Modifications in the Design of Trivium to Increase its Security Level

Mehreen Afzal
College of Signals,
National University of Science and Technology
E-mail: mehreenafzal00@hotmail.com
*Corresponding author

Ashraf Masood
College of Signals,
National University of Science and Technology
E-mail: ashrafm61@gmail.com

Abstract: Inner state of a stream cipher is said to be as large as necessary but at the same time as small as possible. Trivium, a hardware oriented stream cipher, has been selected for the final portfolio of the eSTREAM project. It offers a security level of 80 bits while it has 288 internal state bits. Owing to its simple algebraic structure, it has been proved experimentally that Trivium can provide only a marginal security level of 80 bits. This article presents some modified versions of Trivium to increase its security level from 80 bits. Our objective is to give a better security level with the same number of internal states without changing much the elegant and simple design philosophy of Trivium. The focus is to make its algebraic structure intricate enough to resist the algebraic attack with guess and determine approach, which can recover its secret internal state bits. We have proposed two possible modifications that can increase its security level without any increase in the number of AND gates. Maximov and Biryukov have proposed a tweaked version of Trivium (Trivium/128) (11), with additional AND gates, to increase the security level to 128 bits. In this article, two other modifications with additional product terms proven to have a better security margin than Trivium/128 are also presented.

Keywords: Cryptography, Stream cipher algorithm, Internal state, Algebraic analysis, Trivium

1 Introduction

Stream ciphers, a class of data encryption primitives, are widely applicable both in hardware and in software. Their importance due to their efficient application is behind the success of projects such as NESSIE (12) and eSTREAM by ECRYPT (9). However, the board of the NESSIE project could not select any stream cipher for its final portfolio because most of the proposals underwent successful cryptanalysis. Later the eSTREAM project started in 2004 with 34 submissions and ended after selecting some interesting structures for their final portfolio. In keeping with the theme of the project, four stream ciphers were selected each for hardware and software profiles.

Trivium (7; 6; 8) is also one of the ECRYPT stream cipher project candidates selected for the final portfolio. It is a hardware-oriented synchronous cipher, which supports a key size of 80 bits and an initialization vector (IV) size of 80 bits. It has remained unchanged since it was submitted. A few papers on its cryptanalysis can be found on the eSTREAM website (11; 10; 14; 13; 5). Results given in (10) and (14) have proved Trivium to be in general strong against linear sequential approximation attacks. In another approach, presented in (13), that is algebraic in nature, it has also been established that Trivium can withstand this kind of attack; however, a two-round variant of Trivium is shown to be compromised. Some other suggestions and ideas on attacking Trivium are given by Babbage in (5). In (11), Maximov and Biryukov give theoretical evidence that internal state bits of Trivium can be recovered with a time complexity around $c2^{83.5}$. Their analysis gave a methodological cryptanalysis of the structure, including state recovering and linear distinguishing attacks,

and proved that it has a very thin security margin. It is thus established that the structure of Trivium is not adequate for increasing its key stream length and that the design provides a marginal security of 80 bits. They have also proposed a tweaked version of Trivium, Trivium/128, with three additional AND gates, which can resist the proposed attacks with a 128-bit security level. In another work a practical algebraic analysis of Trivium is presented (1).On one hand it has been validated that secret key cannot be increased to 128 bits with the original structure of Trivium. Moreover, the analysis has also proved that Trivium/128 is not suitable to provide 128-bit security.

Since Trivium has an internal state of 288 bits but provides a security level of only 80 bits, an obvious desire is to raise its security level to at least 128 bits without increasing its internal state bits. The studies and analysis of Trivium available so far in literature have proved that mainly the algebraic structure of the cipher has the propensity that a guess and determine type of attack, to recover the internal state bits of the cipher by solving algebraic equations, can be mounted. In this article, we propose some modifications in the structure of Trivium so that its elegant structure can be used to give a larger security margin. The modified versions show better resistance to the internal state recovery attack. Trivium/128 uses one additional AND gate in each register to improve its security margin. We, however, present two simple modifications without any additional AND gates and show that our proposed modifications can provide better security level than actual Trivium. Other two modifications with additional AND gates, are also proposed but these can provide remarkably better resistance than Trivium/128, and thus can give a larger security margin with the same number of internal state bits. We compare our proposed structures with the original Trivium and Trivium/128 based on the analysis made in (1). As it can be prospected that while trying to increase the complexity of one attack, we may make our cipher vulnerable to any other attack. But it should be noted here that we have not changed the basic design philosophy of the Trivium cipher. Therefore our proposed modifications will not reduce the complexity of any other possible attack on the cipher. To best of our knowledge, there is no other existing successful attack on the cipher and therefore this article mainly discusses the effect of proposed modifications on the recovery of internal state bits by exploiting its algebraic structure of the cipher.

