Kriptografi: Şifreleme algoritmaları kullanarak şifreleme oluşturma bilimidir.

Kriptanaliz: Şifre ya da anahtar kullanmadan deşifre etme yöntemi.

Kriptoloji: Hem kriptografiyi hem de kriptanalizi kapsar.

Kriptografide kullanılan bazı temel kavramlar;

Plaintext : Orijinal, düz metin.

Ciphertext : Şifrelenmiş metin.

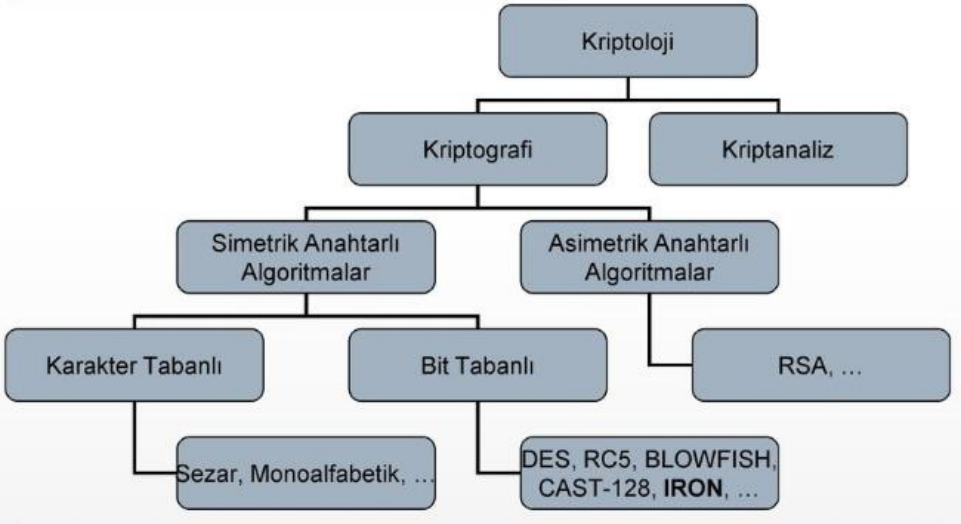
Cipher : Düz metni, şifrelenmiş metne çeviren algoritma. Şifreleme algoritması.

Encipher (encrypt): Düz metni şifrelenmiş metne çevirme.

Decipher (decrypt): Şifreli metinden düz metni kurtarma.

Düz metini (plaintext) (kodlanan mesaj ya da veri parçasını temsil eder) şifrelemek ve şifresini çözmek için aynı anahtar kullanılır. Şifreleme işlemi, bir düz metinin (girdi) şifre (cipher) adındaki bir şifreleme algoritmasından geçirilerek bir şifreli metin (ciphertext) (çıktı) oluşturulması.

Veriyi şifrelerken ve çözerken kullanılan matematiksel metoda ise şifreleme algoritması denilmektedir.



Simetrik Şifreleme

İki ya da daha fazla kullanıcının ortak kullandığı mesajların şifrelenmesi ve çözülmesinde aynı anahtarın kullanıldığı bir şifreleme yöntemi.

Şifreleme sistemlerinin güvenliği, sistemin karşılık gelen anahtarın kaba kuvvet (brute force) uygulanarak tahmin edilmesini ne derece zor hale getirdiğine dayanır.

Örneğin, bir 128-bit anahtarı sıradan bir bilgisayar donanımı kullanarak tahmin etmek milyarlarca yıl alır. Şifre anahtarı ne kadar uzun olursa, bunu kırmak da bir o kadar zor olur. 256-bit uzunluğundaki anahtarlar genellikle çok güvenli olarak kabul edilir ve teorik olarak quantum bilgisayarların kaba kuvvet saldırılarına karşı dayanıklıdır.

Günümüzde en yaygın kullanılan iki simetrik şifreleme düzeni blok ve akış (stream) şifrelerine dayanır. Blok şifreleri, veriyi önceden belirlenmiş boyutlarda bloklar olarak gruplar ve her bir blok karşılık gelen anahtar ve şifreleme algoritması (ör. 128-bit düz metin 128-bit şifreli metne şifrelenir)

kullanılarak şifrelenir. Diğer yandan, akış şifreleri düz metin verisini bloklar olarak şifrelemek yerine 1-bitlik artışlarla (Bir kerede 1-bit düz metin 1-bit şifreli metin olarak şifrelenir) şifreler.

10 yıldır Devletler ve ordular arasındaki gizli iletişimi sağlamak için kullanıldı. Günümüzde çeşitli bilgisayar sistemlerinde veri güvenliğini arttırmak için kullanılır.

