Week-10 : Lecture Note

Topic:

Cryptanalysis

Time-Memory Trade off Attack

More on Differential crypt-analysis.

Linear cryptanalysis.

Attacks on Cryptosystem

- Up to this point we have mainly seen how ciphers are implemented.

- We have seen how symmetric ciphers such as DES and AES use the idea of substitution and permutation to provide security and also how asymmetric system

such as RSA and Diffie-Hellman use other methods.

- What we have not really looked at are attacks on cryptographic systems.

- An understanding of certain attacks will help you to understand the reasons behind the structure of certain algorithms (such as Rijndael) as they are designed to thwart known attacks.

- Although we are not going to exhaust all possible avenues of attack, we will get an idea of how cryptanalysts go about attacking ciphers.

- This section is really split up into two classes of attack: Cryptanalytic attacks and Implementation attacks.

- The former tries to attack on mathematical weakness in the algorithms whereas the later tries to attack on the specific implementation of the cipher (such as a smart card system)

- The following attacks can refer to either of the two classes (all forms of attack assume the attacker knows the encryption algorithm):

  - Ciphertext-only attack: In this attack, the attacker knows only the ciphertext to be decoded. The attacker will try to find the key or decrypt one or more pieces of ciphertext (only relatively weak algorithms fail to withstand a ciphertext only attack)

  - Known plaintext attack: The attacker has a collection of plaintext

ciphertext pairs and is trying to find the key or to decrypt some other ciphertext that has been encrypted with the same key.

- Chosen plaintext attack: This is known plaintext attack in which attacker can choose the plaintext to be encrypted and read the corresponding ciphertext.

- Chosen Ciphertext attack: The attacker has the ability to select any ciphertext and study the plaintext produced by decrypting them.

- Chosen text attack: The attacker has the abilities required in the previous two attacks.

- The following terminology is also useful to know:

  - An encryption scheme is

unconditionally secure if the ciphertext generated does not contain enough information to determine uniquely the corresponding plaintext no matter how much ciphertext is available or how much computational power the attacker has. With the exception of the one time pad, no cipher is unconditionally secure.

— The security of conditionally secure algorithm depends on the difficulty in reversing the underlying cryptographic problem such as how easy it is to factor large primes. All ciphers other than the one time pad fall into this catagory.

— An encryption scheme is said to be computationally secure if:

1. The cost of breaking the cipher exceeds the value of the encrypted information.

2. The time required to break the cipher exceeds the useful lifetime of the information.

Non-Generic Attacks

- Exploits special properties of an algorithm or a class of algorithms.

- Linear Cryptanalysis: Obtain approximate linear relationships between input, output and key bits such that not all key bits are involved. Use such relations to mount a divide and conquer attack.

- Differential Cryptanalysis: Similar idea as above. In this case instead of linear relationships, try to obtain relationship between

inpnt and outpnt differentials.

- Other examples: correlation attack, algebric attack etc.

- The literature abounds in such attacks and ideas.

- Very little (or no) work on
  - hardware implementation of non-generic attacks;
  - exploiting parallelism

Generic Attacks

- Most basic of all attacks.
- Treats the algorithm as a black box.

- The same attack idea can be used for many algorithms.

- One can utilise a large amount of paralleism.

- Most feasible implementation in special purpose hardware.

- Exhaustive search
- Time/memory trade-off (TMTO) attack.

Exhaustive search attack

- A brute force method which will try for all possible keys.
- The resistance against exhaustive search depends on:
  - size of the key space
  - implementation in software or special purpose hardware.
  - The number of parallel processors available.
  - The speed at which each key can be processed.

- The cost of each processor and the overall cost of implementing the attack

- Brute force attacks are also available to the attacker however we will not be concerned with them in this chapter although it is interesting to note the following.

- In 1977 Diffie and Hellman claimed that it would cost twenty million dollars to build a million chip machine that could find a DES key in twelve hours (given a plaintext - ciphertext pair)

- In 1995, it was estimated that advances in chip densities and speeds would permit a several thousand chip machine to do the same job at a cost of well under a million dollars.

- However, in July 1998 a machine was built by EFF, cryptography research and Advanced Wireless technolog that could search 90 billion keys per second which would take a little over 200 hours to search the entire key space.

