Feistel Cipher with Hash Round Function

Cyril Dever
Edgewhere

January 31, 2021

Abstract

We needed an obfuscation tool to secure some data with an almost Format-Preserving Encryption schema.

I. Introduction

When ...

II. Description

We herein define $\mathfrak{F}$ our own implementation of a Feistel block cipher[1] we use for both encryption and decryption stages. It's an almost Format-Preserving Encryption scheme, "almost" because it depends on the size of the input; if the latter is of even length, then the output will preserve its size; otherwise, we'd pad it — see (2).

The formal algorithmic description provided by Wikipedia[1] is as follows:

- Let $N = n + 1$ be the number of rounds, $K_0, K_1, ..., K_n$ the keys associated with each round and $F : \omega \times \mathcal{K} \mapsto \omega$ a function of the $(words \times keys)$ space to the *words* space.
- For each step $i \in [0..n]$, note the encrypted word in step i, $m_i = L_i \parallel R_i$ with

$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= L_i \oplus F(L_i, K_i) \end{aligned}$$

- $m_0 = L_0 \parallel R_0$ is the unciphered text, and $m_{n+1} = L_{n+1} \parallel R_{n+1}$ the ciphered word.

For the round function F_i, we don't actually use one different key per round but rather a single key K and the `SHA-256` hash function applied to the byte array of the right part to which the key is bytewise-added at each round[2]:

$$\begin{aligned} F_i : \omega \times \mathcal{K} &\to \omega \\ (x, K) &\mapsto \texttt{SHA-256}\,(x + \mu(K)) \end{aligned} \tag{1}$$

where $x + \mu(K)$ represents the addition of the UTF-8 charcodes of both terms[3].

As mentioned, in case the length of the input in is odd, we add one padding character to the left beforehand because our cipher is a balanced implementation:

$$in \leftarrow \begin{cases} \textbf{if } |in| \bmod 2 \neq 0 \\ \quad \textbf{then } \texttt{PAD\_CHAR} \parallel in \\ \textbf{else } in \end{cases} \tag{2}$$

$$\begin{aligned} &\Rightarrow in := L \parallel R \\ &\quad \text{with } |L| = |R| = |in| \div 2 \end{aligned}$$

The number of round N is set to 10 anywhere $\mathfrak{F}$ is used in the oblivion process.

Note that it's been proved [2] that, for such an implementation of the Feistel block cipher, four rounds of permutations are enough to make it "strong", making our choice still very fast and two and half times stronger.

Recall. One of the main advantage of using this Feistel block cipher construction is that encryption and decryption are similar:

$$out = \mathfrak{F}(in, K) \iff in = \mathfrak{F}(out, K)$$

Figure 1 provides a graphical representation of $\mathfrak{F}$.

Contents

[1] `https://en.wikipedia.org/wiki/Feistel_cipher`

[2] a "masking" operation $\mu()$ being applied first on K to make sure to use a key of the same length than the input

[3] eg. $\texttt{a} \leftarrow 61, \texttt{b} \leftarrow 62 \Rightarrow \texttt{a} + \texttt{b} \leftarrow 123 \mapsto \texttt{b01111011}$

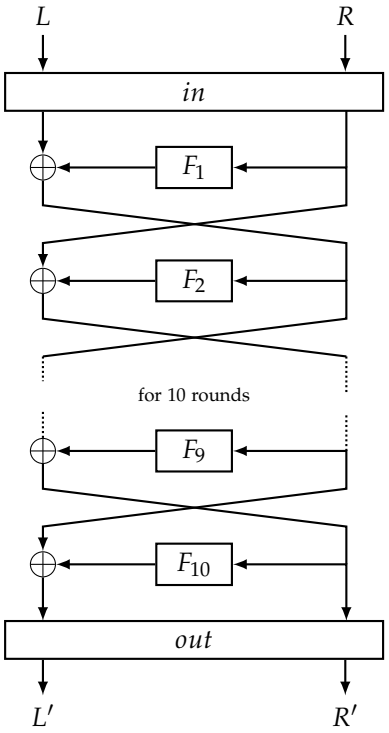

Figure 1: *Feistel block cipher* $\mathfrak{F}$

References

[1] Horst Feistel. *Cryptography and Computer Privacy*, Scientific American, 1973.

[2] Michael Luby, Charles Rackoff. *How to Construct Pseudorandom Permutations from Pseudorandom Functions*, SIAM Journal on Computing, 1988.