Mod n Cryptanalysis, with Applications Against RC5P and M6

John Kelsey*, Bruce Schneier**, and David Wagner***

Abstract. We introduce "mod n cryptanalysis," a form of partitioning attack that is effective against ciphers which rely on modular addition and bit rotations for their security. We demonstrate this attack with a mod 3 attack against RC5P, an RC5 variant that uses addition instead of XOR. We also show mod 5 and mod 257 attacks against some versions of a family of ciphers used in the FireWire standard. We expect mod n cryptanalysis to be applicable to many other ciphers, and that the general attack is extensible to other values of n.

1 Introduction

Nearly all modern statistical attacks on product ciphers work by learning some way to distinguish the output of all but the last rounds from a random permutation. In a linear attack, there is a slight correlation between the plaintext and the last-round input; in a differential attack, the relationship between a pair of inputs to the last round isn't quite random. Partitioning attacks, higher-order differential attacks, differential-linear attacks, and related-key attacks all fit into this pattern.

Mod n cryptanalysis is another attack along these lines. We show that, in some cases, the value of the last-round input modulo n is correlated to the value of the plaintext modulo n. In this case, the attacker can use this correlation to collect information about the last-round subkey. Ciphers that sufficiently attenuate statistics based on other statistical effects (linear approximations, differential characteristics, etc.) are not necessarily safe from correlations modulo n.

1.1 The Rest of This Paper

The rest of this paper is organized as follows. First, in Section 2 we introduce mod 3 cryptanalysis and develop the tools we need to attack RC5P. Next, in Section 3 we develop the attack on RC5P and show how it can be applied in a reasonably efficient way to break RC5P variants with quite a few rounds. Section 4 analyzes M6, a family of ciphers proposed for digital content protection.

* Counterpane Systems; 101 E Minnehaha Parkway, Minneapolis, MN 55419, USA; kelsey@counterpane.com.

** Counterpane Systems; schneier@counterpane.com.

*** University of California Berkeley, Soda Hall, Berkeley, CA 94720, USA; daw@cs.berkeley.edu.

Finally, in Section 5 we discuss what we've discovered so far, consider some generalizations to our techniques, and point out a number of interesting open questions whose answers we hope will be the subject of future research.

Also, in Appendix A we demonstrate why our definition of bias is the right one and recall some important facts about the χ^2 test.

2 Tools for Mod 3 Cryptanalysis

In mod 3 cryptanalysis, we trace knowledge of the mod 3 value of some part of a cipher's block through successive rounds of the cipher, leaving ourselves with some information about the input to the last round or two that lets us distinguish it from a randomly-selected block. The attack is conceptually very similar to Matsui's linear cryptanalysis [Mat94,Bih95,KR94,KR95,KR96], though it is properly included in the class of partitioning attacks [HKM95,HM97] developed by Harpes and Massey. We also draw somewhat from Vaudenay's statistical cryptanalysis [Vau96].

In this paper, we will use the shorthand term "mod 3 value" to stand for the value we get when we take some selected 32-bit part of a block, and reduce it modulo 3. A mod 3 value may thus be only 0, 1, or 2. In a randomly-selected 32-bit block, we would expect 0, 1, and 2 to occur as mod 3 values with almost identical likelihood. (If we automatically discarded any block with the value $2^{32}-1$, we would have perfectly equal probabilities.) As a block cipher's successive rounds operate on its block, the block should become harder and harder to distinguish from a randomly-selected block, without knowledge of the cipher's round keys. Mod 3 cryptanalysis works when the block's mod 3 value is still not too hard to distinguish from that of a random block, very late into the cipher. (In the same sense, linear cryptanalysis works when the parity of some subset of the block's bits is still not too hard to distinguish from that of a random block, very late into the cipher.)

2.1 Approximating Rotations

The insight that first led us to consider mod 3 cryptanalysis at all involved the behavior of the mod 3 value of some 32-bit word, X, before and after being rotated by one bit. When we consider X as a 32-bit integer, and $X \lll 1$ as X rotated left by one bit, we can rewrite the effects of the rotation in terms of integer arithmetic:

$$X \lll 1 = \begin{cases} 2X, & \text{if } X < 2^{31} \\ 2X + 1 - 2^{32}, & \text{if } X \geq 2^{31} \end{cases}$$

The first thing to notice is that $2^{32} \equiv 1 \bmod 3$. Thus, $X \lll 1 \equiv 2X \bmod 3$, because $2X + 1 - 2^{32} \equiv 2X \bmod 3$.

From this, we can derive the effect of any larger number of rotations. For instance,

$$X \lll 2 \equiv (X \lll 1) \lll 1 \equiv 2 \times 2 \times X \equiv X \bmod 3$$

In general, we have

$$X \lll n \equiv 2^n X \equiv 2^{n \bmod 2} X \bmod 3$$

so rotating by any odd number of bits multiplies the mod 3 value by 2, while multiplying by any odd number of bits leaves the mod 3 value unchanged. This means that when we know the number of bits a 32-bit block was rotated, and what its input mod 3 value was, we also know what its output mod 3 value was.

Next let us consider the case where we know the input mod 3 value, but not the rotation amount. We do *not* lose all knowledge of the output mod 3 value. Indeed, some traces of X leak, because we know

$$X \lll n \bmod 3 = \begin{cases} 2X \bmod 3, & \text{if } n \text{ odd} \\ X \bmod 3, & \text{if } n \text{ even} \end{cases}$$

Note that in the case of $X \bmod 3 = 0$, we have $X \lll n \equiv 0 \bmod 3$, regardless of n. Thus $\Pr[X \lll n \equiv X \bmod 3] = 4/6$ when X is uniformly distributed, and we have some incomplete knowledge on $X \lll n$.

