A Weight Based Attack on the CIKS-1 Block Cipher

Brian J. Kidney, Howard M. Heys, Theodore S. Norvell
Electrical and Computer Engineering
Memorial University of Newfoundland
{bkidney, howard, theo}@engr.mun.ca

November 12, 2003

Abstract

In 2002, Moldovyan and Moldovyan introduced a cipher with security based mainly on data-dependent permutations (DDPs) called CIKS–1[1]. The goal of the cipher was to exploit the speed and simplicity of DDPs to create a fast hardware-oriented block cipher. This paper examines the properties of DDPs. In particular, it is noted that these structures do not change the Hamming weight of the data. Using this fact, we introduce an attack on CIKS–1 implementations using weak (i.e. low weight) keys which exploit low weight inputs to allow the determination of individual subkeys in CIKS–1.

1 Introduction

In recent years, data–dependent structures have gained interest in cryptography. The first major cipher to use these primitives was RC5 [2]. RC5 depends heavily on the use of data-dependent rotations (DDRs) for security, incorporating only these primitives and an expanded key array in the cipher. The use of DDRs in this relatively simple cipher have been shown to reduce the ability to determine the cipher key through linear and differential cryptanalysis [3]. Due partly to this fact, Data–Dependent Permutations (DDPs), of which DDRs are a subset, have become increasingly popular in cryptographic study. Two of the final candidate ciphers in the Advanced Encryption Standard (AES) process, RC6 and MARS, used DDRs in combination with other cryptographic primitives to produce strong ciphers, resistant to both linear and differential cryptanalysis.

In January 2002, Moldovyan and Moldovyan proposed a new 8–round cipher based mainly on DDPs [1]. CIKS–1 was presented as a fast, hardware–oriented cipher. It relies on DDPs for their speed in hardware and is designed to lack precomputation of key scheduling. Preliminary analysis of the cipher showed that it can easily obtain speeds of 2Gb/s and was resistant to both linear and differential cryptanalysis.

In [4], a chosen plaintext attack is presented on a reduced 5–round version of the CIKS–1 cipher. The attack uses the data's parity and chosen inputs to cancel out the effect of the first and last rounds of the cipher and reveal the last subkey. The authors estimated the time complexity of this attack to be $2^{65.7}$. Although the attack applied to all possible keys, its success was limited in the number of rounds that could be attacked.

This paper will look at the data dependent permutations used in CIKS–1. Particularly, it will focus on how the DDPs affect the Hamming weight of the ciphertext produced from the cipher. Using the facts that the weight of the ciphertext depends mainly on the subkeys of the cipher and that there is no specified key schedule in the paper, it is shown that there is potential for a class of weak keys which can reveal information about the first or last round subkey. An attack is then presented which uses this information to reduce the search space for a brute–force attack on the first subkey.

2 The CIKS–1 Cipher

The CIKS–1 cipher is a fast, hardware–oriented cipher, with its principle security component being data–dependent permutations. It is a block cipher with block size 32–bits. The cipher is comprised of 8 rounds, each with a 64–bit subkey for a total key size of 256–bits. A single round of the cipher is shown in Figure 1. The solid lines in the diagram show the flow of data and the dashed lines are control vectors. Permutations are label $P_{n/m}$, where n is the number of bits permuted and m is the number of bits of control.

2.1 Data–Dependent Permutations

The label data–dependent permutations refers to a large set of functions. Basically, a data–dependent permutation includes any permutation of data which is directly influenced by another piece of the data. In RC5, the DDP is a simple rotation of one half of the data n bits left, where n is determined by a subset of the bits in the other half of the data. This is known as a Data–Dependent Rotation, one of the simplest DDPs.

The data–dependent permutations in CIKS–1 use a control vector (CV) to determine the permutation of the position of the input bits in the output. For example, the 2–bit $P_{2/1}$ DDP requires a CV of only one bit. If the CV is a 0, the bits are swapped, otherwise they pass through the primitive without changing position. These smaller permutation blocks are layered together to form more complex permutations.

