to be 2-universal with respect to the following well known generalization of the original notion of universality, which is due to Carter and Wegman [1].

**Definition 1.** *A family  $\mathcal{H}$  of hash functions  $h : U \rightarrow [m]$  is called  $c$ -universal if for arbitrary keys  $x \neq y$  and  $h$  chosen uniformly at random from  $\mathcal{H}$ ,*

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

## 1.3 Related Work

Recently, Mitzenmacher and Vadhan [10] approached the question of the behavior of weak hash classes from the other direction: they showed that if the key set  $S$  is produced by a random process that satisfies certain requirements then weak hash classes (including  $\mathcal{H}_{k,l}^{\text{mult}}$ ) will produce a distribution of hash values that is close to uniform. In particular, in such a situation cuckoo hashing will work well.

In a related paper [6], the authors show that cuckoo hashing will not work well with a set  $S$  chosen randomly if  $S$  is very dense in  $U$ , in the sense that  $m \geq |U|^{1-\gamma}$  for some small constant  $\gamma$ . This seems to contradict [10]. However, the hypotheses of the result in [10] lead to the requirement that  $S$  is not too dense in  $U$ , and hence in [10] no statement is made about the situation investigated in [6].

In [3], Cohen and Kane show that even the property of a hash family to be 5-wise independent is not sufficient to guarantee that cuckoo hashing works well. The hash family constructed there as a counterexample is quite contrived and not suited for being used in practice.

## 1.4 Our Result

The purpose of the present paper is to show the following: even if  $U$  is much larger than  $[m]$ , when applied to cuckoo hashing the multiplicative class has deficiencies in comparison to  $\Omega(\log n)$ -wise independent classes, in the sense that there are structured key sets  $S$ , constructed as a mixture of regular patterns and randomly chosen keys, that will make cuckoo hashing fail with constant probability (in place of the  $O(1/n)$ , resulting from the analysis in [11]). The construction and proofs are totally different from those in [6]. Here again, our result is no contradiction to [10], as we allow only very restricted randomness in the data. In fact, our result implies the existence of a set without any random elements for which cuckoo hashing with the multiplicative class behaves badly.

According to a statement in a recent paper by Mitzenmacher, Kirsch, and Wieder [9], our result shows that even in the case where  $S$  is very sparse in  $U$ , the combination of cuckoo hashing with the multiplicative class will be unsuitable for production systems where a constant failure probability is not tolerable, no matter how small it is. The method proposed in [9] (utilizing a small extra storage, called a *stash*, to circumvent the effect of few keys that obstruct suitability) will not help in this situation.

## 2 Preliminaries and Main Result

### 2.1 The Cuckoo Graph and Bad Edge Sets

The *cuckoo graph* (see e. g. [4]) represents the hash values on a set  $S$  of keys in  $U$  for hash functions  $h_1, h_2: U \rightarrow [m]$ . Its vertices correspond to the cells in tables  $T_1$  and  $T_2$ , and an edge connects the two possible locations  $T_1[h_1(x)]$  and  $T_2[h_2(x)]$  for a key  $x \in S$ . Formally, the cuckoo graph  $G(S, h_1, h_2)$  is defined as an undirected bipartite multigraph  $(V_1, V_2, E)$  with vertex sets  $V_1 = [m]$  and  $V_2 = [m]$ , and edge (multi)set  $E = \{(h_1(x), h_2(x)) \mid x \in S\}$ . We refer to  $G(U, h_1, h_2)$  as the *complete cuckoo graph*.

If  $G(S, h_1, h_2) = (V_1, V_2, E)$ , we call  $E' \subseteq E$  a *bad edge set* if  $|E'|$  is larger than the number of distinct vertices that are incident with edges in  $E'$ . The following lemma will be useful.

**Lemma 1.** *The hash functions  $h_1$  and  $h_2$  are suitable for  $S$  if and only if  $G(S, h_1, h_2)$  does not contain a bad edge set.*

*Proof.* In [4] it is shown that  $h_1$  and  $h_2$  are not suitable for  $S$  if and only if  $G(S, h_1, h_2)$  has a connected component that contains two or more different cycles. It is not hard to see that this condition is equivalent to  $G(S, h_1, h_2)$  having a bad edge set.  $\square$

### 2.2 The Main Result

For hash functions  $h_1, h_2$  and  $S \subseteq U$ , all of which may be the result of a random experiment, we denote the probability that  $h_1$  and  $h_2$  are not suitable for  $S$  as *failure probability*  $p_F = p_F(S, h_1, h_2)$ . As building blocks of our “bad” key sets we define “grid sets”, which are arithmetic progressions in  $U$  with step size  $2^{k-l}$ . Let  $\delta = 1/8$  and  $d := \lceil (1 - \delta)m/3 \rceil$ . Define  $x_i(c) := (c + i \cdot 2^{k-l}) \bmod 2^k$ , for  $c \in [2^k]$  and  $i \in [2^l]$ , and the grid sets

$$G_c := \{x_i(c) \mid i \in [d]\} , \text{ for } c \in [2^k] .$$

To get  $S$ , we perform the following random experiment: choose  $c$  at random from  $O_k$ , and choose a random subset  $R_c$  of  $U - (G_0 \cup G_c)$  of size  $d$ . Then

$$S = S(c, R_c) = G_0 \cup G_c \cup R_c . \tag{1}$$

The purpose of this paper is to establish the following theorem.

