Dynamic Substitution in Stream Cipher Cryptoghraphy

Saad M. Khalifa,
M.s.c.
Military Eng. College
Baghdad

Abstract:
Dynamic Substitution is a replacement for the week exclusive-OR combiner normally used in stream ciphers . Dynamic Substitution is generally nonlinear , with substantial internal state . Multiple nonlinear combining levels are practical, as are selections among multiple different combiners . The resulting strength can support the use of weaker but faster keying generators . Additional strength is available in " hardened " and customized versions .

Introduction

Dynamic Substitution is a reversible nonlinear combiner based on substitution . A combiner is a cryptographic mechanism which mixes a confusion or keying sequence with data. A combiner is the heart of a stream cipher ,which generally uses an " additive " combiner such as exclusive - OR . Unfortunately , additive combiners have absolutely no strength at all : If an Opponent somehow comes up with some plaintext and matching ciphertext , an additive combiner immediately reveals the confusion sequence This allows the Opponent to begin work on breaking the confusion sequence generator.

In a similar *Known - plaintext_* situation, Dynamic Substitution hides the confusion sequence (to some extent). This strengthens the cipher and allows the use of weaker (and faster) confusion sequence generator. The sources of the Dynamic Substitution idea were :

*A need to improve the strength of a conventional stream cipher , and
*A desire to negate the attacks on simple substitution .

Attacks on simple substitution usually contain the implicit assumption that the substitution table is static or unchanging . One way to avoid these attacks is to change the contents of the table dynamically . the signal needed to control these changes makes simple substitution into a new type of combiner , thus solving both problems at once .

The Dynamic substitution mechanism consists of one or more invertible substitution tables, and some way to change the arrangement of the values in the tables.

Enciphering and deciphering operations

In operation , a data character is substituted into ciphertext or a result character . Then the substitution table is permuted . In the usual case , the content of the just - used entry is swapped with the content of some entry selected by a keyed random number stream.

Enciphering procedure

Suppose we have a simple four - element substitution table , initialized as : (D , B , A , C) for addresses A..D.
That is , adress A contains D , address B contains B , and so on . In this way , the table transforms any input value in the range A..D into some output value in the same range [1] . This is classical simple substitution

INITIAL STATE Forward table

Table value

D	B	A	C

Table Location

A B C D

Data In (DI) : C
Random In (RI) : D
Data Out (DO) = Fwd (C) = A
Swap (Fwd (DI) , Fwd (RI) = Swap (Fwd © , F wd (D)

FINAL STATE Forward Table

Table STATE

D	B	A	C

Table Location

A B C D

Suppose we have a Data input of C and RNG input of D: The data value uses location C, which contains the value A; this is the result or "cipherext." The RNG value selects location D, and then the values in the "used" and "selected" locations are exchanged. This produces the changed table: (D,B,C,A).(This is perhaps the most efficient of many possible froms of Dynamic Substitution.)

DATA In : C

RNG In : D

D	B	A	C
A	B	C	D

Initial State

Result : A

Swap : C , D

D	B	C	A
A	B	C	D

Final State

Dynamic Substitution Encipher

Dynamic substation Encipler

In the table, each substitution element changs "at random" whenever is used, and the more often it is used, the more often it changes. The effect of this is to adaptively "even out" symbol-frequency statistics
Which are the principle weakness of simple substitution. Also, the elements are changed "behind the scenes" (the exchange is not visible until future data selects one of the exchanged elements), and this tends to hide the confusion sequence.

Deciphering procedure

Dynamic substitution is deciphered by maintaining an inverse table, and by updating the inverse table dynamically whenever elements in the forward table are exchanged. This generally also requires [2] maintaining a forward table.

Suppose we have the same substitution table we had before: (D,B,A,C). For deciphering we also need the inverse table: (C,B,D,A) which we originally compute from the forward table. We assume that we will have the transmitted cipher-text and also exactly the same RNG value as at the farend. The RNG is initialized from the same key and so produces exactly the same sequence on both ends.