The "butterfly" pattern that is used to connect the various levels of these permutations ensures that bits that are grouped together in the input are not continually swapped with each other as they move through the levels. It also guarantees that a CV which is comprised of mostly 1s will not result in a poorly permuted output. It is also important to note that the output consists totally of unchanged bits from the input in a different order, therefore the DDP has done nothing to change the Hamming weight of the data. Further information on the structure of the data–dependent permutation found in CIKS–1 can be found in [1].

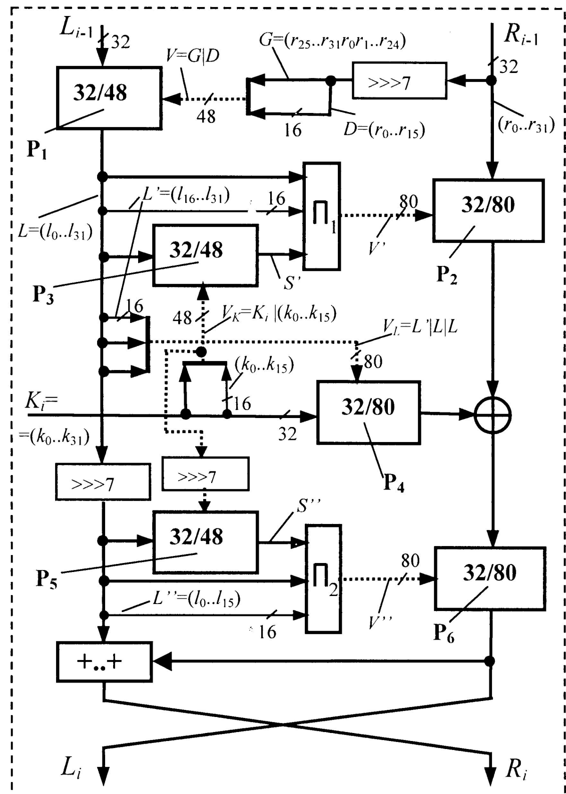

Figure 1: A single round of the CIKS–1 cipher [1]

2.2 Key Scheduling

There is no key schedule specified for CIKS–1. The authors note that there is internal key scheduling (IKS) due to permutation P_4 which scrambles each subkey in a round, controlled by the data on the left hand side of the cipher. This is considered to be beneficial to the cipher, as the key scheduling can be done in parallel with other parts of the cipher, thus adding no time delay due to frequent key changes.

The CIKS–1 paper however does not give any indication as to how these subkeys should be derived from the master key. Leaving this to the implementor of the cipher allows for the increased chance of using weak keys. For example, the cipher can be implemented such that each round uses the same subkey, depending on the IKS to scramble it differently for each round. If a weak (i.e. low Hamming weight) key is chosen in this implementation then the entire cipher will be compromised as the key is repeatedly used. Even more complex key schedules can result in a significant probability of weak keys. For example, the DES key schedule specifies that a single key 64–bit key is chosen for the cipher and the each 48–bit subkey is chosen as a subset of that key. If a similar key schedule was used with CIKS–1, a single low weight main key would result in a high probability of weak subkeys throughout the cipher[5].

3 Analysis of the Cipher

CIKS–1 uses only four different types of primitives: data–dependent permutations, fixed permutations (and rotations), XOR, and mod 2^2 addition. Of these, only the XOR and the addition have the potential to change the Hamming weight of the data as it flows through the cipher. The following sections take a look at each component individually, focusing on their contribution to the change of weight of the data.

3.1 Data–Dependent and Fixed Permutations

All of the data–dependent permutations used in CIKS–1 simply permute the location of data bits as they progress through the cipher. The control–vector input to each DDP block has absolutely no effect on the value of the input bits themselves: it simply determines their new order. Therefore, this primitive has absolutely no effect on the weight of the data.

