CRYPTOGRAPHY AND NETWORK SECURITY (MC-407)

ASSIGNMENT-2

ANISH SACHDEVA
DTU/2K16/MC/13

Scanned with CamScanner

Q1) Differentiate between:-

a) Block cipher and stream cipher

Key	Block Cipher	Stream Cipher
Definition	Block Cipher is the type of Encryption where the conversion of plain text performed by taking it's block at a time.	On the other hand stream cipher is the type of encryption where the conversion of plain text performed by taking one byte of plain text at a time.
Conversion of Bits	As Block cipher takes blocks at a time so comparatively more bits gets converted as compared to in Stream Ciphers specifically 64 bits or more could get converted at a time.	On other hand in case of stream cipher at most 8 bits could get converted at a time.
Principle	Block cipher uses both Confusion and difusion principles for the conversion required for encryption.	On the other hand stream cipher uses only confusion principle for the conversion.

Scanned with CamScanner

Key	Block Cipher	Stream Cipher
Algorithm	For encryption of plain text block cipher uses Electronic Code Book (ECB) and Cipher Block Chaining (CBC) algorithm.	On other hand Stream Cipher uses CFB (Cipher Feedback) and OFB (Output Feedback) algorithm.
Decryption	As combination of more bits gets encrypted in case of Block Cipher so the reverse encryption or decryption is comparatively complex as compared to that of stream cipher.	On other hand stream cipher uses CFB (Cipher Feedback) and OFB (Output Feedback) algorithm. On the other hand stream ciphers use XOR for the encryption which can be easily reversed to plain text.
Implementation	The main implementation of the block cipher is the Feistel cipher which is used in the DES, 3-DES Algorithm.	On the other hand the main implementation of stream cipher is vernam cipher.
Examples	DES, 3-DES, AES, Feistel Cipher	Caeser Cipher, Vigenere Cipher, ~~Transposition~~ Monoalphabetic Substitution Cipher

Scanned with CamScanner

b) Confusion and Diffusion

Confusion

Confusion means that each binary digit (bit) of ciphertext should depend on several parts of the key, obscuring the connections between the two. The property of confusion hides the relationship between ciphertext and the key. This property makes it difficult to find the key from the ciphertext and if a single bit in the key is changed, the calculation of the values of most or of all the bits in the ciphertext will be affected. Confusion increases the ambiguity of ciphertext and it is used by both block and stream cipher.

Diffusion

Diffusion means that if we change a single bit of plaintext, then (statistically) half of the bits in the ciphertext should change, also called the avalanche effect, and similarly if we change a bit of ciphertext then approximately one half of the plaintext bits should change. Since a bit can have only two states, when they are re-evaluated and changed from one seemingly random position to another, half of the bits will have changed states. The idea of diffusion is to hide the relationship between the ciphertext and plaintext. This will make it hard for an attacker who tries to find out the plaintext and it increases the redundancy of plaintext by

Scanned with CamScanner

spreading it across the rows and columns ; it is achieved through transposition of algorithm and it is used by block ciphers only.

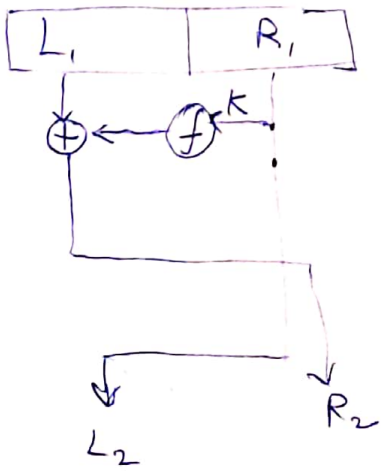
Q.2) Given a Fiestel cipher to encrypt a block of n bits, prove that the number of different reversible mappings for the ideal block cipher is $2^n!$

We know that for a permutation group P_n with n members we have n! permutations that are possible eg.

$$\begin{pmatrix} 1 & 2 & 3 & 4 \ldots & n \\ 1 & 2 & 3 & 4 \ldots & n \end{pmatrix}$$

P_n : Permutation group with n members

We have n! of such permutations when we have n members. In fiestel cipher we have one reversible component and one irreversible function f. We recieve input P. We divide it into 2 parts L_1 and R_1. We send R_1 to left $L_2 = R_1$ and encrypt L_1



Scanned with CamScanner

The number of possible numbers (combinations/values) possible with n bits is the galois field cardinality over n, i.e. 2^n
So we have 2^n possible values with n bits.

