Dynamic Feedback Arrangement Scrambling Technique

David Scott Brown
General Instrument Corporation, VideoCipher Division

Abstract

The Dynamic Feedback Arrangement Scrambling Technique (DFAST) is a method of generating keystream for use in scrambling binary data to prevent unauthorized listeners from recovering that data. The algorithm was initially developed and modeled using the "C" programming language, and then implemented in discrete digital hardware, assembly language code, and several custom integrated circuits. A United States patent was granted in August 1989 for the DFAST Keystream Generator [1]. Two VideoCipher® systems using this technique have received export licenses from the State Department.

INTRODUCTION

Environment

The keystream generated by this technique is added modulo-2 to the binary data on a bit-by-bit basis, creating an encrypted data stream. This encrypted data stream is recoverable if an identical keystream, generated synchronously by an identical keystream generator that begins with an identical initialization number (the key), is added modulo-2 to it. DFAST falls in the general category of "stream ciphers" [4].

Design Constraints

DFAST was developed in order to provide a scrambling technique usable in exportable products. All previous VideoCipher® systems utilized the Data Encryption Standard (DES) [2,3], which is not approved for export by the United States Department of State. This resulted in three major design constraints:

1. The algorithm must be secure against real-time attacks.

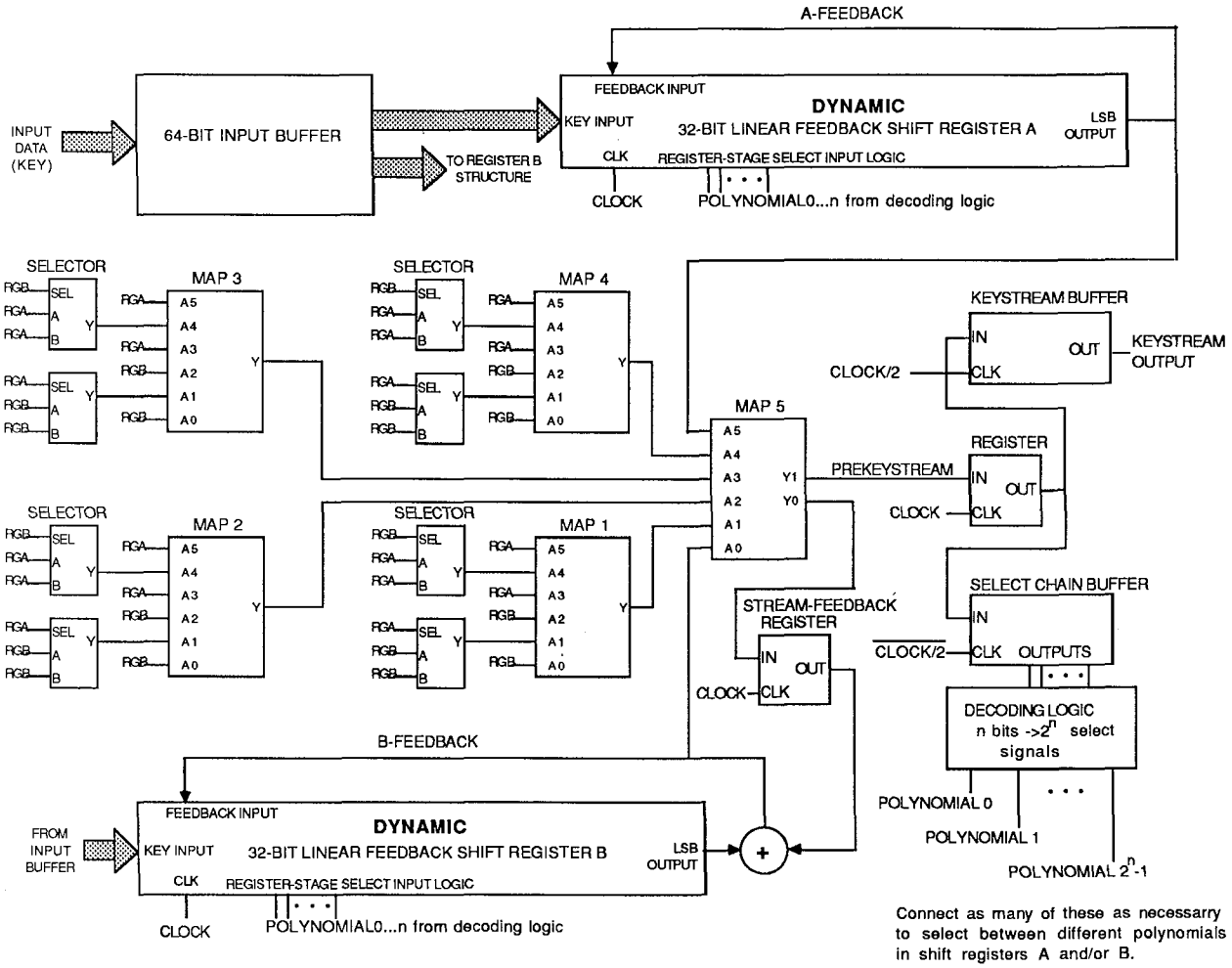
2. It must be readily implementable at high speeds in both hardware and firmware (assembly language).