**Theorem 1.** *If  $l \geq 14$  and  $k - \log k \geq 3l + 5$ , then for the set  $S = G_0 \cup G_c \cup R_c$  of size  $3d \leq (7/8) \cdot m + 2$  formed by the random experiment as just described and for  $h_{a_1}, h_{a_2}$  chosen from  $\mathcal{H}_{k,l}^{\text{mult}}$  uniformly at random we have  $p_F = \Omega(1)$ .*

The rest of the paper is devoted to the proof of this theorem, where we assume

$$l \geq 14 \text{ and } k - \log k \geq 3l + 5 \quad (2)$$

throughout, particularly in Lemmas 2, 3, 4, 5, 7, and 11. The constant lower bound we establish for  $p_F$  is  $2^{-24}$ . Experiments indicate that the failure probability for the sets  $S$  constructed here is much larger.

### 3 Basic Structure of the Proof

Apart from the grid structure of the set  $S$ , a certain property of hash function pairs is vital in our proof: we say that a pair  $(h_{a_1}, h_{a_2})$  of hash functions from  $\mathcal{H}_{k,l}^{\text{mult}}$  has an *almost uniform distribution* of values for the domain  $D \in \{U, O_k\}$ , if for  $x$  chosen uniformly at random from  $D$  we have

$$\forall (i, j) \in [2^l]^2: \quad \frac{1}{4} \cdot 2^{-2l} \leq \Pr((h_{a_1}(x), h_{a_2}(x)) = (i, j)) \leq 4 \cdot 2^{-2l}. \quad (3)$$

In Sections 4 and 5 we prove the following two lemmas, respectively.

**Lemma 2.** *If (2) holds, then a fraction of more than  $1/7$  of all hash function pairs  $(h_{a_1}, h_{a_2})$  has an almost uniform distribution as in (3) for  $D \in \{U, O_k\}$ .*

**Lemma 3.** *Let  $(h_{a_1}, h_{a_2})$  be a pair with almost uniform distribution for  $D \in \{U, O_k\}$ , assume (2), and let  $S = S(c, R_c)$  be chosen randomly as in (1). Then  $p_F(S) > 2^{-21}$ .*

Once these lemmas are proved, we have proved Theorem 1, because  $2^{-24} < (1/7) \cdot 2^{-21} < p_F = p_F(S(c, R_c), h_{a_1}, h_{a_2})$ .

### 4 Proof of Lemma 2: Many Hash Function Pairs Have an Almost Uniform Distribution

The distribution of hash value pairs for  $D \in \{U, O_k\}$  is represented by the cuckoo graph  $G(D, h_{a_1}, h_{a_2})$ . In [6] the following simple observation was made.

**Observation.** *The set  $\{G(D, h_{a_1}, h_{a_2}) \mid a_2 \in O_k\}$  of cuckoo graphs for fixed  $a_1$  and variable  $a_2$  does not depend on  $a_1$ .*

So, we can assume w. l. o. g. that  $a_1 = 1$ , and it remains to identify a suitable set  $A_2 \subseteq O_k$  of parameters  $a_2$ . Let

$$A_2 := \left\{ a \in O_k \mid \exists x \in O_{(k-l)-(l+2)}: ax \bmod 2^k \in \left\{ \frac{2^{k-l}}{4}, \dots, \frac{2^{k-l}}{2} - 1 \right\} \right\}. \quad (4)$$

We will show (Lemma 4) that each pair  $(h_1, h_{a_2})$ ,  $a_2 \in A_2$ , has an almost uniform distribution for  $D \in \{U, O_k\}$ , and that  $|A_2|/|O_k| > 1/7$  (Lemma 5), which concludes the proof of Lemma 2.

**Lemma 4.** *Each pair  $(h_1, h_{a_2})$ ,  $a_2 \in A_2$ , has an almost uniform distribution for  $D \in \{U, O_k\}$ .*

*Proof.* We define  $\min_D$  and  $\max_D$  as the *minimum cardinality* and *maximum cardinality of a preimage with respect to  $(h_1, h_{a_2})$  for the domain  $D$* , respectively, i. e.  $\min_D := \min\{|\{x \in D \mid (h_1(x), h_{a_2}(x)) = (i, j)\}| : i, j \in [m]\}$  and  $\max_D$  accordingly. If  $x$  is chosen uniformly at random from  $D$ , then for all  $i, j \in [m]$  we have

$$\frac{\min_D}{|D|} \leq \Pr((h_1(x), h_{a_2}(x)) = (i, j)) \leq \frac{\max_D}{|D|}. \quad (5)$$