Now, we can find the different one-to-one bijective and surjective mappings, by putting them through the permutation group. For 2^n values we will have 2^n permutations.

A reversible mapping is one that is bijective and hence we have $2^n!$ reversible mappings in Ideal block cipher.

Q3) Principles of block cipher Design. Discuss.

A block cipher is designed by considering it's 3 critical aspects which are listed as below:-

1) Number of Rounds
2) Design of function f
3) Key Schedule Algorithm

1) Number of Rounds

The greater the number of rounds, the more difficult it is to perform crypt-analysis, even for a relatively weak F. In general, the criterion should be that the number of rounds is chosen so that known crypto-analytic efforts require greater effort than a simple brute force key attack. The criterion was certainly used in the design of the DES.

Scanned with CamScanner

Schneier [SCHN96] observes that for 16-round DES, a differential cryptanalytic attack is slightly less efficient than brute force; the differential cryptanalytical attack requires $2^{55.1}$ operations whereas brute force requires 2^{55}. If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute force key search.

This criterion is attractive, because it makes easy to judge the strength of an algorithm that satisfies the criterion can be judged solely on key length.

2) Design of Function F

The heart of a Feistel cipher lies in the function f. We specifically look at S-Box design.

The function f provides the element of confusion in a Feistel cipher, thus it must be difficult to unscramble the substitution performed by f. One obvious criterion is that f be nonlinear, as we discussed previously. The more non-linear f, the more difficult it is to approximate f by a set of linear equations, the more non-linear f is.

Several other criteria should be considered in designing f. We would like the algorithm to have good avalanche properties. Recall that, in general, this means that a change in one-bit of the input should produce a change in many bits of the

Scanned with CamScanner

output.

Another criterion proposed is the bit independence criteria (BIC) which states that output bits j and k should change independently when any single input bit i is inverted for all i, j and k. The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function.

3) Key Schedule Algorithm

With any Fiestel block cipher, the key is used to generate one subkey for each round. In general we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key. No general principal for this has yet been promulgated.

Q4) Explain the avalanche Effect in DES.

Avalanche Effect is a desirable Effect. It means that a very small change in the input will lead to a very big change in the output.

A security algorithm that doesn't provide the avalanche effect can lead to an easy statistical analysis. If the change of one bit in the input leads to only one change of one bit of the output, then it's relatively easier to try to guess the input. In DES there are 16 rounds and 2

Scanned with CamScanner

permutations. The real thing happens in the rounds. For each round input bits are used to lookup for the bits that will be used as the output (S-Box).

In each round, we make a shift and the index bits come from the data bits from previous round.

TO observe the avalanche effect let us encrypt 2 plaintext blocks with the same key that differ only in one bit and observe the differences in the number of bits in each round.

Plain text : $(0000\ 0000\ 0000\ 0000)_{16}$

Key : $(2223\ 4512\ 987A\ BB23)_{16}$

Ciphertext : $(4789\ FD47\ 6E82\ A5F1)_{16}$

Plaintext: $(0000\ 0000\ 0000\ 0001)_{16}$ → difference of only 1

Key : $(2223\ 4512\ 987A\ BB23)_{16}$

Ciphertext: ~~$(4789\ FD47\ 6E82\ A5F1)_{16}$~~

$(0A4E\ D5C1\ 5AG3\ FEA3)_{16}$

Even a difference in 1bit causes the entire ciphertext to change.

Scanned with CamScanner

*5) Consider a block Encryption algorithm that encrypts block of length 'n' and let $N = 2^n$. Assume we have 't' plaintext-ciphertext mappings pairs $P_i C_i = E(k, P_i)$ where we assume that the key K selects one of the N! mappings. Imagine that we wish to find key K by exhaustive search. We could generate key $\bar{K}$ and test whether $C_i = E(k', P_i)$ for $1 \le i \le t$ If k' encrypts each P_i to its proper C_i, then we have evidence that $k' = K$. However, it may be the case that the mappings $E(K, \cdot)$ and $E(k', \cdot)$ exactly agree to the t plaintext-ciphertext pairs $P_i C_i$, and agree on no other pairs. What is the probability that $E(K, \cdot)$ and $E(k', \cdot)$ are in fact distinct mappings..?

Without loss of generality we may assume that $E[K, P_i] = P_i$ since $E_K(\cdot)$ is taken over all permutations. It then follows that we seek the probability that a permutation on $N - t$ objects has exactly t' fixed points is equal to the number of ways t' out of $N - t$ objects can be fixed, while the remaining $N - t - t'$ are not fixed.

