Orhun Kara

Dr., Başuzman Araştırmacı, UEKAE, TÜBİTAK

Kriptografinin Yapıtaşları

Kriptografik Algoritmalar ve Protokoller

Kripto sistemlerinin temellerini kripto algoritmaları ve bu algoritmaların hangi kurallarla kullanılacağını ifade eden kripto protokolleri oluşturur. Kriptolojinin varoluş nedeni olan gizlilik, kimlik doğrulama, inkâr edememe ve veri bütünlüğü gibi bilgi güvenliği hizmetleri, kripto algoritmaları ve protokolleri sayesinde sağlanır. Kriptografik algoritmalardan belki de en çok ilgi çekenler ve en yaygın kullanılanlar şifreleme algoritmalarıdır. İlginç olan ise günümüzde yaygın olarak kullanılan ve standartlaşmış modern şifreleme algoritmalarının hiçbirisiyle ilgili, kırılamayacağına dair henüz matematiksel bir ispat ortaya konamamış olması.

Anahtar Kavramlar

Simetrik şifrelemede, şifreleme yapacak ve çözecek kişiler arasında ortak bir anahtarda anlaşılmış olmalıdır.

Simetrik algoritmalarda şifreleme yapan aynı zamanda şifre de çözebilir. Oysa asimetrik algoritmalarda herkes şifreleme yapabilirken sadece özel anahtar sahibi şifreyi çözebilir.

Girdi olarak anahtar kullanmayan kripto algoritmaları da var. Örneğin, özet fonksiyonları rastgele uzunlukta metinleri girdi olarak alır ve sabit uzunlukta vektörler üretir. Bu vektörler metinlerin parmak izleri gibidir ve birçok kriptografik uygulamalarda uzun metinler yerine onları temsil ettiği düşünülen özetleri kullanılır.

Simetrik algoritmalar asimetriklere nazaran çok daha hızlıdır. Karşılaştırmak gerekirse simetrikler süpersonik uçaklar kadar hızlı ise asimetrikler ancak kağnı hızında olabilirler.

Kriptografik protokoller birçok açıdan resmi davet protokollerine benzer. Her ikisinde de davetli sayısı önemlidir. Kriptografik protokollerin çoğunda iki, üç, dört gibi az sayıda taraf (davetli) vardır.

Çoğu kriptografik sistemin öncelikli hedefi bilgiye yalnızca istenilen kişilerin ulaşabilmesini sağlamak, yani gizliliktir. Gizlilik, şifreleme (ve şifre çözme) algoritmalarıyla sağlanır. Şifreleme algoritması şifrelenecek metni ve şifreleme anahtarını girdi olarak alır. Şifrelenecek metne açık metin denir. Şifreleme algoritması bu iki veriyi kullanarak şifreli metni oluşturur. Şifre çözme algoritmasındaysa şifreli metin ve şifre çözme anahtarı kullanılarak açık metin üretilir. Şifre çözme algoritmasını şifreleme algoritmasının ters fonksiyonu gibi düşünebiliriz. Şifreyi nasıl çözeceğimizi bilmeden şifrelemeyi bilmek işimize yaramayacağı için kriptologlar çoğunlukla ikisine birden "şifreleme algoritması" derler.

JUPITERIMAGES

Şifreleme algoritmaları denilince önce hem şifreleme işleminde hem de şifre çözme işleminde aynı anahtarın kullanıldığı simetrik şifreleme algoritmaları akla gelir. Simetrik şifrelemede kullanılan anahtar başkalarından gizli tutulduğu için bu anahtara gizli anahtar denir. Bu yüzden simetrik şifrelemenin bir diğer adı da gizli anahtarla şifrelemedir.

Simetrik şifrelemede, şifreleme yapacak ve çözecek kişiler arasında ortak bir anahtarda anlaşılmış olmalıdır. Bunu sağlamanın bir yolu anahtarı, şifreleyecek ve şifre çözecek kişilere güvenli bir kanaldan ulaştırmaktır. Burada aklınıza şu soru takılabilir. Anahtarı güvenli bir kanaldan ulaştırıyorsak mesajı neden doğrudan o kanaldan göndermeyelim? Öncelikle, algoritmanız yeterince güçlüyse

>>>

JUPITERIMAGES

Kriptografi sayesinde internette kredi kartı numarası, vatandaşlık numarası gibi hassas bilgilerimizi yetkili kişilere güvenle iletebilir, güvenli alışveriş yapabilir, bankacılık işlemleri gerçekleştirebilir, faturalarımızı ödeyebilir, belge imzalayabiliriz.

ve anahtarınız yeterince güvenli saklanıyorsa şifreleme anahtarını milyarlarca defa kullanabilirsiniz! Sonra anahtarlar mesajlara göre çoğunlukla çok kısa boydadır. Örneğin, 128 bit ya da 256 bit. Bu anahtarla gigabaytlarca veri şifreleyebilirsiniz. Ayrıca güvenli kanal her zaman açık olmayabilir.

60 farklı internet kullanıcısının birbirleriyle simetrik şifrelemeyle haberleşmek istediklerini varsayalım. 60 kullanıcının 60'ı da aynı anahtarı paylaşıyor, yani tek bir anahtarla yetiniyor olabilir. Bu durumda hepsi diğerlerine gelen/giden mesajları okuyabilir. Diyelim ki bu kişiler birbirlerinden gizlisi saklısı olmayan insanlar. Dolayısıyla bu durumdan rahatsız değiller. Fakat içlerinden birisi çok dikkatsiz ve bu anahtarı koruyamamış. Bu durumda tek bir dikkatsiz kullanıcı yüzünden 60'ının da mesajları okunuyor olacak. O zaman bütün kullanıcı çiftleri ayrı bir anahtar paylaşsın. Bakalım ne kadar anahtara ihtiyaçları var? Hesaplayalım: 60×59/2, yani 1770 farklı anahtar!

Peki ya bin kişi birbirleri ile haberleşecekse? Ya biri yüzünden bini de anahtarını kaptıracak ya da yüz binlerce anahtar dağıtılacak. Kırk katır mı, kırk satır mı? İşte bu sorun asimetrik şifreleme algoritmaları sayesinde aşılabilir. Asimetrik şifreleme algoritmalarına sonra tekrar değinmek üzere, şimdi simetrik şifreleme algoritmalarını incelemeye devam edelim.

Simetrik şifreleme algoritmalarını iki grupta incelemek mümkün: Blok şifrele-

Gizli Anahtara Karşı Açık Anahtar

Gizli anahtarla şifrelemenin (simetrik şifreleme) binlerce yıllık geçmişe sahip olmasına karşın, açık anahtarlı şifreleme (asimetrik şifreleme) henüz 32 yaşında! Açık anahtar kriptografisi Diffie ve Hellman'ın 1976'da buldukları anahtar paylaşım protokolüyle doğmuş oldu. Bir sene sonra Rivest, Shamir ve Adleman tarafından tasarlanan tarihin ilk açık anahtarlı şifreleme algoritması RSA yayınlandı.