The cardinality of the preimage of  $(i, j)$  with respect to a uniformly distributing hash function pair is  $\text{avg}_D := |D|/2^{2l}$ , as  $m = 2^l$ . Assume that

$$\frac{\max_D}{\min_D} \leq 4. \quad (6)$$

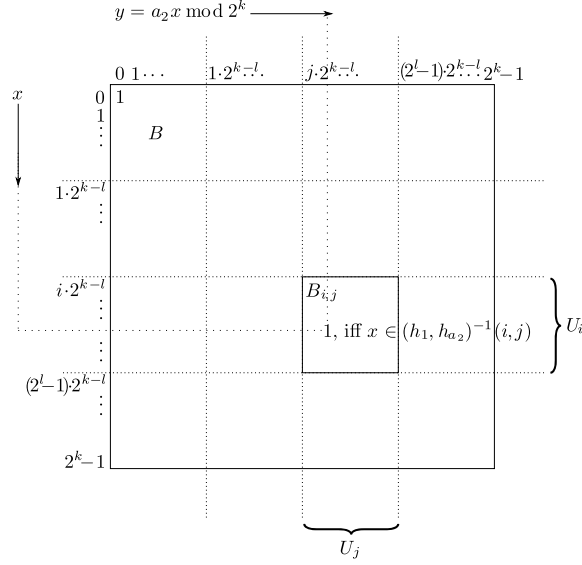
Then we have  $\min_D / |D| \geq \max_D / (4|D|) \geq \text{avg}_D / (4|D|) = (1/4) \cdot 2^{-2l}$ , and similarly  $\max_D / |D| \leq 4 \cdot 2^{-2l}$ , and together with (5) this yields (3). So, it remains to prove (6). Consider the matrix  $B = (b_{x,y})_{x,y \in U}$  that is given by  $b_{x,y} := 1$  if  $y = a_2 x \bmod 2^k$ ,  $b_{x,y} := 0$  otherwise. Let  $U_i := \{i2^{k-l}, \dots, (i+1)2^{k-l} - 1\}$  for  $i \in [2^l]$ . Observe that the number of 1s in the submatrix  $B_{i,j} := (b_{x,y})_{x \in U_i, y \in U_j}$  is  $|(h_1, h_{a_2})^{-1}(i, j)|$  for  $D = U$  (Fig. 1), as well as for  $D = O_k$  if every second row is counted. This follows from the fact that  $z \bmod 2^{k-l} = i$  for all  $z \in U_i$ . Furthermore, observe that the pattern of 1s in all submatrices  $B_i := (b_{x,y})_{x \in U_i, y \in U}$  is equal in the sense that  $B_{i'}$  is just a shifted version of  $B_i$  for arbitrary  $i, i' \in [2^l]$ . This in turn follows from the fact that each row contains exactly one 1, and from the obvious equivalence  $b_{x,y} = 1 \Leftrightarrow b_{(x+1) \bmod 2^k, (y+a_2) \bmod 2^k} = 1$ , which holds for all  $x, y \in U$ .

Thus, for estimating the values of  $\max_D$  and  $\min_D$ , we can restrict ourselves to considering  $B_0$ . We have to show that the number of 1s in arbitrary blocks  $B_{0,j}$  and  $B_{0,j'}$ ,  $j, j' \in [2^l]$ , differs by no more than a factor four.

Fix  $a_2 \in A_2$  and, according to (4),  $x \in O_{(k-l)-(l+2)}$  with  $a_2 x \bmod 2^k \in \{2^{k-l}/4, \dots, 2^{k-l}/2 - 1\}$  arbitrarily. For each  $t \in [x]$ , consider the row sequence  $(x_s^{(t)})_{0 \leq s \leq d_t}$ , where  $x_s^{(t)} := (s \cdot x + t)$ , and

$$d_t = \left\lfloor \frac{2^{k-l} - (t+1)}{x} \right\rfloor \quad (7)$$

is the maximum natural number with  $(s \cdot x + t) \in U_0$ . The 1 in row  $x_s^{(t)}$  resides in column  $y_s^{(t)} := a_2 x_s^{(t)} \bmod 2^k$ , and we refer to the sequence of matrix positions  $(x_s^{(t)}, y_s^{(t)})_{0 \leq s \leq d_t}$  which represent the 1s in rows  $(x_s^{(t)})$  as  $(\text{ones}_s^{(t)})$ . Observe that the set of the 1 positions in  $B_0$  is the disjoint union of the sets  $\{\text{ones}_s^{(t)} \mid s \in \{0, \dots, d_t\}\}$  over all  $t \in [x]$ . Now consider a sequence  $(\text{ones}_s^{(t)})$  for a fixed  $t \in [x]$ . (If for all  $j, j' \in [2^l]$  the number of 1s in  $B_{0,j}$  and  $B_{0,j'}$  given by  $(\text{ones}_s^{(t)})$  differs by no more than a factor four, then the same is true if we sum over all  $t \in [x]$ .)