Rest of the paper is organized as follows: Consequent section presents the brief description of the structure of Trivium along with a discussion on its algebraic structure; we also give a discussion on increasing bit security level of Trivium. Section 3 and 4 presents the proposed modifications of Trivium with and without additional AND gates respectively. The article is concluded in Section 5.

Figure 1: Design of the Trivium Cipher

2 A Brief Description of Trivium

Trivium is a simple hardware-oriented synchronous stream cipher. The proposed design uses an 80-bit secret key and an 80-bit IV. It consists of an iterative process that extracts the values of 15 specific state bits and uses them both to update 3 bits of the state and to compute 1 bit of the key stream. The state bits are then rotated, and the process is repeated. The cipher is shown to be suitable to generate up to 2^{64} bits of the key stream from a pair of key and IV.

Let the 288-bit internal state of the cipher be represented as $(s_1, s_2, ...s_{288})$; then the complete description of the cipher is given by the following simple pseudo-code:

procedure $Trivium-orig(s_1, ..s_{288})$
  for $t = 1$ to n **do**
    $t_1 := s_{66} + s_{93}$
    $t_2 := s_{162} + s_{177}$
    $t_3 := s_{243} + s_{288}$
    $z_t := t_1 + t_2 + t_3$
    $t_1 := t_1 + s_{91}.s_{92} + s_{171}$
    $t_2 := t_2 + s_{175}.s_{176} + s_{264}$
    $t_3 := t_3 + s_{286}.s_{287} + s_{69}$
    $(s_1, s_2...s_{93}) := (t_3, s_1, ...s_{92})$
    $(s_{94}, s_{95}...s_{177}) := (t_1, s_{94}, ...s_{176})$
    $(s_{178}, s_{179}...s_{288}) := (t_2, s_{178}, ...s_{287})$
  end for

For key initialization, the 80-bit key and IV are directly assigned to the internal state of the cipher and the remaining bits (except the last three) are set to zero. Then, the cipher is clocked 4 full cycles without producing any output. Figure 1 shows the original structure of Trivium.

To illustrate why a large number of state bits can be recovered when bits at some appropriate places are guessed, we give here the algebraic relationship of the internal state bits with the output bits. polynomial expressions of the generated output bits in terms of the initial state bits during the first five clocks can be seen as
$y_{66} + y_{93} + y_{162} + y_{177} + y_{243} + y_{288}$,
$y_{65} + y_{92} + y_{161} + y_{176} + y_{242} + y_{287}$,

$y_{64} + y_{91} + y_{160} + y_{175} + y_{241} + y_{286}$,
$y_{63} + y_{90} + y_{159} + y_{174} + y_{240} + y_{285}$,
$y_{62} + y_{89} + y_{158} + y_{173} + y_{239} + y_{284}$

After 66 clocks, second-degree expressions are obtained; consider a few of them:
$y_{243} + y_{288} + y_{286}.y_{287} + y_{69} + y_{27} + y_{96} + y_{111} + y_{162} + y_{177} + y_{175}.y_{176} + y_{264} + y_{222}$,
$y_{242} + y_{287} + y_{285}y_{286} + y_{68} + y_{26} + y_{95} + y_{110} + y_{161} + y_{176} + y_{174}.y_{175} + y_{263} + y_{221}$,
$y_{241} + y_{286} + y_{284}.y_{285} + y_{67} + y_{25} + y_{94} + y_{109} + y_{160} + y_{175} + y_{173}.y_{174} + y_{262} + y_{220}$,
$y_{240} + y_{285} + y_{283}.y_{284} + y_{24} + y_{93} + y_{91}.y_{92} + y_{171} + y_{108} + y_{159} + y_{174} + y_{172}.y_{173} + y_{261} + y_{219}$,
$y_{239} + y_{284} + y_{282}.y_{283} + y_{23} + y_{92} + y_{90}.y_{91} + y_{170} + y_{107} + y_{158} + y_{173} + y_{171}.y_{172} + y_{260} + y_{218}$.