We can express the propagation of partial information using the notation of probability vectors. Let the probability vector p_X represent the distribution of X, so that the j-th component of p_X is $\Pr[X = j]$. Then, for example, if $Y = X \lll 1$ and $p_X = [0, 1/2, 1/2]$, we find that $p_Y = [0, 1/2, 1/2]$. As another example, a uniformly-distributed random variable U is represented by the probability vector $p_U = [1/3, 1/3, 1/3]$.

It is also tempting to think of operations such as $\lll$ in terms of their transition matrix M (where $M_{i,j} = \Pr[f(X) = j | X = i]$). However, as will be discussed below, there are subtle pitfalls with such an approach.

In some cases it is also useful to view rotations as a multiplication modulo $2^{32} - 1$. The key observation is that we have the relation

$$x \lll j \equiv 2^j x \bmod (2^{32} - 1)$$

for rotations left by j bits. Reducing both sides modulo 3, we obtain $x \lll j \equiv 2^j x \bmod 3$. (This is valid because 3 divides $2^{32} - 1$.) This is another way to derive the mod 3 approximation of rotations given above.

We can also see that we get a good mod p approximation $x \lll j \equiv 2^j x \bmod p$ for bit-rotations whenever p divides $2^{32} - 1$. Section 5 explores this direction in more detail.

2.2 Approximating Addition Modulo 2^{32}

A similar analysis works for addition mod 2^{32}. Consider a simple description of mod 2^{32} addition in terms of integer addition:

$$X + Y \bmod 2^{32} = \begin{cases} X + Y, & \text{if there was no carry out} \\ X + Y - 2^{32}, & \text{if there was a carry out} \end{cases}$$

Since $2^{32} \equiv 1$ modulo 3, this can be rewritten as

$$(X + Y \bmod 2^{32}) \bmod 3 = \begin{cases} X + Y \bmod 3, & \text{if there was no carry out} \\ X + Y - 1 \bmod 3, & \text{if there was a carry out} \end{cases}$$

Sometimes, we know the distribution of the carry. For example, we might know that the high-order four bits of Y are all ones, and so know that the carry-out probability is around 0.98. We can then rewrite this approximation as:

$$X + Y \bmod 2^{32} \bmod 3 = \begin{cases} X + Y \bmod 3, & \text{with prob. } 0.02 \\ X + Y - 1 \bmod 3, & \text{with prob. } 0.98 \end{cases}$$

2.3 Biases and the l_2 Norm

As we discussed above, the probability vector $p_U = (1/3, 1/3, 1/3)$ is approximately what we would expect from a random 32-bit block. It would be nice to have some measure of distance from the uniform distribution. In this paper, we use the l_2 norm[1] as our measure of bias. The bias of a probability vector p_X is defined using this distance measure as

$$||p_X - p_U||^2 = \sum_j (p_X[j] - p_U[j])^2.$$

Intuitively, the larger the bias, the fewer samples of a block described by p_X are necessary to distinguish those blocks from a random sequence of blocks. Appendix A motivates and formalizes this measure of bias: we find that $O(1/||p_X - p_U||^2)$ samples suffice to distinguish the distribution p_X from uniform and that the χ^2 test may be used to implement the distinguisher.

3 Mod 3 Cryptanalysis of RC5P

RC5 is a conceptually simple block cipher designed by Ron Rivest [Riv95] and analyzed in [KY95,KM96,Sel98,BK98,KY98]. The cipher gets its strength from data-dependent rotations, a construct also used in Madryga [Mad84], Akelarre [AGMP96], RC6 [RRS+98,CRRY98], and Mars [BCD+98]. Presently, 16 rounds (each RC5 round consists of two Feistel rounds) of RC5 is considered to be secure. RC5P is an RC5 variant described in [KY98] and conjectured to be as secure as RC5. It is identical to RC5, except that the XORs in RC5 are replaced with additions modulo 2^{32} in RC5P.

In this section, we discuss a mod 3 attack on RC5P. We have implemented simplified versions of the attack on RC5P with up to seven full rounds (fourteen half rounds), but without input or output whitening. (This attack took about three hours on a 133 MHz Pentium.) We conjecture that this attack might be extended to as many as nineteen or twenty rounds for at least the most susceptible keys.

The RC5P round function is as follows:

[1] Also called the Euclidian Squared Distance in [HM97].

$L := L + R$
$L := L \lll R$
$L := L + sk_{2i}$
$R := R + L$
$R := R \lll L$
$R := R + sk_{2i+1}$

3.1 Modeling the RC5P Round Function

We initially tried to predict the bias of multiple RC5P rounds using several mathematical tools (transition matrices, matrix norms, second-largest eigenvalues, etc.). However, we found that precise analytical methods were surprisingly difficult to develop for RC5P, because of a lack of independence between rounds: in technical terms, RC5P is not a Markov cipher with respect to mod 3 approximations. As a result, multiplying the biases or transition matrices of each individual round gives incorrect answers.

