Cryptanalysis of Magenta

Eli Biham[1], Alex Biryukov[2], Niels Ferguson[3], Lars R. Knudsen[4], Bruce Schneier[3] and Adi Shamir[5]

[1] Computer Science Dept., Technion, Israel Institute of Technology, Haifa, Israel 32000

[2] Applied Mathematics Dept., Technion, Israel Institute of Technology, Haifa, Israel 32000

[3] Counterpane Systems

[4] Dept. of Computer Science, University of Bergen, Norway

[5] Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel

Magenta [1] is an encryption algorithm submitted for AES by Deutsche Telekom AG. In this note we cryptanalyze Magenta, and any algorithm of the same structure and key schedule.

We refer the reader to the figure on slide 7 of the Magenta presentation (given in the handouts at the first AES candidate conference). This figure describes the block structure and key schedule for 128-bit keys. Magenta is a Feistel cipher with 6 rounds, in which the key is divided into two halves, called K_1 and K_2, and these halves are used in the following way: K_1 is used in rounds 1,2,5,6, and K_2 is used in rounds 3 and 4. We use the following notation for the intermediate values during encryption: X_0 is the plaintext, X_1 is the data after one round, X_i is the data after i rounds, and X_6 is the ciphertext. The data X_i is divided into two halves: X_i^T is the top half of the data, and X_i^B is the bottom half.

We first present a chosen plaintext attack using 2^{64} chosen plaintexts and requiring 2^{64} steps of analysis.

1. Choose an arbitrary plaintext X_0.
2. Request the ciphertext X_6 of X_0 under the unknown key K.
3. Try all 2^{64} possible values K'_1, and for each compute the following:
   (a) Partially encrypt X_0 for the first two rounds to get a candidate for X_2.
   (b) Choose an arbitrary X'_2 such that $X_2'^T = X_2^T$.
   (c) Partially decrypt X'_2 under the trial subkey K'_1 to get X'_0.
   (d) Request the ciphertext X'_6 of X'_0 under the unknown key K.
   (e) Partially decrypt X_6 and X'_6 under the trial subkey K'_1 to get X_4 and X'_4.
   (f) From X_2 and X_4 compute the output of the E function of round 3, and similarly for X'_2 and X'_4.
   (g) Reject the trial subkey K'_1 if the outputs are different.
4. Make a list of all the keys that passed the equality test.

The correct key must be on the list, since equality of the two inputs of the E function must cause the output to be equal as well. It is expected that the list contains only a few candidates, and the wrong candidates can be easily discarded using one additional trial encryption.

This attack can be converted into a known plaintext attack using only 2^{33} known plaintexts but 2^{97} steps of analysis. In this attack the attacker receives the 2^{33} known plaintexts and their corresponding ciphertexts and searches for collisions of X_2^T. The attack is as follows:

1. Try all the 2^{64} possible values K_1' of K_1, and for each compute the following:
   (a) Partially encrypt all 2^{33} X_0's for the first two rounds to get candidates for X_2.
   (b) Search for collisions of X_2^T in the received results.
   (c) Partially decrypt, for the last two rounds, the pairs of ciphertexts X_6 of the colliding X_2^T to get candidates for X_4.
   (d) From the X_2's and X_4's compute the output of the E function of round 3.
   (e) Reject the trial subkey K_1' if the outputs are different.
2. make a list of all the keys that passed the equality test.

It is interesting to note that the same attack applies to the larger key sizes with the same reduction in complexity over brute force. We guess all the subkeys except for the subkey used in the middle two rounds, and perform the same key recovery attack. A 192-bit key can be found using 2^{128} chosen plaintexts within 2^{128} steps, or using 2^{33} known plaintexts within 2^{161} steps. A 256-bit key can be found using 2^{128} chosen plaintexts within 2^{192} steps, or using 2^{33} known plaintexts within 2^{225} steps.

We would also wish to note that due to the symmetry of the key scheduling, encryption and decryption are identical except for the order of the two halves of the plaintexts and ciphertexts. Therefore, given a ciphertext, one can decrypt it by swapping its two halves, reencrypting the result, and swapping again. Note that this attack uses an adaptive access to an encryption device and cannot be used to recover the key. Such a property is undesirable in various scenarios. Furthermore, the symmetry in the key schedule of all versions of Magenta also introduces 2^{64} "weak plaintexts" for each value of the secret key. More precisely, for each value of the secret key there are 2^{64} plaintexts for which the ciphertexts are just the swapped values of the plaintexts. To see this, consider r-round Magenta for $r = 6, 8$. Consider one of the 2^{64} ciphertexts, say $\tilde{c}$, with equal 64-bit halves after $r/2$ rounds of encryption. For any value of the secret key, it follows that the decryption of $\tilde{c}$ from round $r/2$ to the plaintext equals the encryption of $\tilde{c}$ from round $r/2$ to the plaintext except for a swapping of the halves, and the result follows.

We are grateful to Michael Jacobson, Jr. for his lucid presentation of the Magenta algorithm and for patiently answering our questions regarding cryptanalysis.

References

1. M.J. Jacobson, Jr., Klaus Huber. "The Magenta Block Cipher Algorithm". Presented at the first Advanced Encryption Standard Candidate Conference, August 1998. Available from `http://www.nist.gov/aes`.