جامـــــعة المـــــستقبل
AL MUSTAQBAL UNIVERSITY

قســـم الامـــــن الـــــــــسيبرانـــــــي

Department of Cyber Security

Subject:

FEAL

Class:

Second

Lecturer:

Asst. lect. Mustafa Ameer Awadh

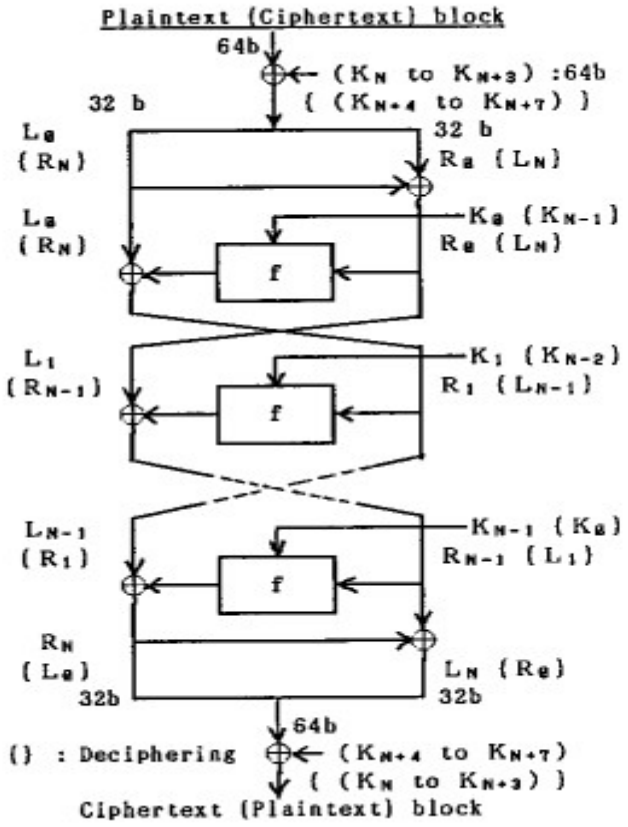
Lecture: (7)

FEAL

FEAL was designed by Akihiro Shimizu and Shoji Miyaguchi from NTT Japan. It uses a 64-bit block and a 64-bit key. The idea was to make a DES-like algorithm with a stronger round function. Needing fewer rounds, the algorithm would run faster. Unfortunately, reality fell far short of the design goals.

Description of FEAL

In the following Figure is a block diagram of one round of FEAL. The encryption process starts with a 64-bit block of plaintext. First, the data block is XORed with 64 key bits. The data block is then split into a left half and a right half. The left half is XORed with the right half to form a new right half. The left and new right halves go through *n* rounds (four, initially). In each round the right half is combined with 16 bits of key material (using function f) and XORed with the left half to form the new right half. The original right half (before the round) forms the new left half. After *n* rounds (remember not to switch the left and right halves after the *n*th round) the left half is again XORed with the right half to form a new right half, and then the left and right halves are concatenated together to form a 64bit whole. The data block is XORed with another 64 bits of key material, and the algorithm terminates.



One round of FEAL.

Function f takes the 32 bits of data and 16 bits of key material and mixes them together. First the data block is broken up into 8-bit chunks, then the chunks are XORed and substituted with each other.

Figure 13.4 is a block diagram of function f. The two functions S_0 and S_1, are defined as:

$S_0(a,b)$ = rotate left two bits $((a + b) \bmod 256)$

$S_1(a,b)$ = rotate left two bits $((a + b + 1) \bmod 256)$

The same algorithm can be used for decryption. The only difference is: When decrypting, the key material must be used in the reverse order.

In the following Figure is a block diagram of the key-generating function. First the 64-bit key is divided into two halves. The halves are XORed and operated on by function f_k, as indicated in the diagram. Figure (f_k) is a block diagram of function f_k. The two 32-bit inputs are broken up into 8-bit blocks and combined and substituted as shown. S_0 and S_1 are defined as just shown. The 16-bit key blocks are then used in the encryption/decryption algorithm.

On a 10 megahertz 80286 microprocessor, an assembly-language implementation of FEAL-32 can encrypt data at a speed of 220 kilobits per second. FEAL-64 can encrypt data at a speed of 120 kilobits per second.