**Fig. 1.** The number of 1s in  $B_{i,j}$  is  $|(h_1, h_{a_2})^{-1}(i, j)|$  for  $D = U$ , because the single 1 in row  $x$  is in row block  $i = x \operatorname{div} 2^{k-l} = h_1(x)$  and in column block  $j = h_{a_2}(x)$ .

Whenever the sequence  $(\text{ones}_s^{(t)})$  passes a block  $B_{0,j}$ , it hits this block with at least two and at most four successive elements, because by the definition of  $A_2$  the step size  $y$  of the column sequence  $(y_s^{(t)})$  is

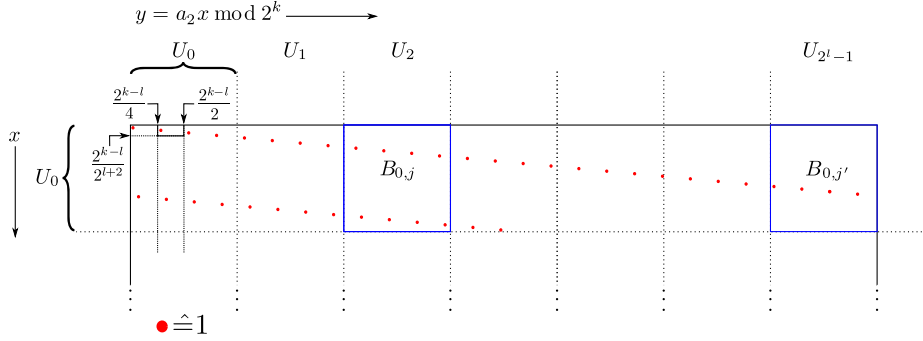
$$y = a_2 x \bmod 2^k \in \{2^{k-l}/4, \dots, 2^{k-l}/2 - 1\}. \quad (8)$$

Furthermore, the sequence  $(\text{ones}_s^{(t)})$  passes each block  $B_{0,j}$ ,  $j \in [2^l]$ , at least once, because the sum of  $d_t$  steps of size  $y$  is greater than  $2^k - 2^{k-l}/2$ , by (7) and (8).

We obtain an upper bound on  $\max_D / \min_D$  as follows. Consider an arbitrary block  $B_{0,j}$ . The sequence  $(\text{ones}_s^{(t)})$  might pass this block only once and hit it with two elements or pass it twice and hit it with four elements. Now assume that there exist blocks  $B_{0,j}$  and  $B_{0,j'}$ ,  $j, j' \in [2^l]$ , such that  $(\text{ones}_s^{(t)})$  passes  $B_{0,j'}$  once and hits it with only two elements, whereas  $B_{0,j}$  is passed twice and each time hit with four elements. In this worst case, the number of 1s, given by  $(\text{ones}_s^{(t)})$ , differs in the two blocks by a factor four, and hence (6) is proved for  $D = U$  (see Fig. 2).

For  $D = O_k$ , the argument is similar, noticing that every second element of  $(x_s^{(t)})$  is odd: if  $t$  is odd, then  $x_s^{(t)}$  is odd for every even  $s$ , and vice versa. Thus, the corresponding sequence of matrix positions which contain the 1s passes every block  $B_{0,j}$  at least once, and hits it with one or two elements, whenever it is passed, and so on.  $\square$

**Lemma 5.**  $|A_2|/|O_k| > 1/7$ .



**Fig. 2.**  $(\text{ones}_s^{(0)})$  in the worst case with respect to  $\max_D / \min_D$

*Proof.* Consider  $A_2$  as in (4). For all  $x \in O_{(k-l)-(l+2)}$  and  $i \in [x]$  we define

$$Q_{x,i} := \left[ \frac{1}{x}(i \cdot 2^k + 2^{k-l-2}), \frac{1}{x}(i \cdot 2^k + 2^{k-l-1}) \right), \quad (9)$$

and  $Q_x := \bigcup_{i \in [x]} Q_{x,i}$  and  $Q := \bigcup_{x \in O_{(k-l)-(l+2)}} Q_x$ . Then  $A_2 = Q \cap O_k$ , where for our purposes the obvious subset relation  $Q \cap O_k \subseteq A_2$  is sufficient. Observe that there exist disjoint half-open intervals  $I_1, \dots, I_t$  of the form  $[a', b')$  and of length  $\geq 2^l$  such that  $Q = \bigcup_{1 \leq j \leq t} I_j$ . This in particular implies that  $t < |Q|/2^l$ . As  $|Q| = |I_1| + \dots + |I_t|$  and each interval  $I_j$  contains at least  $\lfloor |I_j| \rfloor$  natural numbers, of which at least  $\lfloor |I_j|/2 \rfloor$  are odd, we have

$$|A_2| \geq |Q \cap O_k| > \sum_{1 \leq j \leq t} \left( \frac{|I_j|}{2} - 1 \right) = \frac{|Q|}{2} - t > (1 - 2^{-l+1}) \cdot \frac{|Q|}{2},$$

and hence  $|A_2|/|O_k| > (1 - 2^{-l+1}) \cdot |Q|/2^k$ . We show that  $|Q|/2^k > 5/2^5$ . Then  $(1 - 2^{-l+1}) \cdot |Q|/2^k > 1/7$  for  $l \geq 14$  – as desired.

