Uma versão mais forte do algoritmo RC6 contra criptanálise χ^2

Eduardo Takeo Ueda[1], **Routo Terada**[1]

[1]Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo
São Paulo – SP – Brazil

edutakeo@ime.usp.br, rt@ime.usp.br

Abstract. *We analyze the χ^2 attack, one of the most successful cryptanalysis technique against the RC6 algorithm. We apply this type of cryptanalysis as distinction attack as well as key-recovery attack. We present a modified version of RC6 by introducing a swapping function in its structure. The conclusions inferred by stastistical experiments is that this modified version is stronger against the χ^2 cryptanalys technique.*

Resumo. *Neste artigo analisamos uma das técnicas de criptanálise mais bem sucedidas contra o algoritmo RC6, o ataque χ^2. Utilizamos este tipo de criptanálise como ataque de distinção e também como ataque de recuperação da chave. Apresentamos uma versão modificada do algoritmo RC6 que foi projetada através da introdução de uma função de troca em sua estrutura. Constatamos através de experimentos estatísticos que esta nova versão é mais forte contra a técnica de criptanálise χ^2.*

1. Introdução

O algoritmo RC6 é um cifrador de blocos simétrico e foi desenvolvido a partir do RC5, seu antecessor, a fim de ser submetido ao *NIST (National Institute of Standards and Technology)* como candidato ao *AES (Advanced Encryption Standard)*. Ele é de autoria de Ronald L. Rivest, Matthew J. B. Robshaw, Ray Sidney e Yiqun L. Yin [Rivest et al. 1998], pesquisadores do *MIT(Massachusetts Institute of Technology)* e dos Laboratórios RSA.

A técnica de criptanálise χ^2 foi proposta originalmente por Serge Vaudenay [Vaudenay 1996] como um ataque sobre o algoritmo *DES*. Knudsen e Meier [Knudsen and Meier 2000] foram os primeiros a aplicar o ataque χ^2 sobre o algoritmo RC6, e estimaram que é possível atacar o RC6 até 15 iterações com este tipo de criptanálise. Depois disso, outras pesquisas [Isogai et al. 2003, Miyaji and Nonaka 2002, Miyaji and Nonaka 2003, Miyaji and Takano 2005, Takenaka et al. 2004] também fizeram estudos desta técnica de criptanálise contra o RC6. Em [Isogai et al. 2003] e [Miyaji and Nonaka 2002] foram consideradas variantes do RC6 denotadas por RC6W e RC6P, respectivamente. RC6W significa RC6 sem *"pre-"* ou *"post-whitening"* e RC6P significa RC6 sem *"post-whitening"*. A parte *"pre-whitening"* do algoritmo RC6 consiste na adição das subchaves $S[0]$ e $S[1]$ antes do laço de iteração, enquanto a parte *"post-whitening"* é a adição das subchaves $S[2r+2]$ e $S[2r+3]$ logo após o laço de iteração.

Aplicar o ataque χ^2 ao RC6 considerando 16 ou mais iterações foi tratado por Knudsen e Meier como uma questão em aberto. Mas, os estudos teóricos de Miyaji e Takano em [Miyaji and Takano 2005] demonstraram que é possível atacar o algoritmo RC6 com 16 iterações usando $2^{127.20}$ textos legíveis. A criptanálise χ^2 aplicada sobre o algoritmo RC6 em [Miyaji and Takano 2005] apresenta os melhores resultados conhecidos desta técnica atualmente.

A fim de fortalecer o algoritmo RC6 contra criptanálise χ^2 consideramos uma nova versão, denotada por RC6T, que foi obtida através da introdução de uma função de troca na estrutura do algoritmo. Esta função de troca foi utilizada em [Terada and Junior 2003, Terada et al. 1996] e consiste em inverter as duas metades com 16 bits menos e mais significativos de uma seqüência de 32 bits, caso o número de bits 1 desta seqüência seja ímpar.

Essa função de troca é uma permutação de bits que apresenta a propriedade de *difusão* [Terada 2000] e sua utilização na estrutura do RC6 faz aumentar o nível de aleatoriedade do algoritmo. Porém, o acréscimo desta função diminui a performance do algoritmo em cerca de 17% quando comparamos versões do RC6 e RC6T com o mesmo número de iterações.