3. It must be readily implementable in a high-level language so that an export license can be obtained.

GENERAL STRUCTURE

The general form of DFAST is shown in figure 1. There are two 32-bit feedback shift register structures, labeled A and B. Either or both may be a dynamic structure capable of implementing any desired number of polynomials. It is recommended that one of the structures be static and that it implement a primitive, irreducible polynomial of degree 32 to generate a maximal-length binary sequence of length 2^{32}-1. This insures that the length of the keystream output sequence will also be at least 2^{32}-1 before repeating [5].

Twenty out of the 32 register outputs from each of these structures (40 outputs total) are tapped as inputs to the selectors and mapping functions. Each tap is unique; that is, it only connects to one point in the Selector-Map structure. Further rules concerning the connection of the shift register taps are discussed in subsequent sections.



NOTES:

1. The selector places the A or B input on the output Y based on the value of the select (SEL) input (1=A, 0=B).
2. The select input of each selector (SEL) always comes from the register opposite to the inputs A and B.
3. RGA and RGB are all outputs from unique stages of the A and B registers, respectively.
4. There may be any number of different degree-32 polynomials implemented in the A and/or B registers; the select chain buffer will expand to provide enough select bits.
5. The rules for connecting the taps from the A and B register stages to the Selectors and Map functions appear on subsequent pages.
6. The rules for determining the mapping functions (MAP1-5) appear on subsequent pages.
7. Only one of the feedback shift register structures MUST be dynamic, although both may be.

Figure 1

There are five mapping functions, designated MAP1 through MAP5. MAP1 through MAP4 produce one output bit each, while MAP5 produces two output bits at a time. Each MAP function is preceded by two selectors, one of which selects the A4 input to the MAP and the other the A1 input.

There is a holding register at the Y1 output of the MAP5 function, for the purpose of splitting up the PREKEYSTREAM bits. The structure shown here places alternate bits of PREKEYSTREAM into the Keystream Buffer, with the remaining bits stored in the Select Chain Buffer. Many other subdividing schemes are possible; the idea here is to avoid placing contiguous PREKEYSTREAM output bits into the KEYSTREAM. The Select Chain Buffer stores as many successive alternate bits from the PREKEYSTREAM output as necessary to select between the different polynomials available in the

Dynamic Register(s).

There is also a holding register at the Y0 output of the MAP5 function for the purpose of storing the STREAM-FEEDBACK bits. This pseudo-random bitstream is exclusive-OR'ed with the LSB output of the B-register structure and the result is used as both the feedback bit of the B-register structure and as the A0 address input of the MAP5 function. This causes further randomness in the sequence of states that appears in the B-register structure, making that sequence even more difficult to predict. The holding register is necessary to avoid an unstable feedback path around the MAP5 function.

Finally, there is a block of Decoding Logic that uses the stored Select Chain Buffer bits to decode a particular polynomial, setting a single signal corresponding to a particular polynomial true while keeping all other polynomial signals false.

COMPONENT DESCRIPTIONS

Static Feedback Shift Register Structure

If a Static Feedback Shift Register Structure is used and it implements a primitive, irreducible polynomial, it will generate a maximal length sequence of length 2^{32}-1. This will insure that no matter what polynomial is used in the dynamic register, the overall keystream generator for DFAST will have a sequence length of at least 2^{32}-1 before repeating. Figure 2 shows a more detailed view of a Static Structure.

Selector Structure

There are two Selectors in front of each Mapping function, selecting the middle input to that Map from each register structure. This provides a means to use additional taps from the shift register structures into the mapping functions and provides greater nonlinearity in the overall scrambling process. Each selector chooses between its two data inputs, labeled A and B, based on the value of its SEL (select) input. If SEL is a logic TRUE (1), the A input is selected; if SEL is FALSE (0), the B input is selected.

