# Dijital imzalar

Açık anahtarlı kriptografi, hesaplama açısından simetriden daha yoğundur. Bu nedenle, genellikle bir protokolün parçası olarak değiş tokuş edilen birkaç küçük bilgi bloğunu şifrelemek için kullanılır.

Dijital imza, asimetrik kriptografi kullanılarak gerçekleştirilir.

Dijital imza iki şekilde çalışır. Belgeyi gönderen C tarafından belgeyi kendisinin yazdığını kanıtlamak için kullanılabilir ve belgeyi alan kişi tarafından C'nin yazdığına dair kanıt olarak kullanılabilir. İkinci durum, reddedilmeme olarak adlandırılır: imzalı bir çek gönderdikten sonra, daha sonra bir para transferine izin verdiğinizi inkar edemezsiniz. Dijital imza aynı zamanda üçüncü ve çok önemli bir amaca da hizmet eder. Bir belge imzalandıktan sonra, imza geçersiz kılınmadan değiştirilemez. Bu nedenle, dijital imza veri bütünlüğünü garanti eder.

Dijital imzalar, birçok asimetrik şifreleme algoritmasının bir özelliğini kullanır - genel PC ve özel RC anahtarlarının rolleri simetriktir:



C'nin bir belgeyi M'yi RC özel anahtarıyla şifreleyerek imzaladığı basit fikirli bir dijital imza algoritmasıdır. Alıcı, PCkullanarak bir mesajın şifresini çözerek anlamlı bilgileri (örneğin, bir ASCII dizisi) kurtarabilirse, alıcı, mesajın yalnızca C tarafından üretilmiş olabileceği sonucuna varabilir, çünkü yalnızca C, RC'yi bilir. Bu teknik, alıcının C'nin genel anahtarını (PC) bildiğini varsayar. Mesajın gizli olmadığını unutmayın. Mesaj gizleme, dijital imza protokolünün amacı değildir.

Asimetrik şifreleme zaman alıcı olduğu sürece, M'nin bazı fonksiyonları (f) hesaplanır - genellikle bir hash - bu da M'nin kendisinden önemli ölçüde daha küçük bir sonuç üretir. f(M), M'nin mesaj özeti olarak da adlandırılır. f(M), RCile şifrelenir ve M ile birlikte iletilen bir dijital imza olarak adlandırılır.

Böylece, alıcıya iki öğe, ve M, gönderir. Alıcı, PCkullanarak ilk öğenin şifresini çözer ve ardından sonucu, f'nin ikinci öğeye uygulanmasının sonucuyla karşılaştırır. Eğer ikisi aynıysa, alıcı, M'nin yalnızca C tarafından üretilmiş olabileceği sonucuna varabilmelidir. Ancak, böyle bir sonuca güvenli bir şekilde izin vermek için, başka bazı konuları ele almalıyız.

# Dijital İmzalar ( Devam 1)

C'den imzalı bir iletimi (,M) dinleyen ve kopyalayan bir davetsiz misafir düşünün.

1. , bir alıcıyı C'nin M'yi gönderdiğine inandırmak amacıyla farklı bir mesaj olan M'yi imzalamak için imzayı kullanabilir. Davetsiz misafir M'yi f(M)=f(M') olacak şekilde yapılandırabilirse bu saldırıda başarılı olabilir. Bu saldırıyı önlemek için, f'nin tek yönlü bir fonksiyonu olması gerekir: f, bir çıktı verildiğinde, y, bir bağımsız değişken, x oluştururken, f(x )=y hesaplama açısından olanaksız olacak şekilde, özelliğine sahiptir. Örneğin, tek yönlü bir mesaj özeti fonksiyonu, aşağıdaki özellikleri karşılayan bir çıktı dizesi üretebilir:
2. f aralığındaki tüm değerler eşit derecede olasıdır
3. Mesajın herhangi bir biti değiştirilirse, mesaj özetindeki her bitin değişme şansı %50'dir.

Özellik (a), f(M)=f(M') olacak şekilde bir M' bulma olasılığına karşı koruma sağlar, çünkü tüm değerler eşit derecede olasıdır. Özellik (b), M ve M' ilişkili veya benzer mesajlar olduğu için f(M) ve f(M')'nin aynı olmamasını sağlar.

