Security Enhancement Algorithms for Data Transmission for Next Generation Networks

Vikas Kaul
TCET, Mumbai

S K Narayankhedkar
SIGCOE. Mumbai

S Achrekar, S Agarwal, P Goyal, TCET,Mumbai

ABSTRACT

As the speed of processors and knowledge about cryptanalysis goes on increasing day by day, the original encryption methods prove to be insufficient for better security. As large amount of data is transmitted over the network, it is preliminary to secure all types of data before sending it.. The problem with AES, most extensively used encryption, is that it uses many multivariate equations which are linear in nature. Thus it can be broken using algebraic cryptanalysis. This provides a serious threat as AES was considered to be unbreakable and thus it was used in many encryption systems.

The paper presents the design and implementation of a hybrid based 128 bit key AES-DES algorithm as a security enhancement.

Keywords

SBox, XOR, Cipher, Key matrix, Hybrid AES DES, Chaos, Feistal network.

INTRODUCTION

We have chosen integrating AES with Feistal network of DES so as to enhance security. The paper outlines the possible weaknesses within the current AES encryption algorithm especially against algebraic based cryptanalysis. Understanding the need to minimize algebraic attacks into the AES, it proposes the idea on integrating AES within the Feistal network of DES, hence resulting into the development of the Hybrid AES-DES algorithm. The levels of Feistal network as applied into the Hybrid AES-DES structure are retrospective to Quality of Service (QoS) needs.

The main idea of using a hybrid approach is that we combine two algorithms in such a way that the disadvantages of both of them are reduced. Hence we get a system which is much difficult to break by cryptanalysis. Thus the hybrid system will take a 256 bit data and 128 bit key.

Cryptanalysis of AES

AES is based on a design principle known as a Substitution permutation network. It is fast in both software and hardware.Unlike its predecessor DES, AES does not use a Feistel network.

AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The blocksize has a maximum of 256 bits, but the keysize has no theoretical maximum.

AES operates on a 4×4 column-major order matrix of bytes, termed the *state* (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field..The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

AES Implementation:-

A byte in Rijndael is a group of 8 bits and is the basic data unit for all cipher operations. Such bytes are interpreted as finite field elements using polynomial representation, where a byte b with bits b0, b1,...b7 represents the finite field elements. The conversion from plain text to State Matrix is shown in Figure 1.As discussed, bytes are can be represented as polynomials. Finite field operations like addition and multiplication are required for key scheduling and rounding. The addition of two finite field elements is achieved by adding the coefficients for corresponding powers in their polynomial representations, this addition being performed in GF(2), that is, modulo 2, so that 1+ 1 = 0. Addition is nothing but performing XOR between two expressions .Finite field multiplication is more difficult than addition and is achieved by multiplying the polynomials for the two elements concerned and collecting like powers of x in the result. Since each polynomial can have powers of x up to 7, the result can have powers of x up to 14 and will no longer fit within a single byte. This is overcome by using a field generator. The product is divided by this field generator and the remainder is taken asthe result. Since there are 256 possible polynomials, a look uptable can be created for a specific field generator. So the lookup table will have 256 * 256 entries .scheduling:

The round keys are derived from the cipherkey by means of a key schedule with each round requiring Ncwords of key data. For 128-bits keys the key scheduling operates intrinsically on blocks of 4 32-bits words; we can calculate one new round key from the previous one,denote the ithword of the actual round key with K[i], where 0¡i¡4, and the ithword of the next round key with K[i]. K[0] is computed by anXOR between K[0], a constant r (field generator) and K[3], the latter being

pre rotated and transformed. The other three words K[1], K[2] and K[3] are calculated as K[i] = K[i] . K[i - 1].

Rounding: At the start of the cipher the cipher input is copied into the internal state using the conventions described before(Figure 1). An initial round key is then added and the state isthen transformed by iterating a round function in a number of cycles.The steps involvedin AES are shown in Illustration1. It is the algorithm as rinjdael suggested.