Peki ama biz şifreleme yapacaksak ne tür bir algoritma kullanacağız? Açık anahtarlı mı, gizli anahtarlı mı? Her iki türün de kendine göre avantajlı olduğu yerler var.

Simetrik şifreleme hem donanımda hem de yazılımda çok daha hızlıdır. Aralarındaki hız farkını gözünüzde canlandırmak istiyorsanız bir kaplumbağa ile bir jetin hızını düşünün! Sabit diskinizi asimetrik bir algoritma ile şifrelemeye karar verdiyseniz bir kez daha düşünmelisiniz!

Simetrik olanların gerçeklenmeleri de çok daha kolay. Genellikle simetrik algoritmalarda elektronik yongaların sevdiği ve/veya, dışarlayıcı-veya (XOR) gibi basit işlemler kullanılırken, asimetrik algoritmalarda devasal kümelerde çarpma, üs alma, bölme, ters alma gibi yongaları ve işlemcileri zorlayan aritmetikler kullanılır. Üstelik genel olarak asimetrik olanların anahtar boyları çok daha uzundur. Örneğin 80 bitlik bir simetrik algoritmanın sağladığı güvenliği 1024 bitlik bir RSA sağlayabilmektedir. Kütüphanelerdeki kitapların kapaklarına yapıştırılmış RFID etiketlerinde dahi bir simetrik algoritma koşturabilirsiniz. Oysa bir asimetrik algoritmayı gerçeklemek için pahalı ve büyük bir yongaya ihtiyacınız var.

Simetrik sistemler sayesinde hızlı bir şekilde veri bütünlüğü sağlamak da mümkün.

Buraya kadar hep simetrik algoritmaları övdük; sıra asimetrik algoritmalarda! Kullanıcı sayısının çok olduğu bir uygulamada anahtar paylaşımı ve tutulması gereken anahtar sayısı açısından asimetrik algoritmalar oldukça başarılıdırlar.

Kullanıcılardan herhangi ikisinin kendi aralarında, diğerlerinin dinleyemeyeceği kriptolu haberleşmeleri gereksin. Simetrik şifreleme ile kullanıcı sayısının ikili kombinasyonu kadar anahtar çiftinin kullanıcılar arasında güvenli kanallardan paylaştırılması gerekmektedir. Oysa asimetrik sistemde herhangi iki kullanıcı kullanıcı sayısı kadar anahtar çiftiyle kendi aralarında kriptolu haberleşebilir. Aslında simetrik algoritmaların en büyük eksikliği ve asimetrik olanların da ortaya çıkış nedeni bu problemdir.

Asimetrik şifrelemede çok az sayıda anahtarla problemi çözebiliriz. Üstelik güvenli kanaldan gizli anahtar paylaşımına da gerek yok. Çünkü gizli kalması gereken anahtarlar zaten paylaşılmıyor. Yalnızca açık anahtarlar paylaşılıyor, onlar da gizli olmak zorunda değiller. Açık anahtarlı sistemlerdeki bir sorun, açık anahtarın gerçekten sahibine ait olup olmadığını göstermektir. Saldırgan kendi açık anahtarını sizin açık anahtarınız gibi kabul ettirirse, sizin adınıza işlemler yapabilir. Bu nedenle açık anahtarlar genellikle güvenli biri tarafından sertifikalandırılarak dağıtılır.

Asimetrik sistemlerden vazgeçemememizin bir nedeni de, inkâr edememe hizmetinin ancak asimetrik sayısal imza algoritmalarıyla sağlanabilmesi. Asimetrik şifrelemede olduğu gibi, burada da "özel" işlem, yani imzalama işlemi özel anahtarla, herkesin yapabileceği işlem, yani imza doğrulama işlemi açık anahtarla yapılır. Nasıl ki, asimetrik şifrelemede de herkes şifreleme yaparken sadece özel anahtar sahibi şifre çözebiliyorsa, imzayı sadece yetkili atabilirken, herkes doğrulayabiliyor.

Görünen o ki her iki şifreleme türü de farklı alanlarda birbirlerine üstünlük kurmuşlar. Bu nedenle kriptologlar hibrit (melez) sistemler tasarlamayı tercih ederler. Anahtar şifreleme, anahtar anlaşma ve sayısal imza işlemleri genellikle asimetriklerle, yığın veri şifrelemeleri ve imzasız veri bütünlüğü korumaysa simetriklerle gerçekleştirilir.

me algoritmaları ve dizi şifreleme algoritmaları. Blok şifreleme algoritmaları metinleri uzunlukları belli olan bloklar halinde şifreler. Dolayısıyla her bir anahtar belli blok uzunluğunda bir permütasyon belirler. Bu permütasyonlar bir açık metin bloğuna karşılık hangi kapalı metin bloğu çıkacağını ifade eder. Blok şifreleme algoritmalarında içsel bir hafıza yoktur. Dolayısıyla şifreleme zamana bağlı değildir. Bu yüzden blok şifreleme algoritmalarına hafızasız şifreleme de denir. Veri Şifreleme Standardı (DES), Gelişmiş Şifreleme Standardı (AES) ve Uluslararası Şifreleme Algoritması (IDEA) gibi şifreleme algoritmaları birer blok şifreleme algoritmasıdır.

Dizi şifreleme algoritmalarında bir üreteç aracılığıyla, anahtar yardımıyla istenildiği kadar uzun bir dizi üretilir. Bu diziye, kayan anahtar denir. Kayan anahtar üretimi genellikle karmaşık fonksiyonlarla yapılır. Kayan anahtarla açık metnin "toplanmasında" basit matematiksel işlemler kullanılır. Kayan anahtar üretimi sırasında, üreteç içerisinde bir içsel durum vektörü oluşturulur. İçsel durum vektörü zamana bağlı olarak güncellenir ve kayan anahtar üretiminde kullanılır. Dolayısıyla kayan anahtar zamana bağlıdır ve hafızadaki durum vektörü şifrelemede rol oynar. Bu yüzden dizi şifreleme algoritmalarına hafızalı şifreleme de denir.

Dizi şifreleme algoritmalarının en ilginç özelliği kayan anahtar üretimi sırasında açık metnin girdi olarak kullanılmaması ve asıl karıştırıcı fonksiyon olan kayan anahtar üretecine açık metnin girmemesidir. Açık metin şifrelemenin en son adımında şifreleme işlemine basit bir metamatiksel işlemle dâhil edilir. Dolayısıyla şifreli metinde açık metnin karıştırım (confusion) ve yayınımını (diffusion) göremeyiz. Diğer bir deyişle, açık metindeki değişiklikler şifreli metne aynen yansır. Bunun tersi de doğrudur. Şifreli metindeki değişiklikler açık metinde ancak karşılık gelen karakterleri etkiler. Böylece şifreli metin karşı tarafa iletilirken ortamdaki gürültüden kaynaklanan hatalar yayılmaz.