The occurrence of alternate variables in the above expressions reveals the importance of guessing alternate bits. When alternate bits are guessed, a large number of linear equations are obtained, and, therefore, the algebraic equations thus formed can be solved within a few seconds to obtain 144 unknown bits. The number of guessed bits can be further reduced to 120 if the first 10 to 12 state bits from each of the three divisions of state bits $(s_1..s_{93}, s_{94}..s_{177}, s_{178}..s_{288})$ are left as unknowns, and for the rest of the state bits, alternate positions are guessed. For further detail, one may refer to (1). Consequently, with 120 bits guessed, 168 bits can be recovered within a few seconds on a PC of limited computational power. This result also confirms the theoretical results given in (11) that with the actual design of Trivium, the security level cannot be increased to 128 bits.

Trivium/128 (11) has an additional three AND gates that are connected backward. As assumed earlier, if the initial state bits of Trivium/128 are taken as $y_1, y_2, ...y_{288}$, we obtain polynomial expressions of the output bits in terms of the initial state bits. For Trivium/128, after 65 linear expressions, quadratic expressions of the following form are obtained:

$y_{242} + y_{288} + y_{286}.y_{287} + y_{69} + y_{66}.y_{68} + y_{28} + y_{96} + y_{112} + y_{161} + y_{177} + y_{175}.y_{176} + y_{264} + y_{243}.y_{245} + y_{223}$,
$y_{241} + y_{287} + y_{285}.y_{286} + y_{68} + y_{65}.y_{67} + y_{27} + y_{95} + y_{111} + y_{160} + y_{176} + y_{174}.y_{175} + y_{263} + y_{242}.y_{244} + y_{222}$,
$y_{240} + y_{286} + y_{284}y_{285} + y_{67} + y_{64}y_{66} + y_{26} + y_{94} + y_{110} + y_{159} + y_{175} + y_{173}y_{174} + y_{262} + y_{241}y_{243} + y_{221}$,
$y_{239} + y_{285} + y_{283}y_{284} + y_{66} + y_{63}y_{65} + y_{25} + y_{65} + y_{93} + y_{91}.y_{92} + y_{171} + y_{162}.y_{164} + y_{109} + y_{158} + y_{174} + y_{172}y_{173} + y_{261} + y_{240}y_{242} + y_{220}$,
$y_{238} + y_{284} + y_{282}.y_{283} + y_{65} + y_{62}y_{64} + y_{24} + y_{64} + y_{92} + y_{90}.y_{91} + y_{170} + y_{161}.y_{163} + y_{108} + y_{157} + y_{173} + y_{171}y_{172} + y_{260} + y_{239}.y_{241} + y_{219}$

Although alternate guessing of bits cannot simplify the equations, so that the system of equations thus formed can be solved, still, half bits can be recovered within a few seconds, if half of the appropriately selected bits are guessed (1). Because a large number of state bits can be recovered within the limited resources, it can be seen that with better resources the results can be further improved. Thus, it can be concluded that Trivium/128 is not adequate to provide 128-bit security.

The discussion here reveals that the possibility of recovering state bits can be reduced if we can have lesser number of linear equations in the start and also the variables in the product terms are distributed in a way that degrees of equations do not decline on guessing of the bits. The proposed modifications have this inspiration. We, therefore, give here a comparison of the degree of equations, which are formed, from our proposed modifications, with degrees of original Trivium and Trivium/128. In general, the degrees of equations cannot be taken as the only criteria for deciding about a system difficult enough or not to be solved. In this case as the original structure let too many bits to be recovered owing to the presence of many lower-degree equations, therefore, keeping the main design strategy intact, we compare our modified designs with the original design based on these criteria.