For these reasons, we abandoned our pursuit of precise mathematical models and turned to empirical measurements. Let (P_L, P_R) and (C_L, C_R) represent the plaintext and ciphertext after encrypting by R rounds. For convenience, we choose plaintexts so that $(P_L \bmod 3, P_R \bmod 3)$ is fixed: in practice, each of the nine possibilities give about the same test results. Then we empirically compute the probability distribution of $(C_L \bmod 3, C_R \bmod 3)$ and measure its bias using the χ^2 test. More precisely, we count the number of texts needed for the χ^2 score to exceed a certain threshold. Since $(C_L \bmod 3, C_R \bmod 3)$ has 9 possible outcomes, we use a chi-square test with 8 degrees of freedom. To give some baseline figures, a threshold of $\chi^2_{2^{-16},8} = 37$ has a probability of about 2^{-16} of occurring in a random sequence of inputs, while a test value of $\chi^2_{2^{-32},8} = 62$ has a probability of about 2^{-32} of occurring in a random sequence of inputs.

We used this technique to estimate the number of texts needed to distinguish R rounds of RC5P, for $1 \leq R \leq 8$. For each choice of R, we ran 50 trials of the previous test and computed the average number of texts needed as well as a 90% confidence interval. Our measurements are presented in Figure 1; note that the y axis is scaled logarithmically.

3.2 Mounting the Attack

Overview of the Attack Here, we discuss a chosen-plaintext chi-square attack on RC5P without pre- or post-whitening. The attack can clearly be applied with the whitening, or with only known plaintexts; in each case, we require more texts and more processing. In the final version of this paper, we will specify attacks on more versions of the cipher, as well as having more complete experimental data.

We make use of the mod 3 values of the two 32-bit halves of the RC5P block to select one of nine partitions into which to put the block. Given a sequence of N RC5P blocks, we can count how many fall into each of the nine partitions, and

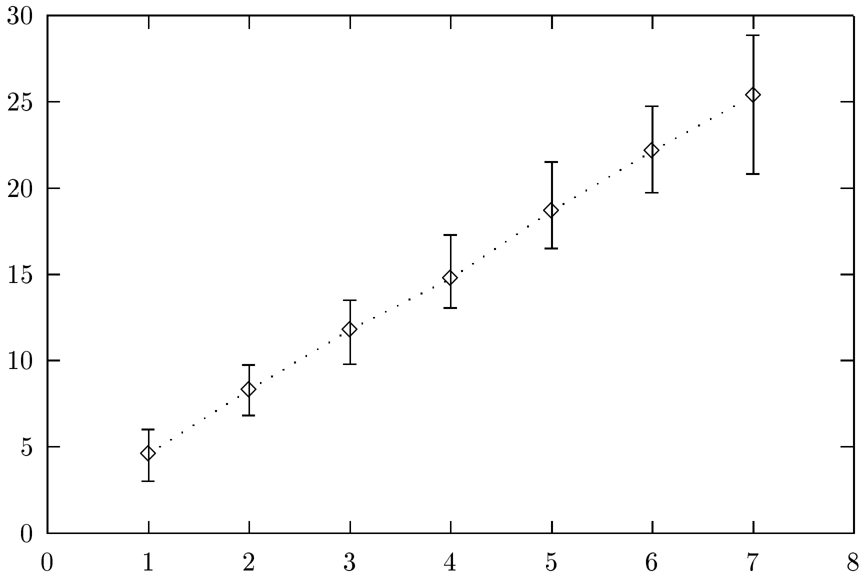

Fig. 1. Number of known texts needed to distinguish R rounds of RC5P from random.

use this count to compute a chi-square score, as discussed above. Informally, the chi-square score allows us to distinguish between a uniformly random selection of these partitions, as we would expect from the output of a random permutation, and a biased selection of these partitions, as we would expect from a cipher with a good mod 3 approximation available.

The attack works as follows:

1. Request the encryptions of N chosen plaintexts, where N is chosen according to the criteria given below.
2. For each resulting ciphertext, try all 48 possible combinations of mod 3 value and high-order four bits for the last half-round's subkey. Use this guess to predict the mod 3 values of both 32-bit halves of the block before the last half round, and keep count of these values for each guess.
3. Use these counts to calculate a chi-square score with eight degrees of freedom, based on splitting the block into nine possible categories based on the two mod 3 values.
4. Select the partial guess with the highest chi-square score as the most likely guess.
5. Assuming the above guess is correct, begin the process again, this time guessing the next six bits of subkey. Continue this process, guessing six bits of subkey at a time, until the last four bits of subkey are remaining. Guess those four bits and test the guesses in the same way.
6. The result is a guess which is likely to be correct of the last half-round's subkey. Using this value, we can peel the last half-round off all the ciphertexts,

and use the resulting values to mount the attack again on a version of the cipher with one fewer half-rounds.

Choosing the Plaintexts We choose the plaintexts to try to maximize the bias in the selection of a partition after N rounds. In practice, this means attempting to bypass most of the effects of the first full round of RC5P. We thus choose P_L, P_R such that:

1. The high-order eight bits of P_L and P_R are all zeros.
2. The low-order five bits of P_L and P_R are all zeros.
3. $P_L \bmod 3 \equiv P_R \bmod 3 \equiv 0$.

To understand why this makes sense, we must consider the first round of RC5P without the whitening in some detail. Recall that the operations are:

1. $L := L + R$
2. $L := L \lll R$
3. $L := L + sk_0$
4. $R := R + L$
5. $R := R \lll L$
6. $R := R + sk_1$

When the high-order eight bits of both P_L and P_R are zeros, the high-order seven bits of L after step one must be zeros, and there is no chance of a carry in that addition. Recall that when there is not a carry in mod 2^{32} addition, the mod 3 values of the addends can be added together mod 3 to get the correct mod 3 value for the sum. Because the low five bits of R are zeros, there is no rotation in step two. Thus, in step three, we know that the high-order seven bits of L are zeros before the addition. Unless the high-order seven bits of sk_0 are all ones, there will never be a carry in that addition, either. *Thus, L will, for 127/128 possible keys, have the same mod 3 value for all inputs chosen as we have described.*

The high-order seven bits of L after the third step are no longer known, but will be closely related for all the texts. There will be only two possible values for these high seven bits, depending on whether there was a carry into them in step three's addition.

