MCMAMBO V1: A NEW KIND OF LATIN DANCE

WATSON LADD

Submitter and Designer: Watson Ladd
Contact Email: watsonbladd@gmail.com
Date: March 4, 2014

1. Specification

We begin by specifying a cipher Mambo's operation on 32-bit words. Mambo operates on a 16-word state, organized in a square that is 4 by 4. Let k_i denote the 8 word key, and t_i the 16 word tweak. Indices will run from 0 to the number of words minus one. The tweak and the key will be treated as parameters for the functions we will define.

We define $\oplus$ to be the XOR function on words,$\vee$ the bitwise OR function, $\wedge$ to be bitwise AND function, $\uparrow$ the bitwise NAND function, and lastly $\downarrow$ the bitwise NOR function. These symbols come from symbolic logic.

We also define $R(x, c)$ where c is a constant and x a 32-bit word to be the left bit rotation of x by c bits, that is the unique word congruent to $2^c u$ modulo $2^{32} - 1$.

Let T be the operation that fed a 16 word state $(m_0, \ldots m_{15})$ returns $(m_0 \oplus t_0, \ldots m_{15} \oplus t_{15})$, that is xors the tweak with the state. Note T is a bijection.

Define $Q(x_0, x_1, x_2, x_3)$ to be (y_0, y_1, y_2, y_3) where $y_1 = x_1 \oplus R(x_0 \wedge x_2, 7)$, $y_2 = x_2 \oplus R(x_0 \vee x_3, 9)$, $y_3 = x_3 \oplus R(y_1 \uparrow x_0, 13)$, $y_0 = x_0 \oplus R(y_1 \downarrow y_2, 18)$. Note that Q is a bijection.

We now define D to conceptually apply Q to each row of the state and then to each column. The state is numbered across and then down, left-to-right.

More formally $D(x_0, \ldots x_15) = (z_0, \ldots z_{15})$, where $(y_0, y_1, y_2, y_3) = Q(x_0, x_1, x_2, x_3)$ and so on, and $(z_0, z_4, z_8, z_{12}) = Q(y_0, y_4, y_8, y_{12})$ and so on.

We now define $K_j(x_0, \ldots x_15)$ as $(z_0 \ldots z_n)$, where $z_{2i+1} = x_{2i+1} \oplus k_i$ if i is congruent to 0 or 1 modulo 4, and $z_{2i} = x_{2i} \oplus k_i$ otherwise. For i divisible by 5, $z_i = x_i \oplus j$. Otherwise $z_i = x_i$, which happens precisely when i is in $\{2, 7, 8, 13\}$. This is adding the key in a checkerboard pattern missing the main diagonal, while adding the round counter in on the diagonal.

The cipher C is then $K_0 D K_{11} D K_{10} \ldots K_6 T D K_5 D K_4 \ldots K_0$. It is obviously an invertible, tweakable block cipher. What remains is to see how attackable it is.

Mambo is C with the additional specification that words should be packed and unpacked to byte arrays in little-endian order, in the order of the indices. (That is, k_0 goes first, then k_1 and so on).

We now specify the authenticated encryption mode McMambo. Let k be fixed as the key and $E_T(M)$ the encryption of the block M with tweak T. Let N be the message number, which we treat soly as a block of 64-bytes. Let $M_1, \ldots M_n$ be the message. The last block has a single 1 byte followed by 0 or more 0 bytes: this is padding which will be removed on recipt and is inserted by the McMambo function on sending. Let $L_1, \ldots L_H$ be the associated data, padded the same way.

Then let U_{-H} be 64 all zero bytes. Define $U_{i+1} = E_{U_i}(L_{H+i+1}) \oplus L_{H+i+1}$ up to U_0, and let $\tau = E_{U_0}(N)$, $U_1 = \tau \oplus N$, and $C_i = E_{U_i}(M_i)$, $U_{i+1} = C_i \oplus M_i$ for i in 1 to n. Then let $C_{n+1} = E_{U_{n+1}}(\tau)$.

The encrypted message is $C_0, \ldots C_{n+1}$.

On reciept of the message the recipient computes U_0 and τ the same way and lets $M_i = D_{U_i}(C_i), U_{i+1} = C_i \oplus M_i$ for i in 1 to n. Finally they compute $\rho = D_{U_i}(C_{n+1})$, and accept the message if $\rho = \tau$.

