Differential Cryptanalysis of REDOC III

Ken Shirriff

Address: Sun Microsystems Labs, 2550 Garcia Ave., MS UMTV29-112, Mountain View, CA 94043. Ken.Shirriff@eng.sun.com

Abstract: REDOC III is a recently-developed block encryption algorithm. This paper describes an attack using differential cryptanalysis that can determine most of the key with about 2^{20} chosen plaintexts and 2^{30} bytes of storage.

Keywords: block code, differential cryptanalysis, REDOC

Introduction

This paper describes an attack on the REDOC III code [3] that uses differential cryptanalysis [1]. This attack can determine most of the key with about 2^{20} chosen plaintexts and 2^{30} bytes of storage.

The REDOC III algorithm is a block cipher that was designed to be efficient when implemented in software. The REDOC III encrypts an input block of ten bytes. It uses a key table of 256 10-byte internal keys, numbered 0 through 255; this key table can be generated from a smaller external key, or can be considered a 2560 byte external key. The REDOC III algorithm is [3]:

(1) Create two 10-byte mask blocks: M_0 is the exclusive-or of the first 128 10-byte keys and M_1 is the exclusive-or of the second 128 10-byte keys.

(2) To encrypt a 10-byte block:

(a) Exclusive-or the first byte (byte 0) of the data block with the first byte of M_0 to obtain an index between 0 and 255 into the key table. Exclusive-or each byte in the data block with the corresponding byte in the key selected from the table, except for the first byte.

(b) Exclusive-or the second byte (byte 1) of the data block with the second byte of M_0. Select the key from the key table corresponding to the value obtained from the exclusive-or. Exclusive-or each byte in the data block with the corresponding byte in the chosen key, except for byte 1.

(c) Continue with the entire block (bytes 2 through 9), until each byte has been used to select a key from the key table after exclusive-oring it with the corresponding M_0 value, and then exclusive-or each byte with the key except for the byte use to select the key.

(d) Repeat steps (a) through (c) with M_1.

Figure 1 illustrates the algorithm and Figure 2 summarizes the notation used in this paper.