Now, we have :

$$P(t' \text{ additional fixed points}) = \binom{N-t}{t'} P\left(\begin{array}{c}\text{No fixed points} \\ \text{in } N-t-t' \\ \text{objects}\end{array}\right)$$

Scanned with CamScanner

$$= \frac{1}{(t')!} \sum_{k=0}^{N-t-t'} \frac{(-1)^k}{k!}$$

We see that this reduces to the solution when we substitute

$$t' = N - t$$

$$P[E(k,\cdot), E(k',\cdot) \text{ not distinct}] = \frac{1}{(N-t)!}$$

$$P[E(k,\cdot) \text{ and } E(k',\cdot) \text{ are distinct}] = 1 - \frac{1}{(N-t)!}$$

Scanned with CamScanner

06) Let x' be the bitwise complement of x. Prove that if the complement of the plaintext block is taken and the complement of an encryption key is taken, then the result of DES Encryption with these values is the complement of the original ciphertext i.e.

if $Y = E(K, X)$ then

$$Y' = E(K', X')$$

Let $DES(K, X)$ represent the encryption of a plaintext x with key K using the DES cryptosystem. Suppose $Y = DES(K, X)$ and $Y_c = DES(\bar{K}, \bar{x})$ where $\bar{x}$ and $\bar{k}$ are bitwise complements, we show that $Y_c = \bar{Y}$.

The permutations, expansion permutations (EP), selection permutation or permuted choice (PC), key rotations and key selection permutation all behave the same regardless of their input, since DES is closed under complement. Thus a bit complemented in the output, at the position assigned to that bit. This is not true of S-Box computation

Let $x \in \{0,1\}^*$ and denoting unary vector on x as

$$1 = (1, 1, 1, 1 \ldots 1)$$

$$x = (0, 1, 1, 0, 1 \ldots 0) \text{ Some random input}$$

Note complement of x is given by x xor 1

$$\bar{x} = x \oplus 1$$

Scanned with CamScanner

We also note that for any 2 binary strings A and B

$$\begin{aligned}\bar{A} \text{ XOR } \bar{B} &= A \text{ XOR } 1 \text{ XOR } B \text{ XOR } 1 \\ &= A \text{ XOR } B \text{ XOR } (1 \text{ XOR } 1) \\ &= A \text{ XOR } (B \text{ XOR } 0) \\ &= A \text{ XOR } B\end{aligned}$$

We also note that -

$$\begin{aligned}\bar{A} \text{ XOR } B &= A \text{ XOR } 1 \text{ XOR } B \\ &= (A \text{ XOR } B) \text{ XOR } 1 \\ &= \overline{(A \text{ XOR } B)}\end{aligned}$$

Now, let us take input $\bar{X}$ with key $\bar{K}$. After initial permutation, we have $\bar{L_0}$ and $\bar{R_0}$. Since $L_i = R_{i-1}$ we will obtain $\bar{L_i}$ after each stage with $\bar{R}_{i-1}$ as input. If we let $A = R_i$, $B = EP(A)$, $C = B \text{ XOR } k_i$, $D = S(C)$ where S denotes an S-Box, and $E = P(D)$, then if $A_c = \bar{A}$ and $B_c = \bar{B}$ (i.e. the EP or expansion permutation preserves complement), then

$$Y_c = \bar{B} \text{ XOR } \bar{k_i} = \bar{B} \text{ XOR } k_i = Y$$

As $Y_c = Y$, we can clearly say that

$$DES(\bar{X}, \bar{K}) = \bar{Y} \text{ where } DES(X, K) = Y$$

We can even see that after each round i in the DES

$$R_i = E_c \text{ XOR } \bar{L}_{i-1} = E \text{ XOR } \bar{L}_{i-1} = \bar{R}_i$$

Scanned with CamScanner

Q7) Show that in DES the first 24 bits of each subkey come from the same subset of 28 bits of the initial key and the second 24 bits of each subkey come from a disjoint set (subset) of 28 bits of the initial key.