Simetrik Şifreleme. Ortak bir anahtar ile hem şifreleme hem de şifre çözme yapılır

Hatanın yayılmaması nedeniyle yüksek frekanslı telsiz haberleşmelerinde olduğu gibi gürültülü ortamlardaki ses iletimini şifrelemek için genellikle dizi şifreleme kullanılır. Hatanın yayılmaması sayesinde ses, ortamdaki gürültüye rağmen alıcı tarafından anlaşılabilir. Diğer taraftan, hatanın yayılmaması açık metindeki bütünlük kontrolünü zorlaştırır. Dolayısıyla bütünlüğün önemli olduğu haberleşmelerde genellikle dizi şifreleme yerine blok şifreleme algoritmaları tercih edilir.

Minik Diziler Mini Minnacık Bloklar

Yaklaşık son beş yıla kadar dizi şifreleme algoritmalarının blok şifreleme algoritmalarına kıyasla daha basit olduğu,

Uğur Kaşif Boyacı
Uzman Araştırmacı,
UEKAE, TÜBİTAK

Kriptonun Olmazsa Olmazı Anahtar

Pahalı ve güvenli bir arabanız var. Arabanızın motor kilidi "immobilizer", anahtarınız olmadan arabanızın çalışmasını olanaksız hale getiriyor. Böylece arabanıza düz kontak dahi yapılamıyor. Arabanızın kapıları da anahtarsız mümkün değil açılmıyor. Camlar kırıldığında ya da kapılar zorlandığında alarm devreye giriyor. Hırsızların hiç şansı yok! Arabanız gerçekten de güvende. Ama bir dakika! Eğer anahtarınız güvende ise! Anahtarınızı kaybederseniz ya da çaldırırsanız araba hırsızları arabanıza sizin kadar yakın demektir. Modern kripto sistemlerinde de güvenlik anahtarın güvenliğine indirgenmiştir. Dolayısıyla anahtarlar kripto sistemlerinin yumuşak karnıdır. Bu nedenle bir anahtarın bütün varoluş süreçleri boyunca özenle korunması şart.

Kripto sistemlerinin kalbi anahtarlardır, bu nedenle anahtarlarımızı gözümüz gibi korumalıyız. Daha teknik bir ifade ile "bir kripto sisteminin güvenliği anahtarların gizliliğine dayanmalıdır". Bu ilke 19. yüzyılda yaşamış Fransız dilbilimci Auguste Kerckhoff tarafından ortaya atılmıştır. Sisteminiz, şifreleme algoritmanız ve yaptığınız her türlü matematiksel işlem ve fonksiyonlar bir şekilde düşmanın eline geçebilir. Bu durumda dahi sisteminiz güvenli olmalı. Güvenliğinizi algoritmanın ya da haberleşme protokolünün gizli olmasına, açık metinlerin tahmin edilemez ve karmaşık olmasına dayandırırsanız ciddi bir risk altındasınız demektir.

Kerckhoff ilkesinin ilginç bir özelliği de dünyada en çok yanlış algılanan ilkelerden biri olmasıdır. İlkeyi yanlış algılayanlar, genellikle algoritmanızı ve protokolünüzü en ince detayına kadar açıklamanız gerektiğini ve sadece anahtarınızın gizli kalması gerektiğini ifade ederler. Oysa Kerckhoff'un anlatmak istediği il-

>>>

donanımda daha az yer kapladığı kanısı hâkimdi. Blok şifreleme algoritmalarının da yazılımda, özellikle masaüstü işlemcilerinde çok daha hızlı olduğu düşünülüyordu. Son yıllarda yapılan araştırmalar ve geliştirilen yeni şifreleme algoritmaları bu ezberi bozacak gibi görünüyor.

Avrupa Birliği 6. Çerçeve Programı Mükemmeliyet Ağları projesi kapsamında 2004'de başlayıp geçen yıl sona eren Estream Projesi dizi şifreleme algoritması tasarımı ve analizi üzerine odaklanmıştı. Projenin bir ayağında özellikle donanımda çok az yer kaplayan, yani olabildiğince az sayıda devre kapısıyla gerçeklenen dizi şifreleme algoritmaları masaya yatırıldı. Bu kategoride en çok ilgi çeken iki algoritma Trivium ve Grain oldu. Wili Meier ve arkadaşları tarafından tasarlanan Grain, yaklaşık 1500 devre kapısıyla, Christophe De Canniere ve Bart Preneel tarafından tasarlanan Trivium ise 2500-3000 devre kapısıyla gerçeklenebilmektedir. Donanımda hız öncelikli bir AES gerçeklenmesinin yaklaşık 100.000 devre kapısı kadar yer kapladığı düşünülürse her iki dizi şifreleme algoritmasının da donanımda ne kadar az yer kapladığı daha iyi anlaşılır.

Grain de Trivium da dizi şifreleme algoritmalarının donanımda ne kadar az yer kaplayabileceğine iyi birer örnek olsa da, bu algoritmalardan birkaç yıl sonra tasarlanan bir blok şifreleme algoritması az yer kaplama açısından dizi şifreleme algoritmalarının tahtını salladı diyebiliriz. Aralarında Lars Knudsen ve Matt Robshaw gibi kriptologların bulunduğu bir grup tarafından tasarlanan ve yaklaşık 1500 devre kapılık yer kaplayan PRESENT adlı blok şifreleme algoritması 2007'de CHES (Cryptographic Hardware and Embedded Systems-Kriptografik Donanım ve Yerleşik Sistemler) konferansında yayınlandı. Geçtiğimiz aylarda Orr Dunkelman ve Christophe De Canniere tarafından tasarlanan KATAN adlı bir blok şifreleme algoritmasının bir sürümüyse sadece 500 devre kapılık yer kaplıyor! Tasarımcılardan alınan bilgiye göre algoritmanın bu yılın ikinci yarısında bir kriptoloji konferansında yayınlanması planlanıyor. Kriptolojideki bu son gelişmeler donanımda dünyanın en küçük algoritmalarının artık dizi şifreleme algoritmaları yerine blok şifreleme algoritmaları olduğunu göstermektedir. Ama yarış devam ediyor. Kim bilir, belki gelecekte dizi şifreleme algoritmaları tahta tekrar oturur.

Masaüstü Bilgisayarlarda Kim Önde?