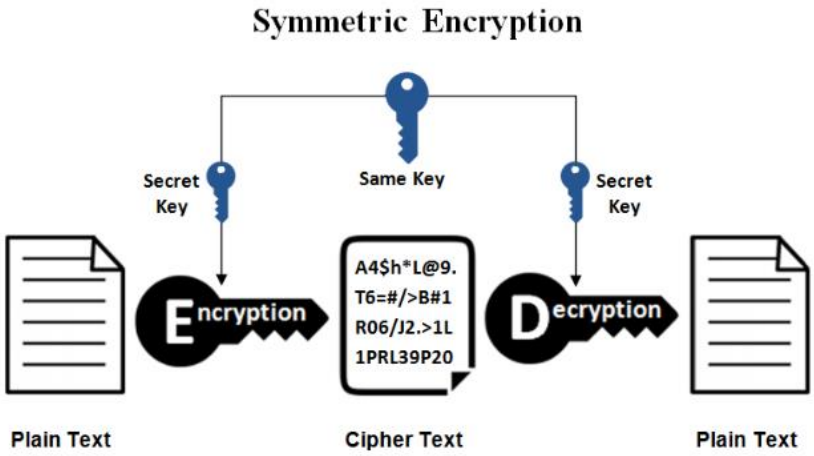
AES, DES, 3DES, RC4 başlıca simetrik şifreleme yöntemlerindendir.

Avantajları

- Şifreleme ve şifreyi çözme işlemleri hızlıdır, donanımla gerçekleştirilmesi kolaydır.
- Taraflar arasındaki iletişimin gizliliği sağlanır.
- Verinin bütünlüğü sağlanır. Şifreli metin çözülemedikçe orijinal metin değiştirilemeyecektir.

Dezavantajları

- Anahtar saklamak zordur. (Key Storage Problem)
- n kullanıcılı bir sistem için [n * (n-1) / 2] anahtar saklanmalıdır. Ölçeklendirilebilir değildir.
- Güvenilir anahtar dağıtımı zordur. (Key Distribution Problem)
- Kimlik doğrulama (authenticity) sağlamaz. Aynı anahtara sahip olan herhangi birisi tarafından veri şifrelenmiş olabilir.
- Bütünlük sağlamaz. Ortadaki bir kişi tarafından veri değiştirilmiş olabilir.
- Kimlik doğrulama ve bütünlük sağlamadığı için inkâr edilememezlik sağlamaz.



1. Karakter Tabanlı Şifreleme Algoritmaları

Sezar Şifrelemesi (Caesar's Cipher)

Sezar şifreleme algoritması bir harf yer değiştirme şifrelemesidir.

Shift=1

Sezar (Gizem)= Hjafn

```
import java.util.*;

class CaesarCipher{

   public static void main(String[]args)

   {

      Scanner scan = new Scanner(System.in);
```

```
        System.out.print("Enter your PlainText: ");
        String message = scan.next();
        int length = message.length();
        System.out.print("Enter the Key: ");
        int key = scan.nextInt();
        String uppercase = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        String lowercase = "abcdefghijklmnopqrstuvwxyz";
        String special = "!#$%&'()*+,-./:;<=>?@[]^_`{|}~";
        String numbers = "0123456789";
        System.out.print("The encrypted Text is: ");
        for(int i = 0 ;i < length ;i++)
        {
           for(int j = 0; j < 26 ;j++)
           {
              if(j < special.length() && message.charAt(i) == special.charAt(j))
               {
                   System.out.print(special.charAt(j)); //print special charecters as it is
               }
              else if(j < numbers.length() && message.charAt(i) == numbers.charAt(j))
              {
                 System.out.print(numbers.charAt(j)); //print numbers as it is
              }
              else if(message.charAt(i) == lowercase.charAt(j))
              {System.out.print(lowercase.charAt((j + key) % 26));}
              else if(message.charAt(i) == uppercase.charAt(j))
              {System.out.print(uppercase.charAt((j + key) % 26));
}}}
        System.out.println();
}}
```

Mono Alfabetik Şifreleme (Monoalphabetic Cipher)

Alfabedeki harfler rastgele permütasyon yapılarak anahtar elde edilir. Örneğin her A yerine Y gibi. Bu şifreleme yöntemi, kullanılan harflerin yerini değiştirir ama harflerin kullanım sıklığını değiştiremez. Dolayısıyla çok kısa sürelerde kırılabilmektedir.

Monoalphabetic (Sanfoundry) = LQFYGXFRKN

```
package com.sanfoundry.setandstring;
import java.util.Scanner;
public class MonoalphabeticCipher
{
   public static char p[]  = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',
```

```
        'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
        'w', 'x', 'y', 'z' };
    public static char ch[] = { 'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O',
        'P', 'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L', 'Z', 'X', 'C',
        'V', 'B', 'N', 'M' };
    public static String doEncryption(String s)
    {
        char c[] = new char[(s.length())];
        for (int i = 0; i < s.length(); i++)
        {
            for (int j = 0; j < 26; j++)
            {
                if (p[j] == s.charAt(i))
                {
                    c[i] = ch[j];
                    break;
                }}}
        return (new String(c));
}
    public static String doDecryption(String s)
    {
        char p1[] = new char[(s.length())];
        for (int i = 0; i < s.length(); i++)
        {
            for (int j = 0; j < 26; j++)
            {
                if (ch[j] == s.charAt(i))
                {
                    p1[i] = p[j];
                    break;
                }}}
        return (new String(p1));
    }
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the message: ");
        String en = doEncryption(sc.next().toLowerCase());
        System.out.println("Encrypted message: " + en);
```

```
    System.out.println("Decrypted message: " + doDecryption(en));
    sc.close();
  }}
```

Doğrusal Şifreleme (Affine Cipher)

y=ax+b fonksiyonunu şifrelemek için kullanılır.