- They managed however to find the key in 56 hours.

- The total budget remained under $250,000 which made the machine the fastest most economical key search device ever known to have been produced.

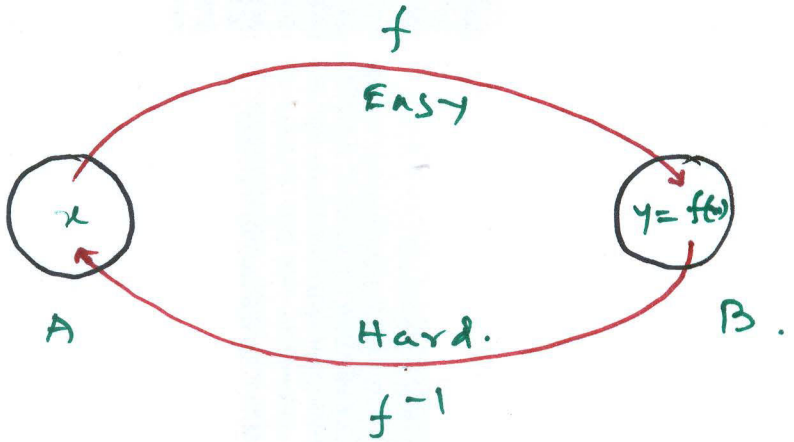
One-way function.

- $f: A \rightarrow B$ satisfies the following two properties:

  - f is easy to compute, i.e. there exist a polynomial time algorithm to compute

$f(x)$ for an input x; and

— f is hard to invert i.e there is no polynomial time algorithm to get x from $f(x)$.



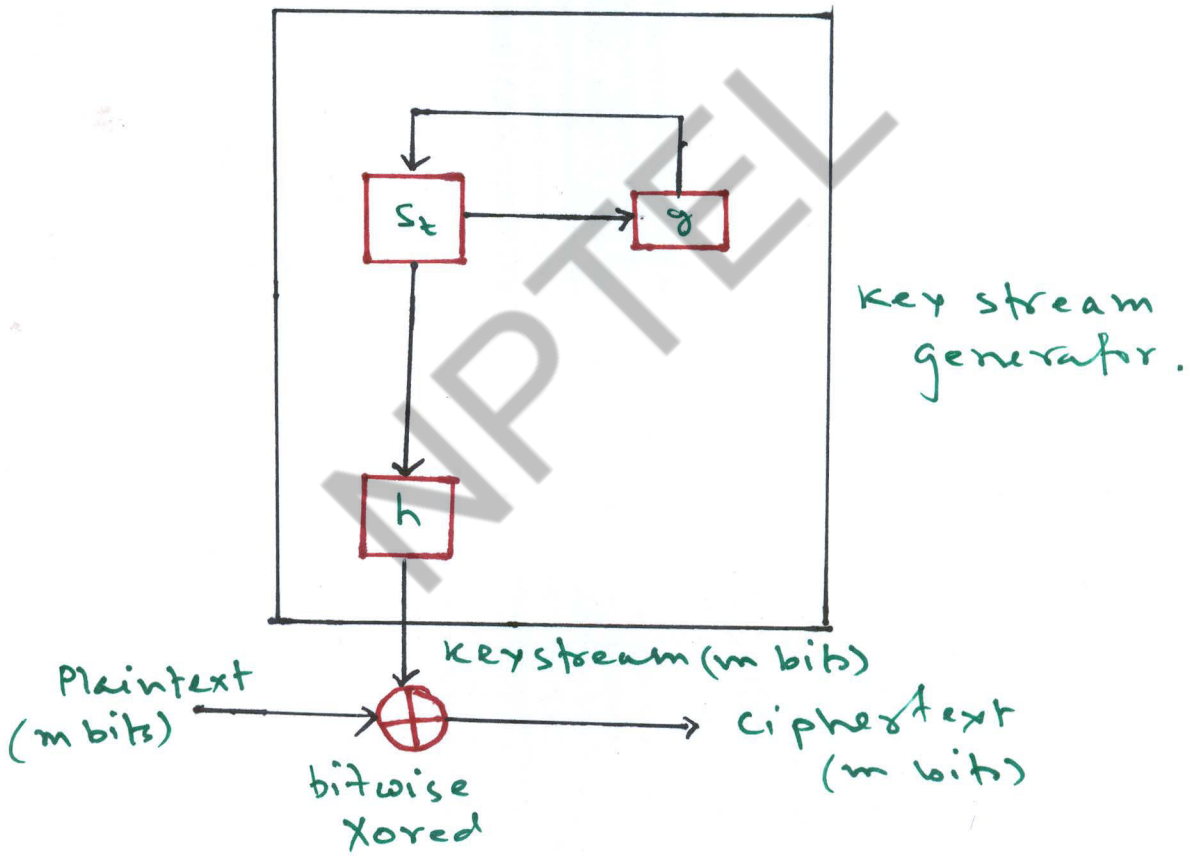
- Block cipher: $E: \{0,1\}^n \times \{0,1\}^s \rightarrow \{0,1\}^n$ where $C = E_K(P)$ denotes the cipher-text C that results from message P under K.