Entrada: Texto legível armazenado em quatro registradores de w-bits: A, B, C, D
Número r de iterações
$2r + 4$ subchaves de w-bits armazenadas em $S[0, \ldots, 2r + 3]$

Saída: Texto cifrado armazenado em A, B, C, D

Procedimento:

$B = B + S[0]$
$D = D + S[1]$
para $i = 1$ até r faça{
  $B = T(B)$
  $D = T(D)$
  $t = (B \times (2B + 1)) \lll \lg w$
  $u = (D \times (2D + 1)) \lll \lg w$
  $A = ((A \oplus t) \lll u) + S[2i]$
  $C = ((C \oplus u) \lll t) + S[2i + 1]$
  $(A, B, C, D) = (B, C, D, A)$
}
$A = A + S[2r + 2]$
$C = C + S[2r + 3]$

Figure 1. Algoritmo de cifração RC6T

Representamos a função de troca por $T()$ e na Figura 1 temos a descrição do algoritmo RC6T. A única diferença do RC6T com relação ao RC6 é o acréscimo de $B = T(B)$ e $D = T(D)$ dentro do laço de iteração. A partir deste ponto vamos nos dedicar a mostrar que está versão modificada do algoritmo RC6 é mais forte contra a técnica de criptanálise χ^2 do que a versão original submetida como candidato ao *AES*.

2. Fatos estatísticos

Nesta seção, explicamos como fazer uso da estatística χ^2 para distinguir um código com distribuição de probabilidade desconhecida $\boldsymbol{p}$ de um código com distribuição de probabilidade uniforme π [Kelsey et al. 1999, Knudsen and Meier 2000, Knuth 1981].

Sejam $X = X_0, X_1, \ldots, X_{n-1}$ variáveis aleatórias independentes tais que $X_i \in \{a_0, a_1, \ldots, a_{m-1}\}$ com distribuição de probabilidade desconhecida $\boldsymbol{p}$, e $N_{a_j}(X)$ o número de vezes que X assume o valor a_j. A estatística χ^2 de X que estima a distância entre a distribuição observada $\boldsymbol{p}$ e a distribuição uniforme esperada $\pi = (\pi_0, \pi_1, \ldots, \pi_{m-1})$ é definida como:

$$\chi^2 = \sum_{i=0}^{m-1} \frac{(N_{a_i}(X) - n\pi_i)^2}{n\pi_i}.$$

É óbvio que $\sum_{j=0}^{m-1} N_{a_j}(X) = n$, e como a probabilidade de cada π_i é $\frac{1}{m}$, pois a distribuição π é uniforme, podemos simplificar a equação anterior obtendo a seguinte expressão:

$$\chi^2 = \frac{m}{n} \sum_{i=0}^{m-1} \left(N_{a_i}(X) - \frac{n}{m}\right)^2.$$

Em um teste χ^2, a estatística χ^2 é comparada a $\chi^2_{a,m-1}$, o valor para o teste χ^2 com $m-1$ graus de liberdade com nível de significância a. Assim, depois de calculada a estatística χ^2 é possível efetuar a decisão no seguinte teste de hipótese:

$$\begin{cases} H_0 : \boldsymbol{p} = \pi & (\textit{hipótese nula}) \\ H_1 : \boldsymbol{p} \neq \pi & (\textit{hipótese alternativa}) \end{cases}$$

A Tabela 1 apresenta valores limiares para a distribuição χ^2 com 63, 255 e 1023 graus de liberdade. Tais valores foram utilizados por Knudsen e Meier [Knudsen and Meier 2000], entretanto neste artigo consideramos apenas o caso com 63 graus de liberdade. Por exemplo, para 63 graus de liberdade (nível, χ^2) = $(0.95, 82)$ na Tabela 1 significa que o valor da estatística χ^2 excederá 82 somente 5% das vezes se a distribuição da observação X for realmente uniforme.

Table 1. Distribuição χ^2 com diferentes graus de liberdade

Nível	0.5	0.60	0.70	0.80	0.90	0.95	0.99
63 graus de liberdade	62	65	68	72	77	82	92
255 graus de liberdade	254	260	266	273	284	293	310
1023 graus de liberdade	1022	1033	1046	1060	1081	1098	1131

Os Teoremas **2.1** e **2.2** [Miyaji and Takano 2005, Ryabko 2003], a seguir, dizem respeito a estatística χ^2 e são resultados importantes para o estudo de criptanálise χ^2.

