5
Stegocontainer
Element
1
2
3
4
Bit index
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
PRNG BIT (n)
4
RSD matrix
Target bit index
RND IN 0..LAST
RND IN 0..LAST
RND IN 0..LAST
RND IN 0..LAST
RND IN 0..LAST
RND IN 0..LAST
RND IN 0..LAST
RND IN 0..LAST
RND IN 0..LAST
RND IN 0..LAST
Source bit index
0 1 2 3 4 5 6 7 8 LAST BIT
3
Bit selector
Target bit index
RND IN 0..7
RND IN 0..7
RND IN 0..7
RND IN 0..7
RND IN 0..7
RND IN 0..7
RND IN 0..7
RND IN 0..7
Source bit index
0 1 2 3 4 5 6 7
2
Source bit block
Index
0 1 2 3 4 5 6 7
1
Stegodata
Source array
Bit index
0 1 2 3 4 5 6 7 8 LAST BIT
Encyption layer
Compression layer

1. Stegodata prepare stage. Using full data size we apply compressor (for example - LZMA), then we apply symmetric cipher (block cipher, or stream cipher). Next, prepared data need to be passed as bit array to the next stage.

2. Then we separate bit array into blocks of 8 bits.

3. In third stage we can use bit mixer. Bit mixer can be user defined. Idea of bit mixer is in passing source bit of range 0..7 into random bit in range 0..7. Mapping Source[n] := Target[Mixer[n]] is secret element of scheme.

4. Ranom secret distribution is based on subtitution table (or matrix) of bit mapping, generated by cryptographic RNG. Idea of RSD is in non-lineral, key-based random mapping source bit into bit of stegocontainer. So, when full source message is M sized as Len, and stego container (usable block) is S, we generate matrix RSD sized as Len, when any element of matrix is in range 0..Len-1. To get position of any bit of stegomessage in RSD mode in target container, we calculate: S[RSD[index]] := M[index]

5. RSD does not impose strict requirements to LSB variant. Typicaly, whe can use 2-3 least significant bits of RGB image colors: when examining image container as bit array (so, 6-9 bits on pixel in 24-bit mode), we can apply both bit mixer and RSD mode. JPEG Steganography can also support RSD mode. We simply need DCT table modification. Regardless of the destination, LSB is based on bit testing. When we need 1 (true) in stegomessage, we set least significant bit in target stegocontainer's numeric value at right index. When apply current selected bit to stegocontainer, we need to encrypt it by simple XORing with current bit, generated by cryptographic-safe RNG (stream cipher).

(c) 2013 Alexander Myasikov
Web: ru-steganograpgy.narod.ru
E-mail: amsoftware@ya.ru