Round Function: One round is termed as a cycle and each cycle has four steps. Each step of transformation is described below. Add Round Key: This is the first step of transformation. The Xor Round Key function declared in Illustration 1 has to perform a bitwise xor of the state matrix and the round key matrix.

Sub-Bytes :Transformation: Inverse of the state matrix is found here and affine transformation is performed.

Shift Rows: The ShiftRows transformation operates individually on each of the last three rows of the state matrix by cyclically shifting the bytes in the row. The second row is shift done time to the left, third row shifted two times and fourth rows shifted three times.

Mix Columns: The mix columns transformation computes the new state matrix S by left-multiplying the current state matrix S by the polynomial matrix P:S = P o S where P is a fixedmatrix as shown below.

The Advanced Encryption Standard (AES), the block cipher ratified as a standard by National Institute of Standards and Technology of the United States (NIST), was chosen using a process markedly more open and transparent than its predecessor, the aging Data Encryption Standard (DES). This process won plaudits from the open cryptographic community, and helped to increase confidence in the security of the winning algorithm from those who were suspicious of backdoors in the predecessor, DES.

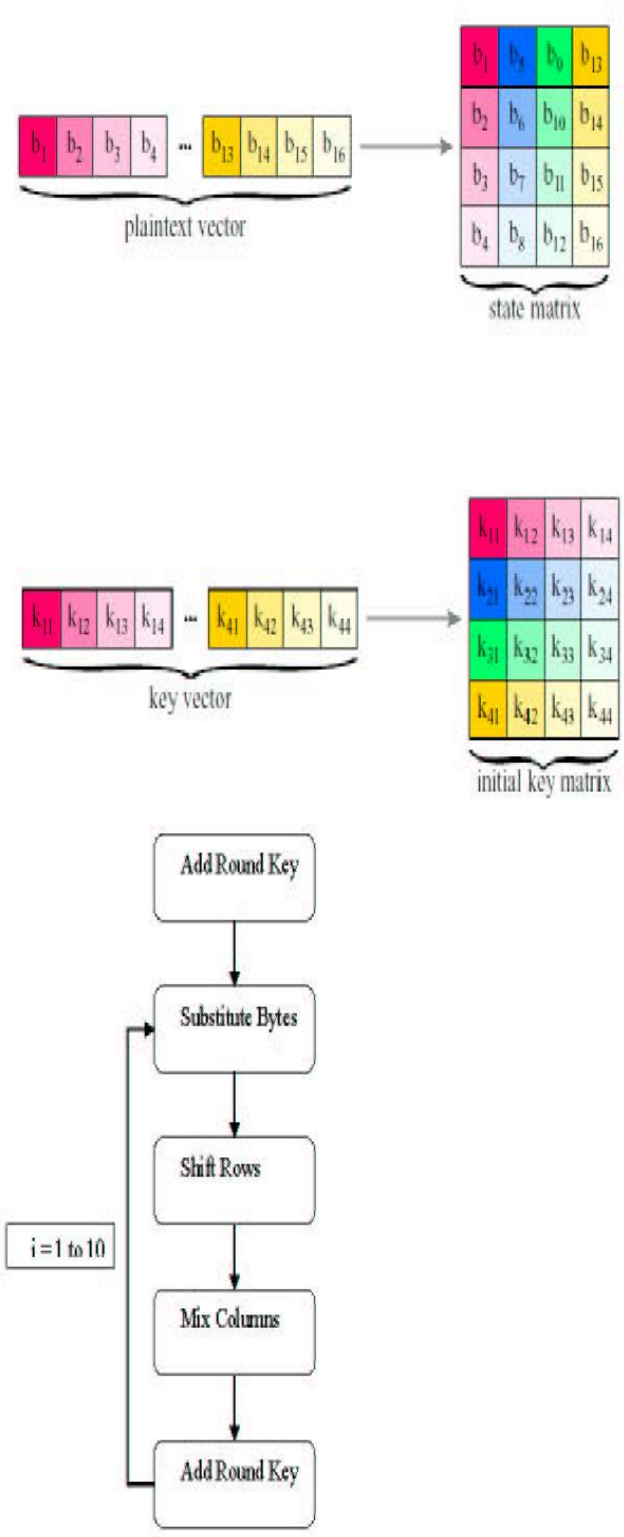

Figure 1 (a) major functions of aes

2 . DES

DES is the block cipher — an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bitstring of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits, and it is never quoted as such. Every 8th bit of the selected key is discarded, that is, positions 8, 16, 24, 32, 40, 48, 56, 64 are removed from the 64 bit key leaving behind only the 56 bit key.