3 Modifications Of Trivium Without Additional Product Terms

In this section, we propose two modified designs of Trivium. Our focus is to increase the degrees of algebraic equations to be higher than the original version without adding any product terms so that the efficiency of the original structure is not declined.

3.1 First Modification of Trivium: Trivium-A

Our first proposal for modification is simple, and the idea is to reduce the number of linear equations; to start with, the system of equations thus formed has a higher degree than the original Trivium. We have seen in the preceding sections that the structure of algebraic equations of the original cipher has the allowance that if some bits are guessed at some selective positions, many linear and other lower degree equations can be obtained. And as a result remaining secret bits can be recovered in a few seconds within the limited resources. The proposed modification tries to avoid this easy recovery of bits even when half of the bits are guessed. This may be achieved if feedback of the product term appears in the output bits earlier. Consider the possible modification as given in Figure 2, keeping the above-mentioned criterion in mind.

The following psuedo-code represents the above mentioned modified version:

for t from 1 to n do
$t_1 := s_{31} + s_{93}$
$t_2 := s_{121} + s_{177}$
$t_3 := s_{214} + s_{288}$
$z_t := t_1 + t_2 + t_3$
$t_1 := t_1 + s_{91}.s_{92} + s_{148}$
$t_2 := t_2 + s_{175}.s_{176} + s_{251}$
$t_3 := t_3 + s_{286}.s_{287} + s_{62}$
$(s_1, s_2...s_{93}) := (t_3, s_1, ...s_{92})$
$(s_{94}, s_{95}...s_{177}) := (t_1, s_{94}, ...s_{176})$

Figure 2: First Modified Version of the Trivium-Trivium-A

Figure 3: A Comparison of the degrees of the Algebraic equations of Trivium-A with the original version of Trivium and Trivium/128

$(s_{178}, s_{179}...s_{288}) := (t_2, s_{178}, ...s_{287})$
end do

In this case we obtain 55 linear equations compared with 66 of the original structure and 65 of Trivium/128 (1). This modification does not involve any additional AND gate; just a change in the placement of the feedback function can help in reducing the number of linear equations. A comparison between the degrees of algebraic equations of Trivium-A with those of the original Trivium and Trivium/128 is presented in Figure 3.

As evident from Figure 3, degrees of Trivium-A are higher than both those of the original Trivium and Trivium/128. Because here the form of equations is not changed, therefore, alternate guessing of bits will simplify the equations. However, it has been experimentally proved that due to the presence of higher-degree equations, the system of equations will not be solved with half bits guessed even if they are selected.

3.2 Second Modification of Trivium: Trivium-B

In this section, we introduce another possible modification in the structure of Trivium. Here also no additional AND gate is used; rather the product of variables is placed due to the mutual dependence introduced within the internal state bits. Due to this, second-degree feedback is generated, and that is why the product of variables in steps 6, 7, and 8 of the procedure of the original Trivium can also be eliminated. However, this second-degree feedback results in complex equations as described next.

In stream ciphers, mutual dependence of registers in clocking can resist algebraic cryptanalysis due to the quick rise in the degree of equations (3; 2; 4). The structure of Trivium, as shown in Figure 1, also has three registers. We propose that if updating of the feedback variables of the three registers becomes mutually dependent, the degrees of equations rise more rapidly. However, it will naturally result in another product of variables or an additional AND gate in implementation of the cipher. Therefore, to maintain the number of AND gates as in the original structure, we eliminate the product terms at subsequent steps.

Given below is the procedure for updating the variables $t_1, t_2 and t_3$ based on mutual dependence of the three portions of the state bits:

1: **if** $s_{162} = 1$ **then**
2: $t_1 := s_{66} + s_{93}$
3: **else**
4: $t_1 := s_{93}$
5: **end if**
6: **if** $s_{243} = 1$ **then**
7: $t_2 := s_{162} + s_{177}$
8: **else**
9: $t_2 := s_{177}$
10: **end if**
11: **if** $s_{66} = 1$ **then**
12: $t_3 := s_{243} + s_{288}$
13: **else**
14: $t_3 := s_{288}$
15: **end if**