- One way function from block cipher: For a fixed message P, we can define a function $f: \{0,1\}^s \rightarrow \{0,1\}^s$ as: $f(x) = R(E_K(P))$, where $R: \{0,1\}^n \rightarrow \{0,1\}^s$ is a function from ciphertext to keys:

  — If $n > s$ (DES has $n = 64, s = 56$) then we remove the first $(n-s)$ bits.

— If $n \leq s$ (AES has $n = 128$ and there are three allowable key lengths, $s = 128, 192$ and 256 bits), then we append $(s-n)$ constant bits.

- stream cipher: Let $S = \{0,1\}^s$ be the set of all possible internal states



- One way function from stream cipher: state to key stream map $f: S \longrightarrow \{0,1\}^s$ is defined as follows:

  state $s \longmapsto$ first s bits keystream prefix.

- Stream cipher with IV :

(key, IV) $\longmapsto$ key stream map. For a k bit stream cipher using an l-bit IV, the following (k+l)-bit one way function f is constructed :

(k bit key, l bit IV) $\rightarrow$ (k+l) bit keystream

- Problem definition: Given a string y, we will have to find a string x (preimage or key) such that $f(x) = y$, where $f: \{0,1\}^s \rightarrow \{0,1\}^s$ is a one-way function.

- Let N be the number of possible keys, then $N = 2^s$

Time/memory trade-off (TMTO) attack

- Introduced by Hellman in 1980.
- consists of two phases:
  - Pre computation : one time

exhaustive search.

— Online search: Can be repeated, much faster than exhaustive search.

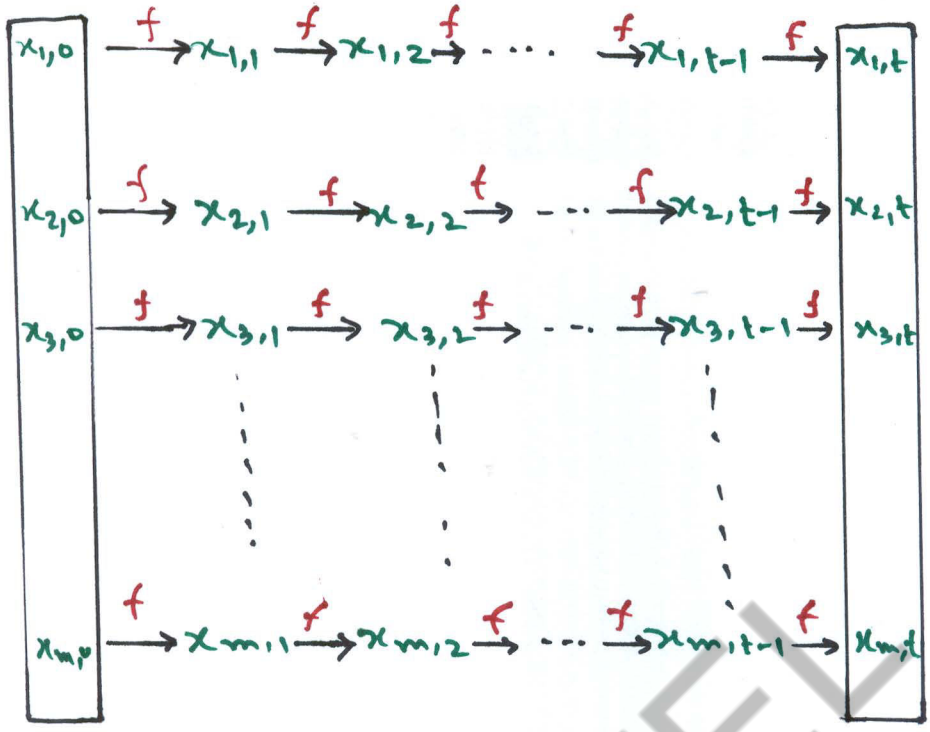
- Rivest introduced the distinguished Point (DP) method.
- Later work by Fiat-Naor, Oechslin, Biryukov-Shamir and others.