Son on yıla kadar, kriptologlar arasında blok şifreleme algoritmalarının masaüstü işlemcilerde dizi şifreleme algoritmalarına kıyasla çok daha hızlı ve verimli çalışacağı kanısı hâkimdi. Kriptologlar aslında böyle bir kanıya varmakta haksız da değiller. Modern masaüstü işlemciler 32 bit ya da 64 bit gibi kelimeler üzerinde işlemler yaparlar ve blok halinde işlemleri başarıyla gerçeklerler. Diğer taraftan bu işlemcilerdeki seri işlem mantığı, yazmaç tabanlı dizi şifreleme algoritmalarında hafızaların güncellenmesi türünden işlemlerin hızlı gerçeklenmesine çok uygun değildir. Gerçekten de 70'li ve 80'li yılların donanıma özel tasarlanmış dizi

ke şöyledir: Kripto sisteminiz öyle bir özelliğe sahip olacak ki, bütün sistem detayları açığa çıksa dahi anahtar gizli kaldığı sürece sisteminiz (kriptografik açıdan) güvenli olacak. Tarihte yaşanmış tecrübelerle Kerckhoff ilkesini benimsemenin ne kadar önemli olduğu defalarca kanıtlanmıştır.

Algoritmanız öyle tasarlanmış olmalı ki, biri nasıl çalıştığını bilse bile anahtarı bulmadan ondan yararlanamamalı. Hatta saldırganın elinde "bol miktarda" algoritma girdisi ve çıktısı bulunsa dahi anahtar hakkında bilgi edinememeli. Bol miktarda derken aynı kategorideki ideal bir algoritmanın karmaşıklığı kastedilmektedir. Örneğin bir simetrik şifreleme algoritması için bu anahtar uzayının neredeyse tamamı demektir. Tabii böyle matematiksel fonksiyonlar tasarlamak tam bir uzmanlık alanı.

Diyelim ki şifreleme algoritmanız sağlam ve saldırgan algoritmayı analiz yoluyla kıramayacağını anladı. O zaman doğrudan anahtarın kendisini hedef alır. Eğer anahtarı daha kolay elde edebilecekse neden yıllarca matematiksel denklemler kurarak, binlerce bilgisayara iş vererek sonuç beklesin ki?

Saldırgan anahtarı "doğumunda", "ölümünde" hatta "mezarda dahi" ele geçirirse yine de avantaj elde edebilir. Evet, anahtarların bir yaşam döngüsü vardır! Anahtarlar sipariş edilir, üretilir, paketlenir, adreslerine teslim edilir, saklanır, kullanılır, işleri bitince de atılır ve gerekirse yok edilir, yerine yenileri gelir. İşte bu yaşam döngüsü boyunca anahtarlara nasıl bakılacağına "anahtar yönetimi" denir.

Visual Photos

Bir anahtarın yaşam döngüsünün ortasına bakalım; yani anahtar saklamaya... Neden ortasından başlıyoruz? Anahtarın saklanması her kullanıcının derdi de ondan. Kişisel bilgisayarımızda anahtarları nasıl saklayabiliriz? Akla ilk gelen cevaplar ya "güvenli bir yerde" ya da "şifreleyerek". Peki bilgisayarınızın güvenli yeri neresi? Günümüzde bilgisayarların sabit diskini sökerek içinden bilgi okunması çok zor değil. Ayrıca internete bağlıysanız saldırgan bilgisayarınıza uzaktan da erişebilir. Peki o zaman bütün anahtarlarımızı başka bir anahtarla şifreleyelim. Bu sefer de anahtar şifreleme anahtarı için aynı soru geçerli. Anahtar şifreleme anahtarını nasıl saklayacağız? Eninde sonunda bir anahtarı güvenli bir şekilde saklamamız gerekir.

Bir kripto sisteminde bütün anahtarlar aynı kıymette olmayabilir. Yukarıdaki çözümde anahtar şifreleme anahtarı diğer anahtarlardan daha kıymetlidir, çünkü anahtar şifreleme anahtarı ele geçirilirse diğerleri de ele geçirilmiş olur. Kıymetli anahtarlarımızı taşınabilir bir

şifreleme algoritmaları masaüstü işlemcilerde bir kağnı kadar yavaştı.

Kriptoloji gibi baş döndürücü bir hızla gelişen bir bilimde, masaüstü işlemcilere uygun ve güvenli birçok dizi şifreleme algoritmasının tasarlanması hiç de şaşırtıcı değil. Hatta öyle dizi şifreleme algoritmaları vardır ki masaüstü işlemcilerde bilindik tüm blok şifreleme algoritmalarından daha hızlı olduklarını söyleyebiliriz. Örneğin Hongjun Wu tarafından tasarlanıp 2004'de Estream projesine sunulan ve şu ana kadar henüz bir zayıflığı keşfedilemeyen HC-128 adlı dizi şifreleme algoritması masaüstünde yaklaşık 2 devirde bir bayt üretebiliyor. Örneğin 2 GHz frekansı olan bir işlemcide saniyede 1 GB (gigabayt) veriyi şifreleyebiliyor. Bu, AES'in yazılımda en hızlı gerçeklenmesinden yaklaşık 6 kat daha hızlı. Tabii, Intel'in bu sene sonunda piyasaya süreceği, içinde AES şifreleme ve şifre çözme komut takımının bulunacağı işlemcileri dikkate almazsak... Bu yeni nesil işlemcide AES çok daha aşağı katmanda, donanımda Intel mühendislerinin özel olarak tasarladığı ve gerçeklediği yonga üzerinde koşuyor olacak. Test sonuçları şimdiden etkileyici: Bu işlemciler sayesinde, AES en az üç kat daha hızlanacak. Ama donanımdan gelen bu ayrıcalığa rağmen AES yine de HC-128 kadar hızlı olamayacak!

Yeri gelmişken HC-128'in güvenliği hakkında bir not ekleyelim. Ünlü Hint kriptolog Maitra öğrencileriyle birlikte yaptığı altı aydan uzun süren yoğun bir çalışma sonucunda HC-128'in iç yapısıyla ilgili "beklenmedik" bazı özellikler keşfetti. Çalışmanın sonuçlarını geçen Mayıs ayında Norveç'te düzenlenen Uluslararası Kodlama Teorisi ve Kriptografi Çalıştayı'nda (WCC) anlattılar. Sunumlarını "Biz algoritmada henüz bir zayıflık keşfedemedik. Ama belki başkaları bizim keşfettiğimiz sapmaları daha da geliştirip HC-128'i kırmayı başarabilir," diyerek sonlandırdılar.

Anahtarsız Algoritmalar

Girdi olarak anahtar kullanmayan kripto algoritmaları da var. Bunlar genellikle tek başlarına bir hedefe ulaştırmıyor fakat sistem içinde diğer algoritmalara çok yardımcı oluyorlar. Anahtarsız algoritmalardan en bilineni özet fonksiyonlarıdır (hash functions). Bu algoritmaların kullanım alanlarında sağlamaları gereken özelliklerle ilgili olarak kendilerine özgü güvenlik ölçütleri bulunur.