Select Chain Buffer/Decoding Logic Structure

The Select Chain Buffer in simply a serial shift register (with no feedback) which must be long enough to provide the capability to select between the different polynomials used in the Dynamic Register structure(s). The general formula is:

Size of chain = highest n such that

2^{n-1} < number of polynomials
and 2^{n} > number of polynomials

For example, if 7 different polynomials were desired, one would use a Select Chain Buffer of length 3 since 2^3=8 which is greater that 7 while 2^2=4 is less than 7. Figure 2 shows the structure.

Structure of Register Select Logic

The Register Select Logic is simply a logical-OR function. For any given Register-Stage within a Dynamic Register structure, selection of the feedback to the selector for that stage is accomplished according to the following formula:

Select = TRUE if that register stage
is part of the polynomial

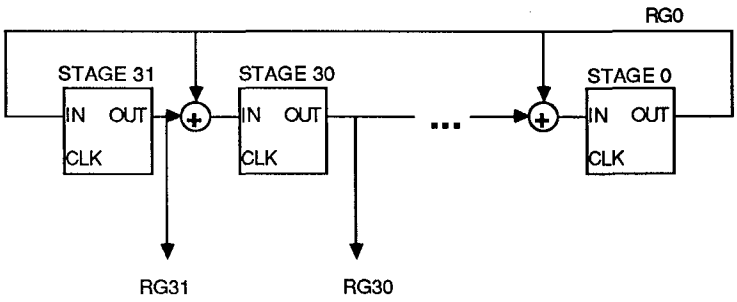
FALSE if not.

So each SEL signal is a logical OR of however many POLYNOMIAL signals that register stage is involved in.

Dynamic Feedback Shift Register Structure

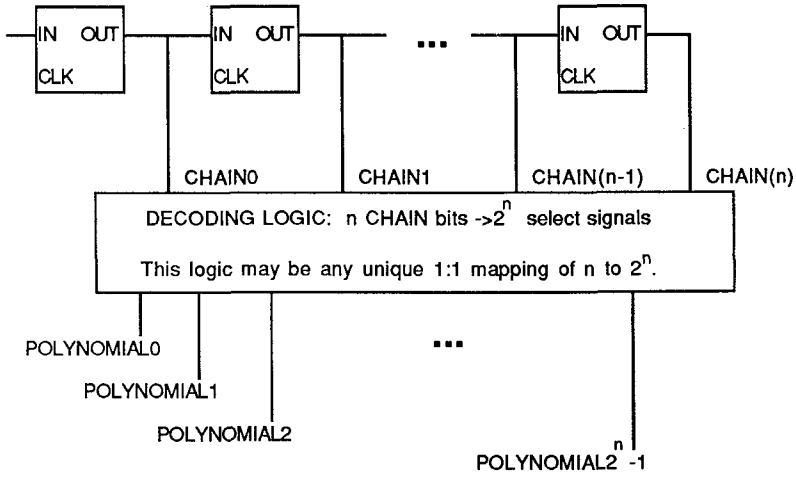
The structure of a Dynamic Feedback Shift

STRUCTURE OF STATIC FEEDBACK SHIFT REGISTER:
(ALL CLOCKS ARE COMMON AT SYSTEM CLOCK RATE)

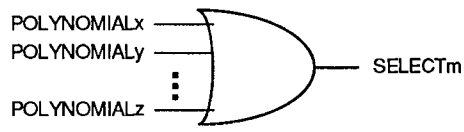


THIS EXAMPLE SHOWS MODULO-2 SUMS (EXCLUSIVE-ORs) AT INPUTS TO 31,30 AND 0. THE SUM AT THE INPUT TO 31 IS IMPLICIT IN THIS STRUCTURE.

STRUCTURE OF SELECT CHAIN BUFFER AND DECODING LOGIC:
(ALL CLOCKS ARE COMMON AT 1/2 THE SYSTEM CLOCK RATE)



STRUCTURE OF REGISTER-STAGE-SELECT INPUT LOGIC:

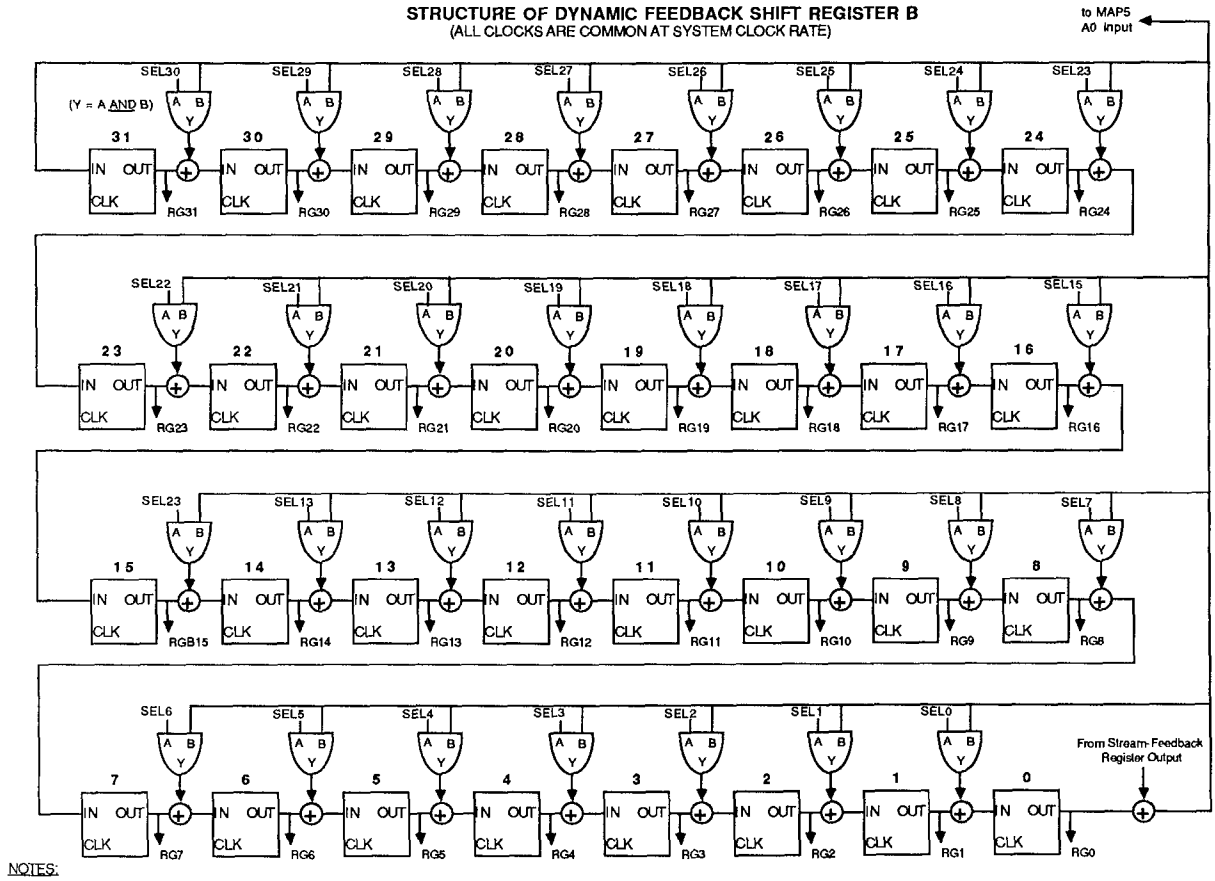


NOTES:

8. The SELECT signal for a given Register-Stage inside the dynamic B-register is the logical-OR combination of all POLYNOMIAL signals that include that given register.

Figure 2

Register is shown in figure 3.



NOTES:

9. The select signals, SEL31...0, are logical 1 (true) if that particular register stage is in the selected polynomial, and that polynomial is selected by the decoding logic. This allows the feedback bit through the logical AND gate to the modulo-2 adder. If a select signal is false, the logical-AND output is also false, and the exclusive-OR is effectively removed from the input to that Register-Stage. Each individual signal is of the form: SELn = POLYNOMIAL0 + POLYNOMIAL1 + ... POLYNOMIALm; where n=the register-stage number,m is the number of different polynomials used, + is logical OR, and only polynomials that the particular register-stage is a part of are included in the sum. The POLYNOMIAL signals are generated by decoding bits from the Select Chain Buffer.

Figure 3

This structure can implement any number of different polynomials of degree 32. The most-significant Register-Stage is numbered 31 and the least is numbered 0. The feedback signal is either:

- The output of the 0-register, or
- The output of the 0-register exclusive-OR'ed with the "STREAM-FEEDBACK" output of MAP5. In figure 1, this is the option chosen for the Dynamic Structure, which is register-structure B.

This feedback signal is connected to the input of the Register-Stage 31 regardless of

which polynomial is used (in other words, all polynomials have a 31 term in them, in order to be of degree 32).

