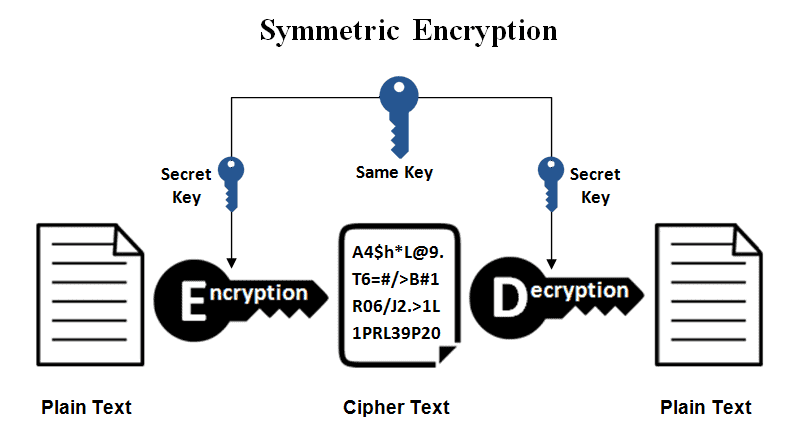
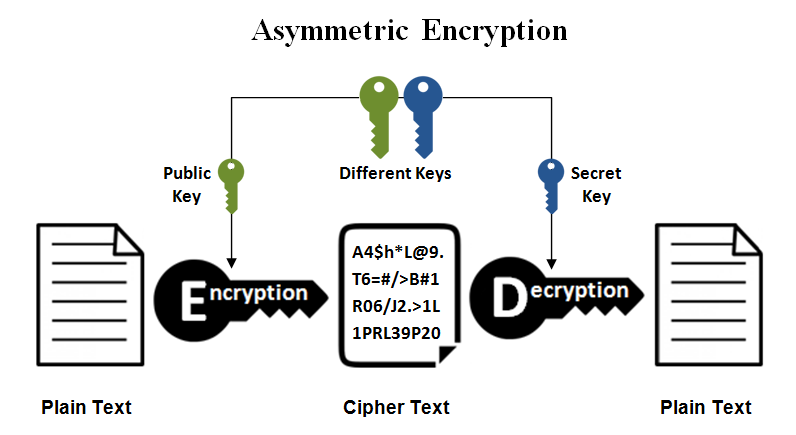
**ASİMETRİK (AÇIK ANAHTARLI) ŞİFRELEME**

Asimetrik şifreleme verileri şifrelemek için farklı, çözmek için farklı olmak üzere iki farklı anahtar kullanan bir şifreleme yöntemidir. Açık anahtar (public key) ve özel anahtar(private key) olarak ayrılan anahtarlardan açık anahtar şifrelemek için kullanılır. Özel anahtar ise şifreyi çözmek için kullanılır. Bu sayede sadece özel anahtara sahip olan kişi şifreyi çözebilme yeteneğine sahip olur. Bu şekilde genel anahtarın deşifre olması verilerin gizliliğini tehlikeye atmaz. Ayrıca veriler özel anahtar ile imzalanmış olur ve böylece kaynağın doğruluğu da tespit edebilir.

**Asimetrik ve Simetrik Şifreleme Farkları**

Simetrik ve asimetrik şifreleme için en büyük fark, simetrik şifrelemede aynı anahtar hem verileri şifreleyip hem de şifreyi çözebilirken asimetrik şifrelemede public key ile şifreleme private key ile şifrenin çözülebilmesi işlemidir. Aslında bu farktan dolayı ortaya yeni farklar çıkmıştır:

* Simetrik şifrelemede kullanılan gizli anahtarın ele geçirilmesi güvenliği yıkabilir fakat asimetrik şifrelemede şifrelenen anahtar zaten herkese açık olduğu için bu anahtarın bulunması güvenlik açısından tehlike yaratmaz.
* Simetrik şifreleme yaptığı şifreleme işleminin daha kısa sürmesi sebebiyle hızın daha önemli olduğu sistemlerde tercih edilir. Asimetrik şifreleme ise hassas verilerin bulunduğu, güvenliğin çok daha önemli olduğu sistemlerde kullanılır.
* Veri bütünlüğü ve doğrulamanın sorgulanabileceği bir sistem isteniyorsa asimetrik şifrelemede kullanılan dijital imzalar bu sorgunun yapılabilmesine olanak tanır bu da bir tercih sebebidir.
* Büyük verileri şifrelemek için simetrik şifreleme, bu şifrelemenin anahtarının gizlenmesi ve daha küçük veriler için asimetrik şifreleme tercih edilir.
* Simetrik şifreleme 🡪 dosya şifreleme, VPN, disk şifreleme; AES, DES protokolleri
* Asimetrik şifreleme 🡪 dijital imza, e-posta güvenliği, SSL/TLS , RSA, ECC gibi sistemlerde kullanılır.



**RSA Algoritması**

RSA, geliştiricileri Rivest, Shamir, Adleman’ın baş harflerini taşıyan bir asimetrik şifreleme algoritmasıdır. RSA private key ve public key asal sayıların bazı işlemler ile birlikte oluşturdukları karmaşık bir şifreleme yapısıdır. Burada asal sayıların kullanılmasının sebebi, asal sayıların çarpanlara ayrılmasının imkansıza yakın olmasından kaynaklanır. Bu sayede decrypt etmek çok zorlaşır. Algoritma uygulanırken:

* Önce p ve q üzerinden iki asal sayı seçilir.
* Sonra bu asal sayıların çarpımından bir n sayısı oluşturulur.
* Sonra p, q ve n sayılarına Euler fonksiyonu uygulanır. 🡪φ(n) = (p-1)\*(q-1)
* Bir sayı seçilir ve buna e diyelim. Bu sayı 1 ile önceki adımda bulduğumuz φ(n) sayısı arasında olmalıdır ve e ile φ(n) sayısı aralarında asal olacak şekilde bu e sayısı belirlenmelidir.
* Son olarak bu sayılarla d \* e = 1 (mod φ(n)) işlemi uygulanarak buradaki d değeri denklem çözümünden bulunur.
* Bu sayılar ile oluşan public key🡪(n,e) , private key 🡪(n,d) şeklinde olacaktır.

Örnek: p=3 , q=11 seçilirse n=33 olur

* φ(n) = (3-1)(11-1)=20 bulunur.
* e yi 1 ile 20 arasında ve 20 ile aralarında asal olacak bir sayı seçelim e=3
* d\*3= 1 mod 20 ise , d buradan 7 olabilir.
* Açık anahtar= (33,3) , Özel anahtar= (33,7)

M=4 şifrelenmek istenirse;

* Şifreli mesaj (C) açık anahtar ile şifrelemek için : C = 4^3 mod33=64 , 64 mod 33=31
* Şifre çözme için : M = C^d mod n = 31^7 mod 33 = 4 şeklinde bulunur

>>> RSA genellikle SSL/TLS gibi HTTPS protokollerinde dijital imzalarda, PGP/GPG gibi e posta güvenlik protokollerinde, kimlik doğrulamada ve kripto para cüzdanlarında kullanılır. Yavaş olması nedeniyle genelde anahtar şifrelemede kullanılır. Ayrıca bit sayısının yüksekliğinden dolayı işlem gücü gerektirir.

**ElGamal Şifreleme**

Asimetrik şifreleme yöntemlerinden biri olan ElGamal şifreleme aşamaları: Anahtar üretimi, şifreleme ve açma şeklindedir. Dairesel gruplardan faydalanılarak ayrık logaritmik işlemlerle şifreleme yapılır.

Anahtar üretimi için:

1. Büyük bir asal sayı (p) ve g üreteci (generator) seçilir.
2. Sonrasında 0 ile p-1 arasında rastgele bir x sayısı seçilir.
3. y = g^x mod p işlemleri yapılarak açık anahtar hesaplanır.
4. Böylelikle açık anahtar (p,g,y), gizli anahtar x olarak belirlenir.