Şifreleme yapılırken a ve b olmak üzere iki tamsayı seçilir. a ve b tamsayıları şifreleme anahtarını oluşturur. Şifrelenecek metnin bütün karakterleri alfabedeki değerine (sırasına) dönüştürülür.

A=0, B=1, C=2, D=3…

Her bir harf yerine bu işlem yapılır.

E(x) = (ax+b) mod n

Mod aldığımız n değeri, alfabedeki karakter sayısıdır. Örneğin Türkçe alfabenin karakter uzayı olarak alındığı varsayılırsa, n değeri 29 olacaktır.

A harfini şifrelemek istersek ve a=3 ve b=1 seçersek,

0.3+1 = 1

Sonuç 1 çıkar ve bu da B harfine denk gelir. A yerine B yazılması ile şifreleme yapılır. Her harf için tek tek yapılmaktadır.

Affine (VAMSIKRISHNA) = LUAQYMJYQRHU

```
def egcd(a, b):
  x,y, u,v = 0,1, 1,0
  while a != 0:
    q, r = b//a, b%a
    m, n = x-u*q, y-v*q
    b,a, x,y, u,v = a,r, u,v, m,n
  gcd = b
  return gcd, x, y
def modinv(a, m):
  gcd, x, y = egcd(a, m)
  if gcd != 1:
    return None # modular inverse does not exist
  else:
    return x % m
def encrypt(text, key):
  #E = (a*x + b) % 26
  return ''.join([ chr((( key[0]*(ord(t) - ord('A')) + key[1] ) % 26) + ord('A')) for t in text.upper().replace(' ', '') ])
def decrypt(cipher, key):
  #D(E) = (a^-1 * (E - b)) % 26
  return ''.join([ chr((( modinv(key[0], 26)*(ord(c) - ord('A') - key[1])) % 26) +ord('A')) for c in cipher ])
```

```
# Driver Code to test the above functions
def main():
  text = 'VAMSI KRISHNA'
  key = [7, 20]
  # calling encryption function
  enc_text = encrypt(text, key)
  print('Encrypted Text: {}'.format(enc_text))
  # calling decryption function
  print('Decrypted Text: {}'.format(decrypt(enc_text, key) ))
if __name__ == '__main__':
  main()
```

2. Bit Tabanlı Şifreleme Algoritmaları

Blok Şifreleme Algoritmaları

Veriyi bloklar halinde (bloklar birbirine bağımlı veya bağımsız olabilir) işlemektedir. Bu algoritmalarda iç hafıza yoktur.

Bütünlük kontrolü gerektiren uygulamalarda ve şifrelenecek olan tüm verinin (mailin, word dosyasının tamamı gibi) şifreleme işlemi öncesinde mevcut olduğu durumlarda tercih edilir.

Genel olarak blokların büyük olması güvenliği arttırır.

DES, AES, Blowfish, IDEA… blok şifreleme algoritmalarındandır

Blok modda 2 önemli implementasyon metodu vardır:

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC): Daha yaygın ve güvenlidir.

Dizi modunda 3 önemli implementasyon metodu vardır:

- Ciphertext Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR): Daha yaygın ve güvenlidir.

Dizi Şifreleme Algoritmaları

Veriyi bir bit dizisi olarak almaktadır. Bir üreteç aracılığı ve anahtar yardımıyla istenilen değişken uzunlukta (zamana bağlı olarak) kayan anahtar adı verilen bir dizi üretilir.

Telsiz haberleşmesi gibi gürültülü ortamlarda ses iletimini sağlamak için tercih edilir.

Enigma makinesi, WEP, WPA vb. dizi şifrelemeyi baz alarak tasarlanmıştır. RC4, Seal, Vernam vb. dizi şifreleme algoritmalarındandır.

ABD'nin kullandığı SIGABA rotor makinesi, 2. Dünya Savaşı sonrasında Enigma'nın zayıflıkları üzerine tasarlanmış olup, 2*5=10 rotorludur.

DES (Data Encryption Standart)

Dünyada en çok kullanılan simetrik şifreleme algoritmalarından birisidir. Feistel şifreleme metodunu kullanır. Blok şifreleme kullanan DES, işlem sırasında 64 bitlik veriyi 56 bitlik anahtar kullanarak

şifreler. Anahtar uzunluğunun kısa olması nedeniyle kırılmıştır. Bunun üzerine Triple-DES (encrypt-decrypt-encrypt) 3DES olarak geliştirilmiştir. 3DES, DES'in üst üste 3 kere kullanılmasıdır. Yani normal DES'e göre 3 kat yavaştır ama günümüzde SSH gibi uygulamalarda kullanılır. AES'in çıkması üzerine DES popülerliğini kaybetmiştir. Çünkü AES'e göre 6 kat daha yavaştır.