Initial Data:	$\mathbf{A_{0,0}}$	$\mathbf{A_{0,1}}$	$\mathbf{A_{0,2}}$	$\mathbf{A_{0,3}}$	$\mathbf{A_{0,4}}$	$\mathbf{A_{0,5}}$	$\mathbf{A_{0,6}}$	$\mathbf{A_{0,7}}$	$\mathbf{A_{0,8}}$	$\mathbf{A_{0,9}}$
$a=A_{0,0} \oplus M_{0,0}$ $\oplus$	0	$K_{a,1}$	$K_{a,2}$	$K_{a,3}$	$K_{a,4}$	$K_{a,5}$	$K_{a,6}$	$K_{a,7}$	$K_{a,8}$	$K_{a,9}$
Step 1 result =	$\mathbf{A_{1,0}}$	$\mathbf{A_{1,1}}$	$\mathbf{A_{1,2}}$	$\mathbf{A_{1,3}}$	$\mathbf{A_{1,4}}$	$\mathbf{A_{1,5}}$	$\mathbf{A_{1,6}}$	$\mathbf{A_{1,7}}$	$\mathbf{A_{1,8}}$	$\mathbf{A_{1,9}}$
$b=A_{1,1} \oplus M_{0,1}$ $\oplus$	$K_{b,0}$	0	$K_{b,2}$	$K_{b,3}$	$K_{b,4}$	$K_{b,5}$	$K_{b,6}$	$K_{b,7}$	$K_{b,8}$	$K_{b,9}$
Step 2 result =	$\mathbf{A_{2,0}}$	$\mathbf{A_{2,1}}$	$\mathbf{A_{2,2}}$	$\mathbf{A_{2,3}}$	$\mathbf{A_{2,4}}$	$\mathbf{A_{2,5}}$	$\mathbf{A_{2,6}}$	$\mathbf{A_{2,7}}$	$\mathbf{A_{2,8}}$	$\mathbf{A_{2,9}}$
...					...					
$j=A_{9,9} \oplus M_{0,9}$ $\oplus$	$K_{j,0}$	$K_{j,1}$	$K_{j,2}$	$K_{j,3}$	$K_{j,4}$	$K_{j,5}$	$K_{j,6}$	$K_{j,7}$	$K_{j,8}$	0
Step 10 result =	$\mathbf{A_{10,0}}$	$\mathbf{A_{10,1}}$	$\mathbf{A_{10,2}}$	$\mathbf{A_{10,3}}$	$\mathbf{A_{10,4}}$	$\mathbf{A_{10,5}}$	$\mathbf{A_{10,6}}$	$\mathbf{A_{10,7}}$	$\mathbf{A_{10,8}}$	$\mathbf{A_{10,9}}$
$k=A_{10,0} \oplus M_{1,0}$ $\oplus$	0	$K_{k,1}$	$K_{k,2}$	$K_{k,3}$	$K_{k,4}$	$K_{k,5}$	$K_{k,6}$	$K_{k,7}$	$K_{k,8}$	$K_{k,9}$
Step 11 result =	$\mathbf{A_{10,0}}$	$\mathbf{A_{10,1}}$	$\mathbf{A_{10,2}}$	$\mathbf{A_{10,3}}$	$\mathbf{A_{10,4}}$	$\mathbf{A_{10,5}}$	$\mathbf{A_{10,6}}$	$\mathbf{A_{10,7}}$	$\mathbf{A_{10,8}}$	$\mathbf{A_{10,9}}$
...					...					
$t=A_{20,9} \oplus M_{1,9}$ $\oplus$	$K_{t,0}$	$K_{t,1}$	$K_{t,2}$	$K_{t,3}$	$K_{t,4}$	$K_{t,5}$	$K_{t,6}$	$K_{t,7}$	$K_{t,8}$	0
Step 20 result =	$\mathbf{A_{20,0}}$	$\mathbf{A_{20,1}}$	$\mathbf{A_{20,2}}$	$\mathbf{A_{20,3}}$	$\mathbf{A_{20,4}}$	$\mathbf{A_{20,5}}$	$\mathbf{A_{20,6}}$	$\mathbf{A_{20,7}}$	$\mathbf{A_{20,8}}$	$\mathbf{A_{20,9}}$

Figure 1. The REDOC III algorithm. This figure shows the twenty steps in encrypting a block of ten bytes. In each step, the mask byte M is exclusive-ored with the appropriate data byte A. The resulting value is used to select the key K that is exclusive-ored with the remaining bytes A. The two rounds of the algorithm consist of steps 1 to 10 and then steps 11 to 20.

$\mathbf{A_y}$	Byte y of the data block A.
$\mathbf{A_{s,y}}$	Byte y of the data block A going into step s. (s=0 for initial data, s=20 for final encrypted data.)
K_w	Key w of the key table ($0 \le w \le 255$). The twenty keys used in successive rounds are denoted a, b, ..., t. X and Y denote keys satisfying desired equalities.
$K_{w,y}$	Byte y of key w ($0 \le y < 10$)
M_r	Mask r (r=0 or 1)
$M_{r,y}$	Byte y of mask r
Ω	A “characteristic”, i.e. the exclusive-or of two corresponding data blocks.

Figure 2. Notation in this paper. This notation is based on [2].

The attack

The fundamental concept behind the attack is that in some circumstances, two blocks will be encrypted with the same keys in the first 19 steps, except that the two keys used in steps 10 and 11 will be used in the opposite order. In this case, because exclusive-or commutes, the keys will yield nearly the same encryption, except that the keys used in the final step will be different. By examining the output of two such blocks, information on the last keys used can be recovered.

