udea
WIRELESS TECHNOLOGIES

WIRELESS TECHNOLOGIES

UN-0506v01_ŞİFRELEME

WIRELESS COMMUNICATION PRODUCTS

ŞİFRELEME ALGORİTMALARI

Kablosuz çözümlerde güvenlik daha çok Askeri ve özellikle WLAN uygulamalarının son yıllardaki artışıyla birlikte ortaya çıkan bir gereksinimdir. Bir çok Askeri ve WLAN uygulamaları için bilginin istenmeyen üçüncü partilere geçme riski, kablosuz networke yetkisiz giriş ihtimali ve networkün doğrudan bloke edilme tehlikesi üç temel tehdit olarak algılanmaktadır. IEEE802.11 ve WLAN uygulamalarının özellikle büyük şirket ağlarında kullanılmaya başlaması kablosuz ağ güvenlik uygulamalarını sıkça sorgulanır hale getirmiştir.

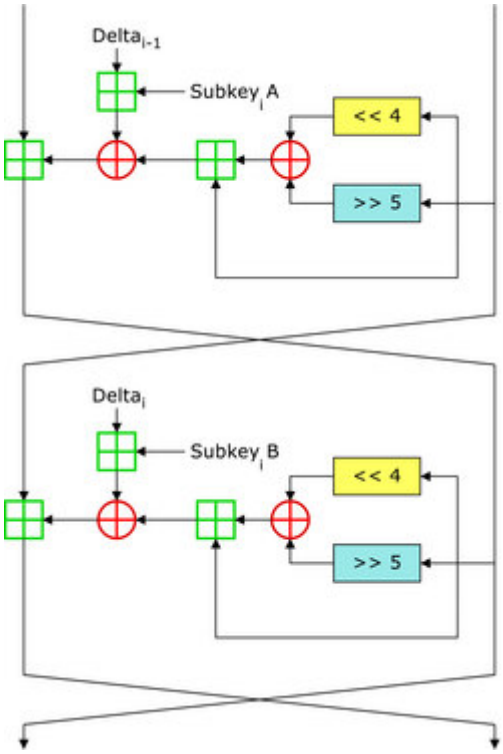
Tarihsel süreçte IEEE 802.11'de güvenlik için WEP - Wired Equivalent Privacy kullanılagelmekteydi. Bu standartla protokolün yetkisiz girişleri önlemek üzere authentication, veri gizliliğini ve çalınmasını önlemek üzere privacy özelliklerinin, kablolu sistemlerdeki ile eşdeğer olması beklenmekteydi. Bununla birlikte durumun öyle olmadığı, WEP key diye adlandırılan anahtarların ve WEP IV diye adlandırılan başlatma vektörleri uzunluklarının yetersiz olduğu görüldü. Geldiğimiz aşamada IEEE 802.11i gibi yakın vadedeki standartlarda kullanımda olan WEP protokollerinin gelişmiş bir versiyonu olarak da görülen TKIP-Temporal Key Integrity Protocol ve orta vadede CBC-MAC gibi AES destekli protokoller kullanılacağı öngörülmektedir.

Genelde RF modul MAC(madie acess control) katmanı yazılımı üzerinde yapılan şifreleme algoritmalarının (Data Encryption Algorithm) seçiminde başka bir çok konuda olduğu gibi; işlem hızı, code boyutu ve güvenlik arasında bir dengeleme yapmak gereklidir. Aşağıdaki tabloda yaygın olarak bilinen ve kullanılmakta olan şifreleme algoritmalarının işlem hızı ve code boyu olarak kıyaslamalarını görebilirsiniz. Uygulamanızda hız güvenlik ihtiyacından daha önemli bir etkense muhtemelen PRBS algoritması en iyi tercih olacaktır. Eğer uygulamanızın güvenlik ihtiyacı code boyutu, maliyet ve işlem hızından daha önemli bir gereklilik ise XTEA 32 (veya daha fazla) veya AES algoritmalarını kullanmak daha doğru olacaktır.

Tiny Encryption Algorithm :

Tiny Encryption Algorithm (TEA), çok kısa olan kod boyutu ve basit algoritması sayesinde özellikle kod boyutunun oldukça sınırlı olduğu gömülü sistemlerde oldukça popüler olan bir şifreleme algoritmasıdır.

