**GELİŞMİŞ ŞİFRELEME STANDARDI**

**AES'in Kökenleri**

3DES'in temel dezavantajı (1999'da Federal Bilgi İşleme Standardı FIPS PUB 46-3'te 168 bit anahtarla yeni standart olarak önerildi), algoritmanın yazılımda nispeten yavaş olmasıdır. İkincil bir dezavantaj, 64 bit blok boyutunun kullanılmasıdır. Hem verimlilik hem de güvenlik nedenleriyle, daha büyük bir blok boyutu arzu edilir.

1997'de Ulusal Standartlar ve Teknoloji Enstitüsü NIST, 3DES'e eşit veya daha iyi güvenlik gücüne ve önemli ölçüde geliştirilmiş verimliliğe sahip olması gereken yeni bir Gelişmiş Şifreleme Standardı (AES) için bir teklif çağrısı yayınladı. Ek olarak NIST, AES'nin 128 bit blok uzunluğuna ve 128, 192 ve 256 bit anahtar uzunluklarını destekleyen simetrik bir blok şifre olması gerektiğini de belirtti.

İlk değerlendirme turunda önerilen 15 algoritma kabul edildi. 2. tur 5 algoritmaya indirildi. NIST, değerlendirme sürecini tamamladı ve Kasım 2001'de nihai bir standart (FIPS PUB 197) yayınladı. NIST, önerilen AES algoritması olarak Rijndael'i seçti. AES'in 2 araştırmacısı Dr. Joan Daemon ve Dr. Vincent Rijmen'dir Belgium.

AES Değerlendirmesi

**Güvenlik** – 128 minimum anahtar boyutu, yeterli güvenlik sağlar

**Maliyet** – AES, yüksek hesaplama verimliliğine sahip olmalıdır

**AES'in Kökenleri (Devam 1)**



Algoritma ve uygulama özellikleri - bu, esneklik, donanım ve yazılım uygulamalarına uygunluk, basitlik gibi çeşitli hususları içerir.

**AES'in Kökenleri (Devamı 2)**



Ek kriterler şunları içerir: genel güvenlik, yazılım uygulamaları, kısıtlı alan ortamları, donanım uygulamaları, uygulamaya yönelik saldırılar (zamanlama saldırıları), şifrelemeye karşı şifre çözme, anahtar çevikliği, esneklik, yönerge düzeyinde paralellik potansiyeli.

**AES'in Kökenleri (Devamı 3)**



**AES'in Kökenleri (Devam 4)**



**AES ŞİFRESİ**



Bir dizi AES parametresi anahtar uzunluğuna bağlıdır (Tablo 5.3). Bu bölümün açıklamasında, anahtar uzunluğunun 128 bit olduğunu varsayıyoruz.

Şekil 5.1, AES'nin genel yapısını göstermektedir.

**AES'İN GENEL YAPISI**





Şifreleme ve şifre çözme algoritmasının girdisi, tek bir 128 bitlik bloktur; bu blok, FIPS PUB 197'de, kare bir bayt matrisi olarak tasvir edilmiştir. Bu blok, şifreleme veya şifre çözmenin her aşamasında değiştirilen Durum dizisine kopyalanır. Son aşamadan sonra Durum, bir çıkış matrisine kopyalanır. Bu işlemler Şekil 5.2a'da gösterilmektedir:

**AES'İN GENEL YAPISI (Devam 1)**



Benzer şekilde, 128 bit, kare bir bayt matrisi olarak tasvir edilir. Bu anahtar, bir anahtar program sözcükleri dizisine genişletilir; her kelime 4 bayttır ve toplam anahtar programı, 128 bitlik anahtar için 44 kelimedir (Şekil 5.2b). Bir matris içindeki baytların sıralaması sütuna göredir.

Ayrıntılara girmeden önce, genel AES yapısı hakkında birkaç yorum yapabiliriz:

1. Bu şifre bir Feistel yapısı değildir.
2. Girdi olarak sağlanan anahtar, w[i] olmak üzere 44 kelimelik (her biri 32 bit) bir diziye genişletilir. 4 ayrı kelime (128 bit), her tur için bir yuvarlak anahtar görevi görür; bunlar Şekil 5.1'de gösterilmiştir.
3. 1 permütasyon ve 3 ikame olmak üzere 4 farklı aşama kullanılır:
* Yedek bayt - Bloğun bayttan bayta değiştirilmesini gerçekleştirmek için bir S-box kullanır
* Satırları kaydır - Basit bir permütasyon
* 8 ) yerine aritmetik kullanan bir ikame .
* Yuvarlak anahtar ekle - Genişletilmiş anahtarın bölümü ile geçerli bloğun basit bir bitsel XOR'u
1. Yapı oldukça basittir. Şekil 5.3, tam bir şifreleme turunun yapısını göstermektedir.

