Kleptografi: Kriptografik Sistemlerde Arka Kapılar

Bu tez Bilgi Güvenliği Mühendisliği'nde
Tezli Yüksek Lisans Programının bir koşulu olarak

Emre CERAN
tarafından

Fen Bilimleri Enstitüsü'ne
sunulmuştur.

İSTANBUL
ŞEHİR
ÜNİVERSİTESİ

Bu tezi okuduk, kapsam ve nitelik açısından Bilgi Güvenliği Mühendisliği alanında Yüksek Lisans derecesi için tümüyle uygun olduğu görüşüne vardık.

ONAYLAYANLAR:

Dr. Mehmet Sabır Kiraz .....
(Tez Danışmanı)

Dr. Osmanbey Uzunkol .....
(Tez Eş-danışmanı)

Prof. Dr. Tahsin Erkan Türe .....

Yrd. Doç. Dr. Seher Tutdere .....

Yrd. Doç. Dr. Erdinç Öztürk .....

Bu tez İstanbul Şehir Üniversitesi, Fen Bilimleri Enstitüsü tarafından belirlenen tüm koşullara uygundur.

ONAY TARİHİ: 26 Mayıs 2016

MÜHÜR/İMZA:

Yazarlık Beyanı

Ben, Emre CERAN, başlığı 'Kleptografi: Kriptografik Sistemlerde Arka Kapılar' olan tezin ve içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik alınak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayınlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Esaslı yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

İmza: 

Tarih: 26.05.2016

Kleptografi: Kriptografik Sistemlerde Arka Kapılar

Emre Ceran

Öz

"Kriptografik bir sistemden, gizli bilgileri farkedilmeden ve sadece algoritmik değişiklerle çalabilme çalışmaları" olarak özetleyebileceğimiz Kleptografi alt disiplinini incelediğimiz bu çalışmada, kleptografik atak senaryolarını, ilgili algoritmaları ve bu algoritmaların, atak barındırmayan standart algoritmaların gerçeklenmesi ile oluşan sonuçların karşılaştırmalı analizleri ele alınacaktır.

RSA şifreleme sistemi, Diffie-Hellman anahtar değişimi, ElGamal şifreleme, DSA dijital imza ve SSL/TLS protokolüne karşı, çeşitli araştırmacıların 90'lardan günümüze kadar sunduklar ataklar ve ilgili diğer çalışmalar tez kapsamında incelenecektir.

Anahtar Sözcükler: Kriptografi, Kleptografi, Kriptografik Arka Kapı, RSA, Diffie Hellman, Eliptik Eğriler, SSL/TLS

Teşekkür

Öncelikle danışmanlarım Mehmet Sabır Kiraz ve Osmanbey Uzunkol'a; bu tez çalışması esnasında verdikleri destek, paylaştıkları bilgi birikimleri ve gösterdikleri yol ve yöntemler dolayısıyla teşekkür ediyorum. Gerektiği zaman motive ederek gerektiği zaman ise konudan sapmamı engelleyecek şekildeki yönlendirmeleri ve ihtiyacım olan odak noktalarını bana göstermeleri, bu çalışmayı tamamlayabilmemde en önemli desteklerin başında geldiğini söylemem gerekir.

Ayrıca hiçbir zaman desteklerini esirgemeyen aileme ve İstanbul Şehir Üniversitesi'ndeki çalışma arkadaşlarıma da teşekkürü bir borç biliyorum.

Son olarak yüksek lisans tezimi '2211 Yurt İçi Lisansüstü Burs Programı" ile destekleyen TÜBİTAK'a şükranlarımı sunuyorum.

İçindekiler

Şekil Listesi

Tablo Listesi

Bölüm 1

Giriş

Kriptoloji uzun yıllardır insanoğlu tarafından mahrem bilginin saklanması amacıyla kullanılmıştır. Bilişim teknolojilerinin ve dolayısıyla internet teknolojilerinin yaygınlaşmasıyla kriptoloji, daha fazla önem kazanmıştır. Şifreleme sistemlerinin geliştirilmesi ve bu sistemlere karşı yapılan saldırı mekanizmalarının araştırılması 1980'lerden sonra üstünde çokça uğraşılan konulardan olmuştur.