TWOFISH

AES kadar hızlıdır. DES gibi Feistel yapısını kullanır. DES'den farkı anahtar kullanılarak oluşturulan değişken S-boxlara (Blok şifrelerde genellikle anahtar ve şifre metni arasındaki ilişkiyi gizlemek için kullanılırlar.) sahip olmasıdır. Metinleri 32 bitlik parçalara ayırarak işleme sokar ve blok algoritması olarak çalışır. Şifreleme ve deşifreleme algoritmalarının birbirinden farklı olması uygulama maliyetini arttırmış, aynı zamanda yazılım uygulamalarını yavaşlatmıştır.

IRON

Feistel yapısını kullanır. 64 bitlik veri bloklarını 128 bit anahtarla şifreler ve 16 ile 32 döngü sayısında çalışır. Alt anahtarların sayısı döngü sayısına eşittir. Bu nedenden dolayı algoritma anahtar bağımlıdır. Bu algoritmanın avantajı, bitler yerine 16 tabanındaki sayılarda kullanılması, dezavantajı ise yazılım için tasarlanmış olmasıdır.

AES (Advanced Encrption Standart)

DES kırıldıktan sonra yeni bir arayışa girilmiş ve AES simetrik şifreleme algoritması oluşturulmuştur. DES'in zayıf yönleri kuvvetlendirilmiş halidir ve blok şifreleme algoritmasını kullanır. DES'e göre daha hızlı ve güvenlidir. Uzunluk olarak 128, 192 ve 256 bit anahtarları destekler. DES'e göre anahtar boyu ve block size daha uzundur. Bu da daha güçlü bir anahtar sağlar. Günümüzde de en popüler algoritmalardan birisidir ve brute force saldırılarına karşı dayanıklı olduğu düşünülmektedir.

Blowfish

Piyasada kullanılan en hızlı blok şifreleyicilerdendir. Karmaşık anahtar çizelgesi kullanarak kırılmasını zorlaştırır. Blowfish, 23'den 448 bite kadar anahtar uzunluklarına sahiptir. Çalışabilmesi için 4 kilobyte RAM'den daha fazla belleğe ihtiyaç duyarlar. Bu nedenle küçük gömülü sistemlerde kullanılamazlar.

IDEA

Açılımı "International Data Encryption Algorithm" olan IDEA bir blok şifreleme algoritmasıdır. Aynı zamanda Ascom tech adlı firmanın tescilli algoritmasıdır. PGP'nin temelini oluşturan algoritmalardan birisidir. Bilinen en güçlü algoritmalardandır. IDEA, şifrelenecek olan 64 bitlik metin ve 128 bitlik anahtarı kullanarak 64 bitlik şifrelenmiş metni oluşturur.

RC4 (Rivest Encryption 4)

Şifrelenecek veriyi akan bir bit dizisi olarak algılar. RC4 belirlenen anahtar ile veriyi şifreleyen bir algoritmadır. Genellikle hız gerektiren uygulamalarda kullanılır. Şifreleme hızı yüksektir ve MB/sn seviyesindedir. Güvenliği rastgele bir anahtar kullanımına bağlıdır. Anahtar uzunluğu değişkendir. 128 bitlik bir RC4 şifrelemesi sağlam bir şifreleme olarak kabul edilir. Bankacılık ve Dökümantasyon (PDF) şifrelemelerinde yaygın olarak kullanılır.

RC5 (Rivest Encryption 5)

Modern şifreleme algoritmaları sınıfında yer almaktadır. 16, 32 ve 64 bitli kelime uzunlukları ile çalışabilmektedir. Anahtar boyutu ve döngü sayısı değişken olarak alınabilir. Böylece, yüksek anahtar boyutu ve fazla döngü sayısı ile uzun çalışma zamanı fakat kırılması neredeyse imkânsız şifrelerdir. Düşük anahtar boyutu ve az döngü sayısı ile kısa çalışma zamanı ve bununla beraber daha güçsüz

şifreler arasında seçim yapılabilme olanağını sağlar. Bellek gereksiniminin de düşüklüğü ile cep telefonlarından süper bilgisayarlara kadar her yerde çalışabilir bir algoritmadır.

CAST-128

GPG ve PGP'nin bazı versiyonlarında varsayılan şifre olarak birçok üründe kullanılan simetrik bir anahtar bloğu şifresidir. 64 bit blok boyutu ve anahtar boyutu 40 ile 128 bit arasında olan (ancak yalnızca 8 bitlik artışlarla) 12 veya 16 yuvarlak Feistel bir ağdır. Anahtar boyutu 80 bit'ten uzun olduğunda 16 turun tamamı kullanılır.