Hellman method.

Two phases,

- offline phase: A (set of) table(s) is prepared and only a part of the table(s) is stored which incurs a cost in the online phase. This leads to a trade-off between the memory (storage) and time required in the online phase.
- Online phase

Hellman table:



start points

end points.

offline phase:

We store (start point, end point) pairs, sorted in the increasing order of end points.

online stages:

$$y \xrightarrow{f} x_1 \xrightarrow{f} x_2 \longrightarrow \cdots \longrightarrow x_{t-2} \xrightarrow{f} x_{t-1}$$

Assuming that the key is in the table.

$$x = x_{i,j} \xrightarrow{f} y \xrightarrow{f} x_{i,j+2} \rightarrow \cdots \rightarrow x_{i,t-2} \xrightarrow{f} x_{i,t-1}$$

Then we come to the corresponding start point and repeatedly apply the function until it reaches y. The previous value it visited is x.

Multiple Hellman Tables:

- **Success Probability:** Above attack will work if the key x is in the table. The probability that the key is in the table measures the success of the method which we called success probability and is measured by:

$$\frac{\text{number of distinct key in the tables}}{N}$$

- **Critical Values of m and t:** By birthday paradox, we know that the set of mt points is disjoint with a set of t point (a row) as long as $t \times mt \leq N$. So we can add rows in

the table until $mt^r = N$. Hence we choose m and t such that $mt^r = N$ which is called matrix stopping rule.

- **Multiple Hellman tables:** Single Hellman table of size $m \times t$ can cover only a fraction $\frac{mt}{N} = \frac{1}{t}$ of N. Hence, we need t unrelated tables to cover all N keys.

- For the i^{th} table, we select a function f_i (say), which is a simple output modification of f such that f_i, $i = 1, 2, \ldots, t$ are unrelated.

- **Trade off curve:** Hellman method can recover the key in time T (number of f invocations at online phase) with memory requirement M such that $TM^r = N^r$. One interesting point on the curve is $T = M = N^{2/3}$.

Implementation of TMTO algorithm

- software implementation:

  - rainbowcrack: crack 99.9% of all aphanumerical MS-Windows password hashes (out of 2^{37}) in 13.6 second using 1.4 GB data.

  - Attack on the stream cipher A5/1 which is used in GSM mobile phone: $P = 2^{48}$, the actual attack can be carried out in real time on a single PC.

- Hardware Implementation:

  - Amirazizi and Hellman proposed time/memory/processor trade off: switching/sorting network is used.

  - Wiener prove that the wiring cost can be less than $\Theta(n^{3/2})$ for any switching/sorting network to

connect n processors with n memory blocks.

- Guisquater and Standaet suggest a pipelined architecture for implementing a multi-round function.

- Nele Mentens et al propose a hardware architecture on rainbow method.

Cryptanalytic Attacks

- All forms of cryptanalysis for symmetric encryption schemes are designed to exploit the fact that traces of structure or pattern in the plaintext may survive encryption and be discernible in the ciphertext.

- Cryptanalysis of public key schemes proceeds from a fundamentally different premise, namely that the

mathematical properties of the pair of keys may make it possible for one of the two keys to be deduced from the other.

Differential cryptanalysis

- One of the most significant advances in cryptanalysis in recent years is differential cryptanalysis.

- Although this appears to have been discovered at least 30 years ago it was not reported in the open literature until 1990.