A simple inclusion-exclusion bound, Boole's inequalities, turns out to be helpful to establish a lower bound for  $|Q|/2^k$ .

**Lemma 6 (Boole's inequalities).** *Let  $D_1, \dots, D_r$ ,  $r \in \mathbb{N}$ , be arbitrary events. Then*

$$\sum_{i=1}^r \Pr(D_i) - \sum_{1 \leq i < j \leq r} \Pr(D_i \cap D_j) \leq \Pr\left(\bigcup_{i=1}^r D_i\right) \leq \sum_{i=1}^r \Pr(D_i).$$

In order to apply it, we normalize all values by shrinking the interval  $[0, 2^k]$  to  $[0, 1)$  and working in the measure space that is given by  $[0, 1)$  with the usual Lebesgue measure  $\lambda$ .

Now, in direct correspondence to (9), we define the sets

$$E_{x,i} := \left[ \frac{1}{x}(i + 2^{-(l+2)}), \frac{1}{x}(i + 2^{-(l+1)}) \right), \quad (10)$$

as well as  $E_x := \bigcup_{i \in [x]} E_{x,i}$  and  $E := \bigcup_{x \in O_{(k-l)-(l+2)}} E_x$ . Then  $|Q|/2^k = \lambda(E)$ . The following lemma allows us to apply Boole's lower bound inequality.

**Lemma 7.** *Let  $x, x' \in [2^{(k-l)-(l+2)}] - [2^{(k-l)-(l+3)}]$ ,  $x < x'$ , be coprime. Then*

$$\lambda(E_x \cap E_{x'}) < \frac{3}{16} \cdot 2^{-2l} .$$

The intuitive meaning of this lemma, whose proof can be found in the full paper, is as follows: for the admitted  $x$  and  $x'$ , the probability  $\lambda(E_x \cap E_{x'})$  of  $E_x \cap E_{x'}$  is not much larger than  $\lambda(E_x) \cdot \lambda(E_{x'}) = (2^{-(l+2)})^2 = (2/16) \cdot 2^{-2l}$ , and hence approximately the same as in the case of independence between  $E_x$  and  $E_{x'}$ .

Consider the following fact that was proved by Finsler [7].

**Lemma 8 (Finsler's inequalities [7]).** *Let  $n \in \mathbb{N}$ ,  $n > 1$ , and define  $\pi(n)$  as the number of distinct prime numbers less than or equal to  $n$ . Then*

$$\frac{n}{3 \ln(2n)} < \pi(2n) - \pi(n) < \frac{7n}{5 \ln(n)} .$$

By Finsler's inequalities, we know that the set  $[2^{(k-l)-(l+2)}] - [2^{(k-l)-(l+3)}]$  contains at least  $2^{k-2l-3}/(3 \ln(2^{k-2l-2}))$  distinct prime numbers. Of course these prime numbers are odd and pairwise coprime. For  $k - \log k \geq 3l + 5$  we have  $2^{k-2l-3}/(3 \ln(2^{k-2l-2})) \geq 2^l$ . So, let PR be a set of exactly  $2^l$  distinct prime numbers in  $[2^{(k-l)-(l+2)}] - [2^{(k-l)-(l+3)}]$ . We complete the proof of Lemma 5 as follows:

$$\begin{aligned} \frac{|Q|}{2^k} &= \lambda(E) = \lambda \left( \bigcup_{x \in O_{(k-l)-(l+2)}} E_x \right) && \text{(see (10))} \\ &\geq \lambda \left( \bigcup_{x \in \text{PR}} E_x \right) && (\text{PR} \subseteq O_{(k-l)-(l+2)}) \\ &> \sum_{x \in \text{PR}} \lambda(E_x) - \sum_{\substack{x, x' \in \text{PR}, \\ x \neq x'}} \lambda(E_x \cap E_{x'}) && \text{(Lemma 6)} \\ &> 2^l \cdot 2^{-(l+2)} - \binom{2^l}{2} \cdot \frac{3}{16} \cdot 2^{-2l} && \text{(Lemma 7)} \\ &> \frac{5}{2^5} . \end{aligned}$$

□

## 5 Proof of Lemma 3: $p_F(S)$ Under the Condition of an Almost Uniform Distribution

For the proof of Lemma 3 we consider the cuckoo graph  $G = (V_1, V_2, E) = G(S, h_{a_1}, h_{a_2})$  of the set  $S = S(c, R_c)$ . We show that if  $c$  is suitably chosen then

a large subset of the edges corresponding to  $G_0$  and  $G_c$  in  $G$  form a set of simple paths with disjoint vertex sets. The number of these paths is a random variable  $\Delta$ . Then we prove a lower bound for the probability that we chose a suitable  $c$  and that under the condition of a suitable  $c$  choosing  $R_c$  yields  $\geq \Delta+1$  edges with endpoints on the  $\Delta$  paths, and hence yields a bad edge set. This will conclude the proof of Lemma 3.