2. Security Goals

The Mambo block cipher with a fixed key cannot be distinguished from an random injection from tweaks to permutations on 512 bit strings by a circuit whose number of gates times the number of gate delays is substantially less then 2^{256} with

probability greater then 1/2. There is no proof of this: it is a conjecture analogous to $sPRP$ security of the AES.

The McMambo mode I claim is CCA secure against nonce respecting adversaries who are limited to 2^{134} bytes of transmitted message and whose area size product is less then 2^{256}, with an attack success probability of 2^{-114}.

I claim the McMambo mode is secure in the $ORPR$ setting of [2] with the same claim on complexity. This claim is pessimistic: I used the easier, weaker claim, from the paper to calculate this. For concreteness, reuse of nonces only leaks a common prefix of the message, the best that any online scheme can attain.

For all desired properties I claim attack cost of 256 with success probability 2^{-114}. Since this is below the cost of brute-forcing the key the mode does not lose any security.

3. Security analysis

3.1. **The reduction strength.** From the Theorem 2 of the paper [2] there is a quadratic reduction in security against adversaries who repeat nonces. However, the large blocksize of Mambo significantly reduces the impact of this reduction. In particular, after sending a total of 2^{134} bytes through McMambo under the influence of an attacker who can repeat blocks, the advantage of an attacker over breaking the Mambo block cipher is only 2^{-114}. In other words, the attacker must have an advantage p in breaking Mambo to break McMambo with an advantage $p + 10^{-38}$.

The security claim of the Mambo cipher leads to an immediate claim for the McMambo mode. By the arguments of the [2] paper, this gives a 114 bit success level to an attacker bounded to size 2^{256} and allowed to send 2^{134} bytes through McMambo, with nonces chosen as they desire. Note that we reveal all common prefixes of the concatantion of message number and message: this is an unavoidable choice of the online structure.

For nonces that are actually nonces, we do not reveal common prefixes, and have the same security level for the standard notion of security.

3.2. **Strength of a cipher.** We begin by investigating the D operation. A change in x_3 propagates to x_2 and x_0, but not to x_1 in the round on the rows. The column round propagates this change comprehensively, but not to the middle column, as nothing has changed there.

Therefore the encryption of the second column does not depend on two of the words of the key. The same is true for each other column, as can be seen by tracing similar properties. Therefore, K_0DK_0 is insecure as a cipher.

It takes two rounds of D before the state is thoroughly mixed. Bitlevel mixing is not yet attained, however, we can conservatively estimate it by examining the bitlevel mixing in a single row or column. This can be determined from Salsa, as the rotation coefficients are the same. I estimate the number of rounds for bit mixing at 5 iterations of D.

Algebraically D is a very complex operation. On bits D is an order 4 polynomial, and after the full 12 iterations the degree should be 2^{24}. This prevents the attacks in [3] and the Cube attacks of [1].

Slide attacks are prevented by the inclusion of the round number in each K operation, which makes each round unique.

Meet-in-the-middle attacks are frustrated by the use of the entire key at each round.

$\wedge$ is a bilinear operation over $GF(2)$. This enables another attack on one round of D based on using the linearity to solve for one word of the key. $\uparrow$ is a bilinear operation plus a constant.

$\vee$ is a much more complicated operation from a $GF(2)$ perspective. However, De Morgan's laws imply that $\oplus$ and any one of the four operations used sufficies to express any cipher using all of them, and so I decided that the cryptanalyst ought to be forced to deal with them all.

Rotational symmetry in the words is broken only by the addition of the round number. This suffices to break all remaining symmetries on flipped bits by virtue of the diffusion properties. Reduced round variants have symmetries, but these are increasingly hard to find and small groups as the number of rounds grows.

Cryptanalysists can remove invocations of D and K_i from either end to get reduced round versions. They can chop out T, but that's easy enough by fixing a tweak of 0.

If two tweaks produce the same permutation, this is due to a linear relation in the first half of the cipher which significantly weakens the cipher. Additional cryptanalysis by third-party cryptographers is desired.

4. Features

4.1. **Nonce reuse resistance.** Implementations on platforms without the ability to reliably count will leak a very limited amount of information about the messages they process.