Özet fonksiyonları girdi olarak rastgele uzunlukta metinleri alır, sabit uzunlukta (genellikle 20-64 bayt arası) vektörler üretir, bir nevi metinlerin parmak izlerini alır ve birçok kriptografik uygulamada uzun metinler yerine onları temsil ettiği düşünülen özetleri kullanılır.

Özet fonksiyonları bütünlük denetiminde ve güvenli parola saklamada yaygın olarak kullanılır. Güvenlik nedeniyle bilgisayarlarda parolalarımızın kendileri saklanmaz. Bunun yerine, parolalarımızın "tuz" denilen, rastgele üretilmiş vektörlerle birlikte özetleri alınır ve bunlar saklanır. Bu yüzden özet fonksiyonları tek yönlü fonksiyonlar olmalıdır, yoksa diğer yönden parolayı elde ederiz. Yani bir metnin özetini almak hesapsal olarak kolayken, verilmiş bir özete sahip bir metin oluşturmak pratikte mümkün olmayacak kadar zor olmalıdır. Bu özelliğe,

cihazda saklayabiliriz. Böylece hem anahtarları başka yerde de kullanabiliriz, hem de anahtarlar "gözümüzün önünde" olur. Özellikle anahtar saklamak üzere üretilmiş taşınabilir cihazlar vardır. Bu cihazlarda anahtarlar şifreli olarak saklanır. Saldırganın erişemeyeceği, erişmeye çalıştığı takdirde silinen küçük bir bellekte ise bu anahtarları şifreleyen anahtar saklanır. Bu anahtar da genellikle parola ile korunur. Böylece anahtar saklayan mini cihaz ele geçse bile anahtarlarımıza parola bilinmeden ulaşılamaz.

Anahtar saklamanın bir diğer yolu da anahtarı ikiye ayırmaktır! Bir parçasını bilgisayarınızda, diğer parçasını ise taşınabilir fakat çok da güvenli olmayan bir ortamda, örneğin bir bellek kartında saklarsınız. Karttan okunan parça ile bilgisayardaki parça bir araya gelince anahtar geri kazanılır. Bir saldırganın parçalardan birini öğrenmiş olma ihtimaline karşı birkaç kullanımdan sonra farklı bir parçalama yapılarak anahtar farklı bir şekilde ayrılır. Böylece saldırgan elini çabuk tutmazsa öğrendiği parça işine yaramaz.

Gelelim anahtarların doğumuna! Eğer anahtar üretimi sonucunda tahmin edilebilir anahtarlar çıkıyorsa saldırgan da bunları tahmin edebilir. Bu nedenle anahtarlar mümkün olduğunca rastsal üretilmelidir.

Meşhur bilgisayar bilimci Donald Knuth'un söylediği gibi "Rastsal sayılar rastgele metotlarla üretilmemelidir." Tam aksine rastsal sayı üreten mekanizmaların tasarımı ve gerçekleştirilmesi büyük özen ister.

Rastsal sayı üreteçleri temel olarak ikiye ayrılır. Bir diyotun anlık elektrik akımı ya da kaotik bir sistem gibi fiziksel olaylara dayalı olarak rastsal sayı üreten mekanizmalara "Gerçek Rastsal Sayı Üreteci" (GRSÜ), matematiksel yollardan çekirdek bir değerden deterministik olarak rastsal sayı dizileri üreten mekanizmalara "Sanki Rastgele Sayı Üreteci" (SRSÜ) denir.

Her ne kadar adı "gerçek" ile başlasa da gerçek rastsal sayı üretecilerin gerçekten rastsal sayı ürettiğinden emin olmak kolay değildir. Aşırı ısı, elektrik yüklemesi, manyetik alan gibi dış etkenlerden dolayı gerçek rastsal sayı üreteci çıktıları tahmin edilebilir dizilere dönüşebilir.

Buna karşılık sanki rastgele sayı üreteci çekirdek bilinirse üretilen rastsal değerlerin hepsi ortaya çıkar. Bu nedenle çekirdek saldırgan tarafından tahmin edilememelidir. Ayrıca saldırgan sanki rastgele sayı üreteciyi aynı çekirdeği yutmaya ikna ederse sanki rastgele sayı üretecilerde aynı dizi ortaya çıkar. Bazı algoritma ve protokollerde anahtar kadar önemli, taze oluşturulmuş değerler gereklidir.

İstatistiksel Testler

Bir üretecin rastsal sayı ürettiğinden emin olabilir miyiz? Rastsallık konusuyla uğraşan matematikçi ve istatistikçiler "Hiçbir test tek başına rastgeleliğe karar veremez" demektedir. Çeşitli istatistiksel ve matematiksel testlerle üreteç çıktısından topladığımız numune dizilerinin beklediğimiz belli "davranış profilleri"ne uyup uymadığını kontrol ederiz. Davranış profilleri genelde

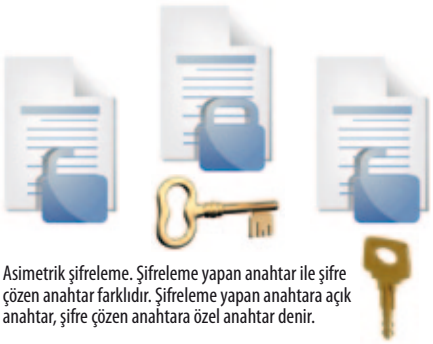
ters-görüntüye dayanıklılık deniyor. Böylece bir parolanın özet değerini ele geçirseniz bile, parolayı ortaya çıkaramazsınız.

Özet fonksiyonlarının kullanıldığı bir başka uygulama ise sayısal imza algoritmalarıdır. Asimetrik algoritmalar kullanıldığı için imza algoritmaları oldukça yavaştır. Dolayısıyla büyük metinlere doğrudan imza atmak uzun zaman alır. Üstelik imza da metin kadar büyük olursa, imzalı metin kaynak metnin iki katı yer kaplayacaktır. Bu nedenle önce metnin özeti alınır, sonra özete imza atılır. Özet almak imza atmaya kıyasla çok daha hızlı bir işlem olduğundan, uzun verilere imza atmak kısa verilere imza atmak kadar hızlı olacaktır. Ancak burada dikkat edilmesi gereken bir güvenlik problemi var. Performanstaki bu kazanım güvenlik açığına neden olmamalı! Herhangi iki metin aynı özeti veriyorsa birine atılan imza diğeri için de geçerli olacaktır. Dolayısıyla bir metinle aynı özeti üretecek ikinci bir metin bulmak hesapsal olarak zor olmalı. Özet fonksiyonların bu güvenlik ölçütüne ikinci ters-görüntüye dayanıklılık denir.