- The first published effort appears to have been the cryptanalysis of a block cipher called FEAL.

- This was followed by a number of papers by Biham and Shamir, who demonstrated this form of attack

on a variety of encryption algorithms and hash functions.

- The most publicised results for this approach have been those that have application to DES.

- Differential cryptanalysis is the first published attack that is capable of breaking DES in less than 2^{55} complexity.

- The scheme can successfully crypt analyse DES with an effort of 2^{47}, requiring 2^{47} chosen plaintext (hence it is a chosen plaintext attack)

- Although 2^{47} is certainly significantly less than 2^{55}, the need to find 2^{47} chosen plaintexts makes this attack of only theoretical interest.

- Apparently this attack was known at the time DES was being designed and played a large part in its design.

- The attack can be summarised as follows: (a more detailed explanation)

- Consider the original plaintext block for DES m to consist of two halves m_0, m_1

- Each round of DES maps the right hand input into the left hand output and sets the right hand output as a function of the left hand input and the subkey for this round.

- So, at each round, only one new 32-bit block is created.

- If we label each new block m_i $(2 \le i \le 17)$, then the intermediate message halves are related as follows:

$$m_{i+1} = m_{i-1} \oplus f(m_i, K_i) \quad i = 1, 2, \ldots, 16$$

- In differential cryptanalysis, one starts with two messages, m and m′ with a known XOR difference

$\Delta m = m \oplus m'$, and considers the difference between the intermediate message halves : $\Delta m_i = m_i \oplus m_i'$. Then we have $\Delta m_{i+1} = m_{i+1} \oplus m_{i+1}'$

$$= [m_{i-1} \oplus f(m_i, k_i)] \oplus [m_{i-1}' \oplus f(m_i', k_i)]$$

$$= \Delta m_{i-1} \oplus [f(m_i, k_i) \oplus f(m_i', k_i)]$$

- Now, suppose that many pairs of inputs to f with the same difference yield the same output difference if the same subkey is used.

- To put this more precisely, let us say that X may cause Y with probability p, if for a fractions p of the pairs in which the input XOR is X, the output XOR equals Y.

- We want to suppose that there are a number of values of X that have high probability of causing a particular output difference.

- Therefore, if we know Δm_{i-1} and Δm_i with high probability, the we know Δm_{i+1} with high probability.

- Furthermore, if a number of such differences are determined, it is feasible to determine the subkey used in the function f.

- The overall strategy of differential cryptanalysis is based on these considerations for a single round.

- The procedure is to begin with two plaintext message m and m' with a given difference and trace through a probable pattern of differences after each round to yield a probable difference for the ciphertext.

- Actually, there are two probable differences for the two 32-bit halves: $(\Delta m_{17} \| \Delta m_{16})$

- Next, we submit m and m' for encryption to determine the actual

difference under the unknown key and compare the result to the probable difference.

- If there is a match,

$$E_K(m) \oplus E_K(m') = (\Delta m_{17} \| \Delta m_{16})$$

then we suspect that all the probable patterns at all the intermediate rounds are correct.

- With that assumption, we can make some deductions about the key bits. This procedure must be repeated many times to determine all the key bits.

- A differential attack against an r round iterated block cipher is the following:

  — step1. Find an r-1 round differential $\alpha \rightarrow \beta$ with high enough propagation ratio.

  — step2: keep a counter for each possible round subkey K_r

at each round r. Initialize the counters to zero.

- step3: Pick a plaintext x uniformly at random and set $x^* = x \oplus \alpha$. Encrypt the plaintexts under the unknown key k obtaining the ciphertexts y and y^*. For each possible round subkey kr compatible with the assumed input difference β and the observed outputs y, y^* at round r, add one to the corresponding counter.

- step4: Repeat step3 until some round subkeys are counted significantly more often than the others. Output these keys as the most likely subkey at the last round.

Boomerang Attack

- The main idea behind the boomerang attack is to use two short differentials with high probabilities instead of one differential of more rounds with low probabilities.

- The motivation for such an attack is quite apparent, as it is easier to find short differentials with a high probability than finding a long one with a high enough probability.