Şifreleme ve şifre çözme için:

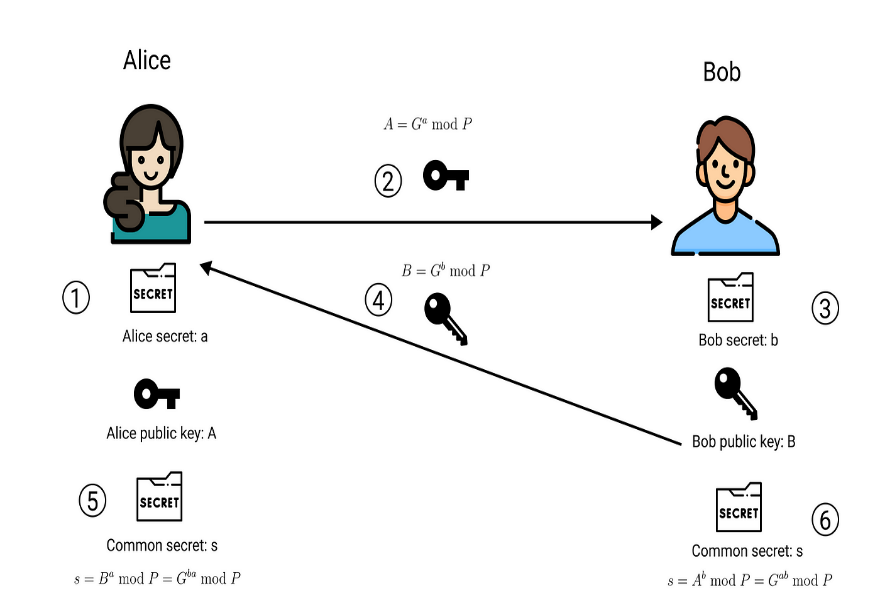
1. Bir M mesajını şifrelemek istiyorsak önce bir k sayısı seçmemiz gerekir
2. Hesaplamak için c1= g^k mod p 🡪 geçici anahtar gibi düşünülebilir
3. c2 = M \* y^k mod p işlemleri yapılır. Bu da asıl mesajın şifrelenmiş halidir.
4. Şifreli mesaj = (c1, c2)
5. Şifreyi çözmek içinse gizli anahtar olarak kullandığımız x ile M =c2 \* (c1^x)^(-1) mod p işlemini uygularız.

ElGamal, şifreleme yaparken alıcıdaki açık anahtarın rastgeleliği ile sürekli farklı şifreli mesaj üretir. Alıcı da bu mesajı özel anahtar kullanarak okuyabilir. Burada bu işlemlerin arkasındaki temel mantık şifreleme işleminin modüler üs alma ile kolayca hesaplanabilmesi şifre çözme aşamasında tersten gittiğimizde logaritmik hesaplama işleminin oldukça zor olmasıdır. ElGamal şifreleme yaparken alıcıdaki açık anahtarın rastgeleliği ile sürekli farklı şifreli mesaj üretir. Alıcı da bu mesajı özel anahtar kullanarak okuyabilir.

* Rastgelelik içermesinden dolayı yüksek güvenliklidir.
* Güçlü bir matematiksel algoritmaya dayanır.
* Dijital imza bulundurur.
* Büyük sayılarla çalıştığı için yavaştır ve şifreli metin de açık mesajın iki katı kadar büyüktür.

**Diffie-Hellman Anahtar Değişimi**

Kriptografik anahtar değişimleri için kullanılan özel bir metot olmakla birlikte bu alanda ilk anahtar değişimi örneklerindendir. Bu anahtar değişimi protokolünde amaç güvenli ve gizli iletişim kurmak isteyen iki kişinin ortak bir simetrik anahtar kullanarak kendi aralarında gizli bir iletişim kurma yolu oluşturmalarını sağlar. Burada konuşma açık ve izlenebilir bir ortamda yapılmasına rağmen iki tarafın konuşması sadece iki tarafın bildiği gizli bir anahtarla şifrelendiği için iletişim de gizli kalacaktır.



