Encryption algorithm as a composition of function

It has similar properties and structure to DES with much smaller parameters. The reader might find it useful to work through an example by and while following the discussion in this Appendix. The S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used to produce that ciphertext as input and produces the original 8-bit block of plaintext. The encryption algorithm involves five functions: an initial permutation (IP); a complex function labeled fK, which involves both permutation and substitution operations and depends on a key input; a simple permutation function that switches (SW) the two halves of the data; the function fK again; and finally a permutation function that is the inverse of the initial permutation (IP–1). As was mentioned in Chapter 2, the use of multiple stages of permutation and substitution results in a more complex algorithm, which increases the difficulty of cryptanalysis.

The function fK takes as input not only the data passing through the encryption algorithm but also an 8-bit key. The algorithm could have been designed to work with a 16-bit key, consisting of two 8-bit subkeys, one used for each occurrence of fK. Alternatively, a single 8-bit key could have been used, with the same key used twice in the algorithm. A compromise is to use a 10-bit key from which two 8-bit subkeys are generated, as depicted in Figure C. 1. In this case, the key is first subjected to a permutation (P10). Then a shift operation is performed.

The output of the shift operation then passes through a permutation function that produces an 8-bit output (P8) for the first subkey (K1). The output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey (K 2). We can concisely express the encryption

algorithm as a composition1 of functions: which can also be written as: IP-1 o fK2 o SW o fK1 o IP (((ciphertext = IP-1 fK 2 SW fK1 (IP(plaintext)) where (K1 = P8 Shift (P10(key)) ! (( )))) K2 = P8 Shift Shift(P10(key)))) Decryption is also shown in Figure C. and is essentially the reverse of encryption: (((plaintext = IP-1 fK1 SW fK 2 (IP(ciphertext)) 1))) Definition:! f f and g are two functions, then the function F with the equation y = F(x) = I g[f(x)] is called the composition of f and g and is denoted as F = g of. C-2 8/5/05 We now examine the elements of S-DES in more detail. C. 2 S-DES Key Generation S-DES depends on the use of a 10-bit key shared between the sender and receiver. From this key, two 8-bit subkeys are produced for use in particular stages of the encryption and decryption algorithm. Figure C. 2 depicts the stages followed to produce the subkeys.

First, permute the key in the following fashion. Let the 10-bit key be designated as (k1, k2, k3, k4, k5, k6, k7, k8, k9, k10). Then the permutation P10 is defined as P10(k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) = (k3, k5, k2, k7, k4, k10, k1, k9, k8, k6) P10 can be concisely defined by the display: 3 5 2 7 P10 4 10 1 9 8 6 This table is read from left to right; each position in the table gives the identity of the input bit that produces the output bit in that position. So the first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on.

For example, the key (1010000010) is permuted to (1000001100). Next, perform a circular left shift (LS-1), or rotation, separately on the first five bits and the second five bits. In our example, the result is (00001 11000). Next, we apply P8, which picks out and permutes 8 of the 10 bits according to the following rule: P8 6 3 7 4 8 5 10 9 The result is subkey 1 (K1). In our

example, this yields (10100100) We then go back to the pair of 5-bit strings produced by the two LS-1 functions and perform a circular left shift of 2-bit positions on each string. In our example, the value (00001 11000) becomes (00100 00011).

Finally, P8 is applied again to produce K2. In our example, the result is (01000011). C. 3 S-DES Encryption Figure C. 3 shows the S-DES encryption algorithm in greater detail. As was mentioned, encryption involves the sequential application of five functions. We examine each of these. Initial and Final Permutations The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function: IP 2 6 3 1 4 8 5 7 This retains all 8 bits of the plaintext but mixes them up. At the end of the algorithm, the inverse permutation is used: C-3 8/5/05 1 3 IP–1 57 2 8 6 It is easy to show by example that the second permutation is indeed the reverse of the first; that is, IP–1(IP(X)) = X. The Function fK The most complex component of S-DES is the function fK, which consists of a combination of permutation and substitution functions.