# RESYNCHRONIZATION ATTACK ON STREAM CIPHERS FILTERED BY MAIORANA-MCFARLAND FUNCTIONS

Guanhan Chew, Aileen Zhang and Khoongming Khoo
*DSO National Laboratories, 20 Science Park Drive, Singapore 118230, Singapore*

Keywords: Stream ciphers, Resynchronization attacks, Maiorana-McFarland functions, Cube attack.

Abstract: In this paper, we present an extension to the resynchronization attack on stream ciphers of (Daemen et al., 1993). The most general attack in (Daemen et al., 1993) on a nonlinearly filtered register with linear resync has attack complexity $\lceil \frac{n}{\phi} \rceil \times 2^{\phi}$, where $n$ is the key length and $\phi$ the input size of the filter function. It was further shown specifically that the attack complexity can be reduced in the case when the filter function is a multiplexer. The attack of (Daemen et al., 1993) is most efficient when the input size is small. We shall show that a large input size may not necessarily guard against this attack, even when a function with good cryptographic properties is used. It may decrease the attack complexity, in the example illustrated in this paper. Boolean functions from the Maiorana-McFarland class make good choices for these filter functions due to their good cryptographic properties such as balance, high nonlinearity and high order of resiliency. However, these functions can become linear when certain input bits are fixed. We shall demonstrate this weakness and use it to achieve lower attack complexities for the general resynchronization attack of (Daemen et al., 1993).

## 1 INTRODUCTION

Resynchronization mechanisms are used to prevent the loss of synchronization in stream ciphers deployed in synchronous communication contexts. The scheme allows multiple parties to dynamically join or leave a secure network by encrypting the communication channel using different key streams generated by different initial states of the cipher. The internal state is repeatedly reinitialized using publicy-known initialization vectors (IV). Since resynchronization is performed multiple times over the duration of communication, the mechanism should be fast. By using publicly known information (e.g. time) to generate the IVs, no additional information (apart from the shared secret key) required for cipher operations need to be transmitted. For efficiency, the internal state of the cipher at the start of each resynchronization is typically linearly generated from the IV and secret key. Nonlinear functions are used to map internal state bits to keystream bits.

While the scheme may appear to enhance security by generating keystream bits from multiple initial states, it was shown in (Daemen et al., 1993) that the cipher becomes vulnerable to the paper's proposed attack when the input size of the filter function is sufficiently small and we have enough keystream bits from different resyncs. As a natural countermeasure, a cipher designer can use a nonlinear function with large input size. We shall demonstrate in this paper that this measure may not guard against the resynchronization attack, even when the nonlinear function is known to have good cryptographic properties. Furthermore, it may improve the efficiency of a resynchronization attack. This can happen when the nonlinear function belongs to Maiorana-McFarland class of Boolean functions, which are known have good trade-offs between various desirable cryptographic properties such as correlation immunity and nonlinearity (Carlet, 2002; Canteaut et al., 2000; Seberry et al., 1993; Sarkar and Maitra, 2000). Maiorana-McFarland functions are used in ciphers such as Toyocrypt (as described in (Mihaljevic and Imai, 2002)) and Grain-128 (Hell et al.).

In this paper, we shall first briefly recount the original resynchronization attack of (Daemen et al., 1993). In Section 2.3, we shall present the resynchronization attack based on Maiorana-McFarland functions. In Section 3, we compare our attack with other known resynchronization attacks before concluding this paper.

Chew G., Zhang A. and Khoo K. (2009).
RESYNCHRONIZATION ATTACK ON STREAM CIPHERS FILTERED BY MAIORANA-MCFARLAND FUNCTIONS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 159-162
DOI: 10.5220/0002261301590162

# 2 RESYNCHRONIZATION ATTACK ON FILTER FUNCTION

## 2.1 Attack Setup

In this paper, we assume the attack model and notation of (Daemen et al., 1993). Resynchronization is achieved by initializing the internal state by an affine transformation on the key $K \in \{0,1\}^n$. If we let $s_i^0 \in \{0,1\}^n$ be the initial internal state of the $i$-th resync, then

$$\begin{aligned} s_i^0 &= L(K, IV_i) \\ &= A(K) \oplus B(IV_i) \\ &= A(K) \oplus R_i, \end{aligned}$$

where $L$ denotes a linear mixing of the secret key $K$ and known initialization vector $IV_i$. $A$ and $B$ are known affine transformations. $R_i$, defined to be $B(IV_i)$, is therefore publicly known.