Similarly, the fixed permutations in the cipher do not change the weight of the data. In fact, these permutations never operate on the actual data of the cipher directly. Π_1 and Π_2 are used specifically on control–vectors, preventing the attacker from pushing data backwards through the cipher to reveal information about the subkeys.

3.2 Exculsive–OR

The XOR combines a permuted version of a subkey with the data on the right side of the cipher. This primitive is one of two in the cipher where the weight of the data can be changed, however its effect on the weight is dependent on the weight of the key. When a binary '0' is XORed with another binary bit x, the output will be x since $x \oplus 0 = x$. Thus, if the key has a particularly low weight, the weight of the data will be only modestly affected.

3.3 Addition

CIKS–1 uses a mod 2^2 addition to combine the left and right data at the end of each round. Eight of these addition blocks are used in parallel, each operating on only two bits of the input data with the carry bit out of each 2–bit block being ignored. An analysis of the mod 2^2 addition shows that it has an influence on the weight of the left side data, although there is still an increased probability this data will have its weight remain the same. If fact, 6 out of 16 of cases for mod 2^2 addition result in an output weight of the left hand side data identical to the input weight.

3.4 Analysis of Weight

When designing a cipher, a desired property is to have an output that looks completely random no matter what the input for all keys. Ideally, it should not be possible to distinguish between the output of the cipher and the output of a random number generator. One quick check for this is to look at the mean weight of the output of the cipher. If random, this weight would fit a binomial distribution, thus giving a 64–bit output an average weight of 32.

Since there are very few elements of this cipher which affect the weight of the data as it is encrypted, this weight grows slowly as the data progresses through the rounds, particularly if the key has a low weight. If you limit the weight of the input to 6, the weight grows slowly enough that the mean does not get close to 32 until the end of the sixth round for a key weight of 6. This can be seen in Table 1.

Another test to see if the cipher output looks random is to do a "goodness–of–fit" test. The output of this cipher was compared to the binomial distribution using the χ^2 test. After five rounds, the CIKS–1

		Rounds							
		1	2	3	4	5	6	7	8
Key Weight	1	9.80562	15.8245	21.9752	27.3286	30.3768	31.5483	31.8731	31.9652
	2	11.3139	18.4168	24.8688	29.3186	31.2509	31.8091	31.948	31.9767
	3	12.8101	20.7126	27.0043	30.4733	31.6386	31.9094	31.9818	32.0041
	4	14.3018	22.7463	28.5732	31.1414	31.8139	31.9601	31.9952	32.0018
	5	15.8009	24.5132	29.6916	31.5112	31.8963	31.9727	31.9909	31.9977
	6	17.2943	26.0585	30.4913	31.7317	31.9477	31.9882	31.9972	31.9988
	7	18.7754	27.3571	31.0336	31.8431	31.9641	31.9904	32.0005	32.0039
	8	20.2617	28.4782	31.4082	31.9131	31.9769	31.998	32.001	32.0054

Table 1: Average output weight of cipher with maximum input weight of 6 over 5000000 encryptions

output for a key weight and maximum input weight of 6 gives a $\chi^2 = \infty$.[1] This indicates there is no fit to the binomial distribution for the cipher constrained to an input weight of 6 and a key weight of 6 or less. In fact, this property holds for CIKS–1 as long as these constraints are used as maxim and is the basis for the attack on the cipher presented in the next section.

4 Proposed Attack

As shown in the last section, CIKS–1 depends on the subkeys to contribute to the growth of Hamming weight for the data. In fact, when low weight keys are used with low weight input data, it is possible to distinguish between a random output that conforms to the binomial distribution and the output of the cipher. This reveals the set of low weight subkeys (weight of 6 or less) should be consider to be weak keys.

