REPORTS IN INFORMATICS

ISSN 0333-3590

Improved Collision Attack on OCB

John Erik Mathiassen

REPORT NO 306 **August 2005**

Department of Informatics
UNIVERSITY OF BERGEN
Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2005-306.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Improved Collision Attack on OCB

John Erik Mathiassen *

11th October 2005

Abstract

In this paper we present an improvement of the collision attack [1] on the authenticated encryption mode of operation OCB. [1] presents a detection of collision method and a way to use the collision, and it is possible to use the information from a collision to change some blocks of the message unnoticed, if they have a special property. We found a way to use the information from a collision to change any future message in any position, without knowing anything about the plaintexts and the nonces. Once a collision is detected the probability of success of cheating is 1, and this part of the attack can be done by hand calculations.

It must also be mentioned that our attack depends on the complexity of the detection of a collision, and this does not violate the security bounds given in [2, 3]. However once a collision is found the attack is simple and devastating.

1 Introduction

Modes of operation have been of interest since the invention of block ciphers. The first modes of operation only focused on the confidentiality of the message being processed. Later on some modes were also used for authentication like the CBC-MAC. If both authentication and confidentiality were required, the encryption mode and the authentication mode are typically run separately and with independent keys.

Lately it has been focused on making modes of operations to handle both encryption and authentication using only one symmetric key. The OCB mode [2, 3] is a mode of operation using an underlying block cipher, and it supports both confidentiality and integrity by only one encryption per message block with only 3 block encryptions overhead. It was submitted to NIST's request for a new mode of operation, but CCM was the only mode recommended by NIST. A reason for OCB not to be considered in this competition is that it is patented, and a recommendation by NIST would be a free marketing campaign.

A collision attack on OCB was first presented in [1], and showed that a collision could be a serious threat to the authentication. Here we present an attack which is much more threatening as soon as a collision occurs and is detected. Both these attack use the same detection method, and the complexity of the detection of a collision does not violate the security bounds given by the authors of [2, 3], but as soon as a collision occurs the authentication is useless unless the key is changed.

*The Selmer Center, Department of Informatics, University of Bergen

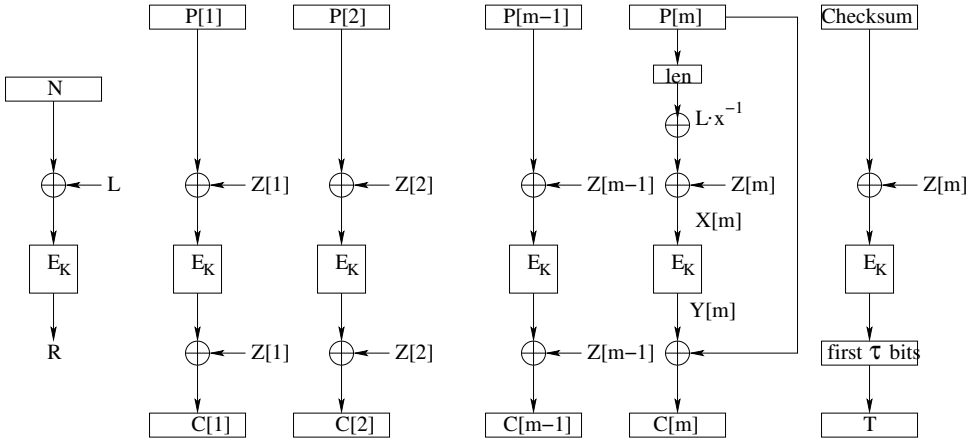

Figure 1: This figure illustrates the OCB Encryption.

2 OCB - Offset CodeBook mode

The OCB (Offset CodeBook), Figure 1, encrypts m plaintext blocks P_i to m ciphertext blocks C_i, and a message authentication tag T. The mode uses a pure block cipher as an underlying cipher, and the block size n is equal to the block size of the underlying cipher, except for the last ciphertext block which can be of any length $1 < len \leq n$ bits. It is also encrypted in a different way than the other blocks. The authentication tag T, if present, is of length $1 \leq \tau \leq n$, and should be of reasonable length to avoid tag guessing attacks. For each message[1] a nonce N is used, and it is never reused within a session, which will be the period where the key K remains the same. It is also required that the message length is not greater than $2^{n/2}$, because the nonce has to be changed before a collision is very likely to occur.