Aynı süreçte bilim adamları şu soruyu sormuşlar ve cevap aramışlardır: "Bir şifreleme sistemini üretirken veya gerçeklerken, kullanımı sırasında saldırgan tarafa istediği zaman mahrem bilgiyi sızdırabilecek şekilde sistemler tasarlanabilir mi?". Bu soru, etik olarak üzerinde çalışılmaması gereken bir konu gibi görünse de şifreleme sistemlerinin kritik yerlerde kullanılması hasebiyle hiçbir devletin uzak duramayacağı kadar hassas bir konudur. Dolayısıyla bu konunun incelenmesi ve varsa tedbirlerinin alınması gerekmektedir.

Saldırgan tarafına şifrelenen bilgiyi sızdıran bir sistemi, arka kapı barındıran bir sistem olarak görebiliriz. Ek olarak çalınacak bilgiyi sadece saldırganın çalabileceği şekilde sızdıracak ve aynı zamanda arka kapının farkedilmemesini de sağlayacak atak mekanizmaları çalışmaları, kleptografi çalışmalarının bütününün oluşturmaktadır.

1.1 Kleptografi Çalışmaları

"Mahrem bilgiyi, güvenli ve farkedilmeden çalabilme çalışmaları" olarak özetleyebileceğimiz Kleptografi tanımını ilk olarak Young ve Yung 1996 yılındaki [1] çalışmalarında yapmışlardır. Ancak yazarların kendi çalışmalarında da belirttikleri gibi bu fikrin temeli, Gus Simmons'ın 1984 yılında [2]'de öne sürdüğü "subliminal kanallar" fikrine dayanmaktadır. Simmons bu çalışmada normalmiş gibi görünen bir iletişim hattında, aslında başka bir haberleşmenin gerçekleşebileceği fikrini öne sürmüştür. Aynı çalışmada bazı dijital

imza algoritmalarında, rastgele seçilmesi gereken değerleri, mesaj taşıyabilecek şekilde belirleyerek, subliminal kanallar için uygulama örnekleri de verilmiştir.

Kleptografi'nin tanımını yapan ve bu alanda en çok çalışma yayınlayan isimler olarak karşımıza çıkan Moti Young ve Adam Yung; 1996 yılında yayınladıkları [1] çalışmalarında Kleptografi tanımıyla beraber RSA [3] ve ElGamal Şifreleme [4], DSA [5] dijital imza ve Kerberos protokolüne [6] karşı kleptografik ataklar sunmuşlardır. Yine bu çalışmadan bir yıl sonra [7] çalışmalarında Kleptografik bir atak için güvenlik seviyeleri belirlemişler ve Diffie Hellman anahtar değişim protokolü [8] için kleptografik atak sunmuşlar ve RSA için sundukları kleptografik atağı geliştirmişlerdir.

2003 yılında Crépeau ve Slakmon [9] çalışmalarında RSA şifreleme sistemine karşı arka kapılar kurgulamışlardır. Çalışmalarındaki arka kapılardan bir tanesi, Coppersmith'in [10] çalışmasında sunduğu, RSA şifrelemede gizli asalların hepsini bilmek yerine, sadece bitsel gösterimlerinin yarısını bilerek gizli anahtarı ele geçirilebileceğini ispatladığı Kısmi Bilgi Atağı'nın kullanıldığı, ilk arka kapı çalışması olarak ön plana çıkmaktadır. Bu atağın detaylı analizi ve çalışmadaki diğer ataklar hakkında özet bilgiler Bölüm 3.4'te ele alınacaktır.

