Cryptology : study of SipHash-2-4

Maxime Raynal,

April 3, 2018

This project is about the implementation and search for collisions for the SipHash-2-4

1 cf sip_h.c

2 The key in SipHash-2-4 is 128 bits large, so we have a key space with a size of 2^{128}. It is not ridiculous, but still too small for today's standarts, since finding a collision would cost only the square root of this size, 2^{64}, which is not big enough. So it is definitly possible to find a collision. To search the entire key space would still be costy, but eventually not out of reach. That's why we think it is wiser to generally chose larger key spaces, with keys at least 512 bits, if not more.

3 We chose a very simple way to build such an injection :
$\Phi : \begin{cases} 2^{64} \to 2^{128} \\ k \mapsto k||k \end{cases}$

4 Here is the implementation we decided to use to store the results. It is a four layers hash table.

```
typedef union node_t{
  union node_t *children[256];
  int present[256];
} node;

typedef node *tree;
```

This structure is a 32-ary tree on the first three levels, the last level being used to store the results in the integer array *present*.

To allow for a simpler memory allocation management, we do store all the tree nodes in lists of arrays.

```
struct node_list_t {
  struct node_list_t *next;
  node *node_array;
};
typedef struct node_list_t *node_list;
```