- We assume that a block cipher $E: \{0,1\}^n \times \{0,1\}^k \longrightarrow \{0,1\}^n$ can be described as a cascade $E = E_1 \circ E_0$, such that for E_0 there exists a differential $\alpha \rightarrow \beta$ with probabilities p, and for E_1 there exists a differential $\gamma \longrightarrow \delta$ with probability q.

- The boomerang attack uses the first characteristic $\alpha \rightarrow \beta$ for E_0 with respect to the pairs $(P_1; P_2)$ and $(P_3; P_4)$ and uses the second characteristic $\gamma \rightarrow \delta$ for E_1 with respect to the pairs $(C_1; C_3)$ and $(C_2; C_4)$

- The attack is based on the following boomerang process:

  - step1: Ask for the encryption of a pair of plaintexts $(P_1; P_2)$ such that $P_1 \oplus P_2 = \alpha$ and denote the corresponding ciphertext by $(C_1; C_2)$

  - step2: Calculate $C_3 = C_1 \oplus \delta$ and $C_4 = C_2 \oplus \delta$ ask for the decryption of the pair $(C_3; C_4)$. Denote the corresponding plaintexts by $(P_3; P_4)$

  - step3: check whether $P_3 \oplus P_4 = \alpha$

- It is easy to see that for a

random permutation the probability that the last condition is satisfied is $\bar{2}^n$. For E, however, the probability that the pair $(P_1; P_2)$ is a right pair with respect to the first differential $\alpha \rightarrow \beta$ is p.

- The probability that both pairs $(C_1; C_3)$ and $(C_2; C_4)$ are right pairs with respect to the second differential is q^r.

- If all these are right pairs, then they satisfies $E_1^{-1}(C_3) \oplus E_1^{-1}(C_4) = \beta = E_0(P_3) \oplus E_0(P_4)$, and thus, with probability p also $P_3 \oplus P_4 = \alpha$.

- Therefore, the total probability of this quartet of plaintexts and ciphertexts to satisfies the boomerang condition is $(pq)^r$. Therefore, $pq > 2^{-n/2}$ must hold for the boomerang attack to work.

Linear Cryptanalysis

- A more recent development is linear cryptanalysis that was presented by Mitsuru Matsui at Eurocrypt'93.

- This attack is based on finding linear approximations to describe the transformations performed in DES (and other block ciphers).

- This method can find a DES key given 2^{47} known plaintexts, as compared to 2^{47} chosen plaintexts differential cryptanalysis (it is therefore a known plaintext attack although it can also work as a ciphertext only attack).

- Although this is a minor improvement (because it may be easier to acquire known plaintext rather than chosen plaintext) it

still leaves linear cryptanalysis infeasible as an attack on DES.

- However, it is useful for an understanding of other similar attacks and gives an insight into why the S-boxes are constructed the way they are.

- To understand this attack we will define a few terms:

- A Boolean function $h: \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ in n variables $s_1, \ldots, s_n$ is linear if it can be represented as

$$h(s) = a_1 s_1 \oplus \cdots \oplus a_n s_n$$ for some $a_i \in \mathbb{Z}_2 = \{0,1\}$, $i = 1, 2, \ldots, n$. The set of all linear Boolean functions in n variables is denoted by

$$L_n = \{ h: \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2 \mid h = a_1 s_1 \oplus \cdots \oplus a_n s_n \}.$$

- A Boolean function $f: \mathbb{Z}_2^n \longrightarrow \mathbb{Z}_2$ is called affine if either $f(s) = h(s)$ or $f(s) = h(s) \oplus 1$, for some $h(s) \in L_n$

The set of all affine Boolean function in n variables is therefore

$$A_n = L_n \cup \{ h \oplus 1 \mid h \in L_n \} = L_n \cup \overline{L_n}$$

- In other words, A_n consists of all linear functions and their negations.

Summary of Attack.

- For a cipher with n bit plaintext and ciphertext blocks and m bit key, let the plaintext block be labeled $P[1], \cdots, P[N]$, the ciphertext block $C[1], \cdots, C[n]$ and the key $K[1], \cdots, K[m]$.