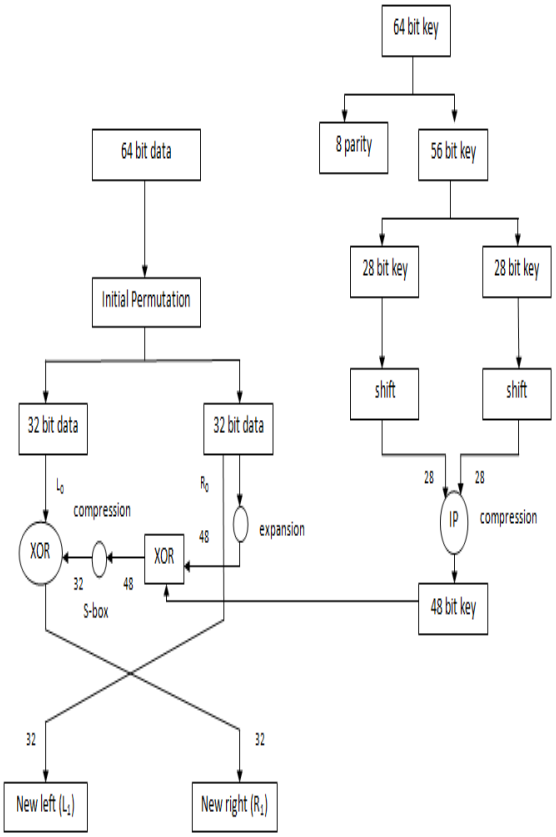

Fig 1(b) One round of DES

3. Concept of Hybrid AES DES

Mathematically, the idea of a hybrid based AES-DES can be construed with reference to basic DES Feistel equations. The repetition of these equations is based on the number of rounds as adapted by the Feistel network, which in the case of DES was standardized at 16. However, by incorporating the AES within this yield the following results.

$L_n = R_{n-1}$ ---------- (1)

$R_n = AES\ (L_n \oplus f(R_{n-1}, K_n))$ ---------- (2)