The high order eight bits of R going into the addition in the fourth step are zeros; thus only if all eight high-order bits of L are ones will there ever be a carry in this addition. For nearly all values of sk_0, it will thus not be possible for there to be a carry in this addition, either.

The low-order five bits of L after the addition in step three will be constant. Thus, the rotation in step five will be by a constant amount. Rotation by a constant amount of a constant mod 3 value will yield a constant mod 3 value. Thus the mod 3 value of R will be constant for nearly all keys after step five. In step six, there is an addition with sk_1. After this step, it will be possible for R to have one of two mod 3 values. Depending on the high-order few bits of sk_1,

R may be nearly balanced between these two values, or may be strongly biased towards one or the other of them.

The result of the way we choose the plaintexts is thus, for nearly all keys, to bypass nearly the first full round of RC5P for the sake of our mod 3 approximation. Instead of the possibility of being in all nine different partitions after the first round, for most keys the texts can only be in one of two partitions after the first round.

Making the Initial Guess Having collected N ciphertexts, corresponding to the N chosen plaintexts, we now make an initial guess of the high-order four bits and mod 3 value of the final half-round's subkey, which we will refer to as sk_f. There are 48 possible values for this guess; we try each guess against all N ciphertexts.

Each guess suggests a value of the right half of the block before the final half-round was applied. Consider the decryption of the final half-round:

1. $R := R - sk_f$
2. $R := R \ggg L$
3. $R := R - L$

In the first step, we must determine, based on the known value of R and the guessed parts of sk_f, what the result of the subtraction will be. This is dependent upon the mod 3 value of sk_f, and also its high-order four bits. In the second step, we must use the known rotation amount from L to determine what the resulting R value will be. The resulting mod 3 value of R is determined only by the low-order bit of L. However, the total rotation amount determines which bits of R from the first step will end up in the high-order bits. Finally, in the third step, we must use the known L value, along with what is known about the R value input, to determine the result of this final subtraction. For most ciphertexts, we know only the mod 3 value of R going into this operation. However, for some rotation amounts in step two, we also know the likely values for the high-order few bits of R going into the third step.

We can model this by noting that there are many different subkey words sk_{guess} which share the same high-order four bits and mod 3 value as a given guess to be tested. We can choose a reasonable representative from the set of these, and get a fair approximation to its behavior. We thus derive sk_{model} by setting its high-order four bits to the value required by the guess, setting the rest of the bits to alternate between zero and one bits, and then incrementing the result as necessary to get the guessed mod 3 value. We then carry out a trial decryption with this partial key guess, get a resulting pair of mod 3 values, and keep count of how many of each value results from the trial decryption.

To distinguish between the right and wrong partial values for sk_f, we use those counts to compute chi-square scores, and choose the highest chi-square score as the most likely value. For sufficiently large values of N, we have seen experimentally that this is very likely to select the right partial value for sk_f.

Weak Key Rounds	Average Key Rounds	Texts	Work
11	8	2^{29}	5×2^{35}
13	10	2^{37}	5×2^{43}
15	12	2^{45}	5×2^{51}
17	14	2^{53}	5×2^{59}
19	16	2^{61}	5×2^{67}

Table 1. Estimates of Difficulty of Attacking RC5P With Many Rounds.

Continuing the Guess Assuming the previous guess was correct, we can extend this guess. We guess the next four or six bits of sk_f, updating sk_{model} appropriately. The new sk_{model} is very little improved at determining whether there will be a borrow in the subtraction of the first step, but for some rotation amounts in the second step, it has a strong impact on determining whether there will be a borrow in the subtraction in the third step. For each guess, we again keep count of the mod 3 values of the two halves after it is applied, and we again select the guess with the largest chi-square value.

Continuing the Attack After the full sk_f value is known, we peel off the last half-round, and apply the attack anew to the resulting cipher.

3.3 Resources Required for The Full Attack

According to our preliminary experiments, the full attack has an acceptably high probability of success using N texts, where N is the number of texts necessary to get a chi-square value high enough that in practice, it simply could not have occurred by chance.

Our experimental data suggest that each additional round requires roughly sixteen times as many texts to get about the same χ^2 score on average, and that there are especially vulnerable keys for which we can expect to get sufficiently high χ^2 scores even with an extra two to three rounds, with the same N.

The work required for the attack is approximately $5 \times 2^6 \times N$. Table 1 shows our predictions for the approximate workfactor and number of texts needed to break RC5P.

3.4 Results and Implications

There are several practical implications of our results:

1. RC5P, the RC5 variant with XORs changed to mod 2^{32} additions, is much less secure than was previously believed [KY98]. We suggest a minimum of 22 rounds for reasonable security when used with a 128-bit key.

