Generalized birthday problem

Application description

This application solves a problem related to the generalized birthday problem. Wagner (2002) proposed a version of the generalized birthday problem together with an algorithm which can solve it. Later the problem and the algorithm were slightly changed and a proof-of-work called Equihash was created. (Biryukov and Khovratovich 2017) A cryptocurrency Zcash consequently used this proof-of-work in its protocol. (Bowe, Hornby, and Wilcox 2018)

My application solves the following problem. Let $n, k \in \mathbb{N}$, $k \geq 3$ and $N = 2^{n/k+1}$. We are given N strings $X_1, \dots, X_N$ consisting of n bits. We are asked to find 2^k strings such that

$$X_{i_1} \oplus X_{i_2} \oplus \cdots \oplus X_{i_{2^k}} = 0,$$

where $\oplus$ is the XOR operator.

The algorithm that solves this problem goes as follows:

1. Create an array of strings $X_1, \dots, X_N$ and an array of indices $1, \dots, N$.

2. In the first step out of k steps search for collisions in the first n/k bits of strings $X_1, \dots, X_N$. Replace the old array of indices with a new one such that each its element is an unordered set $\{i, j\}$ where i and j are the indices of strings whose first n/k bits are the same.

3. In the second step out of k steps search for collisions in the next n/k bits of all strings $X_i \oplus X_j$ where i and j are the indices of strings stored in the array of indices from the first step. Replace the old array of indices with a new one such that each its element is an unordered set $\{i, j, k, l\}$.

4. In all remaining steps out of k steps follow the same rules as in the first two steps.

5. The array resulting from the last step is an array containing solutions of the problem.

The application takes as an input two numbers n and k, respectively. The output is a list of randomly chosen strings $X_1, \dots, X_N$ and a list of solutions of the described generalized birthday problem with these strings.

First, a variable *list* is initialised with random integers from such range that these integers consists of n bits. In each of k steps of the algorithm a new list of indices with collisions in the k-th part of integers is created and put into a variable *tmpIndices*. In the next step of the algorithm only these indices are searched for collisions.

References

Biryukov, Alex and Dmitry Khovratovich (2017). *Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem.* URL: https://ledgerjournal.org/ojs/index.php/ledger/article/download/48/65.

Bowe, Sean, Taylor Hornby, and Nathan Wilcox (2018). *Zcash Protocol Specification.* URL: https://github.com/zcash/zips/blob/master/protocol/protocol.pdf.

Wagner, David (2002). *A Generalized Birthday Problem.* URL: https://link.springer.com/content/pdf/10.1007/3-540-45708-9_19.pdf.