The state gets updated at every clocking time-step via a linear function $F$. If $s_i^t$ denotes the internal state at clock $t$ during the $i$-th resync, we have

$$s_i^{t+1} = F(s_i^t). \tag{1}$$

Keystream bits $z_i^t \in \{0,1\}$ are generated at each clock via a nonlinear filter function $f$, acting on some subset $u_i^t$ of internal state bits $s_i^t$:

$$z_i^t = f(u_i^t), \tag{2}$$

where

$$u_i^t = G(s_i^t), \tag{3}$$

for some known linear transformation $G$ which projects the internal state vector $s_i^t$ onto $u_i^t$.

## 2.2 Basic Resynchronization Attack

If we suppose that $u_i^t \in \{0,1\}^\phi$ and that $u_i^t$ is known, then we can form $\phi$ linear equations involving the key bits of $K$. We can solve for the key $K$ when enough bits from the internal state $s_i^t$ are known. If all linear relations formed are independent, we would need to collect $\lceil \frac{n}{\phi} \rceil$ $u_i^t$ vectors.

To reconstruct $u_i^t$, we make use of the easily verifiable result $u_i^t = u_j^t \oplus G \circ F^t(R_i \oplus R_j)$, for any $j$. Substituting this into $f(u_i^t) = z_i^t$, we get

$$z_j^t = f(u_i^t \oplus G \circ F^t(R_i \oplus R_j)). \tag{4}$$

When multiple keystream bits $z_j^t$ at the same clock $t$ across different resyncs are known, we can solve for $u_i^t$ by performing $2^\phi$ evaluations of $f$, thus forming a set of $\phi$ linear equations in the key bits. Since we need a total of $n$ equations, the total number of $f$ evaluations is about

$$\left\lceil \frac{n}{\phi} \right\rceil \times 2^\phi. \tag{5}$$

For this attack, we need a total of $\lceil \frac{n}{\phi} \rceil \times \phi$ keystream bits. Each group of $\phi$ bits has to come from the same clock.

## 2.3 Extension to Maiorana-McFarland Filter Function

If we suppose $f : \{0,1\}^\phi \to \{0,1\}$ is of the form:

$$\begin{aligned} f(x_1, x_2, \ldots, x_\phi) &= g(x_1, x_2, \ldots, x_r) \\ &\oplus h(x_1, \ldots, x_r) \cdot (x_{r+1}, \ldots, x_\phi), \end{aligned} \tag{6}$$

where $h : \{0,1\}^r \to \{0,1\}^{\phi - r}$ and $g : \{0,1\}^r \to \{0,1\}$, the attack complexity can be improved, in spite of the desirable cryptographic properties functions of this form exhibit (Sarkar and Maitra, 2000). This is mainly brought about by a reduction in search complexity from $2^\phi$ to about $2^r$.

Instead of guessing all $\phi$ bits in $u_i^t$, we guess the $r$ bits in $u_i^t$ that correspond to the inputs to the functions $g$ and $h$. Each such guess linearizes (6) and produces one linear equation in terms of the $(\phi - r)$ unknown bits in $u_i^t$ that are input to $(x_{r+1}, \ldots, x_\phi)$. We collect $(\phi - r)$ resyncs at the same clock $t$ and form a system of $(\phi - r)$ equations in terms of these unknown bits. We then solve this system of equations.

We check for consistency by substituting all bits, both guessed and solved, of $u_i^t$, into the function $f$ for the $r$ additional resyncs at the same clock and compare our result with the actual keystream bits. If they agree, we keep the vector $u_i^t$. Since we have made $2^r$ guesses in $u_i^t$ and we are verifying $r$ keystream bits, we should end up with about one guess out of $2^r$ that passes the consistency check. With this correct solution for $u_i^t$, we form $\phi$ linear equations in terms of the $n$ key bits of $K$.

The steps in the two preceding paragraphs are repeated as necessary to solve for more $u_i^{t'}$s. A total of $\lceil \frac{n}{\phi} \rceil$ $u_i^{t'}$s need to be formed from distinct clocks. When we have formed enough linear equations, we solve for the key bits by Gaussian elimination.

In the guess and verification steps, we require a total of about

$$\left\lceil \frac{n}{\phi} \right\rceil \times \phi \times 2^r. \tag{7}$$

$f$-function evaluations.