2. Other ciphers which use rotations and additions, but no multiplications or XORs, are likely to also be vulnerable. In particular, all elements other than additions and rotations in such ciphers should be carefully reviewed to see whether they have good mod 3 approximations. In some cases modular multiplication may also be vulnerable, depending upon the modulus: for instance, $f(x) = a \cdot x \bmod (2^{32} + 5)$ is vulnerable to mod 3 cryptanalysis, since $2^{32} + 5 = 3^3 \times 47 \times 3384529$.
3. Multiplication mod 2^{32} as done in RC6, and XORing as done in both RC6 and RC5, are both very difficult to approximate mod 3. Hence, these operations generally make ciphers resistant to the attack. However, the specific cipher designs need to be reviewed to verify that they are used in a way that actually helps defeat the attack. RC5 and RC6 seem to be resistant to this attack, as does Mars. But see our analysis of M6 in Section 4 for an example of a cipher that uses rotations, additions, and XORs yet still succumbs to mod n attacks.
4. Placing the multiplication or XORing only at the beginning and end of the cipher is probably not effective in making the cipher resistant to mod 3 cryptanalysis, since there are often clever analytical tricks to bypass these operations.

Mixing operations from different algebraic groups was the guiding principle behind several ciphers—IDEA [LMM91], Twofish [SKW+98], etc.—and that still seems like a good idea.

3.5 RC6

Note that our mod 3 attack suggests a new design principle for RC6-like ciphers. If the f function $x \mapsto x \times (2x + 1) \bmod 2^{32}$ in RC6 had instead been defined as $x \mapsto x \times (2x + 1) \bmod m$ for some other value of m, powerful mod n attacks might be possible.

For instance, if 3 divides m, then we have a probability-1 approximation for the f function. In this case, the RC6 variant obtained by replacing the XORs with additions (and also using the modulus m instead of 2^{32} in the definition of f) could be broken by mod 3 cryptanalysis. For example, with the IDEA-like modulus $m = 2^{32} - 1$, RC6 would be in serious trouble if we replace the XORs by additions.

This suggests the design principle that $\gcd(m, 2^{32} - 1) = 1$. (The mysterious number $2^{32} - 1$ comes from the formulation of rotations as multiplication by powers of two modulo $2^{32} - 1$; see Section 2.1.)

4 M6

M6 is a family of ciphers proposed for use in the IEEE1394 FireWire standard, a peripherals bus for personal computers [FW1,FW2].[2] M6 is used for encrypting

[2] M6 is based on work done in [THN91]. Note that no full description of M6 is publicly available, due to export considerations [Kaz99]. However, a general description of the

copyrighted and other protected content between the computer and the peripheral.

For convenience, we briefly describe an example of a M6 cipher here (specifically, the example given in [FW2], an earlier draft of the standard). See also Figure 2 for a pictorial illustration.

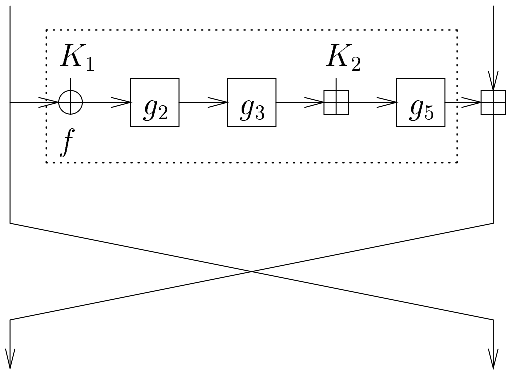

Fig. 2. One round of a M6 block cipher

The cipher uses a 10-round Feistel structure. The f function is defined by

$$\begin{aligned} g_1(x) &= x \oplus K_1 & g_2(y) &= (y \lll 2) + y + 1 \bmod 2^{32} \\ g_3(z) &= (z \lll 8) + z \bmod 2^{32} & g_4(a) &= a + K_2 \bmod 2^{32} \\ g_5(b) &= (b \lll 14) + b \bmod 2^{32} & f(x) &= (g_5 \circ g_4 \circ g_3 \circ g_2 \circ g_1)(x). \end{aligned}$$

The round function F updates a 64-bit block (x, y) according to

$$F((x, y)) = (y + f(x) \bmod 2^{32}, x).$$

M6 typically uses 40 bit keys (although the algorithm also allows for keys up to 64 bits long), and the key schedule is very simple. Let K_1 be the high 32 bits of the key, and W be the lower 32 bits of the key (so that K_1 and W share 24 bits in common). Set $K_2 = K_1 + W \bmod 2^{32}$. Then K_1, K_2 are the output of the key schedule.

The standard also suggests that other variations on the basic construction above can be created by changing the order of the g functions, by swapping additions for XORs (or vice versa), and/or by changing the rotation amounts. Each round may use a different variation of the basic scheme. As a result, we get a family of ciphers, which we will call the M6 ciphers.

For concreteness, we focus on the example cipher given above. However, we note that the same techniques also apply to many other ciphers in the M6 family:

family of ciphers from which M6 is drawn from can be found in [Hit97]. Our techniqes are applicable to most ciphers in this family.

as long as the last g function is of the form $b \mapsto (b \lll \alpha) + b + \beta \bmod 2^{32}$ and the output of the f function is combined with the block by addition, we will be able to apply mod n techniques. Thus, a large fraction of the M6 ciphers can be broken by mod n attacks.

4.1 A mod 5 Attack

We note that f is highly non-surjective, and in fact admits excellent mod 5 approximations. In particular, we have the following theorem.

Theorem 1. $f(x) \bmod 5 \in \{0, 4\}$ *for all* x.

Proof: It suffices to show that $g_5(b) \bmod 5 \in \{0, 4\}$. Note that

$$g_5(b) = (b \lll 14) + b - 2^{32}k \equiv (2^{14} + 1)b - 2^{32}k \bmod 2^{32} - 1$$

using the relation $b \lll 14 \equiv 2^{14}b \bmod 2^{32} - 1$ from Section 2.1. It is not hard to see that $k \in \{0, 1\}$ (just observe that $(b \lll 14) + b < 2^{33}$).