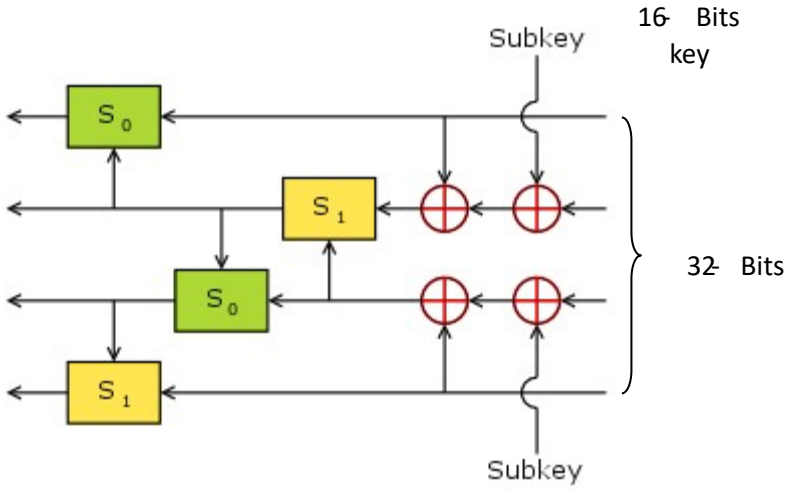


Figure : *Function f.*

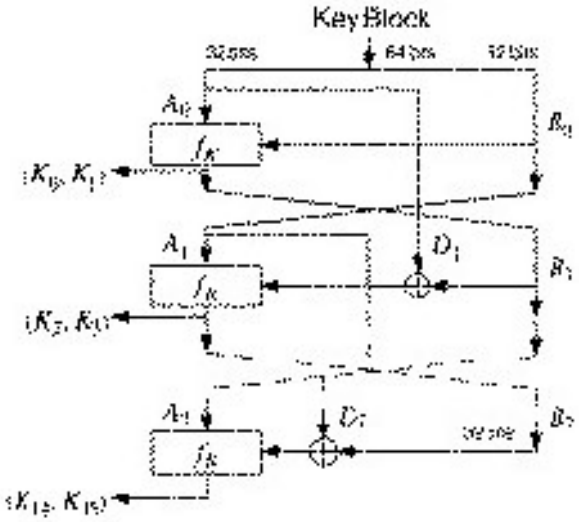


Figure :*Key processing part of FEAL.*

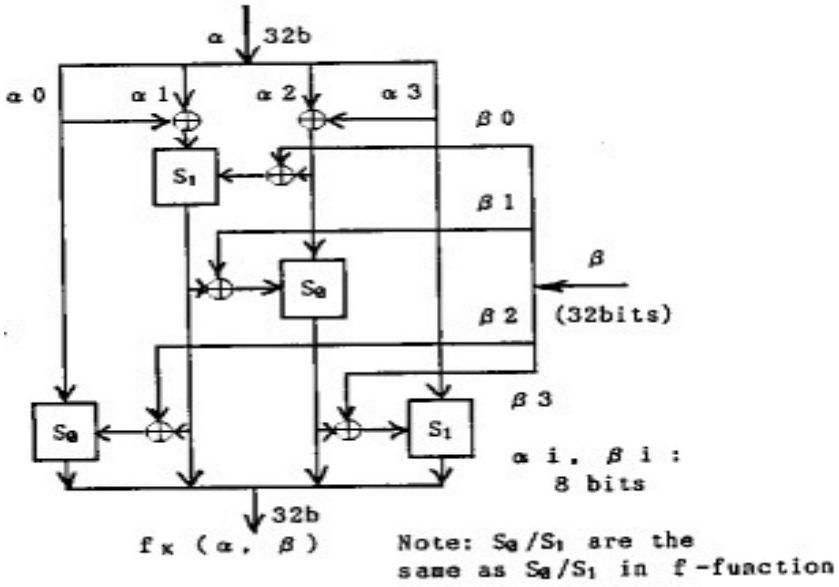


Figure :*Function f_K.*

Cryptanalysis of FEAL

FEAL-4, FEAL with four rounds, was successfully cryptanalyzed with a chosen-plaintext attack in [201] and later demolished. This later attack, by Sean

Murphy, was the first published differentialcryptanalysis attack and required only 20 chosen plaintexts. The designers retaliated with 8-round FEAL [1436,1437,1108] which Biham and Shamir cryptanalyzed at the SECURICOM '89 conference[1427]. Another chosen-plaintext attack, using only 10,000 blocks, against FEAL-8 [610] forced the designers to throw up their hands and define FEAL-*N* , with a variable number of rounds (greater than 8, of course).

Biham and Shamir used differential cryptanalysis against FEAL-*N*; they could break it more quickly than by brute force (with fever than 2^{64} chosen plaintext encryptions) for *N* less than 32 . FEAL-16 required 2^{28} chosen plaintexts or $2^{46.5}$ known plaintexts to break. FEAL-8 required 2000 chosen plaintexts or $2^{37.5}$ known plaintexts to break. FEAL-4 could be broken with just eight carefully selected chosen plaintexts.

The FEAL designers also defined FEAL-*NX*, a modification of FEAL, that accepts 128-bit keys. Biham and Shamir showed that FEAL-*NX* with a 128-bit key is just as easy to break as FEAL-*N* with a 64-bit key, for any value of *N*. Recently FEAL-*N*(X)S has been proposed, which strengthens FEAL with a dynamic swapping function.

There's more. Another attack against FEAL-4, requiring only 1000 known plaintexts, and against FEAºL-8, requiring only 20,000 known plaintexts. The best attack is by Mitsuru Matsui and Atshuiro Yamagishi . This is the first use of linear cryptanalysis, and can break FEAL-4 with 5 known plaintexts, FEAL-6 with 100 known plaintexts and FEAL-8 with 2^{15} known plaintexts. Differentiallinear cryptanalysis can break FEAL-8 with only 12 chosen plaintexts . Whenever

someone discovers a new cryptanalytic attack, he always seems to try it out on FEAL first.