Asimetrik Şifreleme

Asimetrik şifrelemede şifreleme anahtarı ile şifre çözme anahtarı farklıdır. Şifreleme yapan anahtara açık anahtar, şifreyi çözen anahtara özel anahtar denir. Açık anahtar adından da anlaşılacağı gibi açıktadır, dost düşman herkese verilebilir! Herhangi birine gizli mesaj göndermek isteyen, o kişinin açık anahtarı ile açık metni şifreler. Şifreyi çözebilecek olan kişi yalnızca özel anahtarın sahibidir. Özel anahtar kişiye özeldir ve kimseyle paylaşılmaz.

Simetrik şifreleme ile asimetrik şifreleme kavramları arasındaki temel farkı daha açık anlatabilmek için kapı kilitleri ile asma kilitli posta kutusunu örnek verebiliriz. Evimizin dış kapısını kilitlemek ya da açmak için kullandığımız anahtarların simetrik şifrelemede gizli anahtara karşılık geldiğini düşünebiliriz. Bu anahtarlar ile hem kapıyı kilitleyebilir (şifreleme yapabilir) hem de kilitli kapıyı açabiliriz (şifreyi çözebiliriz). Herkesin ulaşabileceği, asma kilitli bir posta kutusunu açık anahtar olarak düşünün. Posta kutusuna herkes mesaj atabilir, ama posta kutusundaki mesajları yalnızca kutuyu açan asma kilit anahtarına sahip olan okuyabilir.



Asimetrik şifreleme. Şifreleme yapan anahtar ile şifre çözen anahtar farklıdır. Şifreleme yapan anahtara açık anahtar, şifre çözen anahtara özel anahtar denir.

KRİPTO PROTOKOLLERİ

Protokol denilince çoğumuzun aklına milli bayramlardaki resmigeçit törenleri ya da smokin veya frak giymiş devlet adamlarının bulunduğu resmi davetler gelir. Resmi davetlerde, kimin kiminle nasıl selamlaşacağı, kimin hangi sırada salona gireceği, yemekte kimin yanına kimin oturacağı sıkı kurallara bağlanmıştır. Protokol belli amaç ve hedefler için, belli bir ortamda, taraflar arasında sırasıyla uyulması gereken iş adımlarını ifade eder.

miktar, büyüklük, sıralanma ya da tekrar etme üzerine kuruludur. Üreteç çıktısı belli profillere uysa da sayıların rastsallığından emin olamayız, ancak şüphelerimizi azaltabiliriz.

İstatistiksel testlerden geçemeyen bir üreteç çok büyük bir ihtimalle kötü tasarlanmıştır. Emin olduğumuz şey: Eğer üreteç istatistiksel testlerden kalıyorsa üreteçte defo vardır! Diğer yandan kötü olduğunu bildiğimiz, yani üreteceği diziyi tahmin edebildiğimiz fakat yapılan istatistiksel testlerden başarıyla geçen üreteçler de vardır. Bu nedenle rastsallıktan uzak üreteçleri yakalamak için istatistikçiler yeni ve pratik testler aramaya devam etmektedir.

Anahtar Üretim/Dağıtım Merkezi

Rastsal sayı üretecinden elde edilen çıktılar, her kriptografik algoritma için anahtar olarak kullanılmaya elverişli değildir. Bazı şifreleme algoritmalarında zayıf, yani saldırganın algoritma girdi-çıktılarından kolayca tahmin edebileceği anahtarlar vardır. Üretilen anahtarın zayıf olup olmadığının kontrol edilmesi gerekir.

Anahtarları gerekli rastsallıkta ve doğru ölçütler altında üretmek, gerekli kişilerce paylaşımını sağlamak, anahtar üretim maliyetini düşürmek ve benzeri nedenlerle, en azından bazı kritik anahtarlar herkes tarafından güvenilen bir anahtar üretim merkezinde üretilmektedir.

Anahtarların doğru adrese teslimini, gittikleri yerde doğru zamanda ve doğru amaçla kullanılmasını sağlamak için anahtarlar etiketlenir. Etiketleme yanlış yapılırsa anahtarlar yanlış kişilerin eline geçebilir ya da kullanıcıya ulaştırılamaz. Etiket üzerinde anahtarın son kullanım tarihi gibi bilgiler de bulunur. Dağıtım sırasında başına bir şey gelmemesi ve kolay taşınması için anahtarlar paketlenmelidir.Genellikle paketin hem içinde, hem dışında birer etiket bulunur.

Saldırgan anahtarı dağıtım sırasında ele geçirmeye de kalkabilir. Eğer anahtar kurye ile elden taşınıyorsa, saldırgan anahtarı ele geçirmek için anahtarı taşıyan kuryeye zarar vermeyi bile göze alabilir ya da kuryeyi kandırmaya kalkabilir. Saldırgan kuryeden elde edeceği anahtarın şifreli olduğunu ve açamayacağını bir şekilde bilirse kurye büyük bir ihtimalle hedef olmayacaktır.

Bir Anahtar Taşıma ve Yükleme Cihazı: KAYC-S

Anahtar üretim merkezinden cihaza güvenli taşımanın emin bir yolu, merkezin paketi güvenli hattan (saldırganın kolayca mü-

Kriptografik protokoller birçok açıdan resmi davet protokollerine benzer. Her ikisinde de davetli sayısı önemlidir. Kriptografik protokollerin çoğunda iki, üç, dört gibi az sayıda taraf (davetli) vardır. Bazı protokollere ise çok sayıda taraf katılır. Hizmet kalitesini düşürmeden ve maliyeti aşırı yükseltmeden kriptografik protokolü işletmeye ölçeklenebilirlik denir. Ölçeklenebilirlik iyi bir "çok taraflı protokol"ün en aranan özelliklerinden biridir.

Resmi davetlerde çoğu zaman davetlilerin katılımını sağlayacak bir davetiye vardır. Protokollere tarafların katılımı kriptografik anahtarlar sayesinde olur. Anahtar ve "davetiye" arasında önemli bir fark vardır. Davetliler davetiyelerini başkalarına gösterebilirler fakat kriptografik protokollerde anahtarları, paylaşanlar dışında kimse görmemelidir. Neden mi? Elektronik ortamda davetsizlerin anahtarı kopyalaması çok kolaydır da ondan. Bazı davetlerde özenle saklanması gereken eşyalar bulunur. Örneğin bir kraliçenin takısı paha biçilemez olabilir. Kriptografik protokollerde de bazı anahtarlar saldırganlar için mücevherlerden daha değerlidir.

Nasıl davetlerin kapalı mekân, maskeli balo ya da resmigeçit töreni olması kuralları değiştirebiliyorsa, kriptografik protokollerde de ortam belirleyici olur. Örneğin kullanılan hattın telsiz, telefon, cep telefonu, kablolu internet, uydu haberleşmesi olması ve bu hatların gürültü oranı gibi karakteristikleri, protokol tasarımını derinden etkileyebilir.