Note that 5 divides $2^{32} - 1$, so we may reduce both sides of the relation modulo 5. Since $2^{14} + 1 \equiv 0 \bmod 5$ and $2^{32} \equiv 1$, we get

$$g_5(b) \equiv -k \bmod 5, \qquad k \in \{0, 1\}.$$

This proves our desired result. □

We next analyze the round function F using Theorem 1. The Feistel function is combined via addition mod 2^{32}, which makes things easy. Let $(y', x) = F((x, y))$, so that $y' - y = f(x) \bmod 2^{32}$. Rewriting the latter equation to eliminate the "mod" gives:

$$y' - y = f(x) + 2^{32}k, \qquad k \in \{-1, 0\}.$$

Reducing both sides modulo five, we get:

$$y' - y \bmod 5 \in \{0, 3, 4\}.$$

It is not hard to see that $f(x) \bmod 5$ is uniformly distributed on $\{0, 4\}$ and k is uniformly distributed on $\{-1, 0\}$, so we see that $y' - y \bmod 5 = f(x) + k \bmod 5$ takes on the values $0, 3, 4$ with probabilities $1/4, 1/4, 1/2$.

This can now be applied to the whole cipher. Let P_L be the left half of the plaintext and C_L the left half of the ciphertext. We see that $C_L - P_L$ is the sum of five independent random variables whose value modulo five has the distribution $(1/4, 0, 0, 1/4, 1/2)$. Thus the distribution of $C_L - P_L \bmod 5$ is the five-fold convolution of $(1/4, 0, 0, 1/4, 1/2)$, or approximately:

$$p_{C_L - P_L \bmod 5} = (0.248, 0.215, 0.161, 0.161, 0.215).$$

The same holds for the right halves.

So a significant amount of information is leaked from the plaintext through to the ciphertext. For instance, $C_L \bmod 5$ has a nearly $1/4$ chance of being equal to $P_L \bmod 5$, which is significantly greater than the $1/5$ chance one would expect from a strong cipher. The bias of the difference is about 0.00577, which indicates that with a hundred or so known texts we could easily distinguish the cipher from random.

4.2 A mod 257 Attack

In fact, the cipher is even worse the analysis above might indicate. The Feistel f function also admits excellent mod 257 approximations. These approximations are easy to use in a key-recovery attack, because they disclose the value of $K_2 \bmod 257$.

Theorem 2. *$f(x) - 194 \cdot K_2 \bmod 257 \in \{0, 62, 63, 256\}$ for all x, and this value is uniformly distributed when x is.*

Proof: Using a similar argument to that found in the proof of Theorem 1, we find that $g_3(z) \bmod 257 \in \{0, 256\}$ since 257 divides $2^{32} - 1$. Also, we have

$$f(x) = g_5(g_3(z) + K_1) \equiv (2^{14} + 1)(g_3(z) + K_1) - 2^{32}k \bmod 2^{32} - 1.$$

Reducing both sides modulo 257 and letting $k' = 194 g_3(z) \bmod 257$ gives

$$f(x) \equiv 194K_1 - k + k' \bmod 257, \qquad k \in \{0, 1\},\ k' \in \{0, 63\}$$

since $2^{14} + 1 \equiv 194 \bmod 257$. Also, k and k' are uniformly distributed over their respective sets of possible values. The theorem follows. $\square$

We can repeat the analysis of the previous attack, noting that the distribution of $X_L = C_L - P_L - 5 \cdot 194 \cdot K_2 \bmod 257$ is highly non-uniform. It has bias 0.056690, so a few dozen known texts should be enough to distinguish the cipher from random using a mod 257 attack.

In fact, the distribution of X_L has only 34 non-zero entries (with most of the probability density concentrated on only a fraction of them), so given one known text we can immediately eliminate all but 34 possibilities for $K_2 \bmod 257$ just by looking at the left half of the plaintext and ciphertext. The expected number of guesses needed to find $K_2 \bmod 257$ from X_L is easily calculated to be about 8. Then we can recover the remainder of the key with a search over the 2^{32} possibilities for K given the guess at $K_2 \bmod 257$. In other words, given one known text we can find the key with expected 2^{35} offline trial encryptions; this is already 16 times faster than exhaustive search.

If more known texts are available, we can do even better. Each text gives us one observation at X_L. If we also look at the right halves of each text, we can double the number of available observations. With several dozen known texts, we expect to be able to recover $K_2 \bmod 257$ with relatively good accuracy. Therefore, when a few dozen known texts are available, we can find the 40-bit key with expected 2^{31} offline trial encryptions. This demonstrates that the level of security afforded by the example M6 cipher is extremely low, even for a 40-bit cipher.

Powerful ciphertext-only attacks may also be possible. If the left half of the plaintext is divided into bytes as $P_L = (w, x, y, z)$, we find that $P_L \equiv z - y + x - w \bmod 257$, so if the bytes of the plaintext are biased $P_L \bmod 257$ will be too. When the plaintext is ASCII-encoded text, we expect very significant biases to remain, and in this case the value of $K_2 \bmod 257$ will leak after a sufficient number of ciphertexts. (For instance, when the plaintext is composed only of the letters 'a'–'z', we have $81 \leq (P_L \bmod 257) \leq 181$.)

M6 could be easily strengthened against mod n attacks with a small change: simply always use XOR instead of addition when combining the output of the f function with the block. (It is worth pointing out that no mere change in rotation amounts can secure M6 against mod n attacks.) With such a defense, some attacks might still be possible given a very large pool of known texts, but since the cipher was only designed for a 40-bit keylength, the results might be good enough for practical purposes.