Figure 3 illustrates the attack in more detail. The attack depends on the existence of two keys numbered X and Y that satisfy $K_{X,0} \oplus K_{Y,0} = X \oplus Y$. Consider two input data blocks A and B with characteristic $\Omega=(0,...,0,X \oplus Y)$, that is, they are the same except for the last bytes, for which $A_9 \oplus B_9 = X \oplus Y$. Both data blocks will be processed identically up until the last step of round 1 (Step 10). At this point, different keys will be selected because the last bytes of A and B differ. Suppose that the encryption of A happens to use K_X in Step 10 and K_Y in Step 11. Then, Step 10 of encrypting B will use K_Y since $A_9 \oplus B_9 = X \oplus Y$. $B_{10,0}$ will equal $A_{10,0} \oplus X \oplus Y$ since $K_{X,0} \oplus K_{Y,0} = X \oplus Y$, so Step 11 of encrypting B will use K_X. The result is that A is exclusive-ored with K_X and then K_Y, while B is exclusive-ored with K_Y and then K_X. Because exclusive-or commutes, both data blocks will be the same except for the first and last byte at this point. As a result, the blocks will continue to be processed identically until the final step. Because the last bytes differ, the final exclusive-or will use two different keys (call them $\alpha = A_{20,9} \oplus M_{1,9}$ and $\beta = B_{20,9} \oplus M_{1,9} = \alpha \oplus X \oplus Y \oplus K_{X,9} \oplus K_{Y,9}$), resulting in the final characteristic: $\Omega = (X \oplus Y \oplus K_{\alpha,0} \oplus K_{\beta,0}, K_{\alpha,1} \oplus K_{\beta,1}, ..., K_{\alpha,8} \oplus K_{\beta,8}, X \oplus Y \oplus K_{X,9} \oplus K_{Y,9})$. Two input blocks A and B that are encrypted in this manner will be called a successful pair and the resulting characteristic will be called a successful characteristic.

Successful characteristics can be used to discover most of the key table. A successful characteristic reveals the first 9 bytes of $K_\alpha \oplus K_\beta$. (The last byte cannot be easily recovered since X and Y are not known; only $X \oplus Y$ is known from A and B.) The computed values of $A_{20,9}$ and $B_{20,9}$ reveal α and β exclusive-ored with the unknown $M_{1,9}$. By combining enough values of $K_\alpha \oplus K_\beta$, most of the key table can be recovered, except that all lines will be exclusive-ored with some unknown key, say K_0. Thus, most of the 2560 byte table can be determined except for 266 bytes of uncertainty. That is, the indices of the recovered table will be permuted by $M_{1,9}$, the lines will be exclusive-ored with an unknown key, and the last column will be unknown.