Davetlilerin niteliği de protokolü değiştirir. Örneğin bazı protokollerde mutlaka herkesin güvendiği biri gerekir. Biz bu davetliye "Güven" diyelim. Güven genellikle bir anahtar dağıtım merkezi ya da sertifika otoritesidir. Bazı protokollerde davetlilerin bir kısmı işlemleri kolayca ve hızlıca yapabilirken bir kısmının eli yavaştır. Örneğin RFID protokollerinde okuyucular hızlı işlem yaparken RFID etiketleri kısıtlıdır. Bazı protokollerdeyse başı çok kalabalık davetliler olacağını hesaba katmak gerekir; örneğin istemci-sunucu protokolleri...

Kripto protokollerinde de tıpkı resmi törenlerdeki protokollerde olduğu gibi davetliler, davetliler arasında hiyerarşi ve uyulması gereken katı kurallar vardır.

Visual Photos

Kriptografik protokolleri asıl ilginç kılan, protokollere katılan davetsiz ya da münasebetsiz katılımcılardır. Davetsizlere saldırgan diyeceğiz. Davetli listesinde olduğu halde protokol kurallarına uymayan ya da uysa bile haksız kazanç peşinde koşan misafirlere ise "düzenbaz" diyeceğiz. Kriptografik protokol düzenlemenin zorluğu da çoğunlukla, davete katılması engellenemeyen saldırgan ve düzenbazlara rağmen "dürüst" tarafların davetin amacına ulaşmasını sağlamaktır. Eğer herkes davetsiz ya da düzenbaz olursa ya da "ortam" davet düzenlemeye uygun değilse, davet elbette amacına ulaşamaz. Bu nedenle kriptografik protokollerde ortam ve katılımcılar üzerinde çeşitli varsayımlarımız olacak. Eğer varsayımlarımız gerçekçi değilse ya çok pahalı bir davet düzenleriz ya da kötü konuklar davetin altını üstüne getirir.

Resmi davetlerin farklı ülkeler arası ilişkileri güçlendirme, belli bir konuda katılımcıları bilinçlendirme gibi hedefleri vardır. Kriptografik protokollerse çoğu zaman aynı anda birçok hedefi sağlamaya çalışır. Veri gizliliği, veri bütünlüğü, kimlik doğrulama, kaynak doğrulama, inkâr ededahale edemeyeceği bir hattan, örneğin kuantum kanalından) ya da güvensiz hattan (örneğin internetten) kriptografik tedbirlerle koruduktan sonra cihaza aracısız yollamasıdır. Peki, anahtar paketini koruyan anahtarlar güvenli bir şekilde nasıl iletilecek? Sonunda mutlaka bir anahtarın güvenli bir şekilde cihaza ulaştırılması gerekir.

Anahtar dağıtmadan, saldırgan da aradaki her mesajı dinliyorken, taraflar arasında taze ve rastsal bir anahtar oluşturabilir mi? İlk anda olanaksız gibi görünüyor. Gerçekten de ilk anahtar anlaşma protokolünün bulunuşu modern kriptolojide bir dönüm noktası olmuştur.

Anahtar anlaşmada kullanılan başka yöntemler de vardır. Bunlardan bir tanesi tarafların daha önceden paylaşılmış bir anahtarı doğrudan algoritmada kullanması yerine, bu anahtardan başka anahtarlar türetilmesidir. Bir diğer yöntem kullanılan anahtarın paydaşlar tarafından önceden bilinen bir fonksiyon ile güncellenmesidir. Bunun için genellikle tek yönlü fonksiyon kullanılır. Bu durumda eski anahtara dönülemediği için, güncel anahtar çalınsa bile hiç olmazsa eski mesajlaşmalar güvende olur.

Su Uyur Saldırgan Uyumaz

Saldırgan anahtarı bulduğunu çoğu zaman hissettirmez. Tedbir olarak anahtarları belli aralıklarla değiştirmemiz gerekir. Anahtarı değiştirme sıklığına anahtarı paylaşan taraf sayısı, anahtarın önemi, anahtar dağıtma maliyeti ve benzeri etkenler göz önüne alınarak karar verilir. Simetrik sistemlerde tek bir kişi bile paylaştığı anahtarı kaptırırsa diğerleri de kaptırmış olur. Bu nedenle çok kullanıcı sistemlerde mümkünse anahtar dağıtımında asimetrik şifreleme ya da taze anahtar oluşturma teknikleri kullanılmalıdır.

Ömrünü doldurmuş anahtarları cihazda saklamaya devam etmek de başka bir risktir, çünkü saldırgan anahtarı cihazdan ele geçirebilirse geçmiş mesajları inceleyebilir ya da anahtarın değiştiğinden haberi olmayan taraflarla mesajlaşabilir. Bu riski önlemek için ilk önce artık kullanılmayacak anahtarların silinmesi gerekir. Anahtarlar silinirken dikkatli olunmalıdır. Birçok kripto cihazı anahtarları silme işini işletim sistemine havale eder, fakat birçok işletim sisteminin silme işlemleri yeterince güvenilir değildir. Anahtar hafızada bir yerlerde siz farkında olmasanız da durmaya devam eder. Bu nedenle kripto sistemlerinde anahtarların imhası ve kullanılmayan anahtar bilgisinin cihaza kaydedilmesi, anahtar yönetiminin önemli bir parçasıdır.

Ele geçmiş ya da süresi dolmuş anahtarlardan diğer cihazların haberdar edilmesi de anahtar yönetiminde özen isteyen bir konudur.

mezlik ve anahtar anlaşma en çok ihtiyaç duyulanlardır. Bunlara son yıllarda önem kazanan mahremiyeti de eklemek gerekir.

Mahremiyet, bir işin kimin yaptığının sadece "gerekli kişiler" tarafından öğrenilmesi demektir. Bunun en çarpıcı örneğini elektronik gizli seçimden verebiliriz. Gizli seçimlerde, geçerli bir oyun kime verildiği belli olmalı fakat kimin tarafından verildiği belli olmamalıdır. Diğer bir deyişle, oy anonim olmalı ve oy verenin mahremiyeti korunmalıdır.

Elektronik oylama protokol tasarımının ne kadar güç olabileceğine güzel bir örnektir. Oy verenin kimliğinin gizlenmesi, oy verenin tekrar oy kullanamaması, oyların gerektiğinde tekrar sayılabilmesi, oy kullananın oyunun sayıldığından emin olabilmesi ve daha birçok hedefin aynı anda sağlanması beklenir. Bu hedeflerin hep birlikte sağlanması her zaman mümkün değildir. Bu nedenle çok uğraşılmasına rağmen herkesin gönül rahatlığıyla "tamam" diyebildiği bir e-oylama protokolü henüz bulunamamıştır. Belki yazımızı okuyanlardan birisi ileride bir çözüm bulur.

