Bilgisayar Mühendisliği Bölümü

BİL 470

Kriptografi ve Bilgisayar Güvenliği

Güz 2021 – 2022

Dönem Projesi Raporu

01.01.2022

Esra Eryılmaz

171044046

ARAŞTIRMA

Hafif sıklet şifreleme algoritmaları sınırlı kaynaklara sahip cihazlar üzerinde güvenli bir haberleşme sağlanması için tasarlanan hafif işlem yüküne sahip tasarımlardır.

Özellikle Nesnelerin interneti gibi kavramların gelişmesi ile hayatımızın her alanında internete bağlı çok fazla sayıda cihaz kullanılmaya başlanmıştır. Bu durum güvenlik ve kişisel mahremiyetin korunmasını çok daha önemli bir duruma getirmiştir.

IoT ağında kullanılan cihazlar genellikle düşük işlem gücüne, hafızaya ve enerji kaynağına sahip cihazlardan oluşmaktadır. Bu sebeple bu cihazlar üzerinde çalışacak olan program ve algoritmaların daha az işlem yükü ve bellek gereksinimine sahip, daha az enerji tüketecek şekilde tasarlanması gerekmektedir. Bununla birlikte kullanılacak olan algoritmaların güvenlik seviyesinden ödün vermemeleri istenmektedir. Geleneksel standart olarak kabul edilen algoritmaların bu platformda kullanılması durumunda aşırı işlem yükünden dolayı gecikmeler, fazla enerji tüketimi ve bellek gereksinimlerinin yetersiz kaldığı görülmektedir.

Hafif sıklet kriptoloji güvenlik zafiyetine sebep olmadan, düşük maliyetli ve performanslı şifreleme gerçekleştirmeyi amaçlamaktadır.

Literatürde çok sayıda hafif sıklet şifreleme algoritması tasarımı bulunmaktadır. Literatürdeki algoritmalar incelendiğinde, anahtar uzunlukları, şifre blok boyutları, mimarileri ve döngü sayısı gibi özellikler açısından birbirinden oldukça farklı oldukları görülmektedir.

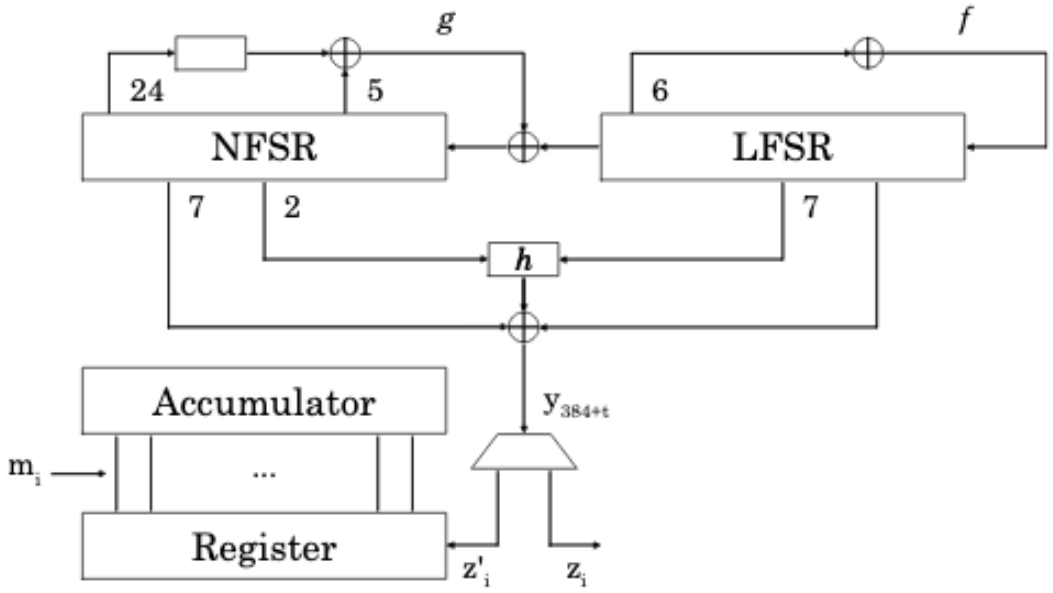
Grain 128-AEAD → stream cipher based simetrik şifreleme algoritmasıdır

Photon-Beetle → permutation based simetrik şifreleme algoritmasıdır

Grain128-AEAD

Grain-128AEAD, ilişkili verilerle kimliği doğrulanmış şifrelemeyi destekleyen bir akış şifresidir. Şu anda NIST hafif kripto standardizasyonu sürecindeki şifrelemelerinden biridir.