In the following we refer to the edge that corresponds to a key  $x_i(c') \in G_{c'}$  as  $e_i(c')$  for arbitrary  $c' \in U$ , and we say that the keys  $x \neq y$  *collide under* the hash function  $h$  if  $h(x) = h(y)$ .

**Lemma 9.** *Each hash function  $h_a \in \mathcal{H}_{k,l}^{mult}$  maps  $G_{c'}$  one-to-one into  $[m]$  for arbitrary  $c' \in U$ .*

*Proof.* Let  $x, y \in G_{c'}$ ,  $x \neq y$ , be arbitrary. We have to show that  $h_a(x) \neq h_a(y)$ . Let  $i$  and  $j$  be the unique numbers in  $[d]$  with  $x = (c' + i \cdot 2^{k-l}) \bmod 2^k$  and  $y = (c' + j \cdot 2^{k-l}) \bmod 2^k$ , and assume w. l. o. g. that  $i < j$ . Then we have  $y = (x + t \cdot 2^{k-l}) \bmod 2^k$  for the positive integer  $t := j - i < 2^l/3$ . Now, on the one hand we have

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

and on the other hand we derive

$$h_a(y) = ((ax \bmod 2^k + at \bmod 2^k \cdot 2^{k-l}) \bmod 2^k) \operatorname{div} 2^{k-l}.$$

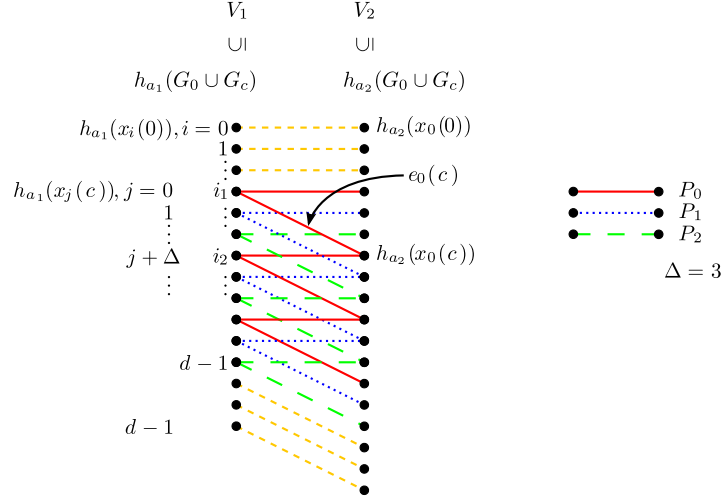
As  $0 < t < 2^l$  and  $a$  is odd,  $t' := at \bmod 2^k$  is neither zero nor a multiple of  $2^l$ . This implies that  $t' \cdot 2^{k-l} \bmod 2^k \geq 2^{k-l}$ , and hence  $h_a(x) \neq h_a(y)$ .  $\square$

Lemma 9 applied for  $c' = 0$  and  $c' = c$ , respectively, yields that the edges corresponding to  $G_0$  and the edges corresponding to  $G_c$  each form a matching of size  $d$ . Imagine each of them as a set of  $d$  parallel lines in increasing order w. r. t. the indices  $i \in [d]$  of the edges  $e_i(0)$  and  $e_i(c)$ , respectively.

Consider  $e_0(c) = (h_{a_1}(x_0(c)), h_{a_2}(x_0(c)))$  and assume that  $x_{i_1}(0)$  and  $x_0(c)$  collide under  $h_{a_1}$ , and that  $x_{i_2}(0)$  and  $x_0(c)$  collide under  $h_{a_2}$ , respectively, for  $i_1 \neq i_2$ , w. l. o. g.  $i_1 < i_2$ . The following Lemma, applied for  $h_a = h_{a_1}$ ,  $\alpha = 0$ ,  $\beta = c$ ,  $i = i_1$ , and  $i' = 0$  as well as for  $h_a = h_{a_2}$ ,  $\alpha = 0$ ,  $\beta = c$ ,  $i = i_2$ , and  $i' = 0$  says that under this assumption there is a sequence of collisions between keys in  $G_0$  and  $G_c$  both with respect to  $h_{a_1}$  and  $h_{a_2}$ .