Young ve Yung ikilisi 2004 yılında Kleptografik atak senaryoları ve ilgili diğer başlıkları detaylıca ele aldıkları "Malicious cryptography: Exposing cryptovirology" isimli kitaplarını yayınlamışlardır [11]. Bundan sonra yazarlar 2006, 2007 ve 2010 yıllarında eliptik eğrilerin kullanıldığı atak çalışmaları üzerine üç çalışma sunmuşlardır [12–14]. Bu çalışmalar da Bölüm 5'te detaylı olarak incelenecektir.

Bu süreçte kleptografi veya özel olarak arka kapılar kurgulama alanında çalışmalar sunan diğer yazarlar, daha çok kriptografik protokollere arka kapı çalışmaları ve arka kapı içeren bir sisteme karşı önlem alma çalışmaları yapmışlardır.

Kriptografik protokollere karşı arka kapı ataklarında SSL/TLS protokolü ön plana çıkmaktadır. Bu yönde Golebiewski ve arkadaşlarının [15] çalışmaları, Goh ve arkadaşlarının [16] çalışmaları ve Young ve Yung'ın [12–14] çalışmalarında SSL/TLS protokolüne karşı kleptografik ataklar bulunmaktadır.

Gogolewski ve arkadaşları 2006 yılında yayınladıkları [17] çalışmalarında e-seçim sistemlerine, 2008 yılındaki [18] çalışmalarında ise online açık attırma sistemlerine karşı kleptografik ataklar sunmuşlardır.

2013 yılına kadar bu çalışmalar, kriptologların uğraştığı ve sadece teorik olan çalışmalar zannedilirken; *The New York Times* [19] ve *The Guardian* [20] gazetelerinde, Edward Snowden'in ortaya çıkarttığı gizli NSA belgelerine dayanarak yayınlanan haberlere göre

NSA, şifreleme sistemleri üreticisi firmalarla ürettikleri sistemlerin, sadece NSA'in faydalanabileceği şekilde zafiyet barındırmaları için yıllık 250 Milyon dolarlık bir projeyi uyguladığını açıklamışlardır. Bu belgede ayrıca NIST'in (Amerika Ulusal Standartlar ve Teknoloji Enstitüsü) 2006 yılında yayınladığı *"Special Publication 900-90"* [21] rastgele sayı üreteçleri önerilerine de atıf yapılmaktadır [22]. Bu belgede yer alan üreteçlerden özellikle bir tanesi dikkat çekmektedir. Bu üreteç *Dual_EC_PRBG* olarak isimlendirilen eliptik eğrilerin kullanıldığı rastgele sayı üretecidir ve araştırmacılar tarafından, Snowden'in ifşa ettiği belgeler yayınlanmadan önce de bu üretecin yavaş olduğu ve yavaş olmasına rağmen standartlaşmış olmasının zafiyet barındırmasından kaynaklandığı hakkında çalışmalar yapılmıştır [23–25] . Sonuç olarak, Snowden haberleri ile de birleştirildiğinde bu üretecin arka kapı barındırdığı söylenebilir.

NSA'in arka kapıları haberinden sonra araştırmacılar, kleptografi ile daha çok ilgilenmişler ve çoğunlukla arka kapı barındıran/barındırabilecek bir sisteme karşı önlemler ile ilgilenmişlerdir. Bunlara örnek olarak Mironov ve arkadaşlarının 2015 yılında öne sürdükleri *Kriptografik Reverse Firewalls*, yani *Kriptografik Ters Ateş Duvarları* olarak çevirebileceğimiz sistemi sunmuşlardır [26]. Bu sisteme göre ateş duvarı normal bir ateş duvarı gibi sistemin dışarıdan gelen paketleri kontrol etmek yerine, içeriden çıkan paketleri rastgelelik katarak arka kapı yerleştirmiş olabilecek bir saldırgana karşı önlem almaktadır. Diğer bir çalışma ise aynı yıl Russel ve arkadaşlarının *Cliptography* ismini vererek öne sürdükleri sistemdir [27] . Yazarlar bu çalışmada, rastgele sayı üreteçleri olarak tek yönlü fonksiyonları incelemişler ve herhangi bir sistemin arka kapı barındırdığı ön kabulü altında alınabilecek önlemler üzerine çalışmalarda bulunmuşlardır.