Teorema 2.1 *Quando H_0 é verdadeira, a estatística χ^2 definida nesta seção segue a distribuição χ^2 cujo grau de liberdade é aproximadamente $m-1$. Em adição, a média ou variância esperada é calculada por $E_{H_0}(\chi^2) = m-1$ ou $V_{H_0}(\chi^2) = 2(m-1)$, respectivamente.*

Teorema 2.2 *Quando H_1 é verdadeira, a estatística χ^2 definida nesta seção segue a distribuição χ^2 não-central cujo grau de liberdade é aproximadamente $m-1$. Em adição, a média ou variância esperada é calculada por $E_{H_1}(\chi^2) = m-1+n\theta$ ou $V_{H_1}(\chi^2) = 2(m-1)+4n\theta$, respectivamente, onde $n\theta$ é chamado de parâmetro não-central e tal que $n\theta = n\sum_{i=0}^{m-1}\frac{(\pi_i - P(a_i))^2}{\pi_i}$, sendo $P(a_i)$ a probabilidade de ocorrência de a_i.*

3. Correlações medidas

Agora, investigaremos a não aleatoriedade dos algoritmos RC6 e RC6T com r iterações. Esta análise é baseada em experimentos sistemáticos, levando em consideração o aumento do número de iterações nos algoritmos com palavras de tamanho $w = 32$ bits. Ressaltamos que nos experimentos realizados a geração de textos legíveis e chaves utilizadas foram feitas com distribuição uniforme através do método de congruência linear.

O método de Knudsen e Meier [Knudsen and Meier 2000] é usado para demonstrar que detecção e quantificação de não aleatoriedade pode ser medida nos algoritmos RC6 e RC6T até 5 iterações através de implementações em software. Para este propósito, destacamos a utilização de dois tipos de testes como descrito a seguir. Em ambos os testes consideramos que (A_0, B_0, C_0, D_0) é um texto legível, $(A_{r+2}, B_{r+2}, C_{r+2}, D_{r+2})$ é um texto cifrado depois de r iterações, $lsb_n(X)$ representa os n bits menos significativos de X, e $a||b$ é a operação de concatenação dos valores a e b.

Teste 1: χ^2 de $lsb_3(A_{r+1})||lsb_3(C_{r+1})$ no caso de $lsb_5(A_0) = lsb_5(C_0) = 0$.

Teste 2: χ^2 de $lsb_3(A_{r+1})||lsb_3(C_{r+1})$ no caso de $lsb_5(B_0) = lsb_5(D_0) = 0$.

É importante frisar que o **Teste 1** aplicado a versões do RC6 e RC6T com número de iterações par acarreta valores χ^2 maiores do que o **Teste 2** para o mesmo número de iterações. Assim como o **Teste 2** aplicado a versões com número de iterações ímpar apresenta valores χ^2 mais elevados que o **Teste 1**.

Um outro ponto que é necessário ser observado com relação aos 2 tipos de testes que estamos considerando é que devido ao fato deles fixarem 10 bits de cada texto legível como zero, o número de textos diferentes possíveis que podem ser gerados nos experimentos que efetuamos é reduzido de 2^{128} para 2^{118}.

As Tabelas 2 e 3 apresentam as correlações medidas no RC6 com o **Teste 1** e **Teste 2**, respectivamente. Note que nestes testes calcula-se a estatística χ^2 de valores inteiros $lsb_3(A_{r+1})||lsb_3(C_{r+1})$ de 6 bits, e então, o valor esperado da estatística χ^2 é 63.

As Tabelas 4 e 5 apresentam os valores das estatísticas χ^2 medidas no RC6T com o **Teste 1** e **Teste 2**, respectivamente. Enfatizando que no algoritmo RC6T também foi calculado a estatística χ^2 de valores inteiros $lsb_3(A_{r+1})||lsb_3(C_{r+1})$ de 6 bits. Quando consideramos versões do RC6T com 4 e 5 iterações os valores da estatística χ^2 foram cal-

Table 2. Teste 1 no RC6-$32/r/16$ com $r = 2, 4$ iterações

r	$\log_2$(# textos)	χ^2	# testes
2	13	54	20
2	14	72	20
2	15	96	20
4	30	59	10
4	31	134	10
4	32	226	10

Table 3. Teste 2 no RC6-$32/r/16$ com $r = 3, 5$ iterações

r	$\log_2$(# textos)	χ^2	# testes
3	14	59	20
3	15	72	20
3	16	90	20
5	32	74	10
5	33	115	10
5	34	205	10

culadas através de apenas 1 teste ao invés de fazer a média de 10 testes, pois o custo computacional nestes casos aumentou consideravelmente comparado com os mesmos testes aplicados ao algoritmo RC6.