This change in the updated variables results in the following complete procedure of the proposed modified version Trivium-B:

procedure $Trivium - B(s_1, ..s_{288})$
for $t = 1$ to n **do**
$t_1 = s_{66}.s_{162} + s_{93}$
$t_2 = s_{162}.s_{243} + s_{177}$
$t_3 = s_{243}.s_{66} + s_{288}$
$z_t = t_1 + t_2 + t_3$
$t_1 = t_1 + s_{91} + s_{171}$
$t_2 = t_2 + s_{175} + s_{264}$
$t_3 = t_3 + s_{286} + s_{69}$
$(s_1, s_2...s_{93}) = (t_3, s_1, ...s_{92})$
$(s_{94}, s_{95}...s_{177}) = (t_1, s_{94}, ...s_{176})$
$(s_{178}, s_{179}...s_{288}) = (t_2, s_{178}, ...s_{287})$
end for

With the modification described above, the output

Figure 4: A Comparison of the degrees of the Algebraic equations of the original version of Trivium with the proposed modified version Trivium-B

bits can be expressed in terms of internal state bits with the algebraic equations of the following form. First five equations are given below for illustration:
$y_{66} * y_{162} + y_{93} + y_{162} * y_{243} + y_{177} + y_{243} * y_{66} + y_{288} + 1$,
$y_{65} * y_{161} + y_{92} + y_{161} * y_{242} + y_{176} + y_{242} * y_{65} + y_{287}$,
$y_{64} * y_{160} + y_{91} + y_{160} * y_{241} + y_{175} + y_{241} * y_{64} + y_{286}$,
$y_{63} * y_{159} + y_{90} + y_{159} * y_{240} + y_{174} + y_{240} * y_{63} + y_{285}$,
$y_{62} * y_{158} + y_{89} + y_{158} * y_{239} + y_{173} + y_{239} * y_{62} + y_{284} + 1$,
$y_{61} * y_{157} + y_{88} + y_{157} * y_{238} + y_{172} + y_{238} * y_{61} + y_{283} + 1$

With this modification, we obtain the first 66 second-degree equations as compared with the 66 linear equations from the original Trivium. Afterward, degrees are increased stepwise. A comparison of the degrees of algebraic equations of the original Trivium with those of Trivium/128 and Trivium-B is presented in Figure 4.

Compared with the original version, Trivium-B not only achieves considerably high degrees but also the alternate guessing of bits is not helpful. However, if some selective bits are guessed by taking into account the structure of equations, some lower-degree equations can be obtained. As described earlier, the internal states of Trivium can be divided into three segments. If alternate bits of two of the segments and some consecutive bits from the third segment are guessed, we may obtain some linear equations. When half of the bits are guessed with this strategy, we may obtain equations with degrees somewhat lesser than the case of alternate guessed bits. Figure 5 shows a comparison of the degrees of original Trivium when state bits at alternate places are guessed with proposed Trivium-B when state bits at alternate and selective positions are known.

It can be seen from the Figure 5 that as compared with the large number of linear equations obtained as a result of guessing state bits at alternate places of original Trivium, the degrees of equations of Trivium-B are much higher. We do not consider here the case of guessing more than half the state bits because in the case of the original Trivium, we can do with fairly less than half to recover the rest of the internal state bits. Thus, we conclude after trying a number of strategies for guessing (with half of the bits guessed), Trivium-B does not give enough lower-degree equations that can be solved.

Figure 5: A Comparison of the degrees of the Algebraic equations of the original version of Trivium with the proposed modified version Trivium-B while half bits are guessed

Both of our proposed versions have degrees higher than even Trivium/128, whereas Trivium/128 also has an additional AND gate to improve the security level to 128 bits. It can be concluded that the Trivium-A and the Trivium-B provide resistance to the algebraic attack with guess and determine approach. Thus the recovery of internal secret state bits of Trivium is more complex than before so that it may have a better security margin.

4 Modifications of Trivium With Additional Product Terms

In this section, we propose some other modified designs of the Trivium again without altering its main design philosophy. Here again our strategy is to increase the degrees of algebraic equations much higher without disturbing the original design of the cipher. We aim to achieve a state where with guessing some of the bits one cannot turn the degrees of equations lower enough to make the system solvable. These modifications with additional AND terms aim to provide security level of 128 bits and we have shown experimentally that our modifications provide better resistance to the algebraic attack with guess and determine approach as compared to Trivium/128.