1.2 Atak Senaryosu

Şifreleme sistemlerinin kullanıcılarından, şifreledikleri mahrem bilgileri çalabilmeyi amaçlayan bir kapalı-kutu (incelemeye kapalı veya zorlaştırılmış sistemler) şifreleme sistemleri üreticisini saldırgan olarak düşünelim. Saldırgan, sistemlere arka kapılar kurgulamaya çalışmaktadır. Öncelikli hedefi arka kapıyı tespit edilemeyecek şekilde üretmektir. Bunun için de kullandığı sistemde arka kapı olduğundan şüphelenen bir kullanıcının ilk kontrol edeceği gösterge olan çıktıları, normal bir sistemin çıktıları ile uyuşmasını sağlamaya çalışacaktır. Yani arka kapı barındıran sistemin çıktılarının olasılık dağılımını, standart bir sisteminki ile örtüşmesini sağlayacak şekilde çalışacaktır. Ancak kullanıcının çıktıların dağılımından sonra bakabileceği gösterge olarak çalışma zamanı olduğundan, arka kapı barındıran sistemi, normal sistemin çalışma zamanı ile aşırı farklılıklar olmadan kurgulaması gerekecektir.

Yukarıdaki senaryo ile beraber Young ve Yung [1] çalışmalarında şöyle bir uygulama senaryosu da sunmuşlardır. Bu senaryoya göre arka kapı devletin (yasalara göre talep edilmesi halinde) istediği zaman kullanabileceği şekilde tasarlanacaktır. Buna göre gerektiği zaman dinlenebilecek böyle arka kapı barındıran sistemleri herkesin kullanması sağlanacak ve gerektiği zaman devlet istediği sistemi dinleyebilecektir. Ancak bu durumda önemli bir güvenlik unsuru daha ortaya çıkmaktadır. Bu da böyle arka kapıların sadece ve sadece saldırgan tarafından kullanılabilmesi gerekliliğidir. Yani bir arka kapı başka sistemlerde bir şekilde (tersine mühendislik) ele geçirilse bile diğer kullanıcıların mahrem bilgilerini çalabilmek ancak saldırgan tarafından mümkün olabilmelidir. Bu faktörleri de göz önünde bulunduran arka kapı barındıran sisteme tasarlayıcıları hem sistemin normal kullanım halindeki gerekli kriptografik güvenlik gerekliliklerini sağlayabilmesini hem de arka kapının normal bir sistemden ayırt edilememesi ve kullanıcının (saldırgan hariç herkese karşı) mahrem bilgisini muhafaza edebilmesini amaçlayacaklardır.

Bu senaryolarda bahsedilen güvenlik gerekçelerini de göz önünde bulundurarak kleptografik bir atağın sağlaması gereken özellikler, tanım olarak aşağıdaki gibi olacaktır.

1.3 Kleptografi

Kleptografi bu alanda en çok çalışma yayınlayan isimler olan Young ve Yung tarafından [1] eserlerinde, *"Bilgiyi subliminal ve asimetrik olarak çalabilme çalışmaları"* olarak tanımlanmışlardır. Burada iki önemli odak noktası karşımıza çıkmaktadır.

1. Kleptografik bir atak asimetrik olmalı: Çalınmak istenen bilgi saldırgandan başkası tarafından ele geçirilememelidir.
2. Kleptografik bir atak subliminal olmalı: Kurban yani kullanıcı, şifreleme sisteminde atak olup olmadığını farkedememelidir.