Initial state

Table value

	Inverse table				Forward table			
Table value	C	B	D	A	D	B	A	C
Table location	A	B	C	D	A	B	C	D

Data In (DI) :A

Random In (RI) : D

Data Out (DO) = Inv [DI] = Inv [A] = C

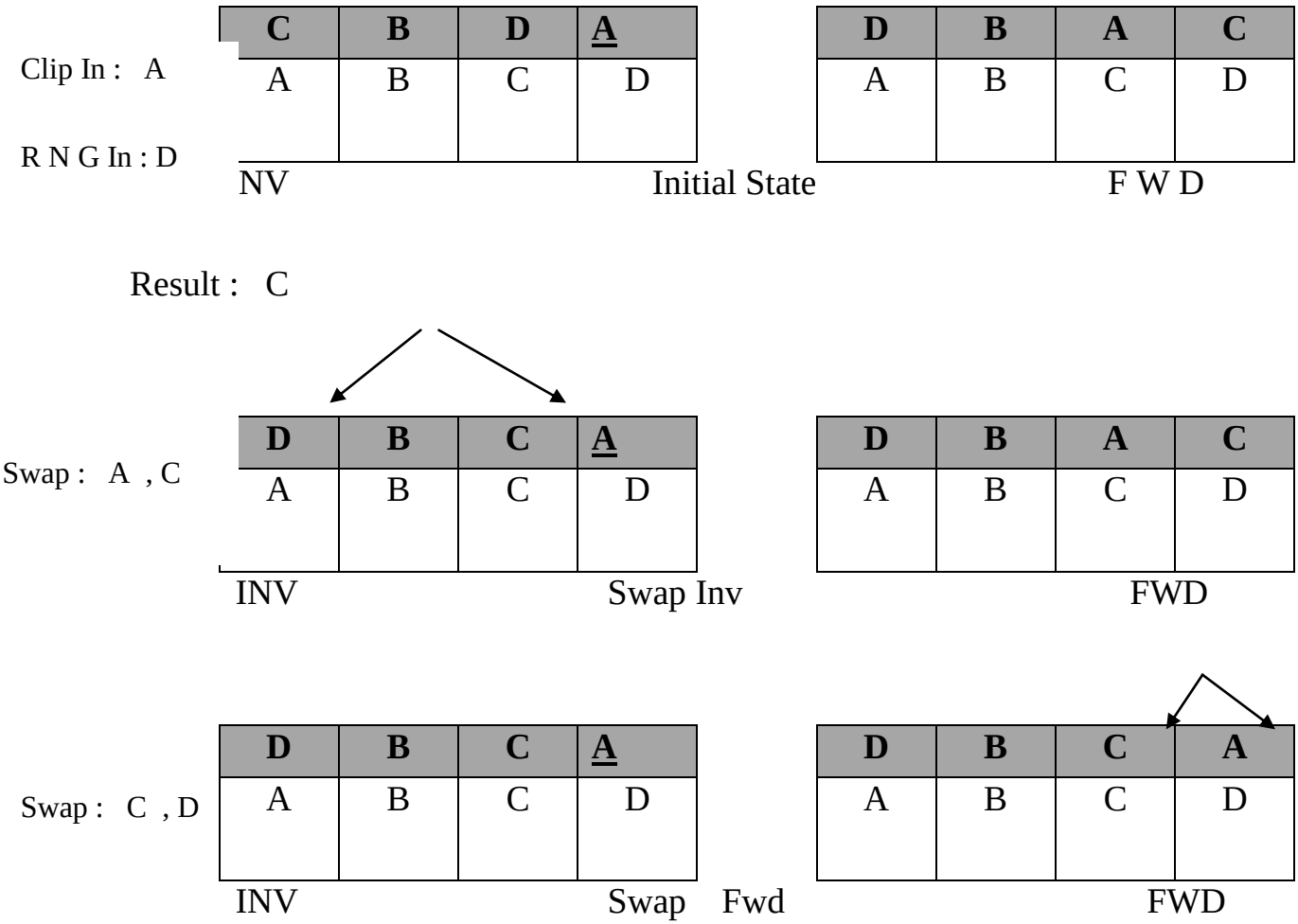
Swap (Inv [DI], Inv [Fwd[RI]] = Swap (Inv [A], Inv [C]

Swap (Fwd [Inv [DI]] , Fwd [RI]) = Swap (Fwd [C], Fwd[D])

FINAL STATE

	Inverse table				Forward table			
Table value	D	B	C	A	D	B	C	A
Table location	A	B	C	D	A	B	C	D