This result can be demonstrated by tracing through the way in which the bits are used. An easy but not necessary way to see this is to number the 64 bits of the key as follows (read each vertical column of 2 digits as a number) :-

2113355 – 1025554 – 0214434 – 1123334 – 0012343 –
1031975 – 1176107 – 2423401 – 7632789 – 7452553 –

2021453 – 0202435 – 0110454 –
0858846 – 6836043 – 9495226 –

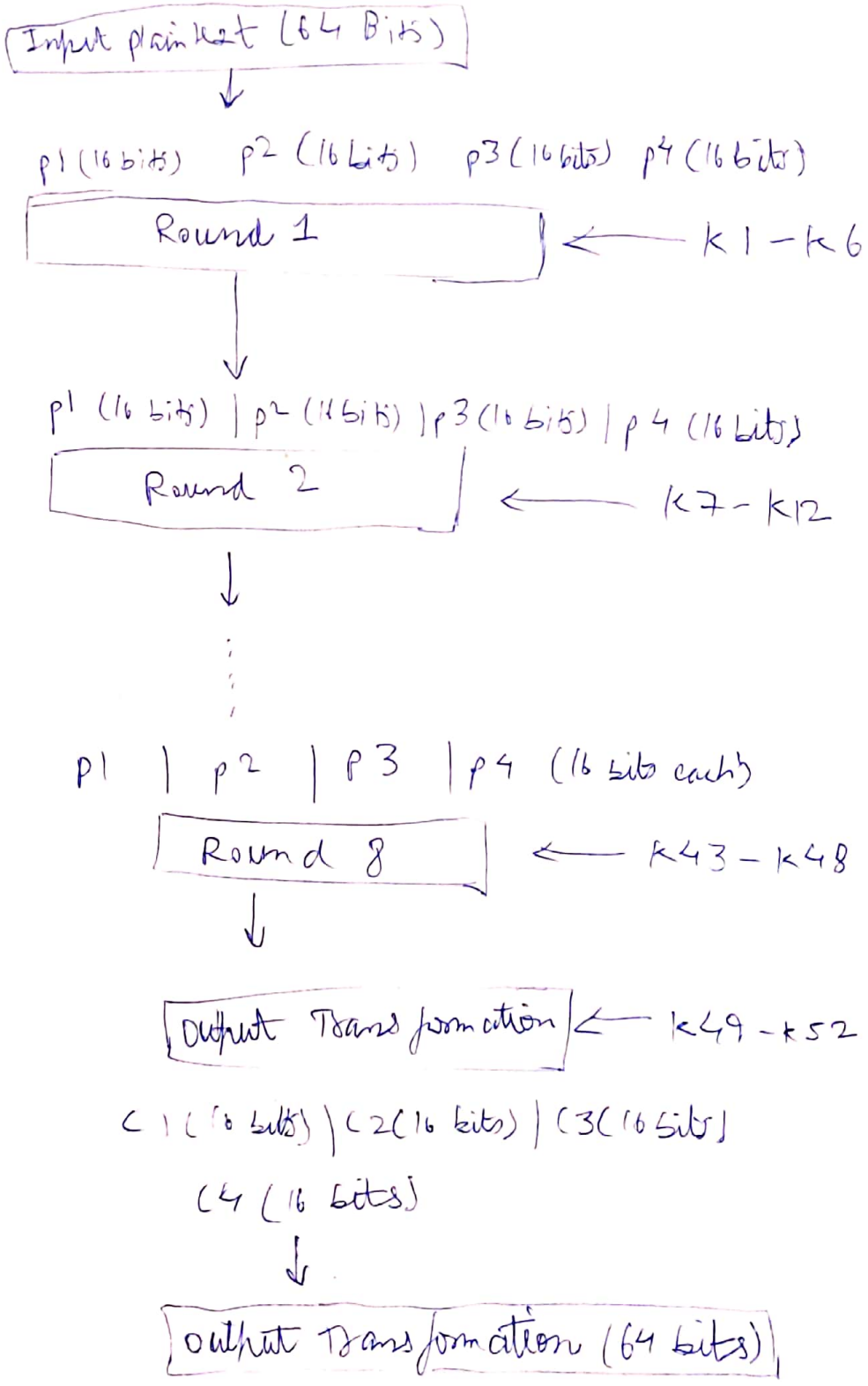
The first bit of the key is identified as 21, the second as 10, the third as 13 and so on. The eight bits that are not used in the calculation are unnumbered. The numbers 01 through 28 and 30 through 57 are used. The reason for this assignment is to clarify the way in which the subkeys are chosen. With this assignment, the subkey for the first iteration contains 48 bits, 1 through 24 and 30 through 53, in their natural numerical order. It is easy in this point to see that the first 24 bits of each subkey will always be from the bits designated 01 through 28 and the second 24 bits of each subkey will always be from the bits designated 30 through 57.