1.4 Kleptografik Bir Atağın Güvenliği

Kleptografik bir atağın güvenlik durumunu incelerken, normal bir kripto sistemin güvenliğini incelemekten biraz daha faklı düşünmemiz gerekecektir. Çünkü kleptografik bir atakta, biri diğerinin içine gizlenmiş iki ayrı şifrelemenin güvenliği ve bu sistemlerin birbiriyle uyumlu olması amaçlanmaktadır. Normal bir kripto sistemde en önemli hedef gizlilik; yani mahrem bilginin korunması iken kleptografik bir sistemde, gizliliğin yanında atağın deşifre olmaması için normal sistemle aynı ölçülebilir özelliklere sahip olmasına dikkat edilecektir. Kleptografik bir atak barındıran bir sistemin çıktılarının normal bir

sistemle aynı özelliklere (çıktıların olasılık dağılımı) sahip olması (ayırt edilemezlik) incelenecek en önemli özelliklerdendir. Bir diğer ölçülebilir özellik ise sistemin çalışma zamanı, çıktıların olasılık dağılımını normal sisteminkine benzetebildiğimiz taktirde sistemin atak barındırıp barındırmadığını test eden bir kullanıcı ikinci olarak bakabileceği tek özellik çalışma zamanının beklenen süre içinde gerçekleşip gerçekleşmeyeceğidir.

1.4.1 Atakların Simülasyonu ve Analizleri

İlerleyen bölümlerde incelenecek ataklarda, ilgili algoritmaları inceledikten sonra; bu algoritmaların gerekli kodlarını yazarak simülasyonlarını ve bu simülasyonlarla üretilecek çıktıların standart algoritmaların çıktıları ile kıyaslamalı analizlerine yer vereceğiz.

Bu simülasyonları, Intel Core2 Duo CPU P8800 işlemci ve 16GB RAM'e sahip, Windows 7 işletim sisteminde çalışan bir bilgisayarda, Python 2.7 programlama dili ile gerçekleştirdik.

Bu çalışmanın diğer bölümlerinin organizasyonu aşağıdaki gibidir:

Organizasyon: Bu çalışmada 2. bölümde okuyucunun diğer bölümlerdeki atakları ve bu atakların uygulamalarının okumada yardımcı olabilecek gerekli matematiksel ve kriptografik altyapı bilgileri verilecektir.

3. bölümde ise RSA şifreleme sistemine kurgulanmış bazı ataklar sunulacak ve bu atakların analizleri ele alınacaktır.

Bölüm 4'de güvenliği Ayrık Logaritma Problemine dayana bazı sistemlere karşı kurgulanmış kleptografik ataklar ele alınacaktır.

Bölüm 5'de ise kleptografik ataklar açısından en verimli sonuçların alındığı eliptik eğrilerle kurgulanan ataklar sunulacaktır.

Bölüm 2

Gerekli Altyapı

Bu bölümde okuyucunun ihtiyaç duyabileceği gerekli matematiksel altyapı özet halinde ele alınacaktır. Bunun için öncelikle özet cebir bilgileri, sonrasında sayılar teorisi ile ilgili bazı tanım ve teoremler ele alınacaktır.

2.1 Cebir ve Sayılar Teorisi

Bu çalışmada ele alınacak şifreleme sistemleri ve ilgili ataklarda birçok cebirsel tanım ve teorem yer alacaktır. Bu bölümde, gerekli cebirsel ve sayılar teorisi ile ilgili bilgileri verilecektir.

Grup, Halka ve Cisim cebirsel yapıları ile ilgili temel tanımlamaları ve kriptografide kullanılan ilgili temel teoremler, Asal Sayılar ve Modüler Aritmetik konularında kriptografi ile ilgili temel tanım ve teoremler bu çalışmada ele alınmayacak ancak okuyucunun hatırlamakta zorlanabileceği bazı özel tanım ve teoremler bu bölümde ele alınacaktır. Yukarıda bahsedilen temel tanım ve teoremler ile ilgili detaylı bilgiler [28], [29] kaynaklarında bulunmaktadır. Türkçe ve İnternet'te bulunabilecek kaynaklar ise [Erhan Güzel Cebir Sayfası `http://web.iku.edu.tr/~eguzel`] ve [Marmara Üniversitesi Fen-Edebiyat Fakültesi Cebir Ders Notları `http://mat.fef.marmara.edu.tr/ogrencilere/cebir-ii-ders-notlari/`] sayfalarında bulunabilir.