Grain-128AEAD iki ana yapıtaşından oluşur. Birincisi, Linear Feedback Shift Register (LFSR), Non-linear Feedback Shift Register (NFSR) ve bir ön çıktı işlevi kullanılarak oluşturulan bir ön çıktı üretecidir, ikincisi ise bir kaydırma yazmacı ve bir akümülatörden oluşan bir kimlik doğrulama üreticisidir.



Grain-128AEAD, değişken uzunluklu düz metin ve değişken uzunluklu ilişkilendirme alır. 96 bit boyutunda sabit uzunluklu bir nonce ve sabit uzunluklu bir anahtar 128 bit boyutunda.

Çıktı, değişken uzunlukta bir şifreli metindir. Düz metin geçerli bir şifreli metinden kurtarılır.

Tek bir anahtar için nonce benzersiz olmalıdır. Nonce benzersiz değilse aynı anahtar için tekrarlanır, algoritma bilgi sızdırır. İki düz metin ve MAC sahte olabilir.
Grain-128AEAD, yalnızca desteklenen parametrelere sahip bir algoritmadır.

Grain-128AEAD, nesnelerin interneti (IoT) ve gömülü sistemler için çok uygun olabilir. Grain-128AEAD ve emsal sürümlerinin güçlü avantajları, endüstriyel olarak uygun görülebilir.

TESTLER :

```
        Grain 128-AEAD cipher
key :           00000000000000000000000000000000
accum init:     c0207f221660650b
register init:  6a952ae26586136f
pre-output :    a0904140c8621cfe8660c0dec0969e9436f4ace92cf1ebb794663f17ab8341ee
keystream :     c800a52f948b89b85cee6cfd8571f90f
macstream :     0498886e288e86666e292d976a77119a
tag :           aab555c073e67664
DONE!
```

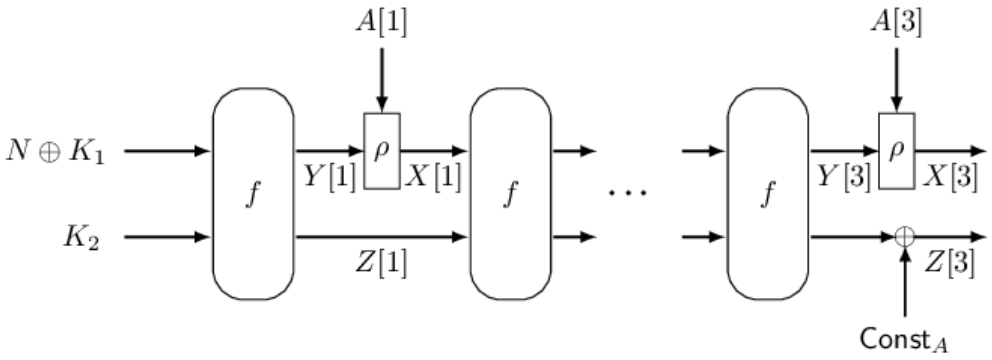
```
PS C:\Users\ESRA\Desktop\171044046\grain128aead> gcc -o g grain128aead.c
PS C:\Users\ESRA\Desktop\171044046\grain128aead> ./g
        Grain 128-AEAD cipher
key :           0102030405abcdef060708099abcdef0
accum init:     7915bb95455f9d68
register init:  1a38f54d886585fb
pre-output :    2168a323bc7c88b8005ed3aa57fe5af2a95d9680d4f91e12c3464bc31227719c
keystream :     46d5e6ae039f1f3de2988e319139154a
macstream :     18116e040ed0fecc1f60ed649a9943d6
tag :           632d4ed8cd3a1893
DONE!
```