Scanned with CamScanner

Q8) Discuss the following with respect to the International Data Algorithm (IDEA):

a) Steps Involved in one round of IDEA:-

The overall structure of the IDEA algorithm is as follows:



Scanned with CamScanner

The 64 bit - plaintext input is divided into 4 parts and there are 8 such rounds. In each round 6 - sub-keys are produced. Each of the subkeys will be put on the 4 input blocks p1 to p4.

The single round is processed as follows: -

i) There are 8 rounds in IDEA

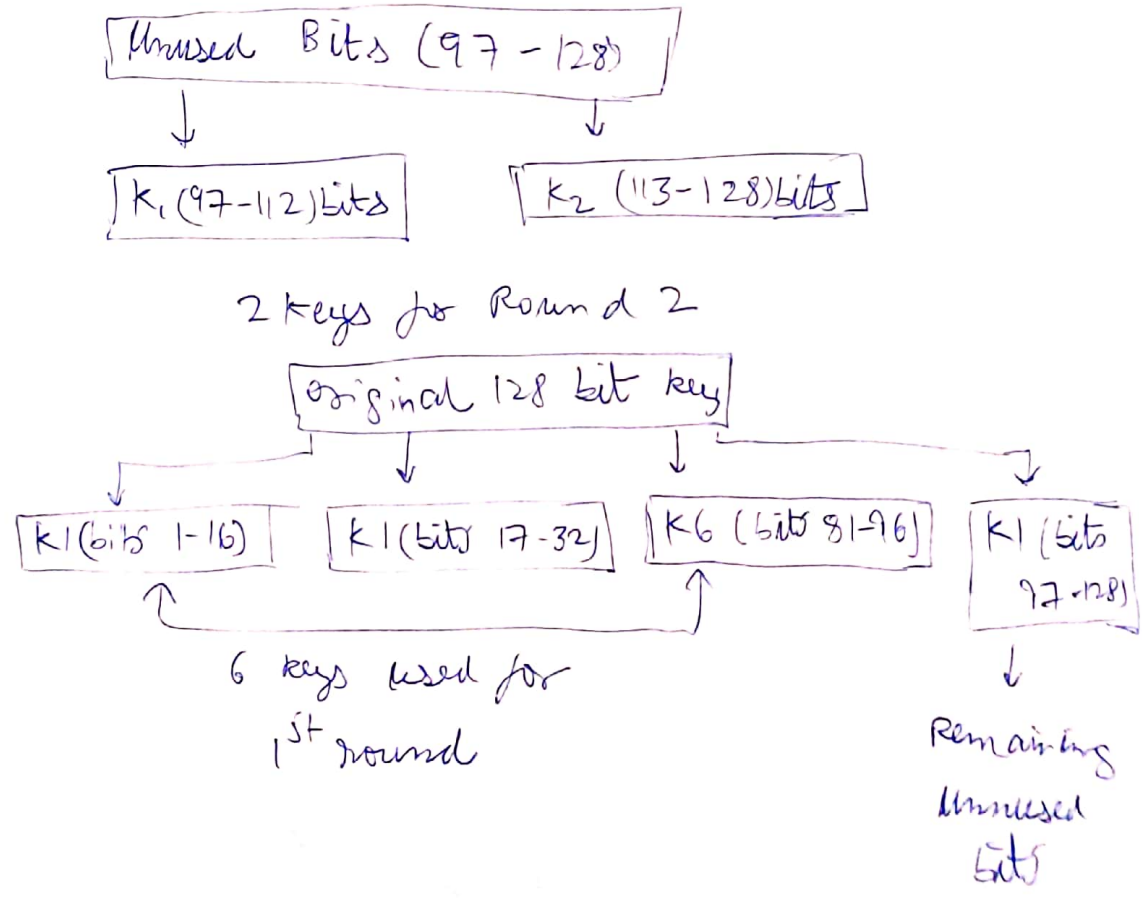
ii) Every single round performs a number of operations around the 4 data blocks applying 6 keys.

iii) These steps works numerous mathematical activities.