The total number of row operations needed is

$$\left\lceil \frac{n}{\phi} \right\rceil \times (\phi - r)^2 \times 2^r + n^2. \tag{8}$$

The above expression can be refined to represent the complexity more accurately. If $(\phi - r) > 64$ and we are performing Gaussian elimination on a 64-bit machine, then the number of row operations needed to solve the $(\phi - r) \times (\phi - r)$ matrix is $\frac{(\phi-r)^3}{64}$, assuming that matrix coefficients are stored in 64-bit words. The same goes for the $n^2$ term, which should be $\frac{n^3}{64}$ when $n > 64$.

We need a total of $\lceil \frac{n}{\phi} \rceil \times \phi$ keystream bits. Each group of $\phi$ keystream bits has to come from the same clock.

# 3 COMPARISON WITH OTHER RESYNCHRONIZATION ATTACKS

## 3.1 Comparison with the Original Resynchronization Attack

Suppose we have a 256-bit LFSR and a 128-bit IV. We let the key size $n = 128$, $\phi = 50$ and $r = 25$. To guard against the Time-Memory-Data Trade-Off Attack, we have chosen the size of the LFSR to be twice the size of the key.

The basic resync attack of (Daemen et al., 1993) requires $\lceil 128/50 \rceil \times 2^{50} \approx 2^{51}$ $f$-function evaluations and $\frac{128^3}{64} = 2^{15}$ row operations.

In comparison, our attack in Section 2.3 requires $\lceil 128/50 \rceil \times 50 \times 2^{25} \approx 2^{32}$ $f$-function evaluations $\lceil 128/50 \rceil \times (50-25)^2 \times 2^{25} + \frac{128^3}{64} \approx 2^{36}$ row operations.

The attack of (Daemen et al., 1993) has a factor of $2^{21}$ less row operations than the attack of Section 2.3, while the latter has a factor of $2^{19}$ less function evaluations. These factors are comparable. However, since function evaluation is a more computationally complex task (more so when the function is of high degree) compared to row operations, we can expect the overall complexity for the attack of Section 2.3 to be less than that for (Daemen et al., 1993). The resource requirements for both attacks are tabulated in Tables 1 and 2.

Table 1: Basic Attack.

| | |
|---|---|
| $f$-function evaluations | $2^{51}$ |
| Row operations | $2^{15}$ |
| Number of resyncs | 50 |
| Number of clocks | 3 |

Table 2: Our Attack.

| | |
|---|---|
| $f$-function evaluations | $2^{32}$ |
| Row operations | $2^{36}$ |
| Number of resyncs | 50 |
| Number of clocks | 3 |

## 3.2 Comparison with Cube Attack

The cube attack is an algebraic attack recently introduced by Dinur and Shamir at Crypto 2008 (Dinur and Shamir, 2009). Given a function $f : \{0,1\}^{\phi} \rightarrow \{0,1\}$ of degree $r+1$, we attempt to find *maxterms*, monomials $t = x_{i_1} \dots x_{i_r}$ of degree $r$. They have the property that $f_0$ can be expressed as

$$f(x) = t \cdot P(x) + Q(x)$$

where $Q(x)$ does not have any terms that are divisible by $t$, and $P(x)$ is nonzero. Then, by summing $f$ over the *cube* where the $x_{i_j}$ are varied over all possible values, and varying the values of the other $x_k$, we can solve for $P(x)$. Since $P(x)$ is linear, by taking $\phi$ maxterms, we get a system of linear equations which we can solve for the $x_i$.

When applied to a LFSR filtered by a Maiorana-McFarland function of degree $r+1$, we get a precomputation complexity of

$$\left\lceil \frac{n}{\phi} \right\rceil \times \phi(\phi+1) \times 2^r + n^2, \qquad (9)$$

and online complexity of

$$\left\lceil \frac{n}{\phi} \right\rceil \times \phi \times 2^r + n, \qquad (10)$$

where $n$ is the key size and $\phi$ the input size of the filter function. In Equations (9) and (10), the $n^2$ and $n$ terms account for complexities of matrix inversion and matrix multiplication respectively. These terms should be changed to $n^3/64$ and $n^2/64$ when $n > 64$ for reasons mentioned in Section 2.3. The online attack complexity is comparable with that of our resync attack above.

However, our resync attack has several advantages over the cube attack. Firstly, it requires only known IVs, whereas the cube attack needs chosen IVs so as to be able to sum over the cube. Furthermore, the cube attack requires $\phi \times 2^r$ resyncs, while our resync attack only needs $\phi - r$ resyncs. (Both methods need $\lceil \frac{n}{\phi} \rceil$ clocks per resync.) These much weaker conditions make our resync attack more suitable than the cube attack for Maiorana-McFarland functions. Finally, the operations used in the resync attack are matrix row operations, which are computationally less demanding than the large number of function evaluations for the cube attack.