Table 4. Teste 1 no RC6T-$32/r/16$ com $r = 2, 4$ iterações

r	$\log_2$(# textos)	χ^2	# testes
2	17	60	20
2	18	79	20
2	19	123	20
4	36	79	1
4	37	137	1
4	38	301	1

Table 5. Teste 2 no RC6T-$32/r/16$ com $r = 3, 5$ iterações

r	$\log_2$(# textos)	χ^2	# testes
3	21	58	20
3	22	73	20
3	23	112	20
5	39	54	1
5	40	128	1
5	41	275	1

Os resultados apresentados nesta seção mostram claramente que é necessário mais textos legíveis para se medir estatísticas χ^2 sobre o algoritmo RC6T equivalentes às medidas no RC6 quando consideramos o mesmo número de iterações. Isso significa que esta

versão do algoritmo RC6 é mais forte contra ataques de distinção e recuperação da chave, como veremos com maiores detalhes nas seções posteriores.

4. Ataque de distinção

É possível explorar os resultados discutidos até agora para distinguir os algoritmos RC6 e RC6T com um certo número de iterações de uma permutação aleatoriamente escolhida no conjunto de todas as permutações. Ataques desta natureza são denominados ataques de distinção e são importantes porque muitas vezes eles auxiliam na criação de ataques de recuperação da chave, sendo que estes últimos costumam ser mais sofisticados. A Figura 2 apresenta o ataque de distinção proposto por Knudsen e Meier [Knudsen and Meier 2000] e que aplicamos aos algoritmos RC6 e RC6T.

Entrada: Algoritmo RC6(RC6T) ou permutação aleatória,
Número n de textos;

Saída: Responde se a entrada é RC6(RC6T) ou não;

1. considere n textos (A, B, C, D) cifrados com RC6(RC6T) ou não;
2. para $i = 1$ até n faça:
   $j = lsb_3(A_i)||lsb_3(C_i)$;
   incremente $array[j]$;
3. calcule χ^2 de $array$;
4. se (χ^2 calculado) ≥ 82
   então retorna *"A entrada é RC6(RC6T)"*;
   senão retorna *"A entrada é uma permutação aleatória"*.

Figure 2. Ataque de distinção

O algoritmo de ataque de distinção da Figura 2 analisa uma seqüência de n textos, decidindo se estes valores foram cifrados com o algoritmo RC6(RC6T) ou não. A condição ≥ 82 indica que o algoritmo retorna a resposta correta com 95% de certeza de acordo com a Tabela 1. É importante ressaltar que a escolha do número de textos n é dependente do número de iterações que se considera nos algoritmos. A seguir, apresentamos os resultados da aplicação do ataque de distinção sobre o RC6 e RC6T.

A Tabela 6 lista os resultados de testes implementados para o RC6-$32/r/16$ com $r = 2, 4$ iterações considerando o **Teste 1**. Temos que 2^{15} textos são suficientes para distinguir uma permutação cifrada com 2 iterações de uma permutação escolhida aleatoriamente em 95% dos casos, pois com 2^{15} textos o valor χ^2 medido 92 é maior que 82. Considerando um fator igual a $2^{15.7}$ $(= 2^{30.7-15})$ textos adicionais a cada 2 iterações, estimamos que para o RC6 com r iterações, valores similares serão alcançados com $2^{15}(2^{15.7})^{\frac{r-2}{2}} = 2^{7.85r-0.7}$ textos e assim, $\log_2(\#textos) = 7.85r - 0.7$.

Os valores apresentados na Tabela 6 indicam que é possível distinguir o RC6 de uma permutação aleatória para versões até 14 iterações quando consideramos o **Teste 1**. Note que para atacar 14 iterações precisamos de $2^{109.2}$ textos e para 16 iterações o número de textos necessários é bem maior que 2^{118}, ou seja, o número de textos disponíveis.

Table 6. Complexidade para distinguir o RC6-$32/r/16$ usando o Teste 1

r	$\log_2$(# textos)	χ^2	Comentário
2	13	59	Implementado, média de 20 testes
2	14	72	Implementado, média de 20 testes
2	15	92	Implementado, média de 20 testes
4	30	66	Implementado, média de 10 testes
4	30.7	107	Implementado, média de 10 testes
6	46.4		Estimado
8	62.1		Estimado
10	77.8		Estimado
12	93.5		Estimado
14	109.2		Estimado
16	124.9		Estimado

Na Tabela 7 temos os resultados de testes implementados para o RC6-$32/r/16$ com $r = 3, 5$ iterações utilizando-se o **Teste 2**. Observamos que 2^{16} textos são suficientes para distinguir uma permutação cifrada com 3 iterações de uma permutação aleatória, uma vez que para 2^{16} textos o valor χ^2 medido é maior que 82. Notando ainda que existe um fator de 2^{16} ($= 2^{32-16}$) textos adicionais para se medir valores χ^2 equivalentes a cada 2 iterações estimamos que, para o RC6 com r iterações, resultados semelhantes serão alcançados com $2^{16}(2^{16})^{\frac{r-3}{2}} = 2^{8r-8}$ textos. Logo, quando nos baseamos em testes implementados sobre o RC6 com 3 e 5 iterações temos que $\log_2(\#textos) = 8r - 8$ para o caso de r iterações.