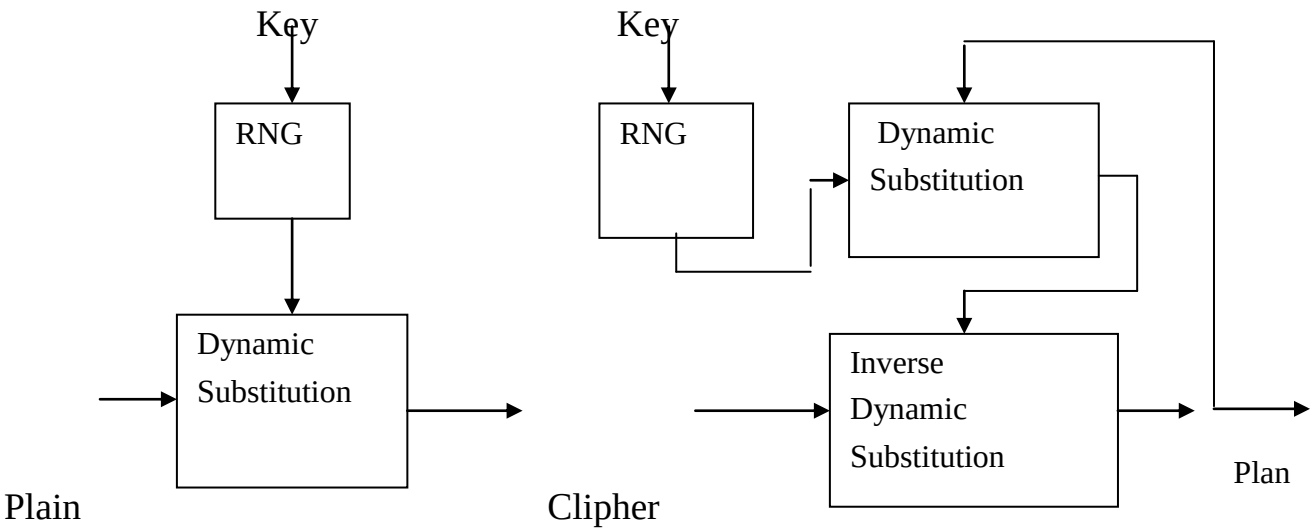
uppose we have a Ciphertext value of A and RNG input of D: The ciphertext uses location A in the inverse table, which contains the value C, which is the result or "plaintext": our original value. The RNG selects location D in the forward table, which contains value C. Then we exchange the values in locations A,C (the enciphered data and enciphered RNG values) in the inverse table. Then we exchange the values in locations C,D (the plaintext data and plaintext RNG values) in the forward table. Just as occurs during enciphering. This serves to keep both the forward and inverse tables "in syn's" with the enciphering system. [3]



Dynamic Substitution Decipher

The point of this is to minimize the amount of work we have to do to maintain the inverse table (which does the deciphering). The locations to be swapped in the inverse table are the enciphered plaintext and the enciphered RNG value. Now , the enciphered plaintext is just the cipherext value, which we have, but the enciphered RNG value is something we do not have. Therefore, we also maitain a forward table to encipher the RNG value so we can update the inverse table. This means that the forward table also must be maintained or "kept in sync." The forward table is updated by swapong elements seleted by the plaintext and RNG values, which we now have, just as occurs during enciphering.

If we consider a substitution table and changes controller as a single component, we find that one such component can be used to encipher, and two such components can be used to decipher: [4]



The difference between the normal and inverse components is just the arrangement of the substitution table.

Hardening Dynamic Substitution

Various constructions can be used to strengthen the basic Dynamic Substitution operation , up to and including re-shuffling the entire table after every character . But the combining operation can be very significantly strengthened with far less overhead than this . Two interesting approaches may be called Doubly Indirect Data and Double Random , and these may be used either separately or together [5].

Doubly Indirect Data

Encipher data through the table twice , and exchange the second selected element with the random selected element.

Enciphering

FINAL STATE

	Forward table			
Table value	C	D	A	B
Table location	A	B	C	D

Deciphering

INITIAL STATE

	Inverse table				Forward table			
Table value	C	A	D	B	B	D	A	C
Table location	A	B	C	D	A	B	C	D

Data In (DI) :B

Random In (RI) :D

Data Out (DO) = Inv [Inv[DI]] = Inv [Inv [B]] = C

Temp = Inv [DI] = Inv [B] =A

Swap (Inv [DI] , Inv [Fwd [RI]]) = Swap (Inv [B], Inv [C])

Swap (Fwd [temp] , Fwd [RI]) = Swap (Fwd [A], Fwd [D])

FINAL STATE

	Inverse table				Forward table			
Table value	C	D	A	B	C	D	A	B
Table location	A	B	C	D	A	B	C	D

The advantage here is the added protection of decoupling the input from the output entry. This generally prevents The Opponent from finding a previous random value when a data in value produces a match to a preceding data out. [6]

The cost of this added strength, being just one more indirect access when enciphering, and perhaps a new temp variable when deciphering.

Double Random

Encipher data through the table once. **To** from an ultimate random value, encipher one random input value and additively combine it with a second random input. Then exchange the selected element with the random selected element.