4.1 Third Modification of Trivium: Trivium-C

This proposed design uses the same idea as in Trivium-B. Here we introduce the same product terms resulting from the interdependent updating of three portions of the inter-

Figure 6: A Comparison of the degrees of the Algebraic equations of Trivium-C with the original Trivium and Trivium/128

nal state of the cipher. Unlike Trivium-B, here we do not remove the product of alternate bits in the original design of Trivium. Thus the AND gates introduced in Trivium-B are additionally included within the design. We compare this design for degrees of equations with not only the original Trivium, from which it must offer higher-degree equations but also with Trivium/128. The comparison with Trivium/128 shows that our proposed version gives much higher degrees of the system of equations, although the proposed version also uses three additional AND gates as in Trivium/128.

Proposed design of Trivium-C is illustrated with the following pseudo-code:

procedure $Trivium-C(s_1,..s_{288})$
for $t = 1$ to n **do**
$t_1 = s_{66}.s_{162} + s_{93}$
$t_2 = s_{162}.s_{243} + s_{177}$
$t_3 = s_{243}.s_{66} + s_{288}$
$z_t = t_1 + t_2 + t_3$
$t_1 := t_1 + s_{91}.s_{92} + s_{171}$
$t_2 := t_2 + s_{175}.s_{176} + s_{264}$
$t_3 := t_3 + s_{286}.s_{287} + s_{69}$
$(s_1, s_2...s_{93}) = (t_3, s_1, ...s_{92})$
$(s_{94}, s_{95}...s_{177}) = (t_1, s_{94}, ...s_{176})$
$(s_{178}, s_{179}...s_{288}) = (t_2, s_{178}, ...s_{287})$
end for

A comparison of the degrees of algebraic equations of Trivium-C with those of the original Trivium and Trivium/128 is presented in Figure 6.

Like Trivium-B, alternate guessing of bits is not helpful here also. However, in this case due to the additional product terms, even the selective guessing does not work. It is concluded after experiments with many options based on the structure of the equations that even after guessing a large number of state bits (a lot more than half), we do not obtain a simplified enough system that could be solved within resources similar to those that were used for the original Trivium and Trivium/128.

4.2 Fourth Modification of Trivium: Trivium-D

Yet another modification in the design of Trivium is proposed here to increase the security level of cipher with the same number of internal state bits. As we have seen in the original design of Trivium, states are updated nonlinearly with the second-degree expressions. However, the output is simple XOR of three state bits. That is why linear equations are obtained in the first 66 clockings, and afterward degrees are increased in steps. If the output function is also made nonlinear, then there will be no linear equation even at the start of the clocking. Here, the nonlinear update and the remaining design strategy are not altered at all.

For illustration, we have selected a simple majority function: $f(x_1, x_2, x_3) = x_1x_2 \oplus x_1x_3 \oplus x_2x_3$. In this case, we have added three product terms or in other words three AND gates, but as a result we obtain much higher-degree equations than those of Trivium/128.

Pseudo-code of Trivium-D with the modification described above is given here:

procedure $Trivium-D(s_1,...s_{288})$
for $t = 1$ to n **do**
$t_1 := s_{66} + s_{93}$
$t_2 := s_{162} + s_{177}$
$t_3 := s_{243} + s_{288}$
$z_t = t_1.t_2 + t_1.t_3 + t_2.t_3$
$t_1 := t_1 + s_{91}.s_{92} + s_{171}$
$t_2 := t_2 + s_{175}.s_{176} + s_{264}$
$t_3 := t_3 + s_{286}.s_{287} + s_{69}$
$(s_1, s_2...s_{93}) = (t_3, s_1, ...s_{92})$
$(s_{94}, s_{95}...s_{177}) = (t_1, s_{94}, ...s_{176})$
$(s_{178}, s_{179}...s_{288}) = (t_2, s_{178}, ...s_{287})$
end for

Increase in the degrees of equations of Trivium-D with the original Trivium and Trivium/128 is given in Figure 7.