MD5 (Message Digest 5, Mesaj Özeti 5)

Verilen dosyanın veya mesajın kendine has parmak izinin oluşturulmasını hash fonksiyonlarına dayalı olarak sağlayan bir algoritmadır. Bir veritabanı yönetimi (database management) tekniğidir. 1991 yılında MIT (Massachusetts Institute of Technology)'de görev yapan Profesör Ron Rivest tarafından geliştirilmiştir. Profesör Rivest MD5'i MD4'ün bir üst sürümü olarak tasarlamıştır. Şifrelenecek metinden 1 karakterin değiştirilmesi, şifrelendikten sonraki hash değerinin tamamen değişmesine yol açabilmektedir. Bu yüzden şifrelenecek metnin iyi bir kontrolden geçmesi gerekmektedir.

MD5 değişken uzunluktaki bir mesajı 128 bitlik bir sabit uzunlukta çıktı olarak işler.

Öncelikle veri 512 bitlik bloklara ayrılır ve her bir bloğa aynı işlem uygulanır. (On altı tane 32-bitlik kelimeler halinde)

İşleme alınacak verinin 512 bit ve katları olması gerekmektedir. Eğer 512 bit ya da katlarından biri değilse ekleme yapılır. Bu işleme padding de denmektedir. Ekleme işleminin kuralları;

512 bitin en yakın katından 64 eksik olacak şekilde verinin binary değerinin sonuna bir adet 1 ve geri kalan kısma ise 0 eklenir. Bu 64 bitlik fark, verinin uzunluğunu belirtmek için kullanılır. Geriye kalan 64 bite de orijinal mesajın uzunluğu mod 2^64 'de yazılır.

Ekleme işleminden sonra MD5 veriyi işlemeye başlar.

A, B, C ve D olarak adlandırılan 4 adet 32 bitlik kelimeyle ayrılmış 128 bitlik parçalar üzerinde çalışır. Bunlar belirli sabit değerlerle başlatılır. Daha sonra ana algoritma, her 512-bit ileti bloğunu durumunu (128 bit) değiştirmek için kullanır. Bir mesaj bloğunun işlenmesi, tur denilen dört benzer aşamadan oluşur; Her tur, doğrusal olmayan bir fonksiyon, modüler toplama işlemi ve bit bazında sola kaydırma işlemlerinden oluşur. Toplamda 16 tur vardır. Figür 1'de her tur içinde yapılan işlemler gösterilmiştir.4 olası F fonksiyonu vardır; her turda farklı bir fonksiyon kullanılır.

Kullanıldığı yerler

- İnternet trafiğinde. "SSL (Secure Sockets Layer - Güvenli Yuva Katmanı)" gibi.
- Özel bilgisayar ağlarında. "VPN (Virtual Private Network- Sanal Özel Ağ)" gibi.
- Güvenli uzaktan ulaşım uygulamalarında. "SSH (Secure Shell- Güvenli Kabuk)" gibi.
- Kimlik belirleme uygulamalarında.
- MD5 algoritması, üzerinde işlem yapılan dosyada herhangi bir değişiklik olup olmadığını tespit eder. Eğer bir değişiklik yapılmışsa, iletilen dosyanın MD5 hash algoritmasından çıkan sonuç ile ilk dosyanın MD5 hashinin sonucu birbirinden farklı olacaktır.
- MD5'ın bir diğer kullanımı da public-key şifrelemededir. Public-key şifreleme, simetrik şifrelemeye göre çok daha fazla hesap gücü ve zaman gerektirdiğinden Public-key sistemlerde bile aslında Simetrik standart şifreleme kullanılır. Daha sonra veri MD5 gibi bir hashten geçirilir ve bu kısa hash değeri asıl olarak asimetrik şifreleme ile şifrelenir. Bu sayede performans ile güvenlik arasında bir denge sağlanmış olur.

Dezavantajları

- Kullanıcı adı-şifre ile girilen sitelerde, şifrenin unutulduğu durumlarda sistem eski şifreyi geri veremez. Şifre MD5 algoritmasından geçirilip saklandığı için sistem kullanıcıya yeni şifre vererek sorunu çözer.
- MD4 e göre yavaş çalışması zaman bakımından dezavantaj olarak sayılabilir.

MD5 algoritmasının Kırılması İçin Yapılan Proje Çalışmaları

1. **RainbowCrack**

RainbowCrack projesi; büyük harf, küçük harf, sayı ve özel karakterlerin kendi arasında oluşturabileceği tüm olasılıklar düşünülerek hazırlanmış, 1 karakterli olanlardan sonsuz karakterli oluşturulan şifrelemelere kadar hepsinin MD5 algoritmasıyla şifrelenmiş hallerinin bir tabloda biriktirilmesidir. Belirlenmiş hashler çözümlemelerde kullanılabilmektedir fakat çok fazla olasılık olduğundan ve çok büyük boyutta verilerin saklanması kolay olmadığından bitirilememektedir.

2. **BruteForce Saldırıları**

BruteForce saldırıları, veri tabanında bulunan MD5 algoritmasından geçirilmiş şifreleri, tahminler yürüterek önceden hazırlanmış karakter setli algoritmalar (dictionary) ile bulmaya çalışan bir saldırı türüdür. Saldırı şekli deneme yanılma şeklindedir.