An attack can be mounted on a 6–round reduced version of the cipher to extract information about the first subkey. A subkey is guessed for the first round. Next, using many random values for the left hand side (lhs) after P_1, the key is permuted and traced back up to the right hand side (rhs) input, which is in turn used to determine the value of lhs at the input. The values for lhs and rhs are then encrypted over 6 rounds. The idea is to produce a value for the right hand input which is close to the actual subkey being attacked. A correct guess nullifies the effect of the key in the first round, thus leaving only the effects of the last five rounds. If these keys are low weight, the χ^2 test shows the output does not match the binomial distribution. Pseudo–code for the attack (using notation from Figure 1) is given in Figure 2.

For all possible 2^{32} subkeys
    For y = 1 TO 1000000
        Create a random value for lhs after with weight ¡= 6
        Run subkey through P_4, call result PSK
        Form Control vector v_l
        Form control vector v_k and v_p
        Permute v_p
        Run PSK back through P_2^{-1} to get rhs
        Form control vector v
        Run vector back through P_1^{-1} to get lhs
        Encrypt rhs and lhs over 6 rounds, record resulting weight
    Calculate the Chi-Squared Value for the results
Compile a list of weights with poor fit to the Binomial Distribution

Figure 2: Proposed weight base attack on CIKS–1 cipher

[1] The result is not strictly ∞, but an overflow of the largest float in Excel.

Tests have shown that the guessed key does not have to be exact to give useful results. Guessed keys with the same weight as the actual key and those with similar weights produce similar results to guessing the actual key. With this knowledge, the search space for the first subkey can be reduced based on the result of the χ^2 tests.

Table 2 is an example of results obtained with the attack. This example used an actual subkey of all–zeros and only 100 random guessed subkeys for each of the possible weights. This tests shows that the search area can be reduce to keys within a Hamming distance of 2 of the correct key. It is important to note that although this test used a low weight first round key, this is not required for the attack to succeed.

Weight Difference	χ^2
0	∞
1	∞
2	∞
3	118.52
4	69.33

Table 2: Preliminary test results for low weight attack

This attack on the 6–round reduced cipher has a time complexity of 2^{52}. This is an improvement over the 5–round attack with time complexity $2^{65.7}$ presented in [4]. However, that attack is makes no assumptions about the cipher keys, whereas the attack presented here requires that all but one of the keys be low weight. The probability of this occurring is highly dependent on the subkey weights and thus dependent on the strength of the key schedule.

5 Conclusion

Due to the choice of primitives with limited effect on the Hamming weight of the cipher data, the CIKS–1 cipher depends on the weight of subkeys to produce change in the data weight. This means that the class of low weight keys should be considered weak keys for the cipher. These keys produce outputs easily detectable using the χ^2 test. Using this fact an attack is proposed to distinguish the first subkey by dramatically reducing its entropy.

Preliminary testing has been done on the attack which has shown to reduce the search area for the first subkey to within a Hamming distance of 2 from the actual weight. At this point in time, the attack has not been extended to finish the search for the actual subkey. Also, more work will done on extending this attack to the full 8–round version of the cipher.

References

[1] A. Moldovyan and N. Moldovyan, "A cipher based on data–dependent permutations," *Journal of Cryptology*, vol. 15, pp. 61–72, January 2002.

[2] R. L. Rivest, "The RC5 encryption algorithm," in *K. U. Leuven Workshop on Cryptographic Algorithms*, December 1994.

[3] B. S. Kaliski Jr. and Y. L. Yin, "On differential and linear cryptanalysis of the RC5 encryption algorithm," in *Advances in Cryptology – CRYPTO '95* (D. Coppersmith, ed.), vol. 963 of *Lecture Notes in Computer Science*, pp. 171–184, Springer–Verlag Berlin, 1995.

[4] C. Lee, D. Hong, S. Lee, S. Lee, H. Yang, and J. Lim, "A chosen plaintext linear attack on block cipher CIKS–1," in *Information and Communications Security: 4th International Conference, ICICS 2002, Singapore, December 9-12, 2002. Proceedings*, vol. 2513 of *Lecture Note in Computer Science*, pp. 456–468, Springer-Verlag Heidelberg, January 2002.

[5] National Bureau of Standards, "Data encryption standard," 1977.