Os resultados da Tabela 7 indicam que é possível distinguir o RC6 de uma permutação aleatória para versões até 15 iterações quando consideramos o **Teste 2**. Note que para 17 iterações o número de textos necessários é maior que 2^{118}.

Table 7. Complexidade para distinguir o RC6-$32/r/16$ usando o Teste 2

r	$\log_2$(# textos)	χ^2	Comentário
3	14	55	Implementado, média de 20 testes
3	15	72	Implementado, média de 20 testes
3	16	91	Implementado, média de 20 testes
5	31	53	Implementado, média de 10 testes
5	32	95	Implementado, média de 10 testes
7	48		Estimado
9	64		Estimado
11	80		Estimado
13	96		Estimado
15	112		Estimado
17	128		Estimado

A Tabela 8 apresenta os resultados de testes implementados para o RC6T-$32/r/16$ com $r = 2, 4$ iterações considerando o **Teste 1**. Temos que $2^{18.2}$ textos são suficientes para distinguir uma permutação cifrada com 2 iterações de uma permutação escolhida aleatoriamente em 95% dos casos. E notando a existência de um fator igual a $2^{17.8}$

$(= 2^{36-18.2})$ textos adicionais para se medir valores χ^2 equivalentes a cada 2 iterações estimamos que, para o RC6T com r iterações, valores similares serão conseguidos com $2^{18.2}(2^{17.8})^{\frac{r-2}{2}} = 2^{8.9r+0.4}$ textos. Assim, nos baseando nos testes implementados e apresentados na Tabela 8 temos que $\log_2(\#textos) = 8.9r + 0.4$ para r iterações.

Table 8. Complexidade para distinguir o RC6T-$32/r/16$ usando o Teste 1

r	$\log_2$(# textos)	χ^2	Comentário
2	17	57	Implementado, média de 20 testes
2	18	76	Implementado, média de 20 testes
2	18.2	95	Implementado, média de 20 testes
4	35	64	Implementado, apenas 1 teste
4	36	87	Implementado, apenas 1 teste
6	53.8		Estimado
8	71.6		Estimado
10	89.4		Estimado
12	107.2		Estimado
14	125		Estimado
16	142.8		Estimado

Os valores mostrados na Tabela 8 demonstram que é possível distinguir o RC6T de uma permutação aleatória para versões até 12 iterações quando consideramos o **Teste 1**. Já havíamos visto que 14 iterações do RC6 podem ser atacadas fazendo uso do **Teste 1** com $2^{109.2}$ textos. Mas note que para 14 iterações o número de textos necessários para atacar o RC6T é maior que 2^{118}.

Table 9. Complexidade para distinguir o RC6T-$32/r/16$ usando o Teste 2

r	$\log_2$(# textos)	χ^2	Comentário
3	21	51	Implementado, média de 20 testes
3	22	78	Implementado, média de 20 testes
3	22.6	92	Implementado, média de 20 testes
5	39	66	Implementado, apenas 1 teste
5	39.4	84	Implementado, apenas 1 teste
7	56.2		Estimado
9	73		Estimado
11	89.8		Estimado
13	106.6		Estimado
15	123.4		Estimado
17	140.2		Estimado

Na Tabela 9 temos os resultados de testes implementados para o RC6T-$32/r/16$ com $r = 3, 5$ iterações utilizando-se o **Teste 2**. Observamos que $2^{22.6}$ textos é um valor aceitável para distinguir uma permutação cifrada com 3 iterações de uma permutação aleatória. Também é possível notar um fator de $2^{16.8}(= 2^{39.4-22.6})$ textos adicionais para se medir valores χ^2 equivalentes a cada 2 iterações. Assim, estimamos que para o RC6T com r iterações, resultados semelhantes serão alcançados com $2^{22.6}(2^{16.8})^{\frac{r-3}{2}} = 2^{8.4r-2.6}$

textos. Logo, nos baseando em testes implementados sobre o RC6 com 3 e 5 iterações temos que $\log_2(\#textos) = 8.4r - 2.6$ para o caso de r iterações.