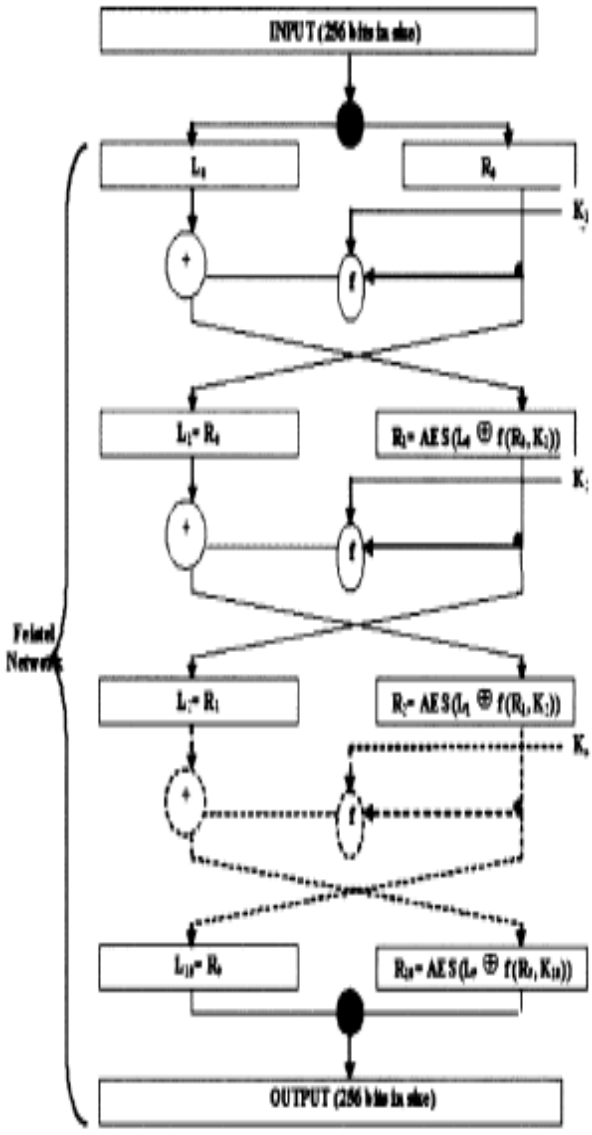

Fig 1:(c) Hybrid AES DES flow structure

From Equation (2), each R_{n-1} and K_n is channeled into the round function, which basically revolves on a XOR function between these variables. The output of the round function is then XORed with n- before being channeled as input data for the AES algorithm. The result from the AES process represents R_n. The AES operations include the byte substitution, shift row, mix columns and add round key operations.

Equations (1) and (2) are then iterated over a period of rounds, retrospective to the number of keys as generated from the key schedule process. The key schedule process for the hybrid system is directly adapted from the AES standard and the round keys for the AES process are directly adapted from the expanded keys. The scheme of using a common set of keys for the round functions within the Feistel network and the AES functions as called during the operation reduces the computational complexity in creating multiple sets of keys for each overall hybrid operation. However, the mathematical modeling for the key expansion of the AES limits its generation at 10 keys based
on the number or rounds as applied for a 128 bit based AES encryption process. Hence, the proposed number of iterations for the hybrid system is set at 10. Nonetheless, the iterations also vary based on the level of security implementation for the Hybrid AES-DES. Note as seen from Equation (2), with every round of iteration, the AES function is performed with different sets of generated results based on varying sets of expanded keys.