		Block A					Block B					Characteristic Ω
Initial:		$\mathbf{A_{0,0}}$	$\mathbf{A_{0,1}}$	...	$\mathbf{A_{0,8}}$	$\mathbf{A_{0,9}}$	$\mathbf{A_{0,0}}$	$\mathbf{A_{0,1}}$	...	$\mathbf{A_{0,8}}$	$\mathbf{A_{0,9} \oplus X \oplus Y}$	$(0, ..., 0, X\oplus Y)$
Step 1	$\oplus$	0	$K_{a,1}$	...	$K_{a,8}$	$K_{a,9}$	0	$K_{a,1}$	...	$K_{a,8}$	$K_{a,9}$	
	=	$\mathbf{A_{1,0}}$	$\mathbf{A_{1,1}}$	...	$\mathbf{A_{1,8}}$	$\mathbf{A_{1,9}}$	$\mathbf{A_{1,0}}$	$\mathbf{A_{1,1}}$	...	$\mathbf{A_{1,8}}$	$\mathbf{A_{1,9} \oplus X \oplus Y}$	$(0, ..., 0, X\oplus Y)$
...				...					...			
Step 10	$\oplus$	$K_{X,0}$	$K_{X,1}$	...	$K_{X,8}$	0	$K_{Y,0}=$ $K_{X,0}\oplus X\oplus Y$	$K_{Y,1}$	...	$K_{Y,8}$	0	
	=	$\mathbf{A_{10,0}}$ $=Y\oplus M_{1,0}$ *	$\mathbf{A_{10,1}}$	...	$\mathbf{A_{10,8}}$	$\mathbf{A_{10,9}}$ $=X\oplus M_{0,9}$ *	$\mathbf{A_{10,0} \oplus X \oplus Y}$ $=X\oplus M_{1,0}$ *	$\mathbf{B_{10,1}}$	...	$\mathbf{B_{10,8}}$	$\mathbf{A_{10,9} \oplus X \oplus Y}$ $=Y\oplus M_{0,9}$ *	$(0, ..., 0, X\oplus Y)$
Step 11	$\oplus$	0	$K_{Y,1}$	...	$K_{Y,8}$	$K_{Y,9}$	0	$K_{X,1}$	...	$K_{X,8}$	$K_{X,9}$	
	=	$\mathbf{A_{11,0}}$	$\mathbf{A_{11,1}}$	...	$\mathbf{A_{11,8}}$	$\mathbf{A_{11,9}}$	$\mathbf{A_{11,0} \oplus X \oplus Y}$	$\mathbf{A_{11,1}}$	...	$\mathbf{A_{11,8}}$	$\mathbf{B_{11,9}}$	$(X\oplus Y, 0,...,0, X\oplus Y\oplus K_{X,9}\oplus K_{Y,9})$
Step 12	$\oplus$	$K_{l,0}$	0	...	$K_{l,8}$	$K_{l,9}$	$K_{l,0}$		...	$K_{l,8}$	$K_{l,9}$	
	=	$\mathbf{A_{12,0}}$	$\mathbf{A_{12,1}}$	...	$\mathbf{A_{12,8}}$	$\mathbf{A_{12,9}}$	$\mathbf{A_{12,0} \oplus X \oplus Y}$	0$\mathbf{A_{12,1}}$	...	$\mathbf{A_{12,8}}$	$\mathbf{B_{12,9}}$	$(X\oplus Y, 0,...,0, X\oplus Y\oplus K_{X,9}\oplus K_{Y,9})$
...				...					...			
Step 19	$\oplus$	$K_{s,0}$	$K_{s,1}$	...	0	$K_{s,9}$	$K_{s,0}$	$K_{s,1}$	...	0	$K_{l,9}$	
	=	$\mathbf{A_{19,0}}$	$\mathbf{A_{19,1}}$	...	$\mathbf{A_{19,8}}$	$\mathbf{A_{19,9}}$	$\mathbf{A_{19,0} \oplus X \oplus Y}$	$\mathbf{A_{19,1}}$	...	$\mathbf{A_{19,8}}$	$\mathbf{B_{19,9}}$	$(X\oplus Y, 0,...,0, X\oplus Y\oplus K_{X,9}\oplus K_{Y,9})$
Step 20	$\oplus$	$K_{\alpha,0}$	$K_{\alpha,1}$	...	$K_{\alpha,8}$	0	$K_{\beta,0}$	$K_{\beta,1}$	...	$K_{\beta,8}$	0	(see text)
Final	=	$\mathbf{A_{20,0}}$	$\mathbf{A_{20,1}}$	...	$\mathbf{A_{20,8}}$	$\mathbf{A_{20,9}}$	$\mathbf{B_{20,0}}$	$\mathbf{B_{20,1}}$	...	$\mathbf{B_{20,8}}$	$\mathbf{B_{20,9}}$	

Figure 3. The attack. This figure shows the successful encryption of two blocks A and B that differ only in the last byte. This attack depends on the existence of two keys that satisfy $K_{X,0}\oplus K_{Y,0} = X\oplus Y$ for some X and Y. Asterisks indicate fortuitous equalities based on A and B that must occur for the attack to succeed. Due to these equalities, steps 10 and 11 will use K_X and K_Y for block A and for block B, but in the opposite order. Steps 12 through 19 will be identical for the two blocks due to this cancellation. The final step uses two different keys: K_α and K_β. The resulting characteristic reveals $K_\alpha \oplus K_\beta$, allowing part of the key table to be recovered.