Between each pair of Register-Stages is an exclusive-OR (modulo-2 addition) function, with inputs from the preceding Register-Stage and a selector gate and an output to the next Register-Stage. The selector gate (a logical AND) provides one of the following to the exclusive-OR:

1. A logical FALSE (0) if the SEL signal for that stage is FALSE (0).

2. The feedback term if the SEL signal for that stage is TRUE (1).

When the FALSE is selected (SEL=0), the exclusive-OR function is effectively disabled, and the term corresponding to the following Register-Stage is removed from the polynomial. When the feedback term is selected, the following Register-Stage is part of the polynomial; thus by connecting the SEL inputs as the output of a logical-OR of all desired polynomials, that Register-Stage is included in those polynomials.

SUMMARY OF DFAST STRUCTURE RULES

At least one of the Feedback Shift Register Structures must be **Dynamic**, implementing any number of degree 32 polynomials, with a modulo-2 sum capability between each of the 32 Register-Stages. The sums are invoked depending on the polynomial selected, and this selection takes place on a clock-by-clock basis.

No tap (individual register output) from either the A-Register Structure or the B-Register Structure is used more than once as a Selector or Map function input.

The Selectors are arbitrarily arranged as inputs to the Maps. The Selector Select inputs, however, must be taps from the Register Structure opposite to that supplying the Selector Data Inputs. The number of Selectors used is only limited by the above rule restricting taps to be used once.

Any two inputs to a particular Map function must have at least one exclusive-OR function between them in their source Register-Structure. This limits the polynomial combinations that may be chosen for implementation by that Register-Structure. This includes taps that enter the Map function through a selector.

The second-level Map function must use all of the first-level Map outputs as inputs as well as the feedback signals from both the A and B register structures.

FUNCTIONAL DESCRIPTION

The operation of DFAST is as follows. A 64-bit initial value, called a key, is loaded into the two feedback shift registers. The registers then shift with every subsequent cycle of CLOCK, and in fact the entire machine is synchronous to this CLOCK. During each clock cycle, or state of the machine, an individual PREKEYSTREAM bit is computed. Various points in each feedback shift register structure are tapped into the Selectors and MAPs as shown in figure 1. Each Selector selects its output between its two data inputs based on the value of its SEL input, and is wired so that the SEL signal comes from the feedback register *opposite* to that of the A and B data signals. Note further that no tap is used in more than one place.

Each MAP function is a pseudo-random distribution of logic 0 and logic 1 values, selected according to the previously defined rules. The outputs of the first-level MAPs (MAP1 through MAP4) are used as the middle four inputs to the second-level MAP. The top and bottom inputs to the second-level map come from the feedback bits of the A and B register structures, respectively. The outputs from this second-level MAP are the PREKEYSTREAM and the STREAM-FEEDBACK, each of which are generated every CLOCK cycle.

The STREAM-FEEDBACK is modulo-2 summed (exclusive-or'ed) with the 0-register output (LSB) of the Dynamic B-Register Structure (RGB0) to produce the B-FEEDBACK signal. This puts further uncertainty in the pattern that the B-Register Structure will produce. The PREKEYSTREAM, which is the result of all these computations, is then subdivided to produce actual KEYSTREAM for use in encryption and to produce the contents of the Select Chain Buffer to select between the various A and B Polynomials.

Conclusions

DFAST is a hardware-efficient, cryptographically strong keystream generation algorithm that has received export license approval. By fixing one of the two feedback shift register structure polynomials, it is possible to guarantee a minimum cycle length for the keystream without repetition. After this, the key may be changed to maintain nonrepeatability. The general structure allows a large number of possible specific implementations (varying the polynomials, the taps, etc.), each of which will meet the previously outlined design constraints while remaining distinct from each other. This allows production of deliberately incompatible systems with only minor changes to the hardware or software implementations.

References

1. "Dynamic Feedback Arrangement Scrambling Technique Keystream Generator", David Scott Brown, U.S. Patent Number 4,860,353, issued August 22, 1989.

2. "Data Encryption Standard", Federal Information Processing Standards Publication 46, U.S. Department of Commerce, January 1977.

3. "DES Modes of Operation", Federal Information Processing Standards Publication 81, U.S. Department of Commerce, December 1980.

4. "Cryptography - A Primer", Alan G. Konheim, John Wiley and Sons, New York, 1981.

5. "Error Correcting Codes", W.W. Peterson and E.J. Weldon, Jr., MIT Press, Cambridge, MA, 1972.