To illustrate the complexities of the two attacks, we consider the following example:

$$\begin{aligned} f(x_0,\ldots,x_{49}) = \;& x_0 \cdot x_{25} + x_1 \cdot x_{26} + \cdots + x_{23} \cdot x_{48} \\ & + S(x_{25},\ldots,x_{48}) + x_{24} + x_{49} \end{aligned}$$

where $S$ is a degree 24 function comprising the sum of many high degree monomials such that it is difficult to find a low degree multiple.

This is a degree 24, 1-resilient (balanced and 1-correlation immune) boolean function with a high nonlinearity of $2^{49} - 2^{25}$, and it is also resistant to algebraic attack due to the lack of a low degree multiple. Suppose the key size $n = 128$. We let the input to $f_0$ be a linear mixing of key and IV.

Our resync attack requires $3 \times 50 \times 2^{24} \approx 2^{31}$ function evaluations, and $3 \times 26^2 \times 2^{24} + \frac{128^3}{64} \approx 2^{35}$ row operations, as well as approximately 26 resyncs.

The cube attack needs $3 \times 50 \times 51 \times 2^{23} \approx 2^{36}$ function evaluations and $\frac{128^3}{64} = 2^{15}$ row operations for the precomputation stage, as well as $3 \times 50 \times 2^{23} \approx 2^{30}$ function evaluations and $\frac{128^2}{64} = 2^8$ multiplications for the online phase. It also requires approximately $2^{29}$ chosen IV resyncs.

Table 3: Our Attack.

| | |
|---|---|
| $f$-function evaluations | $2^{31}$ |
| Row operations | $2^{35}$ |
| Number of resyncs | 26 |
| Number of clocks | 3 |

Table 4: Cube Attack.

| | |
|---|---|
| $f$-function evaluations (precomp) | $2^{36}$ |
| Row operations (precomp) | $2^{15}$ |
| $f$-function evaluations (online) | $2^{30}$ |
| Multiplications (online) | $2^{8}$ |
| Number of chosen IV resyncs | $2^{29}$ |
| Number of clocks | 3 |

As shown in the Tables 3 and 4, the number of function evaluations required for our attack is comparable to that for the cube attack. However, our attack requires a much smaller number of resyncs. Furthermore, the IVs do not need to be of a chosen form.

## 4 CONCLUSIONS

We have applied the resynchronization attack on stream ciphers with linearly clocked registers filtered with Maiorana-McFarland functions. While Boolean functions with large input sizes, nonlinearities, resiliencies and algebraic degrees may be ideal choices for the cryptographic components in a synchronous stream cipher we have described, it is not the case for the class of functions we have studied. Despite their good trade-off between cryptographically desirable properties, their simple algebraic form has made them prone to guess-and-linearize-like attacks such as that we have described. Our study has also affirmed the common view that the internal state should not be linearly resynchronized from the key and IV.

## REFERENCES

Canteaut, A., Carlet, C., Charpin, P., and Fontaine, C. (2000). Propagation characterisics and correlation-immunity of highly nonlinear boolean functions. In *Eurocrypt 2000*. LNCS 1807:507-522.

Carlet, C. (2002). A larger class of cryptographic boolean functions via a study of the Maiorana-McFarland construction. In *Crypto 2002*. LNCS 2442:549-564.

Daemen, J., Govaerts, R., and Vandewalle, J. (1993). Resynchronization weakness in synchronous stream ciphers. In *Eurocrypt 1993*. LNCS 765:159-167.

Dinur, I. and Shamir, A. (2009). Cube attacks on tweakable black-box polynomials. In *Eurocrypt 2009*. LNCS 5479:278-299.

Hell, M., Johansson, T., and Meier, W. Grain - a stream cipher for constrained environments. In *The eStream Project - eStream Phase 3: http://www.ecrypt.eu.org/stream/grainp3.html*.

Mihaljevic, M. and Imai, H. (2002). Cryptanalysis of Toyocrypt-HS1 stream cipher. In *IEICE Trans. Fundamentals, vol. E85-A no. 1, pp. 66-73*.

Sarkar, P. and Maitra, S. (2000). Nonlinearity bounds and constructions of resilient boolean functions. In *Crypto 2000*. LNCS 1880:515-532.

Seberry, J., Zhang, X., and Zheng, Y. (1993). On constructions and nonlinearity of correlation immune functions (extended abstract). In *Eurocrypt 1993*. LNCS 765:181-199.