3. **Çakışmalar**

MD5 işlemlerinden sonra hash değerlerinde çakışma olduğunda yani 2 veri aynı şifreye sahip olduğunda, verinin hash tablosundaki yeni yerinin hesaplanabilmesi için doğrusal sınama, ikinci dereceden sınama ya da ikili sınama yöntemlerinden biri uygulanır. Doğrusal sınamada; veri hash tablosunda hemen bir sonraki yere yerleştirilir. İkili sınamada; şifrenin bulunduğu yerin nümerik karesi alınarak yeni yer belirlenir. İkinci dereceden sınamada ise; 2 hash fonksiyonu iç içe kullanılır. Eğer belli bölgelerde birikme olmuşsa buna kümelenme denir. Zaten sınama yöntemlerindeki amaç da kümelenmeyi önlemektir. Ayrıca homojen dağılım olması için hash tablolarının büyüklüğü asal sayı tercih edilmelidir.

128 bit (16 baytlık) MD5 hashleri (ileti özetleri olarak da adlandırılır) genellikle 32 hexadecimal sayı ile gösterilir.

MD5(ingilizler ben izlemem.) = 2837552910f7b935e719bbe53e1c5f7c

MD5(ingilizler ben izlemek.) = 8c21543d008d5c63d83ec24841bb35b8

Asimetrik Şifreleme

Şifrelemede ve şifre çözmede farklı anahtarlar kullanılır. Bu anahtarlar genel/açık (public) ve özel/kapalı/gizli (private) anahtar olarak geçmektedir.

Genel anahtar şifreleme ve doğrulama için kullanılırken, özel anahtar ise şifre çözme ve imzalama için kullanılmaktadır.

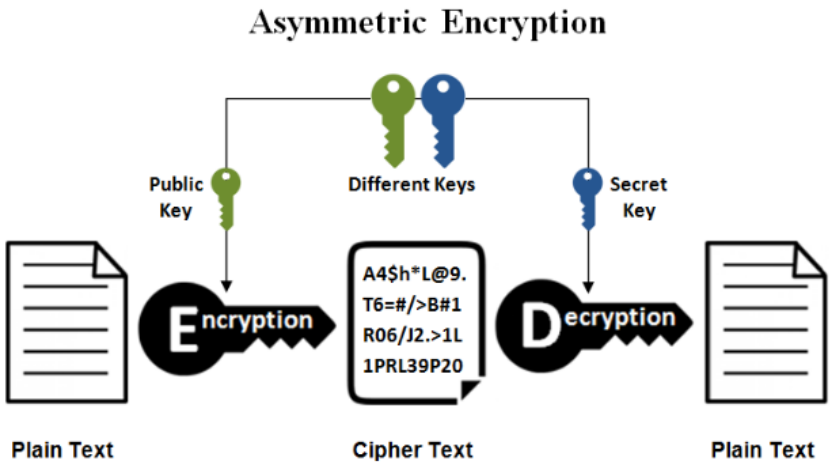
Alıcının genel anahtarı ile şifrelenen veri, sadece alıcının özel anahtarı ile açılabilir. Bu sebeple gönderici taraflar, alıcının genel anahtarı ile şifreleyerek veriyi gönderir. Bu şifreli verinin, sadece özel anahtara sahip alıcı tarafından okunabileceğinden emin olunur.

Göndericinin özel anahtarı ile imzalanan veri, alıcı ve herkes tarafından, göndericinin genel anahtarı ile doğrulanabilir. Bu sebepler gönderici taraf, kendi özel anahtarını kullanarak veriyi imzalar. Bu imzalı verinin, sadece özel anahtara sahip gönderici tarafından gönderildiğinden emin olunur.

Genel anahtar herkese açılabilirken, özel anahtar sadece sahibinde bulunmalıdır. Bir başkasının gizli anahtarını elde eden bir saldırgan, o kullanıcının genel anahtarı ile şifrelenmiş verilerini okuyabilir veya o kullanıcıdan gönderiliyormuş gibi imzalı veriler gönderebilir.

Özel anahtar kullanılarak genel anahtar elde edilebilirken, genel anahtar kullanılarak özel anahtar elde edilemez. Bu sebeple, genel anahtarın başkasının elinde olmasının bir önemi yoktur. Diğer taraftan, özel anahtarın kaybedilmemesi büyük önem taşır.

Asimetrik algoritmalar gizlilik, inkâr edilemezlik (imzalama) ve anahtar paylaşımı için kullanılırlar. Asimetrik algoritmalar simetrik algoritmalara göre çok daha yavaş çalışırlar. Bu sebeple asimetrik algoritmalar güvenilir anahtar değişimi için kullanılırlar. Bu anahtar, simetrik şifreleme için kullanılarak verinin güvenilir olarak iletilmesi sağlanabilmektedir. En yaygın kullanılan asimetrik algoritmalar, RSA (Rivest-Shamir-Adleman), DSA (Digital Signature Algorithm) ve Eliptik eğri algoritmasıdır.



Avantajları

Asimetrik şifreleme, simetrik şifrelemedeki iki ana probleme çözüm bulmuştur.