**Lemma 10.** *Let  $h_a \in \mathcal{H}_{k,l}^{mult}$ , as well as offsets  $\alpha, \beta \in U$ , and indices  $i, i' \in [2^l]$  be arbitrary. If  $x_i(\alpha)$  and  $x_{i'}(\beta)$  collide under  $h_a$  then  $x_j(\alpha)$  and  $x_{j'}(\beta)$  collide under  $h_a$  for all  $j, j' \in [2^l]$  with  $j - j' = i - i'$ .*

The proof of Lemma 10 is a straightforward calculation and can be found in the full paper. So we have  $h_{a_t}(x_j(c)) = h_{a_t}(x_{i_t+j}(0))$  for  $0 \leq j \leq d - i_t - 1$ ,  $t \in \{1, 2\}$ , and hence the two matchings given by the edges of  $G_0$  and  $G_c$  can be merged as depicted in Fig. 3. This reveals the existence of  $\Delta := i_2 - i_1$  simple paths  $P_0, \dots, P_{\Delta-1}$  in  $G$  with disjoint vertex sets. Furthermore, the total number  $|V'_1|$  and  $|V'_2|$  of vertices in  $V_1$  and in  $V_2$  covered by these paths is at least  $d - i_1$ , respectively.



**Fig. 3.** The  $\Delta$  paths in  $G$  for  $e_0(c) = (h_{a_1}(x_{i_1}(0)), h_{a_2}(x_{i_2}(0)))$

Let  $d' := \lceil d/2^7 \rceil + 1$ , and

$$Z := \{(h_{a_1}(x_{i_1}(0)), h_{a_2}(x_{i_2}(0))) \mid i_1, i_2 \in [d'], i_1 \neq i_2\}. \quad (11)$$

For  $e_0(c) \in Z$  we have just proven the existence of  $\Delta \leq \lceil d/2^7 \rceil$  simple paths  $P_0, \dots, P_{\Delta-1}$  in  $G$  with disjoint vertex sets, which in total cover the vertex set  $V'_1 \subseteq V_1$  and  $V'_2 \subseteq V_2$  of size  $\geq d - \lceil d/2^7 \rceil$ , respectively. We assume w. l. o. g. that  $S$  is the result of a random experiment where first  $c \in O_k$ , and then  $R_c \subseteq U - (G_0 \cup G_c)$  is chosen uniformly at random. So,

$$\begin{aligned} \Pr(e_0(c) \in Z) &= \sum_{(i,j) \in Z} \Pr((h_{a_1}, h_{a_2})(c) = (i, j)) \\ &\geq |Z| \cdot \frac{1}{4} \cdot 2^{-2l} && \text{(Lemma 2)} \\ &\geq \left(\frac{d}{2^7}\right)^2 \cdot 2^{-2l-2} && \text{((11), Lem. 9)} \\ &\geq \frac{(1-\delta)^2}{9 \cdot 2^{16}}. && (d = \lceil (1-\delta)2^l/3 \rceil) \quad (*) \end{aligned}$$

If  $e_0(c) \in Z$  and if the uniform random choice of  $R_c \subseteq U - (G_0 \cup G_c)$  yields  $\geq \Delta + 1$  edges in  $V'_1 \times V'_2$ , then these edges together with the edges of  $P_0, \dots, P_{\Delta-1}$  obviously form a bad edge set. We refer to the event that choosing  $R_c$  yields  $\geq \Delta + 1$  edges in  $V'_1 \times V'_2$  as  $F$ .

**Lemma 11.**  $\Pr(F \mid e_0(c) \in Z) \geq 1 - \exp\left(-\left(\frac{1-\delta}{3}\right)^3 2^{l-9}\right)$ .

This is proved by an application of Chernoff bounds. For the details, see the full paper. Now we complete the proof of Lemma 3 as follows.

$$\begin{aligned}
p_F(S) &\geq \Pr(e_0(c) \in Z) \cdot \Pr(F \mid e_0(c) \in Z) \\
&\geq \frac{(1-\delta)^2}{9 \cdot 2^{16}} \cdot \left(1 - \exp\left(-\left(\frac{1-\delta}{3}\right)^3 2^{l-9}\right)\right) && ((*), \text{ Lemma 11}) \\
&\geq \left(\frac{7}{3}\right)^2 2^{-22} \left(1 - \exp\left(-\left(\frac{7}{3}\right)^3 2^{l-18}\right)\right) && (\delta = 1/8) \\
&> 2^{-21} . && (l \geq 14)
\end{aligned}$$

□

## 6 Experiments

We implemented cuckoo hashing in a straightforward way, using the random number generator class Mersenne Twister from the colt distribution for both hash functions and key sets.<sup>1</sup> We carried out experiments that were meant to obtain estimates of the failure probability  $p_F$  by counting average failure frequencies among 5 independently and uniformly random chosen grid based sets  $S(c, R_c)$ , as considered in the proof of Theorem 1, of size  $(1-\delta)m$ , each set inserted 10 times with independently and uniformly random chosen hash functions. This was repeated several times for fixed  $k = 126$  and  $\delta = 0.1$ , and for each  $l \in \{1, 2, \dots, 22\}$ . The result is depicted in Fig. 4.