TEA 1995 yılında, Roger M. Needham ve David J. Wheeler tarafından yayınlandı. 1997 yılında algoritma üzerinde iyileştirme yapılarak XTEA, algoritması duyuruldu. http://www.cix.co.uk/~klockstone/ adresinden XTEA ve TEA hakında detaylı bilgiye ve dökümanların orjinallerine ulaşılabilir. XTEA algoritması basit bit kaydırma ve toplama işlemlerinin tekrarlanmasından oluşan bir algoritmadır. Şekil 1'de algoritmanın birbirini tekrar eden 2 bloğu görülmektedir. Bu blokların sayısı artıkça yapılan şifrelemenin çözülmesi gittikçe daha zorlaşmaktadır. XTEA'nın geliştiricileri sağlam bir şifreleme için bu işlemin en az 64 kez yapılaması gerektiğini belirtemekte fakat 32 tekrarında çogu durumda yeterli olacağını belirtmişlerdir.



Şekil 1

Kaynak Kodu

```
tean( long * v, long * k, long N) {
        unsigned long y=v[0], z=v[1], DELTA=0x9e3779b9 ;

        if (N>0) {
        /* coding */
                unsigned long limit = DELTA * N, sum = 0 ;
                while (sum != limit)
                y +=  (z << 4 ^ z >> 5) + z ^ sum + k[ sum & 3 ],
                sum += DELTA,
                z +=  (y << 4 ^ y >> 5) + y ^ sum + k[sum >> 11 & 3] ;
                }
                else {
                        /* decoding */
                        unsigned long sum = DELTA * ( -N ) ;
                        while (sum)
                        z -= (y << 4 ^ y >> 5) + y ^ sum + k[sum >> 11 & 3];
                        sum -= DELTA,
                        y -= (z << 4 ^ z >> 5) + z ^ sum + k[ sum & 3 ]      ;
                        }
        v[ 0 ] = y, v[ 1 ] = z ;
        return ;
        }
```

Kaynak kodunu incelediğimizde ‘k’, ‘v2, ‘N’ olmak üzere 3 değişken kullanıldığını görmektedir. ‘v’ şifreleneceke verinin adresidir ve 64 bitlik bir bloğu gösterir, ‘k’ şifrelemede kullanılacak anahtardır. 128 bitlik bir bloğu gösterir. ‘N’ ise şifreleme işleminin kaç kez tekrar edileceğini gösterir. Fonksiyon ‘N’ pozitifse şifreleyici, negatif ise şifre çözücü olarak çalışır.

Algoritma Kıyaslama Tablosu :

İŞLEM HIZI :

Algortima	**Tekrarlama** (Iteration) **Sayısı**	**Güvenlik Seviyesi**	**Encoding** Cycles per Byte (yaklaşık)	**Decoding** Cycles per Byte (yaklaşık)	**Encode** Inst./Sec Bytes/Sec	**Decode** Bytes/Sec
PRBS XOR encryption with skipping key	KeyJump = 1	Zayıf	92-146 (**,*)	92-146 (**,*)	68493	68493
XTEA (Tiny Encryption Algorithm)	16 iteration	Yüksek	1075 (*,***)	1280 (*,***)	9302	7813
XTEA (Tiny Encryption Algorithm)	32 iteration	Yüksek/ Çok Yüksek	2133 (*,***)	2194 (*,***)	4688	4558
AES (RijndaelAlgorithm)		Yüksek/ Çok Yüksek	2153	2940 (****)	4645	3401

* **Datanın uzunluğuna bağlıdır**
** **Kullanılan Anahtar(key) değerine bağlıdır.**
*** **Tekrarlamaya bağlıdır.**
**** **Decode anahtarının gömülü veya üretiliyor olmasına bağlıdır.**

KOD BOYU :

Algoritma	**ROM**	**RAM**
PRBS XOR encryption with skipping key	226	12*
XTEA (also referred to as TEAN or TEA-N)	1950	38*
AES (Rijndael Algorithm)	6104	33*

* **şifrelenecek datayı içermemektedir.**

Sahada uzun süre kullanılacak bir ürün işin şifreleme algoritması tasarlarken dikkat edilmesi gerekli temel nokta teknolojinin ve şifreleme (kriptografi) metodlarının sürekli geliştiği ve bu nedenle uygulanacak algoritmanın ilerde çok zayıf hale gelme ihtimalidir.

Sonuç olarak sahada firmware uzun yıllar değiştirilemeyecek uygulamalar için en yüksek güvenlik elde edilebilecek algoritmaların tercih edilmesi önerilmektedir.