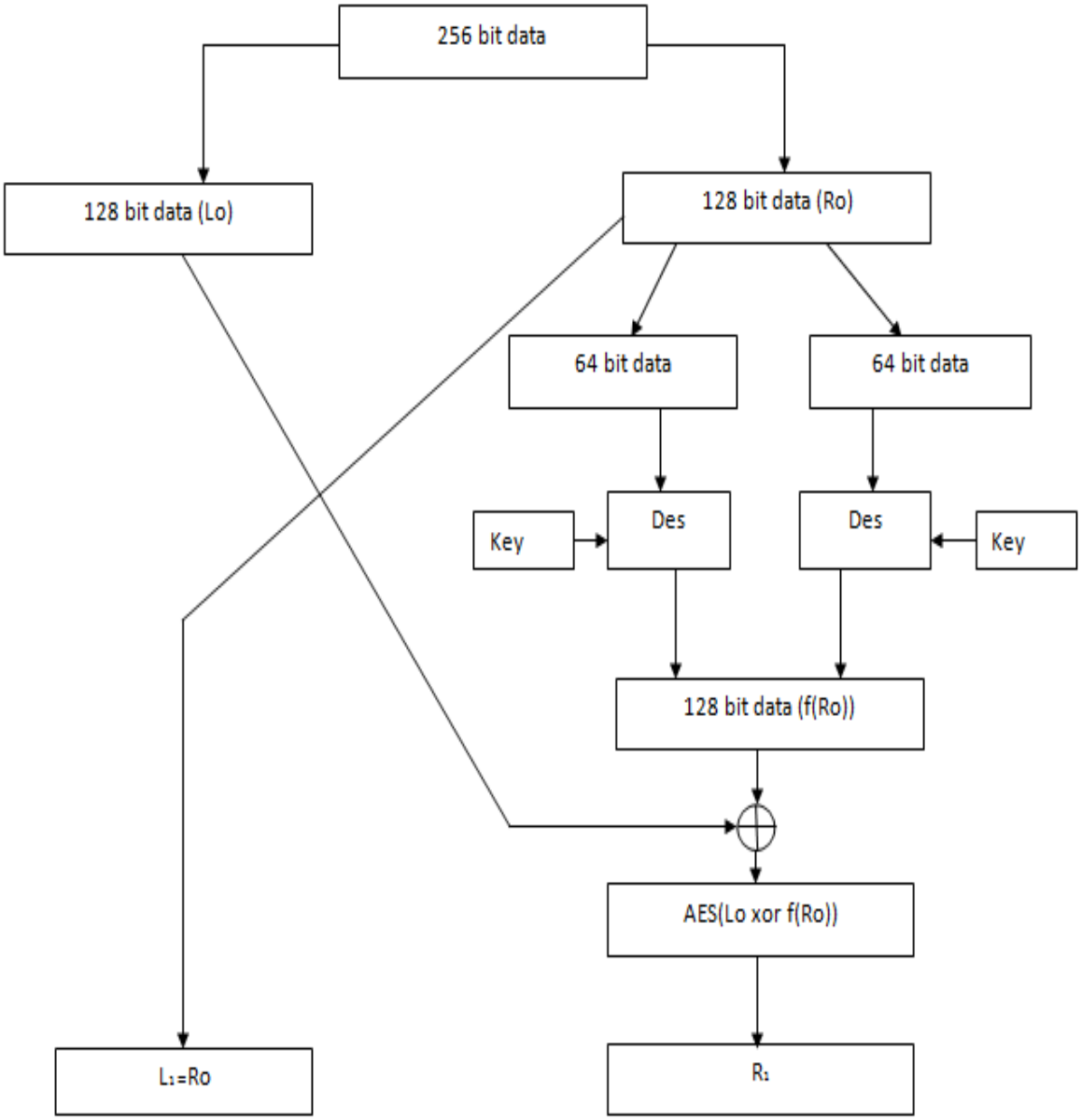

Fig 3: Detailed flow of Hybrid model

As shown in the figure, we use AES within the Feistal network and while doing so, increase the complexity of the algorithm. This helps in securing the data in an enhanced way.