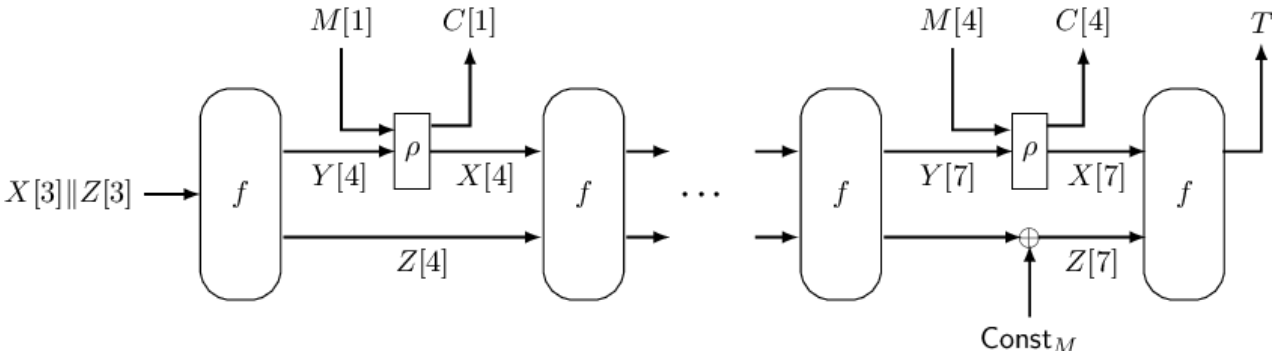
```
        Grain 128-AEAD cipher
key :           0123456789abcdef123456789abcdef0
accum init:     f88720c13f228e98
register init:  8f077fe53cf34e09
pre-output :    e108ca9703c0f4799a99ab3cb6ad8c40947d73191540b2202d843781d84b2adb
keystream :     c2b918c6baf6dea0865200d46858a37b
macstream :     908718ed451663286fd578403271c90d
tag :           77805f2403d1c091
DONE!
```

Photon-Beetle

Photon-Beetle, hafif kriptografi için NIST yarışmasının 2. Turuna ulaşan kripto yöntemlerinden biridir. Sponge authenticated encryption ve sponge hash kullanır ve 256 bitlik bir permütasyona sahiptir.

Genel olarak hafif bir blok şifrelemedir.





Özellikleri :

- Yüksek Güvenlik Sınırı : Photon-Beetle, yüksek güvenlik sınırına ulaşır. Bu da durum boyutunu en aza indirerek modu hafif hale getirir. Aslında tasarımda yalnızca 256 bitlik bir permütasyon kullanılır. Geleneksel dubleks sünger üzerinde birleşik bir geri bildirim tekniği kullanarak güvenliği artırma yaklaşımını izlemektedir.
- Son Derece Esnek Mod : Photon-Beetle yüksek esneklik sağlar. Bu yapıya herhangi bir permütasyonu sığdırmak kolaydır.

- Tek Durum : Photon-Beetle, temel permütasyonun blok boyutu kadar küçük bir durum boyutuna sahiptir ve hem hafif hem de yüksek performanslı platformlarda iyi uygulama özellikleri sağlar.
- Ters-Serbest : Photon-Beetle, tersten bağımsız, kimliği doğrulanmış bir şifreleme algoritmasıdır. Hem şifreleme hem de şifre çözme algoritmaları, temeldeki ince ayarlı blok şifresine herhangi bir şifre çözme çağrısı gerektirmez.
- Neredeyse Yüksüz : Permütasyon çağrısı dışında , veri bloğu başına sadece *2r-bit XOR + r/2-bit* durumunun 1 bit sağa dönüşü gerektirir , bu çok küçük bir ek yük gibi görünmektedir.
- Süper Optimal : Photon-Beetle, bir m blok mesajını işlemek için yalnızca (m-3) birçok ilkel çağrı gerektirir. Bir blok ilişkili veri ve bir m blok mesajı için yalnızca a+m+1 permütasyon çağrılarını gerektirir.
- Kısa Mesaj Verimliliği : Çağrı sayısının optimal düzeyde olması ve ek yükün neredeyse olmaması, kısa mesajlar için çok yüksek performans elde etmesine yardımcı olur. Ayrıca, hash fonksiyonu, kısa mesaj için de verimli olması amaçlanan ilk vektör olarak düz metnin ilk 128 bitini emer.