Os resultados da Tabela 9 indicam que é possível distinguir o RC6T de uma permutação aleatória para versões até 13 iterações quando consideramos o **Teste 2** apresentado na seção 3. Com o mesmo teste pode-se atacar 15 iterações do algoritmo RC6 com 2^{112} textos, como já foi discutido. Note que para 15 iterações do RC6T o número de textos necessários é bem maior que 2^{118}.

5. Ataque de recuperação da chave

Apresentamos nesta seção um algoritmo de ataque desenvolvido por Isogai, Matsunaka e Miyaji [Isogai et al. 2003] e que recupera alguns bits das subchaves utilizadas na última iteração de cifração dos algoritmo RC6 e RC6T sem *"post-whitening"*, que denotamos por RC6P e RC6TP, respectivamente. Este ataque é baseado no **Teste 2**, que fixa $lsb_5(B_0)$ e $lsb_5(D_0)$ como zero e considera $lsb_3(A_{r+1})$ e $lsb_3(C_{r+1})$ para o cálculo da estatística χ^2. Assumimos que, para obtermos valores semelhantes em um teste χ^2 sobre $r+2$ iterações comparado a r iterações, é requerido um fator aproximadamente igual a 2^{16} textos adicionais.

Em [Miyaji and Nonaka 2002] foi demonstrado que não necessariamente é preciso considerar o nível de significância 0.95 como em [Knudsen and Meier 2000] para se recuperar uma chave correta. No caso do **Teste 2** um nível maior que 0.57 é suficiente para a recuperação da chave. É evidente que bem menos textos são necessários para o algoritmo de ataque com um nível aproximadamente pouco maior que 0.57 se comparado com 0.95.

1. Escolha um texto legível (A_0, B_0, C_0, D_0) tal que $lsb_5(B_0) = lsb_5(D_0) = 0$ e o cifre por r iterações.
2. Para cada (s_a, s_c), decifre $y_b||y_d$ com a chave $0||s_a, 0||s_c$ por 1 iteração para $z_a||z_c$, sendo $z = z_a||z_c$ um inteiro de 6 bits.
3. Para cada s, x_a, x_c e z, atualize cada vetor incrementando $count[s][x_a][x_c][z]$.
4. Para cada s, x_a e x_c, calcule $\chi^2[s][x_a][x_c]$.
5. Calcule a média $med[s]$ de $\{\chi^2[s][x_a][x_c]\}_{x_a,x_c}$ para cada s e retorne o valor s com maior $med[s]$ como $lsb_2(S[2r])||lsb_2(S[2r+1])$.

Figure 3. Ataque de recuperação da chave

A Figura 3 descreve o algoritmo de ataque de recuperação da chave aplicado aos algoritmos RC6P e RC6TP. Intuitivamente, o algoritmo fixa $lsb_3(B_0)$ e $lsb_3(D_0)$, calcula a estatística χ^2 sobre valores inteiros de 6 bits obtido da concatenação de 3 bits de A_r com 3 bits de C_r e recupera $lsb_2(S[2r])$, $lsb_2(S[2r+1])$ usadas na *r-ésima* iteração dos algoritmos RC6P e RC6TP. Adotamos as seguintes notações no algoritmo de ataque: $(y_b, y_d) = (lsb_3(B_{r+1}), lsb_3(D_{r+1}))$, $(x_a, x_c) = (lsb_5(F(C_{r+1})), lsb_5(F(A_{r+1})))$, $(s_a, s_c) = (lsb_2(S[2r]), lsb_2(S[2r+1]))$ e $s = s_a||s_c$, onde x_a (respectivamente x_c) é a quantidade de rotação sobre A_r (respectivamente C_r) na *r-ésima* iteração de RC6P ou RC6TP e $F(x) = [x(2x+1) \pmod{2^w}] \lll \lg w$.

O algoritmo de ataque de recuperação da chave da Figura 3 pode ser generalizado para recuperar e bits da chave, sendo e um número par. Neste caso, z é um número de $(e+2)$ bits, sobre o qual o valor χ^2 é calculado. Os textos legíveis no algoritmo de ataque são classificados em 2^{10} grupos através de $\{x_a, x_c\}$ e a média $med[s]$ é calculada sobre cada grupo. Em outras palavras, todos os textos legíveis são distribuídos uniformemente em cada grupo desde que eles sejam gerados aleatoriamente para os experimentos.