4.3 MX

We also note that our analysis techniques can be applied to MX [ATFS98], another cipher with an internal structure similar to that of the example M6 cipher described above. The primary difference is that MX allows for secret round-dependent rotation and addition constants inside the round function. For instance, MX's version of g_5 is defined as $g_5(x) = (x \lll s_3) + x - \gamma \bmod 2^{32}$ where s_3, γ are fixed round constants (not dependent on the secret key). It is not hard to see that g_5 has a good mod n approximation no matter the value of s_3, α, and therefore the MX round function will always be sufficiently biased to allow for excellent mod n attacks.

5 Generalizations, Conclusions, and Open Questions

In this paper, we have discussed a new cryptanalytic attack that is extremely powerful against ciphers based only upon addition and rotation. We have also demonstrated an apparent weakness in RC5P, and given a successful attack against a substantial fraction of the M6 family of ciphers.

This shows that the strength of RC5 relies heavily on the mixture of XORs and additions—differential cryptanalysis breaks the variant with only XORs [BK98], and we have shown that mod 3 cryptanalysis is a very powerful attack against the variant with only additions. We conclude that the mixing of additions and XORs in RC5 is not just a nice touch: it is absolutely essential to the security of the cipher.

Note that we can also consider mod p attacks, for any prime p dividing $2^{32}-1$. The prime factorization of $2^{32} - 1$ is $3 \times 5 \times 17 \times 257 \times 65537$, so there is no shortage of potential candidates for p.

One of the potential problems with using values of $p > 3$ is that a rotation can now involve a multiplication by any value in the set $S_p = \{2^j \bmod p : j = 0, 1, \ldots\}$. Fortunately (for the attacker), when $p = 2^k + 1$ divides $2^{32} - 1$ we have the nice property that $|S_p| = 2k$ (since $2^k = -1 \bmod p$ and $2^{2k} = 1 \bmod p$), so generalized mod p attacks might be even more successful against RC5P than our mod 3 attack was.

Of course, we can also consider mod n attacks where n is not necessarily prime. When n is composite, a mod n attack is the rough equivalent of mod p cryptanalysis with multiple approximations. If the prime factorization of n is $p_1 \times \cdots \times p_m$, then by the Chinese remainder theorem all mod n attacks may be decomposed into m attacks modulo each p_j.

A number of open questions remain, which we hope to see investigated in the near future. Among them:

1. Other moduli than 3, 5, and 257 might have some advantages in this kind of attack. Section 5 contains some early work in this direction, but more is needed.
2. Other ciphers that might be vulnerable to mod p attacks. As a rule, ciphers that use only addition and rotation are likely to be vulnerable, as are ciphers that use addition and some nonlinear operations (such as S-boxes) which have a good mod p approximation.
3. We have made a number of observations in this paper based on experiments; we would like to find improved mathematical models to explain these observations, and solidify our predictions about events we can't verify experimentally.
4. We suspect that a variant of differential cryptanalysis can be defined for some ciphers, using differences mod n instead of mod 2^{32}. Much of the same mathematical apparatus can be used for this class of attack as for our attack.
5. Data-dependent rotations are handled poorly by the standard analytic techniques available to us (namely, linear attacks and differential attacks with XOR or mod 2^{32}-based differences). We are interested in other properties that can be used in differential- or linear-like attacks, but which will survive rotation (and particularly data-dependent rotation) better than fixed XOR differences.

6 Acknowledgements

We are very grateful to Doug Whiting for useful discussion early in our development of mod n attacks, and to Niels Ferguson for his helpful comments.

References

[ATFS98] M. Aikawa, K. Takaragi, S. Furuya, M. Sasamoto, "A lightweight Encryption Method Suitable for Copyright Protection," *IEEE Trans. on Consumer Electronics*, vol.44, n.3, pp.902–910, 1998.

[AGMP96] G. Álvarez, D. De la Guia, F. Montoya, and A. Peinado, "Akelarre: A New Block Cipher Algorithm," *Workshop on Selected Areas in Cryptography (SAC '96) Workshop Record*, Queens University, 1996, pp. 1–14.

[BCD+98] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S.M. Matyas, L. O'Connor, M. Peyravian, D. Safford, and N. Zunic, "MARS — A Candidate Cipher for AES," NIST AES Proposal, Jun 98.

[Bih95] E. Biham, "On Matsui's Linear Cryptanalysis," *Advances in Cryptology — EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, pp. 398–412.

[BK98] A. Biryukov and E. Kushilevitz, "Improved Cryptanalysis of RC5," *Advances in Cryptology — EUROCRYPT '98 Proceedings*, Springer-Verlag, 1998, pp. 85–99.

[BW99] A. Biryukov and D. Wagner, "Slide Attacks," this volume.

[CRRY98] S. Coniti, R. Rivest. M. Robshaw, and Y.L. Yin, "The Security of the RC6 Block Cipher," Version 1.0, RSA Laboratories, 20 Aug 1998.

[FW1] *Content Protection for Digital Transmission System*, 5C compromise proposal version 0.91, 17-Feb-1998, Hitachi, Intel, Matsushita, Sony, and Toshiba.

[FW2] *Response for Data Protection System for Digital Transmission of Copy Protected Information*, Version 0.99, pp. 8–12, Hitachi, Matsushita, and Sony.

[HKM95] C. Harpes, G. Kramer, and J. Massey, "A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma," *Advances in Cryptology — EUROCRYPT '95 Proceedings*, Springer-Verlag, 1995, pp. 24–38.