The only difference from ECB mode encryption is that a hopefully unpredictable offset Z_i is added both before and after encryption. The Z_is is different in each round because of the way it is defined. Two parameters decide the Z_is: A "random" session string L and a "random" string R per message, which is dependent on L and the nonce N. Denote the encryption of the plaintext P with the underlying cipher as $E_K(P)$ which gives the ciphertext C. The L is defined as

$$L = E_K(0^n)$$

and will not change within a session. R is defined as

$$R = E_K(N \oplus L)$$

and is therefore changed in a new message, because the nonce N changes in for each message. The offsets are then defined as

$$Z_i = R \oplus (L \cdot \gamma_i)$$

where γ_i is the ith element of the Gray code[2], and $\gamma_1 = 1$. That gives us $Z_1 = R \oplus L$, and the ith one is easy and efficient to calculate from the previous one by $Z_i = Z_{i-1} \cdot x^{ntz(i)}$ using polynomial representation or $Z_i = Z_{i-1} \cdot (0^{n-1}1 << ntz(i))$, where $ntz(i)$ is the number of trailing zeroes in the binary representation of i.

[1] A message may consist of several plaintext blocks, and the encrypted message is appended the authentication tag τ.

[2] The Gray code is and ordering of the 2^n n-bit numbers where all the adjacent elements have hamming weight 1.

Next the encryption for the $m-1$ first plaintext blocks is

$$C_i = E_K(P_i \oplus Z_i) \oplus Z_i$$

which might be written as

$$Y_i = E_K(X_i)$$

where $X_i = P_i \oplus Z_i$ and $Y_i = C_i \oplus Z_i$. The encryption is simply

$$D_K(C_i \oplus Z_i) \oplus Z_i = D_K(Y_i) \oplus Z_i = X_i \oplus Z_i = P_i$$

The last block of length len bits is encrypted to a ciphertext of len bits

$$C_m = MSB_{len}(E_K(len \oplus (L \cdot x^{-1}) \oplus Z_m)) \oplus P_m$$

where len is represented as a n-bit string, and where the function $MSB_b(S)$ returns the b most significant bits of S assuming a big endian system. To see why this works one might look at the decryption

$$MSB_{len}(E_K(len \oplus (L \cdot x^{-1}) \oplus Z_m)) \oplus C_m =$$

$$[MSB_{len}(E_K(len \oplus (L \cdot x^{-1}) \oplus Z_m))] \oplus [MSB_{len}(E_K(len \oplus (L \cdot x^{-1}) \oplus Z_m))] \oplus P_m =$$

$$P_m$$

and the decryption gives P_m of length len as expected.

Next a $Checksum$ dependent on all the messages is encrypted as follows

$$T = MSB_\tau(E_K(Checksum \oplus Z_m))$$

where $Checksum = P_1 \oplus P_2 \oplus \cdots \oplus P_{m-1} \oplus C_m \oplus Y_m$. Since the length of C_m could be less than n we should strictly have written $C_m|0^{n-len}$ and $P_m|0^{n-len}$ in the above expressions, but we will continue to use the shorter notations for simplicity. Notice that $C_m \oplus Y_m = P_m \oplus LSB_{n-len}(Y_m)$, and $LSB_{n-len}(Y_m) = LSB_{n-len}(E_K(len \oplus (L \cdot x^{-1}) \oplus Z_m)$ is added the $Checksum$ to avoid somebody manipulating the content of the last block by changing the length. This is avoided because Y_m depends on len, the length of P_m.

To check that the message is not changed on the way to the sender, the $Checksum'$ of the received messages must be evaluated

$$T' = MSB_\tau(E_K(Checksum \oplus Z_m))$$

and if T' is equal to the received T the message is assumed to be authentic.

The security of the OCB relies on the difficulty of changing the ciphertexts in a way that at the same time changes the plaintexts so that the $Checksum$ remains the same. Without knowing anything about the key K it is hard to predict a change in the plaintexts by a certain change in the ciphertext, even if the plaintext is known.