No caso da aplicação do ataque sobre o algoritmo RC6P os experimentos realizados indicam que é necessário $2^{21.8}$ textos em cada experimento a fim de que o algoritmo recupere a chave correta com 95% de probabilidade de sucesso. Este fato foi constatado com a realização de 100 experimentos, onde o ataque foi bem sucedido 95% das vezes, e a média dos valores χ^2 das chaves corretas retornadas pelo algoritmo de ataque foi 64.684, o que corresponde um nível de 0.57 aproximadamente.

Pelo uso dos resultados medidos em 100 experimentos e sabendo que é preciso 2^{16} textos adicionais para se conseguir valores χ^2 equivalentes em $r+2$ iterações comparado a r iterações, temos que o número de textos necessários para atacar o algoritmo RC6P com r iterações e probabilidade de sucesso de 95% é dado por:

$$2^{-8}2^{21.8}(2^{16})^{\frac{r-3}{2}} = 2^{8r-10.2}.$$

Note que o fator 2^{-8} decorre do fato do algoritmo de ataque realizar uma decifração por 1 iteração apenas, implicando uma diminuição do número de textos. Assim, com $\log_2(\#\,\text{textos}) = 8r - 10.2$ o algoritmo de ataque recupera a chave correta com probabilidade de sucesso de 95%.

Para investigar a complexidade de tempo, ou seja, a quantidade de trabalho do algoritmo de ataque, consideramos que uma unidade de trabalho é equivalente a um incremento do vetor $count[s][x_a][x_c][z]$. Como temos 2^4 pares (s_a, s_c) para cada texto legível e para cada par destes corresponde um incremento, então a quantidade de trabalho é dada por:

$$(\#\ \textit{de textos})\times 2^4 = 2^{8r-10.2} \times 2^4 = 2^{8r-6.2}.$$

Então, substituindo o número de textos legíveis disponíveis 2^{118} na expressão que determina a quantidade de textos necessários para atacar o algoritmo RC6P, concluímos que o ataque pode quebrar 16 iterações do RC6P usando para isso $2^{117.8}$ textos legíveis e $2^{121.8}$ unidades de trabalho.

Aplicamos o mesmo ataque de recuperação da chave que foi utilizado contra o RC6P sobre o algoritmo RC6T sem *"post-whitening"*, que estamos denotando por RC6TP. Segundo as medidas estatísticas dos testes χ^2 da seção 3, considerando-se o **Teste 2**, é preciso aproximadamente 2^{17} textos adicionais para se alcançar valores χ^2 equivalentes em $r+2$ iterações comparado a r iterações.

A realização de sistemáticos experimentos indica que é necessário $2^{27.2}$ textos em cada experimento para que o algoritmo de ataque recupere a chave correta do RC6TP com 95% de probabilidade de sucesso. Isto foi verificado através de 100 experimentos, onde o ataque teve êxito 95% das vezes, e a média dos valores χ^2 das chaves corretas retornadas pelo algoritmo de ataque foi 64.534, o que corresponde um nível de 0.57 aproximada-

mente. Considerando os resultados obtidos nestes 100 experimentos e sabendo que é preciso 2^{17} textos adicionais para se conseguir valores χ^2 equivalentes em $r+2$ iterações comparado a r iterações, temos que o número de textos necessários para atacar o algoritmo RC6TP com r iterações e probabilidade de sucesso de 95% é dado por:

$$2^{-8.5}2^{27.2}(2^{17})^{\frac{r-3}{2}} = 2^{8.5r-6.8}.$$

Salientamos a existência de um fator igual a $2^{-8.5}$ que é conseqüência do fato do algoritmo de ataque realizar uma decifração por 1 iteração apenas, provocando uma diminuição do número de textos. Assim, com $\log_2(\#\,\text{textos}) = 8.5r - 6.8$ o algoritmo de ataque recupera a chave correta com probabilidade de sucesso de 95%. Também investigamos a complexidade de tempo, ou seja, a quantidade de trabalho. Para tal análise consideramos que uma unidade de trabalho é equivalente a um incremento do vetor $count[s][x_a][x_c][z]$. Como temos 2^4 pares (s_a, s_c) para cada texto legível e para cada par destes corresponde um incremento, a quantidade de trabalho é dada por:

$$(\#\ \textit{de textos})\times 2^4 = 2^{8.5r-6.8} \times 2^4 = 2^{8.5r-2.8}.$$

Substituindo o número de textos legíveis disponíveis 2^{118} na expressão que determina a quantidade de textos necessários para atacar o algoritmo RC6TP, concluímos que o ataque pode quebrar 14 iterações do RC6TP usando para isso $2^{112.2}$ textos legíveis e $2^{116.2}$ unidades de trabalho.