It can be seen from Fig. 4 that for a randomly chosen grid based set  $S$  and hash functions  $h_1, h_2$  chosen uniformly at random from the multiplicative class  $p_F$  at least for some  $l$  appears to be much larger than the constant from Theorem 1: the bound of Theorem 1 is  $2^{-24}$ , whereas we see a failure rate of 8 or 9 percent for set sizes of about one million. The deviation from this failure rate for  $l < 20$  may to some extent be due to the small table sizes tested. We do not yet have a good explanation for the variation in the failure rate.

## References

1. Carter, L., Wegman, M.N.: Universal Classes of Hash Functions. *J. Comput. Syst. Sci.* 18, 143–154 (1979).
2. Chor, B., Goldreich, O.: Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity. *SIAM J. Comput.* 17, 230–261 (1988).
3. Cohen, J., Kane, D.M.: 6.856 Project: Bounds on the Independence Required for Cuckoo Hashing. <http://web.mit.edu/dankane/www/Independence%20Bounds.pdf>.
4. Devroye, L., Morin, P.: Cuckoo Hashing: Further Analysis. *Inf. Process. Lett.* 86, 215–219 (2003).

<sup>1</sup> <http://acs.lbl.gov/~hoschek/colt/>

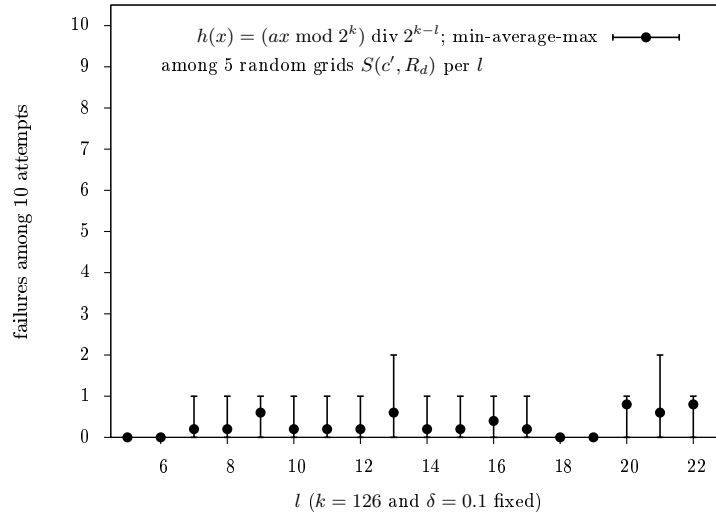


Fig. 4. Constant failure probability for the multiplicative class

5. Dietzfelbinger, M., Hagerup, T., Katajainen, J., Penttonen, M.: A Reliable Randomized Algorithm for the Closest-pair Problem. *J. Algorithms* 25, 19–51 (1997).
6. Dietzfelbinger, M., Schellbach, U.: On Risks of Using Cuckoo Hashing with Simple Universal Hash Classes. *To appear in: Proc. 20th Annual ACM-SIAM Symp. on Discrete Algorithms*. SIAM, Philadelphia (2009).
7. Finsler, P.: Über die Primzahlen zwischen  $n$  und  $2n$ . *Festschrift zum 60. Geburtstag von Prof. Dr. Andreas Speiser*, 118–122. Füssli, Zürich (1945).
8. Hagerup, T., Rüb, C.: A Guided Tour of Chernoff Bounds. *Inf. Process. Lett.* 33, 305–308 (1990).
9. Kirsch, A., Mitzenmacher, M., Wieder, U.: More Robust Hashing: Cuckoo Hashing with a Stash. In: *Proc. ESA 2008*. LNCS, vol. 5193, pp. 611–622. Springer, Heidelberg (2008).
10. Mitzenmacher, M., Vadhan, S.: Why Simple Hash Functions Work: Exploiting the Entropy in a Data Stream. In: *Proc. 19th Annual ACM-SIAM Symp. on Discrete Algorithms*, pp. 746–755. SIAM, Philadelphia (2008).
11. Pagh, R.: On the Cell Probe Complexity of Membership and Perfect Hashing. In: *Proc. 33rd Annual Symp. on Theory of Computing*, pp. 425–432. ACM Press, New York (2001).
12. Pagh, R., Rodler, F.F.: Cuckoo Hashing. In: *Proc. ESA 2001*. LNCS, vol. 2161, pp. 121–133. Springer, Heidelberg (2001).
13. Pagh, A., Pagh, R., Ruzic, M.: Linear Probing with Constant Independence. In: *Proc. 39th Annual ACM Symp. on Theory of Computing*, 318–327. ACM Press, New York (2007).
14. Pagh, R., Rodler, F.F.: Cuckoo Hashing. *J. Algorithms* 51, 122–144 (2004).
15. Zuckerman, D.: Simulating BPP Using a General Weak Random Source. *Algorithmica* 16, 367–391 (1996).