[HM97] C. Harpes and J. Massey, "Partitioning Cryptanalysis," *Fast Software Encryption, 4th International Workshop Proceedings*, Springer-Verlag, 1997, pp. 13–27.

[Hit97] Hitachi, "Symmetric key encipherment method 'M6' for IEEE 1394 bus encryption/authentication," Submission 1997-4-25, Proposal for IEEE 1394, Copy Prorection Technical Working Group, 1997.

[Kaz99] T. Kazuo, personal communication, 19 Mar 1999.

[KM96] L.R. Knudsen and W. Meier, "Improved Differential Attacks on RC5," *Advances in Cryptology — CRYPTO '96*, Springer-Verlag, 1996, pp. 216–228.

[KR94] B. Kaliski Jr., and M. Robshaw, "Linear Cryptanalysis Using Multiple Approximations," *Advances in Cryptology — CRYPTO '94 Proceedings*, Springer-Verlag, 1994, pp. 26–39.

[KR95] B. Kaliski Jr., and M. Robshaw, "Linear Cryptanalysis Using Multiple Approximations and FEAL," *Fast Software Encryption, Second International Workshop Proceedings*, Springer-Verlag, 1995, pp. 249–264.

[KR96] L. Knudsen and M. Robshaw, "Non-Linear Approximations in Linear Cryptanalysis," *Advances in Cryptology — EUCROCRYPT '96*, Springer-Verlag, 1996, pp. 224–236.

[KY95] B. Kaliski and Y.L. Yin, "On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm," *Advanced in Cryptology — CRYPTO '95*, Springer-Verlag, 1995, pp. 171–184.

[KY98] B. Kaliski and Y.L. Yin, "On the Security of the RC5 Encryption Algorithm," RSA Laboratories Technical Report TR-602, Version 1.0, Sep 98.

[LMM91] X. Lai, J. Massey, and S. Murphy, "Markov Ciphers and Differential Cryptanalysis," *Advances in Cryptology — CRYPTO '91 Proceedings*, Springer-Verlag, 1991, pp. 17–38.

[Mad84] W.E. Madryga, "A High Performance Encryption Algorithm," *Computer Security: A Global Challenge*, Elsevier Science Publishers, 1984, pp. 557–570.

[Mat94] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *Advances in Cryptology — EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 386–397.

[Riv95] R.L. Rivest, "The RC5 Encryption Algorithm," *Fast Software Encryption, 2nd International Workshop Proceedings*, Springer-Verlag, 1995, pp. 86–96.

[RRS+98] R. Rivest, M. Robshaw, R. Sidney, and Y.L. Yin, "The RC6 Block Cipher," NIST AES Proposal, Jun 98.

[Sel98] A.A. Selcuk, "New REsults in Linear Cryptanalysis of RC5," *Fast Software Encryption, 5th International Workshop*, Springer-Verlag, 1998, pp. 1–16.

[SKW+98] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish: A 128-Bit Block Cipher," NIST AES Proposal, 15 June 1998.

[THN91] K. Takaragi, K. Hashimoto, and T. Nakamura, "On Differential Cryptanalysis," *IEICE Tranactions*, vol E-74, n. 8, Aug 1991, pp. 2153-2158.

[Vau96] S. Vaudenay, "An Experiment on DES Statistical Cryptanalysis," *3rd ACM Conference on Computer and Communications Security*, ACM Press, 1996, pp. 139–147.

A The χ^2 Test

In this section we study how to distinguish a source with distribution p_X from a source with the uniform distribution p_U. The optimal algorithm is the χ^2 test, and we briefly recall its definition, as well as several standard results, here.

The χ^2 test allows one to test the hypothesis that the source has distribution p_U. Suppose we have n (independent) observations, and let n_i denote the number of times the source took on the value i. Treating each n_i as a random variable (subject to the constraint that $\sum n_i = n$), the χ^2 statistic is defined as:

$$\chi^2(n_1, \ldots, n_k) = \sum_i \frac{(n_i - \mathbf{E}_U n_i)^2}{\mathbf{E}_U n_i}.$$

Here $\mathbf{E}_U n_i$ denotes the expected value of n_i under the assumption that the source has distribution p_U. It is not hard to see that $\mathbf{E}_U n_i = n/k$, so the χ^2 statistic is just $k/n \sum_i (n_i - n/k)^2$. In the χ^2 test, we compare the observed χ^2 statistic to $\chi^2_{a,k-1}$ (the threshold for the χ^2 test with $k-1$ degrees of freedom and with significance level a).

We can easily compute the expected value of the χ_2 statistic.

Theorem 3. $\mathbf{E}_X \chi^2(n_1, \ldots, n_k) = nk||p_X - p_U||^2 + k - k||p_X||^2$.

Corollary 4. $\mathbf{E}_U \chi^2 = k - 1$.

We can see that if $n = c/||p_X - p_U||^2$, then $\mathbf{E}_X \chi^2 = ck + k - k||p_X||^2$. Since we will usually be interested in the case where $p_X \approx p_U$, we find $\mathbf{E}_X \chi^2 \approx (c+1)k - 1$. Thus $\mathbf{E}_X \chi^2$ differs from $\mathbf{E}_U \chi^2$ by a significant amount when $c = \Omega(1)$.

In summary, we can conclude that $n = \Theta(1/||p_X - p_U||^2)$ observations suffice to distinguish a source with distribution p_X from a source with distribution p_U. This shows that our definition of the bias of p_X as $||p_X - p_U||^2$ was well-chosen.