The modified version Trivium-D also has second-degree terms of alternate state bits, but here alternate guessing of bits does not reduce the overall degrees of output equations. The reason naturally lies in the fact that there are many second-degree terms in the output polynomial attributed to the combining second-degree Boolean functions. Thus, even if a large number of state bits are guessed, linear equations or lower-degree equations are not obtained. Thus, the system is not solvable.

In all the modifications discussed here, naturally the authors did not attempt to guess more than half the state bits and, therefore, have not found the limit to the minimum number of state bits that need to be guessed to make the system of equations feasible to be solved. All the situations discussed here show better results in generating complex equations than original Trivium. Moreover, the proposed versions with additional AND gates are more difficult to solve than Trivium/128.

Figure 7: A Comparison of the degrees of Algebraic equations of Trivium-D with the original Trivium and Trivium/128

5 Conclusion

Trivium, a selected candidate for the final portfolio of the eSTREAM project, has a simple and elegant structure. Since its internal state is quite larger than its key-stream length, a natural question arises whether its bit security level can be increased with the same number of internal state bits. Based on the experiments, it was found that this can be achieved with small design modifications. We have proposed here some modified versions of Trivium to increase its bit security level. Compared with an earlier proposed tweak, Trivium/128, our modifications show better resistance against the algebraic cryptanalysis of Trivium with a larger bit security level. Two of our proposed modifications do not use any additional AND gates but still achieve better security margin than the original Trivium. Whereas the other two proposed versions with additional AND gates show a much better security level than Trivium/128.

REFERENCES

[1] Mehreen Afzal and Ashraf Masood, 'Experimental results on algebraic analysis of trivium and tweaked trivium', in: *International Conference on Golbal e-Security*, volume 12 of *Communications in Computer and Information Science(CCIS)* (Springer Berlin / Heidelberg, 2008) pp. 93–101.

[2] ——, 'On generating algebraic equations for a5-type key stream generator', in: *Trends in Intelligent Systems and Computer Engineering*, volume 6 (Springer-Verlag, 2008) pp. 443–452.

[3] ——, 'Algebraic attack on a5-type irregularly clocked key stream generator', in: *International Multiconference of Engineers and Computer Scientists-IMECS07, IAENG* (March, 2007) pp. 670–674.

[4] Sultan Al-Hiana, Lynn Margaret Baten and Bernard D. Colbert, 'Mutually clock-controlled feedback shift registers provide resistance to algebraic attacks', in: *8th International Conference on Finite Fields and Applications(FQ8)* (July 2007).

[5] Steve Babbage, 'Some thoughts on trivium' (available at http://www.ecrypt.eu.org/stream/, 2007).

[6] Christophe De Cannière, 'Trivium: A stream cipher construction inspired by block cipher design principles', in: *Information Security*, volume 4176 of *Lecture Notes in Computer Science* (Springer, 2006) pp. 36–55.

[7] Christophe De Cannière and Bart Preneel, 'Trivium: A stream cipher construction inspired by block cipher design principles', in: *http://www.ecrypt.eu.org/stream/papersdir/2006/021.pdf* (2005).

[8] ——, 'Trivium specifications', in: *ECRYPT Stream Cipher Project Report 2005/030* (2005).

[9] eSTREAM, 'Ecrypt stream cipher project', 2005.

[10] Shahram Khazaei and M. Hassanzadeh, 'Linear sequential circuit approximation of the trivium stream cipher' (available at http://www.ecrypt.eu.org/stream/, 2005).

[11] Alexander Maximov and Alex Biryukov, 'Two trivial attacks on trivium', in: *Selected Areas in Cryptography*, volume 4876 of *Lecture Notes in Computer Science* (Springer Berlin / Heidelberg, 2007).

[12] NESSIE, 'New european schemes for signatures, integrity, and encryption', 1999.

[13] Håvard Raddum, 'Cryptanalytic results on trivium' (available at http://www.ecrypt.eu.org/stream/, 2006).

[14] Meltem Sönmez Turan and Orhun Kara, 'Linear approximation for 2-round trivium', in: *Proc. First International Conference on Security of Information and Networks, SIN 2007* (Trafford Publishing, 2007) pp. 96–105.