**AES'İN GENEL YAPISI (Devamı 2)**



1. Yalnızca Yuvarlak Anahtar Ekle aşaması anahtarı kullanır. Anahtar bilgisi olmadan başka herhangi bir aşama tersine çevrilebilir.
2. Add Round Key, bir Vernam şifresi biçimidir ve kendi başına zorlu olmaz. Diğer 3 aşama birlikte karışıklık, yayılma ve doğrusal olmama sağlar, ancak anahtarı kullanmadıkları için kendi başlarına güvenlik sağlamazlar. Şifreyi, sırayla değişen XOR şifreleme işlemleri (Add Round Key) ve ardından bloğun karıştırılması olarak görebiliriz.
3. Her aşama kolayca tersine çevrilebilir
4. Şifre çözme aynı anahtarları ancak ters sırada kullanır. Şifre çözme, şifreleme ile aynı değildir
5. Şekil 5.1'deki her bir yatay noktada (örn. kesikli çizgi), Durum hem şifreleme hem de şifre çözme için aynıdır
6. Hem şifreleme hem de şifre çözmenin son turu yalnızca 3 aşamadan oluşur; AES'nin özel yapısının sonucudur.

**AES'İN GENEL YAPISI (Devamı 3)**

indirgenemez polinom ile GF(2 8 ) sonlu alanında aritmetik kullanır .

**Yedek Bayt Dönüşümü. İleri ve Ters Dönüşüm**

SubBytes olarak adlandırılan İleri ikame bayt dönüşümü, basit bir tablo aramasıdır (Şekil 5.4a).



**Yedek Bayt Dönüşümü. İleri ve Ters Dönüşüm (Devamı 2)**

AES, olası tüm 256 8 bitlik değerlerin bir permütasyonunu içeren, S-kutusu (Tablo 5.4a) adı verilen 16x16'lık bir bayt değerleri matrisi tanımlar. Durumun her bir baytı, aşağıdaki şekilde yeni bir bayta eşlenir: En soldaki 4 bit, bir satır değeri olarak kullanılır ve en sağdaki 4 bit, bir sütun değeri olarak kullanılır. Bu satır ve sütun değerleri, benzersiz bir 8 bitlik çıkış değeri seçmek için S-kutusuna dizinler olarak hizmet eder. Örneğin, {95} onaltılık değeri, {2a} değerini içeren S-kutusunun 9. satırı, 5. sütununa başvurur:

**Yedek Bayt Dönüşümü. İleri ve Ters Dönüşüm (Devamı 3)**



S-kutusu aşağıdaki şekilde oluşturulmuştur:

1. S-kutusunu bayt değerleriyle satır satır artan düzende başlatın. Böylece, x satırı, y sütunundaki baytın değeri {xy} olur.

**Yedek Bayt Dönüşümü. İleri ve Ters Dönüşüm (Devam 4)**

1. S kutusundaki her baytı sonlu alan GF(2 8 ) içindeki çarpımsal tersiyle eşleyin; {00} değeri kendisine eşlenir.
2. S kutusundaki her baytın (b7,b6,b5,b4,b3,b2,b1,b0) etiketli 8 bitten oluştuğunu düşünün. S kutusundaki her baytın her bir bitine aşağıdaki dönüşümü uygulayın:

 (5.1)

burada c i , c baytının {63} değerine sahip i. bitidir, yani (c7c7c5c4c3c2c1c0)=(01100011). Asal , değişkenin sağdaki değerle güncelleneceğini gösterir. AES standardı bu dönüşümü matris biçiminde şu şekilde gösterir:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B 0 ' |  | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |  | B 0 |  | 1 |  |
| B 1 ' |  | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |  | B1 \_ |  | 1 |  |
| 2 ' \_ |  | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |  | B2 \_ |  | 0 |  |
| 3 ' \_ | = | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | X | B3 \_ | + | 0 | (5.2) |
| 4 ' \_ |  | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |  | B4 \_ |  | 0 |  |
| B 5 ' |  | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |  | B5 \_ |  | 1 |  |
| B 6 ' |  | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |  | B6 \_ |  | 1 |  |
| B 7 ' |  | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |  | B7 \_ |  | 0 |  |

Çarpım matrisindeki her öğe, bir satır ve bir sütundaki öğelerin bit düzeyinde XOR'udur. Ayrıca, (5.2)'de gösterilen son ekleme, bit düzeyinde bir XOR'dur.

Örnek olarak, {95} giriş değerini ele alalım. GF(2 8 ) 'deki çarpımsal ters, ikili sistemde 10001010 olan {95} -1 ={8a}'dır. Denklem (5.2) kullanılarak,

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |  | 0 |  | 1 |  | 1 |  | 1 |  | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |  | 1 |  | 1 |  | 0 |  | 1 |  | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | X | 1 | + | 0 | = | 1 | + | 0 | = | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |  | 0 |  | 1 |  | 0 |  | 1 |  | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |  | 0 |  | 1 |  | 1 |  | 1 |  | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |  | 1 |  | 0 |  | 0 |  | 0 |  | 0 |

Sonuç, S-box'ın satır {09} sütununda {05} görünmesi gereken {2a}'dır. Bu, Tablo 5.4a kontrol edilerek doğrulanır.