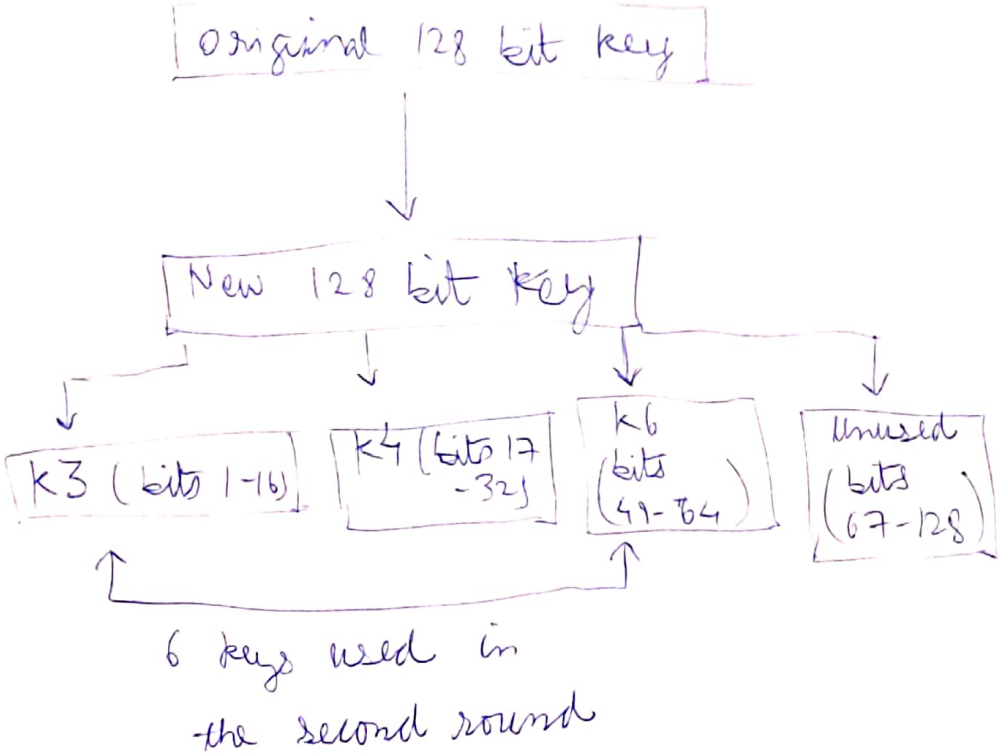
iv) There are multiple *, and 8 XOR(^) procedures.

v) Multiply * means modulo multiplication

vi) Add $+_m$ is addition modulo.



Scanned with CamScanner



b) Output Transformation

Output transformation is a one-time procedure.

It requires keys by the end of 8th round. The input towards the output transformation is a 64-bit value divided into 4-sub blocks (state R1 to R4 every block among 16 bits).

The 4 sub-keys K1 to K4 are used here. The process of the outcome transformation can be as follows.

Step 1: Multiply R1 and K1 $R1 *_m K1$

↓

Step 2: Add R2 and K2 $R3 = R2 +_m K2$

↓

Step 3: Add R3 and K3 $R4 = R3 +_m K3$

↓

Step 4: Multiply R4 and K4 $R4 \times_m K4$

Scanned with CamScanner

c) Strength of IDEA

The designers analyzed IDEA to measure its strength against differential cryptanalysis and concluded that it is immune under certain assumptions. No successful Linear or Algebraic weaknesses have been reported as of ~~2017~~ 2007. The best attack applied to all keys could break IDEA reduced to 6 rounds (The full cipher uses 8.5 rounds). The 6 round attack requires 2^{64} known plaintext and $2^{126.8}$ operations.

Bruce Schneier thought highly of IDEA in 1996, writing "In my opinion, it is the best and most secure block algorithm available to the public right now". However in 1999 he was no longer recommending IDEA due to the availability of faster algorithms, some progress in its cryptanalysis, and the issue of patents.

In 2011 full 8.5 round IDEA was broken using a meet in the middle attack. Independently in 2012, full 8.5 round IDEA was broken using a narrow-bicliques attack with reduction of cryptographic strength of about 2 bits, similar to previous bi-cliques attack on AES previously.

The very simple key-schedule makes IDEA subject to a class of weak keys: some keys containing a large number of 0 bits

Scanned with CamScanner

hundreds weak encryption. These are of little concern in practice, being sufficiently rare that they are unnecessary to avoid explicitly when generating keys randomly. A simple fix was proposed: XORing each subkey with 16-bit constant such as 0x0DAE

Larger classes of weak keys were found in 2002. This is still of negligible probability to be a concern for a randomly chosen key and some of the problems are fixed by the random XOR introduced earlier.

Scanned with CamScanner