Bu koşullar altında davetsiz misafirin f(M)=f(M') olacak şekilde bir M' mesajı oluşturması ve dolayısıyla imzanın  eklenebileceği bir M' mesajı bulması son derece zordur. Ayrıca davetsiz misafir, R C'yibilmediği için M', ile kullanılabilecek imzayı taklit edemez .

1. Davetsiz misafir, yeniden yürütme saldırısı olarak adlandırılan bir durumda, imzalı bir iletiyi kopyalayıp ikinci kez yeniden göndermeye çalışabilir. İmzalayanın bir zaman damgası oluşturması ve bunu M'ye dahil etmesiyle bir yeniden oynatma saldırısının üstesinden gelinebilir. Dijital imza mesajın tamamı üzerinden hesaplandığından, izinsiz giren kişi zaman damgasını değiştiremez. Ağdaki tüm sitelerdeki saatlerin kabaca senkronize olduğunu varsayarsak, eğer alıcı yakın zamanda alınan tüm mesajların zaman damgalarının bir listesini tutarsa ve listede zaman damgaları içeren gelen mesajları reddederse (zaman damgalarının genellikle benzersiz olduğunu unutmayın), bir iletinin ikinci gelişi tespit edilebilir. Buradaki kilit nokta, belirli bir zaman damgasının asla iki kez kullanılmamasıdır. Bu nedenle, her mesajın benzersiz bir sıra numarası varsa, bir yeniden oynatma saldırısı da ele alınabilir.

# Secure Socket Layer (Güvenli Soket Katmanı) Protokolü: Sertifikalar

Bir İnternet işleminin parçası olarak kendilerini diğer taraflara doğrulamak isteyen sunucular (belki işletmeyi temsil ediyor), güvenilir bir üçüncü taraf olarak hareket eden bir sertifika yetkilisi kullanabilir.

Bir CA, bir müdürün adı (örn. Macy) ile genel anahtarı arasındaki ilişkiyi onaylayan sertifikalar oluşturmak için genel anahtar şifrelemesini kullanır. Sertifika, (diğer öğelerin yanı sıra) sorumlunun adını ve genel anahtarı içerir ve CA'nın özel anahtarıyla imzalanır. CA'nın genel anahtarı iyi bilindiğinden, sistemdeki herhangi bir işlem sertifikanın geçerliliğini belirleyebilir. Bu nedenle, bir müşteri Macy ile güvenli bir şekilde iletişim kurmak isterse, "Macy" adını içeren geçerli bir sertifikada bulunan genel anahtarı kullanarak bir mesajı şifreleyebilir ve yalnızca Macy'nin özel anahtarına sahip bir işlemin bu mesajın şifresini çözebileceğinden emin olabilir. Böylece sertifikalar, asimetrik şifreleme için anahtar dağıtım sorunu olan genel anahtarların güvenilir bir şekilde dağıtılması sorununu çözer.

Bir CA'dan sertifika almak isteyen herhangi bir İnternet sunucusu S, önce bir genel ve özel anahtar oluşturur ve ardından genel anahtarı ve diğer bilgileri CA'ya gönderir. CA, sunucunun kimliğini doğrulamak için çeşitli yöntemler kullanır (belki de sunucunun iş yerindeki personelle telefon ve normal posta yoluyla iletişim kurarak) ve ardından ona diğer öğelerin yanı sıra şunları içeren bir sertifika verir:

* CA'nın adı
* S'nin adı
* S'nin URL'si
* S'nin genel anahtarı
* Zaman damgası ve son kullanma bilgileri

CA, sertifikayı imzalar ve imzalanan sertifikayı S'ye gönderir; S, sertifikanın doğruluğunu onaylar.

# SSL Protokolü

Secure Socket Layer (SSL, Güvenli Soket Katmanı) protokolü, bir istemci ile bir İnternet sunucusu (veya sunucular arasında) arasında İnternet üzerinden güvenli iletişimi ve kimlik doğrulamayı desteklemek için sertifikalar kullanır. SSL, sertifikaları kullanarak, saniyede binlerce işlemi işleyen işlem sistemlerinde bir darboğaz olabilen çevrimiçi bir anahtar sunucusuna (Kerberos'ta olduğu gibi) olan ihtiyacı ortadan kaldırabilir.