Another straight forward approach is to guess the authentication tag T, and it has probability $2^{-\tau}$. This is why it is recommended to choose τ big enough to achieve the level of security wanted in the system. This and the chance of collisions is also a reason why the block size should be at least 128 bits, and it is suitable for use together with AES, which has block size 128. We will assume in the rest that the OCB is used with a block cipher with 128 bits block size.

3 Collision attack on OCB

This chapter describes a collision attack on OCB, and a new way to exploit a collision. First we will start by describing the detection phase, which was also shown in [1]. The second phase is how to use the data found in the detection phase. Ferguson has an attack where it is possible to change four[3] blocks of the message used in the detection phase, but we will show that it is possible to change any block of future messages encrypted with the same key K, without knowing anything about the plaintext.

3.1 Collision Detection and Extraction of L

Suppose we have a known message $P_1, P_2, \ldots, P_m$ and its encryption $C_1, C_2, \ldots, C_m, T$. What we look for is two plaintext blocks P_i and P_j where the two inputs to the encryption X_i, X_j are equal and thereby the outputs $Y_i = E_K(X_i)$ and $Y_j = E_K(X_j)$ are equal. Since we know that $P_i = X_i \oplus Z_i$ and $C_i = Y_i \oplus Z_i$ the obvious way to cancel the offset Z_i is to calculate and store $W_i = P_i \oplus C_i = X_i \oplus Y_i$ for all messages P_i. Next observe:

$$X_i = X_j \;\Leftrightarrow\; Y_i = Y_j$$

$$\Downarrow$$

$$X_i \oplus Y_i = X_j \oplus Y_j$$

$$\Updownarrow$$

$$P_i \oplus C_i = P_j \oplus C_j$$

$$\Updownarrow$$

$$W_i = W_j$$

That means that if we have $X_i = X_j$ we also have $W_i = W_j$, and

$$P_i \oplus P_j = X_i \oplus X_j \oplus Z_i \oplus Z_j = L \cdot (\gamma_i \oplus \gamma_j).$$

That means that we easily can calculate L by $L = (P_i \oplus P_j) \cdot (\gamma_i \oplus \gamma_j)^{-1}$.

[3] It might be more changes, but the main difference is that only one message might be changed, and that the blocks to be changed need to fulfill a special criteria to avoid to be detected by the authentication.

3.2 Exploit the knowledge of L

Knowing L we are able to cheat by simply changing the position of any two blocks except the last block, since it is encrypted in a different way. Say we want to change the plaintext at two arbitrary positions i and j. Remember that $C_i = Y_i \oplus Z_i = Y_i \oplus R \oplus (\gamma_i \cdot L)$ and define $\Delta_{i,j}L = (\gamma_i \oplus \gamma_j) \cdot L$, which is possible to calculate for any position i and j only by a prior knowledge of L. Notice that $Z_i \oplus Z_j = \Delta_{i,j}L$, and to change the offset from Z_i to Z_j we must add $\Delta_{i,j}L$. Then we form the new $C_i^{'}$ and $C_j^{'}$ as follows:

$$C_i^{'} = C_j \oplus (\Delta_{i,j}L) = Y_j \oplus Z_j^{'} \oplus (\Delta_{i,j}L) = Y_j \oplus Z_i^{'}$$

$$\Downarrow$$

$$Y_i^{'} = Y_j = C_i^{'} \oplus Z_i^{'}$$

and

$$C_j^{'} = C_i \oplus (\Delta_{i,j}L) = Y_i \oplus Z_i^{'} \oplus (\Delta_{i,j}L) = Y_i \oplus Z_j^{'}$$

$$\Downarrow$$

$$Y_j^{'} = Y_i = C_j^{'} \oplus Z_j^{'}$$

and the decryption of Y_j and Y_i will give us the new preimages $X_i^{'} = X_j$ and $X_j^{'} = X_i$. The exclusive or of the new messages gives us

$$P_i^{'} \oplus P_j^{'} = (X_j \oplus Z_i^{'}) \oplus (X_i \oplus Z_j^{'}) =$$

$$(X_j \oplus X_i) \oplus (Z_i^{'} \oplus Z_j^{'}) = (X_j \oplus X_i) \oplus \Delta_{i,j}L =$$
$$(X_i \oplus Z_i) \oplus (X_j \oplus Z_j) = P_i \oplus P_j$$

which proves that the $Checksum = P_1 \oplus P_2 \oplus \cdots \oplus P_{m-1} \oplus C_m 0^* \oplus Y_m$ and therefore also the tag T remains unchanged. It is therefore possible to swap any two ciphertexts this way (except the last block), and if changed the way we showed the message is still authentic.

