Rainbow Table File Format

File format of rainbow table with .rt file name extension

To explain the file format of .rt rainbow table used by RainbowCrack, we generate a simple rainbow table with following command:

```
rtgen md5 loweralpha-numeric 1 7 0 3800 3 0
```

This command finishes instantly, and the binary data of the generated rainbow table will be:

```
00000000: 00 00 00 00 00 00 00 00 30 02 3C 61 01 00 00 00
00000010: 01 00 00 00 00 00 00 00 77 09 F0 98 06 00 00 00
00000020: 02 00 00 00 00 00 00 00 14 49 40 CB 0A 00 00 00
```

Rainbow table consists of lots of rainbow chains. The size of each rainbow chain is 16 bytes, so the size of the table above with 3 rainbow chains will be 48 bytes. Each rainbow chain consists of an 8 byte start point (marked as red) and an 8 byte end point (marked as green).

Both the start point and the end point are 64-bit unsigned integer in little endian, representing a plaintext.

In this example, the charset is "abcdefghijklmnopqrstuvwxyz0123456789" with the plaintext length range from 1 to 7. So 0 stands for plaintext "a", 1 stands for plaintext "b", 35 stands for plaintext "9", 36 stands for plaintext "aa", and 80603140211 stands for plaintext "9999999".

The start point is generated by the rtgen program, and the end point is computed based on the start point.

File format of rainbow table with .rtc file name extension

The .rt rainbow table uses 64 bits for the start point, and another 64 bits for the end point.

The .rtc rainbow table introduced in RainbowCrack 1.4 uses configurable number of bits for the start point and end point. As an example, if the start point uses 25 bits and the end point uses 31 bits, one rainbow chain requires 7 bytes.

To support this level of flexibility, a simple 32 bytes file header is used in .rtc rainbow table:

```
struct RTCFileHeader

{

    unsigned int  uVersion; // 0x30435452

    unsigned short   uIndexSBits; // number of bits (1 to 64) used
to store the encoded start point
```

```
    unsigned short   uIndexEBits; // number of bits (1 to 64) used
to store the encoded end point
    uint64    uIndexSMin;
    uint64    uIndexEMin;
    uint64    uIndexEInterval;
};
```

Assume the encoded value of start point is s, and the encoded value of end point is e. Then the actual start/end point will be:

```
actual start point = uIndexSMin + s
actual end point = uIndexEMin + uIndexEInterval * i + e
```

In the expression above, i is the index of the rainbow chain in a .rtc rainbow table. Value of the first rainbow chain will be 0, value of the second rainbow chain will be 1, and so on.

All encoded & packed rainbow chains follow the file header. Size of each rainbow chain will be:

```
byte number of each encoded & packed rainbow chain = (uIndexSBits
+ uIndexEBits + 7) / 8
```

The start point generation algorithm

In 1.3 and later versions of RainbowCrack, the **chain_num** and **part_index** parameters of rtgen command are used to generate start points:

```
rtgen hash_algorithm charset plaintext_len_min plaintext_len_max
table_index chain_len chain_num part_index
```

The start points used in a rainbow table are:

```
(chain_num * part_index + 0) % key_space
(chain_num * part_index + 1) % key_space
(chain_num * part_index + 2) % key_space
......
(chain_num * part_index + chain_num-1) % key_space
```

Maximum key space

All versions of RainbowCrack software use 64-bit to store the start point and end point of rainbow chains. So the maximum key space is:

```
2^64 - 1 = 18446744073709551615
```