- Anahtar dağıtım sorununa yeni bir bakış açısı getirilmiştir. Bunun için iki anahtarlı bir yapı kullanılmaktadır. Herkese açık olan bir genel anahtar, sahibinde bulunması gereken özel anahtar
- Kimlik doğrulama problemi giderilmiştir. Bunun için doğrulama sistemleri (Kişi doğrulama – User Authentication) kullanılmaktadır.

Dezavantajları

Yavaş olduğundan dolayı kullanılan sistemlerde çok fazla CPU harcanmasına sebep olmaktadır.

(D-H) Diffie-Hellman anahtar değişimi

Diffie ve Helman tarafından bulunmuş ilk asimetrik şifreleme algoritmasıdır. DH iki katılımcının öncesinde herhangi bir bilgi alışverişi yapmadan güvenli olmayan bir kanal vasıtasıyla (güvenli bir şekilde) ortak bir şifrede karar kılmalarına yarayan bir protokoldür. Algoritma anahtar değişimi ile asıl amacı, iki kullanıcının bir anahtarı güvenli bir şekilde birbirlerine iletmeleri ve daha sonrasında da bu anahtar yardımı ile şifreli mesajları birbirlerine gönderebilmelerini sağlamaktır. Diffie–Hellman algoritması oluşturularak simetrik şifreleme algoritmaları için büyük problemi olan gizli anahtarı koruma ve dağıtım büyük ölçüde aşılmıştır. Bununla birlikte Diffie-hellman algoritması sadece ortak gizli anahtarı belirlemekte kullanılmaktadır.

RSA (Rivest-Shamir-Adleman)

Üç bilim adamının baş harflerinden oluşan RSA, dijital imzalama içinde kullanılmaktadır. Güvenilirliği, çok büyük asal sayıların işlem yapma zorluğuna dayanan bir algoritmadır. Günümüzde bankacılık sistemleri ve ticari sistemlerde öncelikli tercih edilen şifreleme tekniğidir. Bu büyük sayılar nedeniyle oldukça güvenilirdir ama işlemler yavaştır. Bu nedenle fazla bant genişliği harcaması yüzünden kablosuz ağ sistemlerinde kullanılması bazı sorunlara yol açabilir.

El Gamal

Diffie-Hellman anahtar alışverişine dayanan bir açık anahtarlı şifreleme yöntemidir. Anahtar üretimi ve şifreleme/açma olarak iki aşamadan oluşur. Matematiksel zorluk olarak dairesel gruplar üzerindeki ayrık logaritmalara dayanan bir dijital imzadır.

DSA (Digital Signature Algorithm)

NIST tarafından sayısal imza standardı olarak tasarlanmıştır. DSA algoritması da, RSA gibi açık anahtarlı bir kriptografik algoritmadır. Dijital imza algoritması, ElGamal imza algoritmasının bir varyantıdır.

Merkle-Hellman

RSA asimetrik şifreleme sisteminden farkı şifreleme işleminin tek yönlü çalışmasıdır. Açık anahtar sadece şifreleme yaparken, gizli anahtar sadece şifre çözme işlemini gerçekleştirir. Bu nedenle de Dijital İmzalama için kullanılamaz. Düşünce olarak RSA'dan daha basit ve zekice olmasına rağmen kırılmıştır.

(ECC) Elliptic Curve Cryptography

Eliptik Eğri Kriptolojisi (Elliptic Curve Cryptography), sonlu cisimler üzerindeki eliptik eğrilerin cebirsel topolojisine dayanan bir açık anahtar şifrelemesidir. Eliptik eğri kriptografisinin en büyük özelliği depolama ve iletme gereksinimlerini azaltarak daha küçük anahtar boyutuna sahip olmasıdır. Bir eliptik eğri grubu, büyük modülerli ve buna bağlı olarak büyük anahtar boyutlu RSA tabanlı sistem ile aynı güvenlik seviyesi sunabilir. Örneğin; Eliptik eğri ile 256-bitlik anahtar boyutunda elde edeceğimiz güvenliği RSA 'de 3072-bitlik anahtar ile sağlanabilir. Bu algoritma IHA'larda güvenlik açısından kullanılabilir. Ayrıca ECC smart kartlar,cep telefonları, PDA'lar (personal digital assistant), sayısal posta işaretleri gibi zorunlu ortamlara uygundur.

Bazı güvenlik sistemleri 1024-bit RSA genel anahtarlama planının uygulamasını yaymaya çalışır, çünkü kuruluşlar bunun yeterince iyi olduğunu düşünürler. Bununla birlikte bu tehlikeli bir yaklaşımdır. Çünkü genel anahtarlama sisteminin güvenliği kullanılan simetrik şifrelemeyle birebir eşleşmiş olmalıdır. Tabloda görüldüğü gibi, 1024-bit RSA simetrik şifrelemede kullanılan 128-bit güvenlik seviyesiyle uyuşmuyor.Bu gereksinimi karşılamak yani genel anahtarlama planını eşleştirmek için istenen 3072- bit RSA ya da 256-bit ECC kullanılmasıdır. Bu sayede işlemci gücü, saklama kapasitesi, bant genişliği, güç tüketimi gibi durumlarda RSA'ya göre avantaj sağlar.