TESTLER :

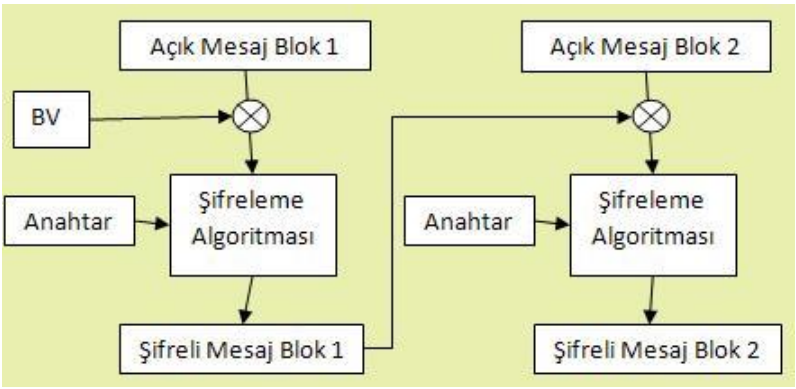
```
PS C:\Users\ESRA\Desktop\171044046\photonbeetle> ./a
        Photon-Beetle light-weight cipher
Plaintext : esra
Key : 0123456789ABCDEF0123456789ABCDEF
Nonce : 000000000000111111111111
Cipher: 7140B8C38A933540074D, Len: 20
Plaintext: esra, Len: 4
DONE!
```

```
        Photon-Beetle light-weight cipher
Plaintext : proje
Key : ABCDEF0123456789ABCDEF0123456789
Nonce : 111111111111000000000000
Cipher: D9ECECD9C6077361BB8EF4, Len: 21
Plaintext: proje, Len: 5
DONE!
```

```
        Photon-Beetle light-weight cipher
Plaintext : bubirmesajdir
Key : 0123456789ABCDEF0123456789ABCDEF
Nonce : 000000000000111111111111
Cipher: 7646A8CB3DE096C11AFB1F44DF3A2E, Len: 29
Plaintext: bubirmesajdir, Len: 13
DONE!
```

- **Cipher Block Chaining (CBC)**

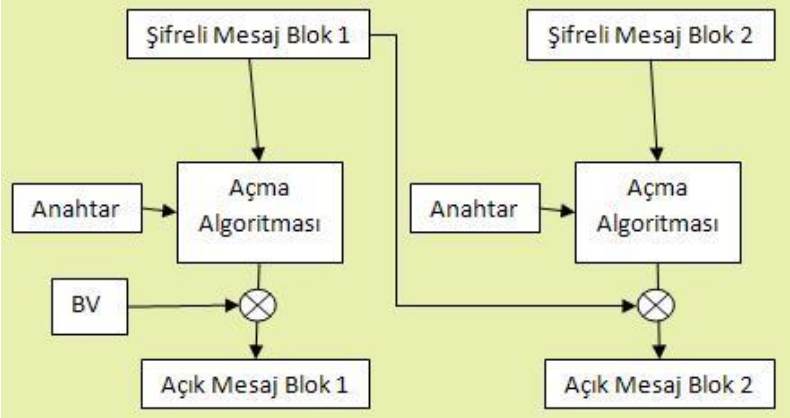
  Blok şifrelemede (block cipher) kullanılan yöntemlerden birisidir. ECB (electronic code book) yöntemindekinin aksine, şifrelenen mesajın bir önceki mesajla XOR işlemine sokulması sonucunda çıkan mesaj şifrelenir.
  Bu durumu aşağıdaki şekil üzerinden daha net anlamak mümkündür.



Yukarıdaki şekilde de görüldüğü üzere, Açık mesaj bloklara bölünmüştür. Her blok ayrı ayrı şifreleme işlemine tabi tutulur ve neticede şifreli mesaj her bloğun şifrelenmesinden elde edilen şifreli mesajların birleşimidir.

Burada dikkat edilecek husus, mesajların şifrelenmeden önce XOR işlemine tabi tutulmasıdır. Her mesajın XOR işlemi bir önceki şifreli mesaj bloğudur. İlk şifreli mesaj bloğu ise Başlangıç Vektörü (yöney) (initial vector) ile XOR işlemine tabi tutulur.

Mesajın açılması sırasında yukarıdaki işlem tersten uygulanır:

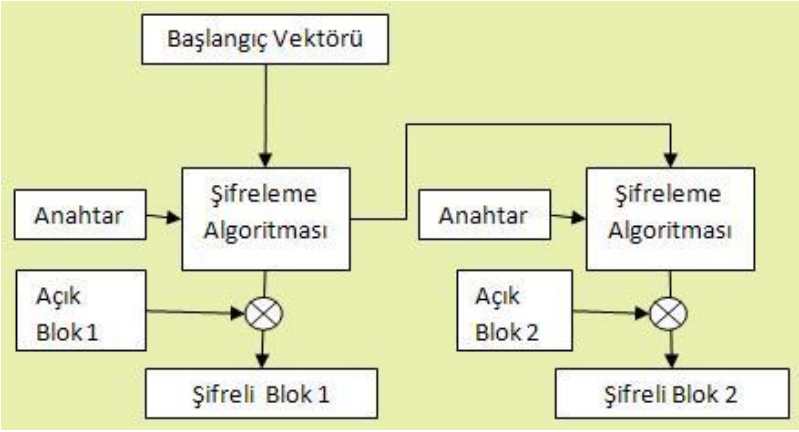


Yukarıda görüldüğü üzere her mesaj açılması işleminden sonra, bir önceki bloğun şifreli hali açma algoritmasının sonucu ile XOR işlemine tabi tutulur. Sonuçta elde edilen değer o blok için açık mesajdır.

- **Output Feedback Mod (OFB)**

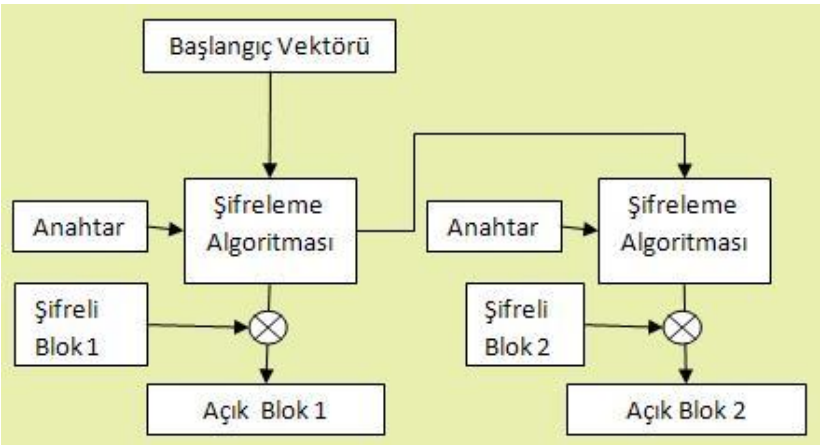
Blok şifreleme (block cipher) yöntemlerinden birisi olan ve bir bloğun şifreleme algoritmasından çıktısı ile bir sonraki bloğu beslemeye yarayan bu algoritmada amaç elektronik kod kitabına (electronic code book, ECB) göre daha başarılı şifreleme elde etmek hedeflenmiştir.

Şifreleme yönteminin çalışması aşağıdaki şekil üzerinden anlaşılabilir:



Yukarıdaki şekilde de görüldüğü üzere şifreleme algoritmasına giren değer başlangıç vektörü (initialization vector) olmaktadır. Şifreleme sonucunda elde edilen değer bir sonraki bloğun başlangıç vektörü olarak kullanılmaktadır. Şifreli mesaj blokları ise, açık blokların, şifreleme algoritması çıktıları ile XOR lanmış halidir.

Bu yöntemin açılması sırasında izlenen yol aşağıdaki şekilden anlaşılabilir:



Yukarıdaki şekilde dikkat edileceği üzere açma işlemi sırasında yine şifreleme algoritması kullanılmaktadır. Başlangıç vektörünün şifrelenmiş hali ile şifreli bloğun XOR lanması sonucunda elde edilen değer açık bloğu oluşturmaktadır.

- **Özet Alma İşlemi**

Kriptografik özet fonksiyonu çeşitli güvenlik özelliklerini sağlayan bir özet fonksiyonudur. Veriyi belirli uzunlukta bir bit dizisine, (kriptografik) özet değerine, dönüştürür. Bu dönüşüm öyle olmalıdır ki verideki herhangi bir değişiklik özet değerini değiştirmelidir. Özetlenecek veri mesaj, özet değeri ise mesaj özeti veya kısaca özet olarak da adlandırılır.

İdeal bir kriptografik özet fonksiyonu şu dört özelliği sağlamalıdır:

- Herhangi bir mesaj için özet hesaplamak kolay olmalıdır.
- Bir özete karşılık gelecek mesajı oluşturmak zor olmalıdır.
- Özeti değişmeyecek şekilde mesajı değiştirmek zor olmalıdır.
- Aynı özete sahip iki farklı mesaj bulmak zor olmalıdır.

Kriptografik özet fonksiyonları, bilgi güvenliği konuları olan sayısal imza, mesaj doğrulama kodu ve diğer doğrulama yöntemlerinde yaygın olarak kullanılmaktadır.

NOT : Projede dosya ile yapmam gereken işlemleri yapamadım.