4.2. **Software Performance.** On a MacBookPro with an Intel Core i7-3720QM processor the system clang complies reference code to a binary measured to encryption at 16 cycles/byte for long messages, without use of SSE. GCC is not able to perform as well, or the MacPorts clang for reasons unknown. For reference OpenSSL performs AES encryption at 25 cycles/byte in software (without AES instructions) on this very same machine, as determined by the built-in benchmarking features of OpenSSL.

Encrypting one byte takes 3952 cycles on this hardware.

Following Salsa, one goal was a high level of instruction level parallelism. The use of vector processors should accelerate Mambo just as well. However, there is no substitution for measurement and implementation.

The use of simple rotate and boolean instructions means that all architectures with 32-bit words implement instructions required for high performance. On 16-bit and 8-bit architectures the rotations are the sole sticking point.

The high level of instruction level parallelism means that multiple functional units will provide performance gains, even without vectorization. The state fits largely into registers on many architectures, and definitely fits into $L1$ cache. No auxiliary tables are used, eliminating a known source of side channels as well as reducing cache pressure.

4.3. **Hardware Performance.** No hardware has been designed or made, but there are some obvious features that will make the hardware fast. First off all operations are simple bitwise operations and rotations, reducing the minimum clock timing of naive designs significantly. Secondly there are several choices of representation for the circuits computing the Q function, and one can decide how many Q functions to run in parallel. As a result tradeoffs between area and time are easy to implement.

Functions like Salsa and Cubehash with similar structures of repeated simple operations have had very small hardware implementations, and Mambo promises to be no different.

For concreteness, it is easy to imagine a design with a single 3-ported register set where each Q operation requires 8 clock cycles, after pipelining. Computing D thus requires 128 clock cycles, and K_i would require 12. T would require 16. The total number of clock cycles is 2816 to calculate the entire cipher, and the number of gates likely to be approximately 4000 by comparison to the Cortex M1, which has similar capabilities.

4.4. **Online.** Online permutations are good for people who can count, as then they are as strong as needed. For people who cannot they have some weakness. However, the online property can be useful when memory pressure is a consideration or when messages are streamed.

4.5. **Software Side-Channel Resistant.** The structure of McMambo eliminates many opportunities for side channel attacks. All instructions will execute in constant time on any sane processor.

5. Design Rationale

On seeing [2] I decided that designing a tweaked cipher would be easier than continuing the search for an authentication system. Salsa is well-known and respected, so I wholesale copied the design, changed the quarterround function, added the tweak and key, and doubled the number of rounds. This let me inherit the good features of Salsa while making a few tweaks.

Hiding a weakness in the quarterround function would be difficult, and that is really one of very few places to hide it. I certainly haven't hidden any weaknesses in the cipher, and I am the sole designer.

6. Intellectual Property

No patents are being sought by me on any element of this cipher. If this changes the submitter/submitters will announce promptly and within one month this change on the *crypto-competitions* mailing list.

7. Consent

The submitter/submitters hereby consent to all decisions of the CAESAR selection committee regarding the selection or non-selection of this submission as a second-round candidate, a third-round candidate, a finalist, a member of the final portfolio, or any other designation provided by the committee. The submitter/submitters understand that the committee will not comment on the algorithms, except that for each selected algorithm the committee will simply cite the previously published analyses that led to the selection of the algorithm. The submitter/submitters understand that the selection of some algorithms is not a negative comment regarding other algorithms, and that an excellent algorithm might fail to be selected simply because not enough analysis was available at the time of the committee decision. The submitter/submitters acknowledge that the committee decisions reflect the collective expert judgments of the committee members and are not subject to appeal. The submitter/submitters understand that if they disagree with published analyses then they are expected to promptly and publicly

respond to those analyses, not to wait for subsequent committee decisions. The submitter/submitters understand that this statement is required as a condition of consideration of this submission by the CAESAR selection committee.

References

[1] Aumasson, et al. "Cube Testers and Key Recovery Attacks On Reduced-Round MD6 and Trivium".
[2] Fleischmann, Ewan, Christian Forler, Stefan Lucks and Jakob Wenzel. "McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes".
[3] Jackobsen, Thomas and Lars R. Knudsen. "Atacks on Block Ciphers of Low Algebraic Degree".

University of California Berkeley, Berkeley, California
E-mail address: watsonbladd@gmail.com