Asimetrik Şifreleme ve Simetrik Şifreleme Arasındaki Farklar

- Simetrik şifreleme algoritmaları güvenli bir anahtar değişimine ihtiyaç duyarlar, asimetrik şifreleme algoritmaları için böyle bir durum yoktur.
- Simetrik şifreleme algoritmaları tek bir anahtar kullanırlar, asimetrik şifreleme algoritmaları iki anahtar kullanırlar.
- Simetrik şifreleme algoritmaları oldukça hızlı bir şekilde çalışırlar, şifrelemede kullanılan anahtar boyu (bit sayısı) asimetrik şifreleme algoritmalarına göre oldukça düşüktür.

- Asimetrik şifreleme algoritmaları ile bütünlük, kimlik doğrulama ve gizlilik güvenli bir şekilde sağlanabilir.
- Asimetrik şifreleme algoritmalarında public (açık) anahtar herkesin erişimine açıktır bu sayede az sayıda anahtar kullanımı ile çok sayıda kullanıcıya sahip uygulamalarda kapasite sorunun önüne geçebilir.

SHA (Secure Hash Algorithm)

Amerika'nın ulusal güvenlik kurumu olan NSA tarafından tasarlanmıştır.

SHA-0, 1, 2 ve 3 olarak dörde ayrılır. SHA 256, 384, 512 gibi çeşitli bit uzunluklarına sahip olan protokoller, özellikle e-imza ve POS cihazı gibi bankacılık uygulamalarında tercih edilir.

SHA-1, 160 bit özet değeri üreten bir fonksiyona sahiptir. E-posta güvenliği için PGP uygulamalarında, web sitesi güvenliği için ise SSL sertifikalarında kullanılmaktadır. Tek yönlü şifreleme algoritmasına sahiptir. SHA-1 protokolünde özet değerleri 40 basamaklı onaltılık sayılar halinde üretilir. Bu algoritmada çeşitli güvenlik açıklıkları tespit edildiği için tarayıcılar tarafından desteklenmemektedir.

SHA yazılım ile kodlanan bilgiler, anlamsız karakterler kümesi halinde görünür. Bu algoritmalarla şifrelenen veriler, aynı altyapıya sahip olan decoder aracılığıyla çözülür.

SHA algoritmaları, geri dönüştürme işlemine karşı güçlü bir altyapıya sahiptir. Bilginin kendisini değil özetini kriptolayan sistem, iletişim ve operasyonel maliyetleri daha düşük seviyelere indirebilmenize yardımcı olur. Mesaj doğrulama, parola saklama, sayısal imza, veri gizliliğini sağlama ve veri boyutunu küçültme Secure Hashing Algoritması protokollerinin en çok kullanıldığı alanlar arasında yer alır.

SHA-2'nin Farkı Nedir?

SHA-2 Dijital imzalar gibi çakışma direnci bulunan alanlarda tercih edilir.

SHA-2; SHA-224, 256, 384, 512, 512/224, 512/256, 256 ve 512 olmak üzere altı fonksiyondan oluşur. Bu protokoller 32 ve 64 bitlik kelimelerle kriptolanan algoritmalardır. Birbirinden farklı değişkenlere sahip olsa da yapıları neredeyse aynıdır. SHA-512/256 ve SHA-512/224 özet fonksiyonları, SHA-512'nin kesilmiş versiyonlarıdır. Bu algoritmaların başlangıç değerleri Federal Bilgi İşleme Standardına göre belirlenmiştir.

SHA-2 algoritması, Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından 2001 yılında geliştirilmiştir. SSL, TLS, SSH, PGP, S/MIME, IPsec uygulamalarında kullanılan bu güvenlik ve kriptolama sistemi hem Unix hem de Linux üreticileri tarafından da tercih edilen bir altyapıya sahiptir. SHA-1'e göre daha hızlı çalışan bir fonksiyondur.

SHA-1'de tespit edilen güvenlik açıklarından dolayı Bitcoin ve Ethereum gibi birçok kripto para biriminde doğrulama yapmak, hesaplama yapmak ya da para gönderme işlemini kanıtlamak için SHA-2 standardı kullanılmaya başlanmıştır. Onaltılık sayı sisteminde 64 karakterlik bir çıktı verdiği için bu tür işlemlerde genel olarak SHA-256 protokolü tercih edilir.

Güvenlik açıklarından dolayı Google, Microsoft ve Mozilla gibi tarayıcılar SHA-1'i desteklemeyi bırakmışlardır. Bu protokol yerine SSL sertifikalarında SHA-2 özetleme protokolü kullanılmaya başlanmıştır. Hız, güvenlik, gizlilik açısından iyi bir performans sergileyen bu algoritmalar, ödeme sistemleri ve sosyal medya platformları gibi büyük çevrim içi hizmetler veren uygulamalarda da tercih edilmektedir.