Arka Pencere

Bir senaryo üzerinden kriptografik protokolün önemini anlatmaya çalışalım. Alfred Hitchcock'un yönettiği "Arka Pencere" filmini görmüş müydünüz? İzlemediyseniz önemli değil. Sonunu söylemeyeceğiz ama senaryoyu biraz değiştiriyoruz. Örneğin başkahramanımız bir kadın. Haftalardır ayağı kırık bir şekilde, tekerlikli sandalyesinde oturan Ayşe can sıkıntısından evinin arka penceresinden etrafı gözetlemektedir. Ayşe bir gece karşı komşusunda korkunç bir olaya tanık olur. Komşusu evinin mutfağında ağır bir torba sürümektedir. Diğer taraftan komşunun karısı günlerdir ortalıkta gözükmemektedir. Ayşe cinayetten şüphelenerek dedektif Bora'yı aramaya karar verir. Ayşe'nin fazla vakti yok çünkü acele etmezse, komşusu delilleri yok edecek ve kaçacak. Önemli bir sorun daha var. Komşusu başkalarının hatlarından açık giden mesajları dinleyebilmekte ve dışarıda belalı arkadaşları kol gezmektedir.

Verdiğimiz örneğin, protokollerin (ya da kriptolojinin) önemini anlatmak için abartılı olduğunu iddia edebilirsiniz fakat gerçek hayatta düşman, hatları dinleyebiliyorken haberleşmeye çalışan askerler daha az tehlike altında değildir. Ya da ucunda ölüm olmayabilir ama "mal, canın yongasıdır" diyorsanız, güvensiz bir internet bankacılığı yüzünden aileniz bütün malvarlığını kaybedebilir.

Senaryodaki gibi günlük yaşamdaki kriptografik protokollerde de saldırganlar çoğu zaman dürüstlerden daha güçlü kuvvetli, yani işlem gücü çok daha yüksek ve daha beceriklidir. Ayrıca kim olduklarını tahmin edemediğimiz başka işbirlikçileri olabilir. Kriptologlar protokol ya da protokollerin yapıtaşlarını tasarlarken, kendilerini saldırgan yerine koyup buldukları çözümü alt etmeyi denerler. Saldırganın işlem gücünü, bilgisini ve işbirlikçilerini modelleyip buldukları çözümün güvenilir olduğunu ispatlamaya çalışırlar. Saldırgan modellemede en yaygın kullanılan modeller "standart" ve "(rastsal) kâhin" modelleridir.

Şimdi "Arka Pencere"mize geri dönelim. Amacımız Ayşe ile Bora arasında güvenli bir ihbar mekanizması kurmak. Bora'nın kötü niyetli olmadığını, örneğin katil zanlısı ile işbirliği yapmayacağını ve Ayşe'yi tanıdıktan sonra dediklerine kulak vereceğini varsayıyoruz. Protokolün sağlaması gereken hedefler arasında gizlilik, kimlik doğrulama ve veri bütünlüğünü korumanın yanı sıra mahremiyeti de sayabiliriz. Çünkü ihbarı kimin yaptığının komşunun arkadaşları tarafından anlaşılması Ayşe için can sıkıcı olurdu. Protokol ortamı, Ayşe ile Bora arasında telefon, cep telefonu veya internet hattı olabilir. Protokolün davetli misafirleri en azından Ayşe ile Bora. Davetsiz misafirler komşu ve işbirlikçileri.

Ne dersiniz; sizce Ayşe komşusunu yakalatabilecek mi?

Özellikle asimetrik sistemlerde imza anahtarının ele geçirildiğinin acilen bildirilmesi gerekir. Aksi takdirde saldırgan sizin adınıza geçerli imzalar atar. Elektronik ticarette birkaç günlük gecikmenin nelere yol açabileceğini siz düşünün!

Aslında anahtar yönetimi konusunda burada bahsedemediğimiz başka sorunlar da var. Örneğin imza sistemlerinde süresi dolmuş anahtarların kontrol edilmesi, açık anahtarların kullanıcılar ile ilişkilendirilmesi, geçmişe yönelik mesajların okunabilmesi için asimetrik şifreleme özel anahtarlarının arşivlenmesi, büyük gruplar için verimli anahtar oluşturma protokolü tasarlanması bunlardan sadece birkaçı.

Kripto algoritmamız sağlam. Anahtarı güzelce ürettik; sağ salim ulaştırdık; cihazın hafızasında korunaklı bir şekilde sakladık. Anahtarın yanlış kullanımını engelledik. Saldırgan anahtara cihazın içinde ya da hattan giden veriyi inceleyerek ulaşamıyor. Acaba anahtarımız güvende mi? Unutmayın saldırgan her yolu deneyecektir. Son yıllarda gelişen "yan kanal analizi" denilen yeni bir saldırı tekniği sayesinde saldırgan, cihazın kripto işlemleri sırasında harcanan zaman ve enerji gibi değerleri ölçerek anahtarı ortaya çıkarabiliyor. Yani algoritmamızın kriptoanalize karşı güvenli olması yetmez, aynı zamanda yan kanal analizlerine dayanıklı bir şekilde gerçeklenmesi gerekir.

Sağlam bir anahtar yönetiminin olduğu bir sistemde saldırgan ne anahtarı defolu üretiminden dolayı tahmin edebilir, ne saklandığı ya da "toprağa verildiği" yerden ele geçirebilir, ne taşıma ya da paylaşım sırasında çalabilir. Ne yazık ki, anahtar yönetimi anahtar güvenliği için mutlaka gerekli fakat tek başına yeterli değildir.

Kaynaklar

Koblitz, N., *Algebraic Aspects of Cryptography*, Springer, 1998.

Knuth, D., *The Art of Computer Programming*, Addison-Wesley, 1969.

Menezes, A. J., Oorschot, P. C., Vanston, S. A., *Handbook of Applied Cryptography*, CRC, 1997.

Vaudenay, S., *A Classical Introduction to Cryptography: Applications for Communications Security*, Springer, 2006.

Kaynaklar

Bogdanov, A. ve diğerleri, *PRESENT: An ultra lightweight block cipher*, CHES 2007, LNCS 7427, s.450-466, Springer, 2007.

Koblitz, N., *Algebraic Aspects of Cryptography*, Springer, Berlin, 1998.

Menezes, A. J., Oorschot, P.C. ve Vanston, S.A., *Handbook of Applied Cryptography*, CRC, NY, 1997.

Vaudenay, S., *A classical Introduction to Cryptography: Applications for Communications Security*, Springer, NY, 2006.

http://www.estream.org

http://www.iacr.org