4. Result

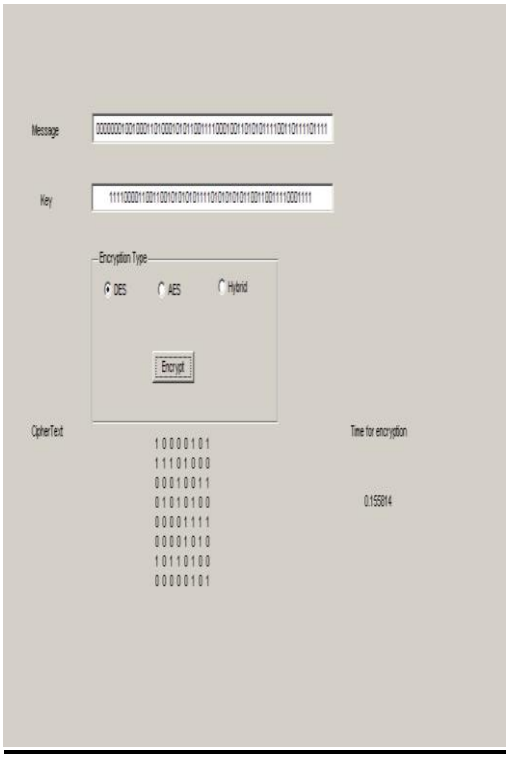

Fig. 4.1: DES output

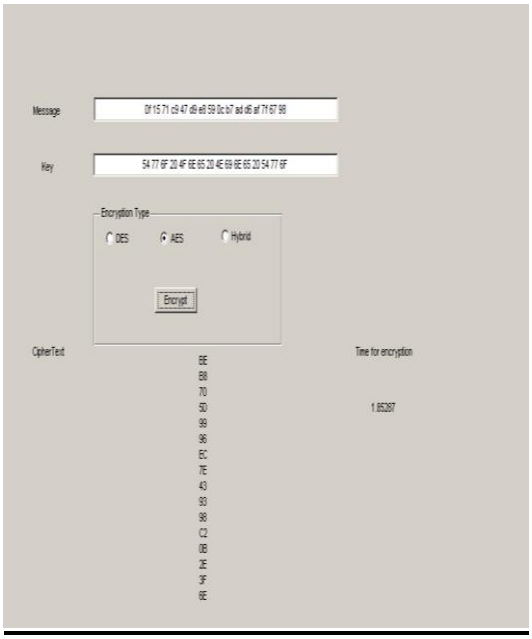

Fig. 4.2: AES output

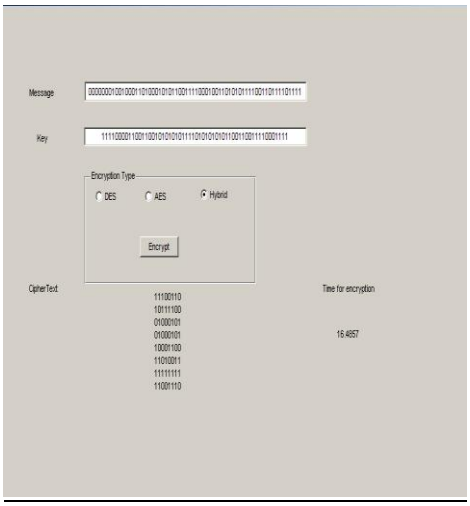

Fig. 4.3: Hybrid Model output

Table 1: Comparison of encryption time

Encryption technique	Encryption time(secs)
DES	0.15581
AES	1.85287
Hybrid AES DES	16.4857

5. CONCLUSION

Thus the paper uses combined concept of AES and DES to obtain a hybrid model which can be used for encrypting various kinds of data.

Nowadays it is very important to design strong encryption algorithms as the power of computers is growing day by day. Thus the hybrid model gives a better non linearity to the plain AES and as it is merged with DES, there is better diffusion. Hence the possibility of an algebraic attack on the hybrid model is reduced.

From the table 1, we can see that the encryption time for the hybrid model is much greater than the times for AES and DES. Thus it can be inferred that the hybrid model will take longer time to be broken by cryptanalysis

.

6. FURTHER ENHANCEMENTS

Being a hybrid of two powerful encryption standards the algorithm will act as efficient and reliable encryption technique for data. This hybrid model also is free of algebraic attacks where AES fails as it is more prone to algebraic attacks. The paper gives the performance criteria between three algorithm techniques i.e. DES, AES and HYBRID AES-DES. This makes it easy to compare the efficiency of all the three techniques clearly with respect to time.

We can design a modified AES and/or DES so that we could make the hybrid model even better compared to the present system. The proposed algorithm can also use a double key approach using chaos concept which makes it resistant to linear attacks. In this case, the safety of encryption algorithm can be further improved.

7. REFERENCES

[1] An Improved AES algorithm based on chaos Yuan Kun, Zhang Han, Li Zhaohui

[2] An Overview of Cryptanalysis Research for the Advanced Encryption Standard

[3] An Efficient MPEG Video Encryption Algorithm

[4] H. Nover, "Algebraic Cryptanalysis of AES: Overview", University of Wisconsin, USA, 2005.

[5] S. Murphy, MJ.B Robshaw, "Essential Algebraic Structure within the AES", Advances in Cryptology CRYPTO 2002, Vol. 2442 of Lecture Notes in Computer Science, Springer-Verlag, August 2002

[6] N. Courtois, A. Klimov, 1. Patarin, A. Shamir, "Efficient algorithm for solving overdefmed systems of muhivariate polynomial equations", Proceedings for Eurocrypt 2000, LNCS 1807, pp 392407, Springer-Verlag, 2000.

[7] MATLAB Description, Available:

http://www.mathworks.in/help/

[8] AES Description, Available:

http://people.eku.edu/styere/Encrypt/JS-AES.html

[9] DES Description, Available:

http://orlingrabbe.com/des.htm

[10] AES mix column

http://www.angelfire.com/biz7/atleast/mix_columns.pdf