- Then define : $A[i,j,\cdots,k] = A[i] \oplus A[j]$

$\oplus \cdots \oplus A[k]$

- The objective of linear cryptanalysis is to find an effective linear equation of the form

$$P[\alpha_1, \alpha_2, \cdots, \alpha_a] \oplus C[\beta_1, \cdots, \beta_b] = K[\gamma_1, \cdots, \gamma_c] \quad \cdots (1)$$

that holds with probability $p \neq 0.5$

- Here we have $x = 0, 1$; $1 \leq a, b \leq n$, $1 \leq c \leq m$ and where α, β and γ terms represent fixed, unique bit locations.

- The further p is from 0.5, the more effective the equation.

- Once a proposed relation is determined, the procedure is to compute the results of the left hand side of the preceding equation for a large number of plaintext-ciphertext pairs.

- If the result is 0 more than half the time, assume $K[\gamma_1, \gamma_2, \cdots, \gamma_e] = 0$
- If it is 1 most of the time, assume $K[\gamma_1, \gamma_2, \cdots, \gamma_e] = 1$.
- This gives us a linear equation on the key bits.
- Try to get more such relations so that we can solve for the key bits.
- Because we are dealing with linear equations, the problem can be approached one round of the cipher at a time, with the result combined.
- The fact that we desire equation (1) to hold with a probability $P \neq 0.5$ implies that it can be $0 \geq P < .5$ or $0.5 < P \leq 1$.

- This leads us to the idea of a linear probability bias which is given by $\epsilon = |p - 0.5|$.

- The larger this bias is (in other words the closer p is to 0 or 1) the better the applicability of linear cryptanalysis with fewer known plaintext.

- If $p=1$ this implies that the eq^n (1) is a perfect representation of the cipher behaviour and the cipher has a catastrophic weakness.

- If $p=0$ then equation (1) represents an affine relationship in the cipher which is also a catastrophic weakness.

- Both linear and affine approximations, indicated by $p > 0.5$

and $p < 0.5$ respectively, are equally susceptible to linear cryptanalysis.

- For an ideal cipher what we would like is that the plaintext be mapped to the ciphertext in such a way that the mapping is random.

- In other words there is no correlation between the plaintext and ciphertext.

- By choosing $a+b$ random variables (number of plaintext plus ciphertext bits in equation (1)) equation (1) should hold with a probability of exactly 0.5.

- This would be the case if it were a perfect cipher.

- However as ciphers are not perfect the probability contains a bias ϵ.

- The bias not only exists for the overall cipher but also for each of the individual non-linear element (S-boxes in case of DES)

- We can therefore find linear approximations for certain S-boxes and use what's known as the piling up lemma to concatenate the results to gain an expression for the whole cipher.

- With this expression, it is possible to take some plaintext and ciphertext pairs and a target partial subkey (this is your guess at a part of the subkey) and deduce the target partials subkeys values.

- More detailed information is given in lecture video.

Slide Attack.

- Slide attack was proposed by Alex Biryukov and David Wagner in 1999.

- In the simplest case, we have an r-round block cipher E whose round functions are same and use the same subkey, so that

  $E = F \circ F \circ \cdots \circ F = F^r$.

- Let (P, C) be a known plaintext-ciphertext pair for E. The crucial observation is, if $P' = F(P)$ then

  $C' = E(P') = F^r(F(P)) = F(F^r(P)) = F(C)$

- In a slide attack, we try to find pairs (P, C) and (P', C') with $P' = F(P)$, we call such a pair a slide pair, and then we will get the extra relation $C' = F(C)$.

- A slide attack provides a very general attack on block cipher with repeating round subkeys.

- The only requirement on F is that it is weak against known-plaintext attack with two pair. More precisely, we call $F_K(x)$ a weak function if given the two equations $F_K(x_1) = y_1$ and $F_K(x_2) = y_2$ it is easy to extract the key K (for example 3-round DES is a weak function).

Implementation Attacks

- Implementation attacks take on a different approach to the above for discovering the secret key.

- Instead of attacking the mathematical properties of the algorithm these form of attacks (also known as side channel attacks) take

advantage of the physical phenomena that occurs when a cryptographic algorithm is implemented in hardware.

- Four side channel attacks are listed in the FIPS standard 140-2 "Security Requirements for Cryptographic Modules", power Analysis, Timing Analysis, Fault Induction and TEMPEST.