🡪 Resimde görüldüğü üzere önce Alice gizli bir sayı (a) seçer. Bu onun gizli anahtarıdır.

🡪 Alice açık anahtarını hesaplar g (üreteç) ve p (asal sayı) olmak üzere A =g^a mod P

🡪 Hesapladığı açık anahtarı Bob’a gönderir. Bu sefer Bob gizli bir sayı (b) seçer.

🡪 Bob da B= g^b mod P hesaplaması ile açık anahtarını oluştur ve Alice’e gönderir.

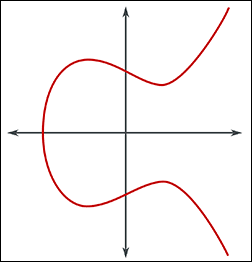
🡪 İki tarafın elinde de kendi gizli anahtarı ve karşı tarafın açık anahtarı var

🡪 5. Adımda Alice, Bob’dan gelen gizli anahtarı kendi belirlediği gizli anahtar ile şifreler. 6.adımda da Bob, Alice’den gelen A’yı kendi gizli anahtarıyla işleyerek aynı ortak sıraya ulaşırlar.

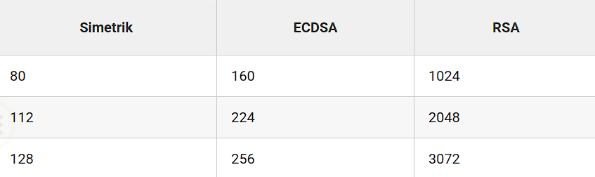
Bu şifrelemenin güvenli olma sebebi dışardan biri sadece public keyleri (A ve B) , g ve p yi görebilir. Fakat a ve b verilerini görmeden şifreyi tamamlamak mümkün değildir. Onları bulmak içinse ayrık logaritma işlemi gerekir fakat bu işlem büyük sayıların bu logaritmik işlemlerde kullanılması sebebiyle günümüzdeki bilgisayarların hesaplama kapasitesinden çok daha fazlasını ister. Bu da, Diffie Hellman şifrelemenin güvenli olduğunu gösterir.

* TLS/SSL gibi HTTPS güvenlik protokollerinde, her oturumda yeni anahtar oluşturulmasında yani tarayıcı ile sunucu arasında gizli bir anahtar oluşturulmasında etkilidir.
* VPN istemcisi ile sunucusu arasında ortak anahtar belirlenmesinde ve bu anahtarla tüm trafiğin gizlenmesi için kullanılır.
* GPG, eposta şifrelemek için bu yönteme benzer şekilde gizli anahtar değişimi yapar.
* SSH oturumlarında istemci ile sunucu arasında ortak şifreleme anahtarı oluşturmak için
* Mobil uygulamalarda da uçtan uca şifreleme yapan uygulamalar için bağlantı kuran cihazlar arasında ortak anahtar oluşturur.

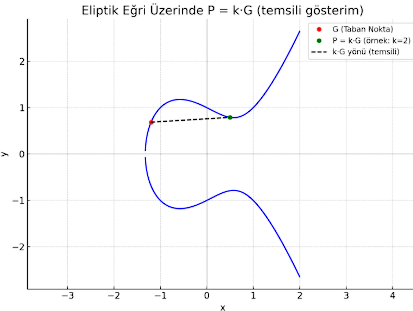
**Elliptic Curve Cryptography (ECC)**

Elliptik eğri şifreleme, kriptografik sistemlerin elliptik eğrilerin matematiksel özelliklerini kullanarak oluşturulduğu açık anahtarlı bir şifreleme örneğidir. Diğer asimetrik şifreleme algoritmalarındaki gibi şifrelemesi kolay fakat tersten çevirme işleminin oldukça zor olduğu bir matematiksel zorluk içerir. Bu zorluk ECC’nin güvenliğini sağlayan temel zorluktur. İçerisinde bir imzalama algoritması olan ECDSA (elliptic eğri dijital imza algoritması) taşımasıyla veri bütünlüğünü ve orijinalliğini kanıtlayıcı bir özelliği de vardır.