INTERNAL STATE

	Forward table			
Table value	D	B	A	C
Table location	A	B	C	D

Data In (DI) :C
Random In A (RIA) :D
Random In B (RIB) :C

Data Out (DO) = Fwd[DI] = FWd [C] = A

Temp = RIB + Fwd [RIA] = C + Fwd [D] = C+C =A
Swap (Fwd [DI] , Fwd [temp]) = Swap (Fwd [C], Fwd [A])

FINAL STATE N

	Forward table			
Table value	A	B	D	C
Table location	A	B	C	D

Deciphering :

INITIAL STATE **Inverse Table** **Forward Table**

Table Value	C	B	D	A		D	B	A	C	
Table Location	A	B	C	D		A	B	C	D	

Datat In (DI) : A
Random In A (RIA) : D
Random In B (RIB) : C

Data Out (DO) = Inv (DI) = Inv (A) = C
Temp 1 = RIB + Fwd (RIA) = C + Fwd (D) = C + C + A
Temp 2 = Inv (DI) = Inv (A) = C
Swap (Inv (DI) , Inv (Fwd)(temp 1)1) = Swap (Inv (A) , Inv (D))
Swap (Fwd (temp 2) , Fwd (temp 1) = Swap (Fwd (C) , Fwd (A))

FINAL STATE **Inverse Table** **Forward Table**

Table Value	A	B	D	C		A	B	D	C	
Table Location	A	B	C	D		A	B	C	D	

The advantage of this method is that we absolutely Know that a single byte of ciphertext cannot possibly resolve two bytes of random input [7]. The cost , of course , is the need for twice as many random values (which may be available anyway , when using a wide Additive RNG), plus another indirect access and exclusive - OR, the cost may be just the extra indirect access and exclusive - OR .

As these two hardening methods show , a wide variety of simple manipulations can be used **to** strengthen the fundamental combining operation . These methods can be used in addition to the ability to use a wide range of more extensive permutations far beyond just one or two swap operations . And all of these Dynamic Substitution flavors can be used in a sequence of combinings **or** a selection among multiple combinings .

Summary

An effective cryptographic system should be seen as a system designed **to** protect the inevitable weaknesses in each of the components. For example , the fact that exclusive-OR is trivially weak does not prevent it from being used to good advantage as a balanced mixer in many ciphering designs . But a stronger component needs less protection .

Perhaps the biggest advantages of Dynamic Substitution follow from a construction based on substantial internal state . It is the Dynamic Substitution internal state which supports the use of (nonlinear) combinings in sequence . In contrast , multiple combining rarely improves strength when using (linear) exclusive - OR . It is the Dynamic Substitution internal state which also makes each combiner different , and thus supports the use of a selection from among several combiners . And , of course , there are no " different " exclusive - OR combiners . The cost is a need to store and initialize this state information .

By making use of the Dynamic Substitution internal state , a cipher can remain secure even if weakness is later found in the confusion generator subsystem . Indeed , a simpler and faster RNG can be used from the start. These are significant advantages .

References

1. **R-C** Tausworthe "Random numbers generated by linear mod. 2" Math. Computation vol. 19 pp. 201-209 -1965
2. E- J Groth "Generation of binary sequences with controllable complexity" IEEE Trans. Infrom. Theory vol. IT-17 pp. 288-296 May 1961
3. **E.L- Key** "Ananalysis of the structure and complexity of non linerar binary sequence generators "IEEE Trans. Inform. Theory vol. IT- 22 pp. 732-736. Nov. 1976
4. Blum and micali, S. "How to generate crypttographically strong sequences of pseudo- Random Bits "SIAMJ. complexity vol. 13 no.4pp.850-864.Nov. 1984
5. Siegenthaler T. "Decrypting a class of stream complexity Using Ciphertext only "IEEE Trans. Computers C-34 1985
6. Brian Beckett "introduction to cryptography and PC security" Published by me Graw-Hill publishing company 1997.
7. Jan **C.A** VAN DER LUBBE "Basic methods of cryptography "CAMBRIDGE university press 1998.

الخلاصة

نقطة الضعف الرئيسية في التحفيز الانسيابي هي بوابة (EX-Or) التجميعية التي تستخدم عادة في توليد المفتاح في هذه الأنواع من التجفير. لمعالجة هذا الضعف في هذا النوع من التجفير فقد استخدمنا " التعويض الديناميكي " ليحل محل بوابة (EX- OR) التجميعية. يقبر التعويض الديناميكي عملية لا خطية والتي تتطلب جهداً من قبل المسترق لكسر هذا النوع من التجفير.

عملياً يتكون هذا النوع من عدة مراحل من المجمعات اللاخطية والتي تستخدم عدداً منها في توليد المفتاح .كنتيجة ان هذه الطريقة سوف تقوي عملية التجفير من ناحية السرية وتكون في توليد المفتاح. وكتقوية إضافية في متانة هذا النوع من التجفير فقد أضفنا عدة مراحل من عملية التعويض الديناميكي.