We might add that the two new message blocks will be:

$$P_i^{'} = P_j \oplus Z_j \oplus Z_i'$$

$$P_j^{'} = P_i \oplus Z_i \oplus Z_j'$$

Notice that "swapping" two ciphertext blocks also change the swapped plaintexts blocks. However the attack enables us to change future messages - encrypted with the same key K - at any position without knowing the plaintext or the nonce, and the probability of being detected is 0.

Actually any even number of ciphertexts in a message might be permuted in any order, and by adding to them the appropriate[4] constants $\Delta_{i,j}L$. The resulting plaintexts will be $P_i^{'} = P_j \oplus Z_j \oplus Z^{'}{}_i$, where $C_i^{'}$ is the ciphertext taken from the j'th position in the sent message. The only difference in the plaintext blocks are the positions where the ciphertexts have changed. Even if the plaintexts in those positions are changed the resulting *Checksum* is not changed. Assume that 4 positions i, j, k, l have changed and the new checksum becomes

$$Checksum' = Checksum \oplus Z_i \oplus Z_j \oplus Z_k \oplus Z_l \oplus Z_i^{'} \oplus Z_j^{'} \oplus Z_k^{'} \oplus Z_l^{'} =$$

$$Checksum \oplus L \cdot (\gamma_i \oplus \gamma_j \oplus \gamma_k \oplus \gamma_l \oplus \gamma_i \oplus \gamma_j \oplus \gamma_k \oplus \gamma_l) = Checksum$$

and is equal to the old one. So the tag τ is still a valid tag.

3.3 Probability of Success

The probability that $X_i = X_j$ and thereby $W_i = W_j$ is 2^{-128}, but having m plaintexts and ciphertexts the probability of at least collision is about $m^2 2^{-129}$ by the birthday paradox.

Notice that $W_i = W_j$ is true also with $\Delta_{i,j}X = X_i \oplus X_j = Y_i \oplus Y_j = \Delta_{i,j}Y \neq 0$ because

$$\Delta_{i,j}X = \Delta_{i,j}Y$$

$$\Updownarrow$$

$$X_i \oplus Y_i = X_j \oplus Y_j$$

$$\Updownarrow$$

$$W_i = P_i \oplus C_i = P_j \oplus C_j = W_j.$$

But the event that $\Delta_{i,j}X = \Delta_{i,j}Y$ also has probability 2^{-128}. That means that both $X_i = X_j$ and $\Delta_{i,j}X = \Delta_{i,j}Y$ gives us $W_i = W_j$, since the probabilities are equal the probability that $W_i = W_j$ actually is a collision is therefore 0.5.

One might add that the collision attack is costly, and does not violate the security bounds of OCB. But if a collision event occurs the attack threatens the authenticity of future messages.

4 Conclusion and Future Work

In this paper we describe a collision attack on OCB, where the detection of collision is exactly the same as in [1]. We show that once a collision occurs it is possible to

[4] $\Delta_{i,j}L$ is added to the ciphertext taken from the i'th position to the j'th position.

swap and thereby change any blocks of future messages using the same key, even if the nonce is changed. The probability of success of this attack is 1 once a collision is found. The detection of a collision does not have better probability and complexity than the proved security bounds, but it will be devastating once a collision is found.

Once L is found it is easy to change any message of length 3 blocks or more, without being caught by the authentication mechanism. Therefore it would be very interesting to find a way to force collisions to occur more often than at random. A better way to find L would prove the security bounds of OCB is wrong.

References

[1] N. Ferguson. Collision attacks on OCB. Comments to NIST, February 2002, Available at NIST's webpage at.

[2] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. Available from.

[3] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. *Eight ACM Conference on Computer and Communications Security (CCS-8)*, pages 195–205.