🡪 y²=x³ + ax + b

Daha yaygın kullanılan RSA algoritmasına göre farkı ECDSA imza algoritmasının RSA’ya göre çok daha düşük anahtar boyutlarıyla RSA’ya eşdeğer bir kriptografik güç sağlamasıdır. Günümüzde büyük verilerin simetrik şifrelemelerle en performanslı şifreleme olduğunu görebiliriz. ECC algoritması da simetrik şifreleme performansına en yakın asimetrik şifreme örneğidir. Çünkü ECDSA anahtar boyutu simetrik şifreleme anahtar boyutunun 2 katı kadar büyüklükte olabilirken RSA anahtar boyutunun simetrik şifrelemenin çok daha büyük katlarına ulaşmaktadır

ECC algoritma olarak gizli anahtar (k) ve açık anahtar P= k\*G denkleminin kullanarak şifreleme yapar. Burada ECDLP, elliptik eğri ayrık logaritma problemi denilen matematiksel problemin zorluğu ECC şifreleme güvenliğinin temelidir. Buradaki matematiksel problem, taban noktası olarak seçilen G’nin onun katı olan başka bir P noktası verildiğinde P=k\*G denkleminde k’yı bulmasının nerdeyse imkansız olduğu bir problemdir. Bu da k gizli anahtarının güvenliğini sağlayacak en büyük etkendir.



* TLS/SSL gibi HTTPS bağlantılarında,
* Kripto paralarda kullanılan imzalama algoritmalarında,
* Dijital imzalarda,
* NFC, IOT cihazlar, akıllı kartlarda,
* SSH,VPN, eposta şifrelemelerinde kullanılır.

**Saldırılara Karşı Güvenlik Önlemleri**

Asimetrik şifreleme koruma ve güvenlik açısından her ne kadar deşifre etmesi zor algoritmalar kullansa da bu şifrelerin yanlış kullanımı ve bazı farklı durumlar güvenlik açıkları meydana getirebilir. Bu durumları ortadan kaldırmak için bu durumları ortadan kaldırmak ve bazı önlemler almak önemlidir. Bu durumlara örnek olarak:

* Brute force saldırılarına karşı algoritmaların ürettiği şifreler her ne kadar uzun ve tahmin edilmesi güç şifreler olsa da yeterince uzun seçilmeyen anahtarlar brute force saldırılarıyla keşfedilebilir. Bu yüzden RSA için 2048 bit, ECC için en az 256 bit uzunluğunda anahtar kullanılmalıdır.
* Matematiksel saldırılara karşı örneğin RSA algoritmasında kullanılan asal sayıların bulunması anahtarı açığa çıkaracağı için güçlü asal sayıların kullanımı çok önemlidir. Diğer algoritmalarda kullanılan matematiksel işlemlerde de kullanılan sayıların güçlü olması önemlidir.
* Man in the Middle atak olarak bilinen ataklarda aradaki adam ağa girerek trafiği kendi üzerinden yönlendirebilmektedir. Bu da kullanılan şifreleri de bu trafikle birlikte görüntüleyebileceği anlamına gelir. Bu yüzden doğrulama yapmak, parmak izi gibi bir anahtarlar kullanmak, dijital sertifikalar ile bağlantı güvenilirliğini test etmek önemlidir.
* Aynı anahtarların tekrar tekrar kullanılması güvenliği tehlikeye atan bir durumdur. Bu yüzden sayı üreteçlerinde rastgele sayı üretme gücünün yüksek olması önemlidir.
* Özet olarak anahtarlar güvenle saklanmalı, her işlem için tek kullanımlık anahtar üretilmeli, şüpheli durumlar izlenmeli, şüpheli işlemlerin olması durumunda loglar izlenmelidir.
* Protokollerin doğrulamasının düzgün yapılabilmesi için güncel sürümler kullanılmalıdır, sertifikaların güvenilirliği ve geçerlilik süreleri kontrol edilmelidir.