A partir dos resultados obtidos podemos afirmar que o algoritmo RC6TP é mais forte contra o ataque de recuperação da chave que o algoritmo RC6P, uma vez que a introdução da função de troca $T()$ fez com que 14 iterações pudessem ser atacadas ao invés de 16 iterações.

6. Conclusão

Baseados em resultados experimentais estimamos que um ataque de distinção pode ser aplicado a versões do RC6 com até 15 iterações. No caso de 15 iterações é necessário 2^{112} textos legíveis para atacar o algoritmo RC6. Porém, para o algoritmo RC6T os experimentos indicam que versões com até 13 iterações podem ser distinguidas de uma permutação aleatória. Para 13 iterações $2^{106.6}$ textos legíveis são exigidos para se atacar o algoritmo RC6T.

Implementamos um ataque de recuperação da chave do tipo *texto-legível-escolhido* sobre o algoritmo RC6 sem *"post-whitening"* e que foi proposto em [Isogai et al. 2003]. A análise dos resultados experimentais deste ataque indica que 16 iterações do algoritmo RC6P podem ser atacadas usando-se $2^{117.8}$ textos legíveis e a probabilidade de se recuperar a chave correta neste caso é de 95%. Aplicamos o mesmo ataque de recuperação da chave no algoritmo RC6T sem *"post-whitening"* e verificamos que neste caso, menos iterações podem ser atacadas. Quando consideramos o RC6TP, os experimentos sugerem que 14 iterações podem ser atacadas usando $2^{112.2}$ textos legíveis e com probabilidade de sucesso de 95%.

Por fim, enfatizamos que a introdução da função de troca na estrutura do algoritmo RC6 fortalece ele contra a técnica de criptanálise χ^2. Com o RC6T é necessário

um número maior de textos legíveis para se alcançar valores χ^2 semelhantes aos que foram obtidos na versão do RC6 submetida como candidato ao *AES*. Este mesmo efeito se estende aos ataques de distinção e de recuperação da chave, dificultando distinguir o algoritmo RC6T de uma permutação aleatória e recuperar uma chave correta com alta probabilidade, respectivamente.

References

Isogai, N., Matsunaka, T., and Miyaji, A. (2003). *Optimized χ^2-Attack against RC6*. *Applied Cryptography and Network Security*, pages 16–32.

Kelsey, J., Schneier, B., and Wagner, D. (1999). *Mod n Cryptanalysis, with Applications Against RC5P and M6*. *Lecture Notes in Computer Science*, 1636:139–155.

Knudsen, L. R. and Meier, W. (2000). *Correlations in RC6 with a Reduced Number of Rounds*. *Proceedings of the 7th International Workshop on Fast Software Encryption*.

Knuth, D. E. (1981). *The Art of Computer Programming, Volume 2*, volume 2. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2 edition.

Miyaji, A. and Nonaka, M. (2002). *Cryptanalysis of the Reduced-Round RC6*. *International Conference on Information and Communications Security*, pages 480–494.

Miyaji, A. and Nonaka, M. (2003). *Cryptanalysis of Reduced-Round RC6 without Whitening*. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, E86-A(1):19–30.

Miyaji, A. and Takano, Y. (2005). *On the Success Probability of χ^2-attack on RC6*. *Australasian Conference on Information Security and Privacy*, pages 61–74.

Rivest, R. L., Robshaw, M. J. B., Sidney, R., and Yin, Y. L. (1998). *The RC6 Block Cipher. Version 1.1*.

Ryabko, B. (2003). *Adaptive Chi-Square Test and Its Application to Some Cryptographic Problems*. *Cryptology ePrint Archive*.

Takenaka, M., Shimoyama, T., and Koshiba, T. (2004). *Theoretical Analysis of χ^2 Attack on RC6*. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, E87-A(1):28–35.

Terada, R. (2000). *Segurança de Dados - Criptografia em Redes de Computador*. Editora Edgard Blücher, São Paulo, SP, 1 edition.

Terada, R. and Junior, I. C. (2003). *A Stronger Version of RC6 Against Differential Cryptanalysis*. *Symposium on Cryptography and Information Security*, pages 11D04–11D09.

Terada, R., Pinheiro, P. G., and Koyama, K. (1996). *A New Version of FEAL, Stronger Against Differential Cryptanalysis*. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E79-A(1).

Vaudenay, S. (1996). *An Experiment on DES Statistical Cryptanalysis*. *ACM Conference on Computer and Communications Security*, pages 139–147.