SSL'nin amacı, bir istemciye bir sunucunun kimliğini doğrulamaktır. Bu bir sertifika kullanılarak yapıldığından, kimliği doğrulanmak isteyen her sunucunun önce bir sertifika alması gerekir.

Müşteriler ise genellikle sertifika yetkililerine kayıtlı değildir ve bu nedenle sertifikalara veya bunlarla ilişkili şifreleme anahtarlarına sahip değildir. Oturum açmış bir istemci, genellikle kendi özel anahtarı olmayan bir tarayıcı tarafından temsil edilir. Bunun yerine, tarayıcı, o tarayıcının satıcısıyla anlaşmalar yapmış olan tüm sertifika yetkililerinin genel anahtarlarını içerir. Tarayıcı, SSL protokolü sırasında bir CA ile fiilen iletişim kurmaz; bir CA, bir tarayıcı hakkında herhangi bir özel bilgi de bilmez.

SSL protokolü, sunucuyu istemciye doğrular ve kullanımları için bir oturum anahtarı oluşturur. Bir tarayıcı, URL'si https: ile başlayan ( http: yerine) bir sunucuya bağlandığında , SSL ile şifrelenmiş bir HTTP protokolünü belirtir.

Bir tarayıcının (C) belirli bir işletmeyi (örneğin, Macy's) temsil ettiğini iddia eden bir sunucuya (S) bağlandığını varsayalım. Bu durumda, protokol aşağıdaki adımlardan oluşur:

1. S, C'ye CA tarafından imzalanmış sertifikasının bir kopyasını açık bir şekilde gönderir.
2. C, CA'nın genel anahtarını (tarayıcısında bulunan) kullanarak sertifikanın imzasını doğrular ve dolayısıyla sertifikadaki genel anahtarın, sertifikada adı geçen kuruluşa ait olduğunu bilir.
3. C, sertifikadaki genel anahtarla şifrelenmiş bir oturum anahtarı oluşturur ve S'ye gönderir.

Oturum anahtarını S'nin değil C'nin oluşturduğuna dikkat edin çünkü protokolün bu noktasında C, sertifikadaki genel anahtarı kullanarak S ile güvenli bir şekilde iletişim kurabilir, ancak S, C ile güvenli bir şekilde iletişim kuramaz. Oturum anahtarı oluşturulduktan sonra, C ve S bunu şifreli mesajları değiş tokuş etmek için kullanabilir.

Birçok uygulamada S, belirli bir müşteriyle konuştuğundan emin olmak ister. Bu tür bir kimlik doğrulama sağlamanın bir yolu, istemci ve sunucunun, oturum anahtarı oluşturulduktan sonra sunucunun sakladığı ve istemcinin sağladığı bir parola üzerinde anlaşmasıdır.

Transport Layer Security(TLS, Aktarım Katmanı Güvenliği) protokolü ayrıca Diffie-Hellman (DH) anahtar değişimi ve Dijital İmza Algoritması (DSA) kullanır..

**DH Anahtar Değişimi**

jeneratör olarak seçilir .

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 2 | 4 | 8 | 5 | 10 | 9 | 7 | 3 | 6 | 1 |
| 3 | 3 | 9 | 5 | 4 | 1 | 3 | 9 | 5 | 4 | 1 |





Anahtar değişim protokolü, katılımcıların kimliğini doğrulamadığı için bu tür bir saldırıya karşı savunmasızdır. Bu güvenlik açığı, dijital imzalar ve genel anahtar sertifikaları kullanılarak aşılabilir.

**Digital Signature Algorithm (DSA, Dijital İmza Algoritması)**



DSA'nın doğruluğu ( [https://en.wikipedia.org/wiki/Digital\_Signature\_Algorithm'den](https://en.wikipedia.org/wiki/Digital_Signature_Algorithm) ):

Daha sonra

Çünkü g'nin q sırası vardır( ),

Sonunda,

.

<http://www.herongyang.com/Cryptography/DSA-Introduction-Algorithm-Illustration-p7-q3.html>adresinde mevcuttur.

# DSA Örneği

P=11, q=5, q |( p-1), h=3, , , , ,

İmzalama , H(M)=4, , imza

Doğrulama

, , ,

biz alırız