Two random blocks A and B have about 2^{-15} probability of being successful. To see this, note that there is 1 chance in 256 that the encryption of A will use X in step 10 and 1 chance in 256 that A will use Y in step 11. There is 1 chance in 256 that $A_{9,0}\oplus B_{9,0} = X\oplus Y$. Thus, the odds are 2^{-24} that A and B will work successfully with a particular X,Y combination. An average key table contains about 2^8 successful X,Y combinations. Besides the use of K_X followed by K_Y by A and K_Y followed by K_X by B as described above, successful cancellation will also occur if A uses K_X twice and B uses K_Y twice. Therefore, the chance is $2^{-24}x2^8x2 = 2^{-15}$ that some particular A and B will be successful with any X and Y.

One way to implement this attack is to process groups of 256 blocks that differ only in the last byte. The 256 encrypted blocks can be combined to generate the characteristics for 256x255/2$\approx 2^{15}$ pairs. Thus, there are about 2^7 pairs generated per encryption, and $2^7x2^{-15}=2^{-8}$ successful pairs on the average. Note that successful pairs are not immediately obvious. Successful pairs can be determined, however, by finding two characteristics that match in the middle eight bytes; this will almost never happen by chance. Matching characteristics can be determined by entering all computed characteristics into a hash table. A collision indicates (almost certainly) that the characteristic corresponds to a successful attack.

The attack is relatively efficient. To solve for the key table requires key pairs that overlap and use every line of the key table. Based on random allocation theory [4, p. 12], an average of 1568 randomly distributed entries would be required to hit all 256 lines, that is, about 784 successful pairs. Empirical measurements showed that since the distribution isn't totally random, an average of 980 key pairs had to be determined to complete the attack, which required about 4800 ($\approx 2^{12}$) successful entries to be placed in the hash table. Since only about 1 in 2^{15} characteristics will correspond to a successful attack, about 2^{12}x2^{15}=2^{27} characteristics must be entered into the hash table, taking about 2^{30} bytes. Generating 2^{12} successful pairs requires about 2^{8}x2^{12}=2^{20} encryptions in total.

The attack can be modified to use less memory but more encryptions. One method would be to fix a value of X⊕Y and only enter those pairs into the hash table, rather than all pairs. Then hash hits are more likely since the space of possible entries is much smaller. In addition, the hash table could be flushed after each hit to prevent it from growing excessively. With these techniques the attack will take about 2^{22} bytes of memory but 2^{30} encryptions.

There are several ways REDOC III could be modified to avoid this attack. One way would be to shift the keys that are used in successive steps of the encryption; this would avoid the cancellation that allows the differential cryptanalytic attack to succeed. If the entries were byte-shifted, the effect on performance would be minimal. Performing more than two encryption rounds would make the attack unfeasibly expensive.

Some key tables are resistant to this attack. An entirely resistant key table would have no m and n satisfying $K_{m,0} \oplus K_{n,0} = m \oplus n$. That is, for every i, $K_{i,0} \oplus i$ will be unique, so the values of $K_{i,0} \oplus i$ must form a permutation of 0 through 255. Thus 256! out of every 256^{256} key tables will not be subject to attack. A restriction to only strong key tables would prevent the differential cryptanalytic attack. This would reduce the key table space considerably, but would still leave a very large key space.

Conclusions

REDOC III is subject to a differential cryptanalytic attack because it combines data with the keys with a commutative operation, it doesn't perform any shuffling of bits, and it only contains two passes. As a result, a relatively small number of chosen plaintexts allows most of the key table can be recovered.

References

[1] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. In *Advances in cryptology, proceedings of CRYPTO '90*, pages 2–21, 1990.

[2] T. W. Cusick and M. C. Wood. The REDOC II cryptosystem. In *Advances in cryptology, proceedings of CRYPTO '90*, pages 545–563, 1990.

[3] B. Schneier. *Applied Cryptography*. Wiley, New York, 1994.

[4] V. Kolchin, B. Sevast'yanov, and V. Chistyakov, *Random Allocations*, V. H. Winston & Sons, Washington, 1978.

Acknowledgments

Most of this research was performed at U. C. Berkeley.

Biographical Sketch

Ken Shirriff is a staff engineer at Sun Labs. He received his Ph.D. in computer science from U.C. Berkeley in 1995.