Some Collisions for the FNV2

Yet another application of LLL

Hosein Hadipour[1]

University of Tehran, Iran, hosein.hadipour@protonmail.com

Abstract. In this document we discuss about the solution of the problem 4 of the NSUCRYPTO 2018 competition. This problem is about a plain hash function called FNV2, which is derived from a real non-cryptographic hash function FNV-1a.

Keywords: LLL algorithm, FNV1-a, FNV2, SageMath

Introduction

In this document we want to find some collisions for a plain non-cryptographic hash function called FNV2. The FNV2 is a simplified version of the FNV1-a which uses modular addition instead of the XOR operation.

1 FNV2 hash function

The FNV2 hash function is derived from the hash function FNV-1a. FNV2 processes a message x composed of bytes $x_1, x_2, \ldots, x_n \in \{0, 1, \ldots, 255\}$ in the following way:

- $h \leftarrow h_0$
- for $i = 1, 2, \ldots, n:\ h \leftarrow (h + x_i)g \mod 2^{128}$;
- return h.

Here $h_0 = 144066263297769815596495629667062367629$, and $g = 2^{88} + 315$.

We want to find some collisions for the FNV2, that is, two different messages x and x' such that $\texttt{FNV2}(x) = \texttt{FNV2}(x')$. This is actually the problem number 4 of the second round of the NSUCRYPTO-2017 olympiad.

2 From the hash collision problem to the LLL algorithm

Firstly, it is clear taht

$$\texttt{FNV2}(x_1, x_2, \ldots, x_n) = (h_0 g^n + x_1 g^n + x_2 g^{n-1} + \ldots + x_n g) \mod 2^{128}.$$

Next, it is sufficient to solve the equation

$$z_1 g^{n-1} + z_2 g^{n-2} + \ldots + z_n g^0 \equiv 0 \mod 2^{128} \tag{1}$$

in $z_1, z_2, \ldots, z_n \in \{-255, \ldots, 255\}$ not equal to zero simultaneously. Indeed, $z_i = x_i - y_i$ for some $x_i, y_i \in \{0, \ldots, 2555\}$, and

$$\texttt{FNV2}(x_1, x_2, \ldots, x_n) - \texttt{FNV2}(y_1, y_2, \ldots, y_n) = g(z_1 g^{n-1} + z_2 g^{n-2} + \ldots + z_n g^0) \equiv 0 \mod 2^{128}.$$

Since $\gcd(g, 2^{128}) = 1$, we can multiply two sides of the above equivalence by g^{-1} and derive the 1.

Therefore the purpose is to construct a polynomial such that g is its root. Note that this is not a natural integer relation problem, since the addition in the above equations is modular addition.

Let us define integer vectors $e^0, \ldots, e^n$ of length $n + 1$ in the following way:

$$e^0 = (\underbrace{0, \ldots, 0}_{n}, t.2^{128}), \text{where } t \text{ is a small integer},$$

$$e^i = (\underbrace{0, \ldots, 0}_{i-1}, 1, \underbrace{0, \ldots, 0}_{n-i}, g^{n-i} \mod 2^{128}), \text{where} i \in \{1, \ldots, n\}.$$

Let us add z_0 to $z_1, \ldots, z_n$ and consider the linear combination

$$l_z = z_0 e^0 + \ldots + z_n e^n = (z_1, \ldots, z_n, z_0.t.2^{128} + z_1.g^{n-1} + z_2.g^{n-2} + \ldots + z_n.g^0).$$

To solve the problem it is sufficient to find a liner combination l_z with $z_1, \ldots, z_n \in \{-255, \ldots, 255\}$ and zero last coordinate. This can be done using `LLL` algorithm. It is a lattice reduction algorithm that can find a short nearly orthogonal basis $\langle e^0, \ldots, e^n \rangle$. Obtaining such an `LLL`-reduced basis, we check if it contains a vector l_z with desired properties[1]. A `SageMath` [2] code which in the above solution has been implemented can be found here: `https://github.com/hadipourh/fnv2`.

References

[1] Anastasiya Gorodilova, Sergey Agievich, Claude Carlet, Evgeny Gorkunov, Valeriya Idrisova, Nikolay Kolomeec, Alexandr Kutsenko, Svetla Nikova, Alexey Oblaukhov, Stjepan Picek, et al. Problems and solutions of the fourth international students' olympiad in cryptography nsucrypto. *arXiv preprint arXiv:1806.02059*, 2018.

[2] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 8.2.0)*, 2019. `https://www.sagemath.org`.