Bu çalışmada yoğunluklu olarak RSA şifreleme sistemine karşı kurgulanan kleptografik ataklar ele alınacaktır. RSA şifrelemede bir m mesajını şifrelemek için, p ve q rastgele seçilecek asal sayılar, $n = p \cdot q$ açık anahtar olmak üzere, diğer açık anahtar e ve gizli anahtar d değeri, Euler Phi fonksiyonu adı verilen bir fonksiyon olmak üzere;

$$d \equiv e \bmod \varphi(n)$$

şeklinde belirlenmektedir.

Tanım 2.1 (Euler Phi Fonksiyonu)**.** $n \in \mathbb{Z}^+$ tam sayısı için, n'den küçük ve n ile aralarında asal olan sayıların sayısı $\varphi(n)$ ile gösterilir ve Euler Phi fonksiyonu adı verilir.

Euler Phi fonksiyonu özellikleri [28] :

(i) p asal ise $\varphi(p) = p - 1$

(ii) $EBOB(m, n) = 1$ ve $m, n \in \mathbb{N} \to \varphi(mn) = \varphi(m)\varphi(n)$

(iii) $n = p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n}$ ise

$$\varphi(n) = n\,(1 - \frac{1}{p_1})\,(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_n})$$

RSA şifreleme sisteminde, açık anahtarlar (n, e) ve gizli anahtar d değerleri belirlendikten sonra; m mesajını şifrelemek için

$$c = m^e \bmod n$$

operasyonu kullanılmaktadır. Şifrelenmiş mesajı açmak isteyen kullanıcı

$$m \equiv c^d \bmod$$

işlemiyle açık metine ulaşabilecektir.

Bu işlemin doğru olmasının sebebi aşağıda vereceğimiz Fermat Teoremi'dir.

Teorem 2.1. $n \geq 2$ tam sayı olmak üzere;

(i) (Euler Teoremi) : $a \in \mathbb{Z}_n^*$ ve $EBOB(a, n) = 1$ ise $a^{\varphi(n)} \equiv 1 \bmod n$ [28].

(ii) (Fermat Teoremi) : $EBOB(a, p) = 1$ ise $a^{p-1} \equiv 1 \bmod p$ [28].

Fermat Teoremi'ni kullanarak RSA şifreleme sisteminde şifrelenmiş mesajın çözümünün açık metine eşit olacağını aşağıdaki gibi görebiliriz;

$$Dec_d(Enc_e(m)) = Dec_d(m^e \bmod n) = (m^e)^d \bmod n = m^{ed} \bmod n$$

Yukarıdaki denklemde $Enc_e(m)$ notasyonu, m mesajını e açık anahtarıyla şifreleme işlemini; $Dec_d(c)$ ise c şifreli metninin d gizli anahtarı ile şifre çözme işlemini temsil etmektedir.

Ayrıca güvenliği Ayrık Logaritma Problemi'nin zorluğuna dayanan kriptografik sistemleri (Bölüm 4'te ele alınacak) incelerken karşılaşacağımız Döngüsel Grup ve Üreteç eleman kavramları aşağıdaki gibidir.

Tanım 2.2 (Döngüsel grup, Mertebe)**.** Bir G grubunda $\forall b \in G$ için $b = \alpha^i$, $i \in \mathbb{Z}$ olacak şekilde bir $\alpha \in G$ elemanı bulunabilirse bu gruba döngüsel bir grup denir ve α elemanına bu döngüsel grubun üreteci denir.

G bir grup ve $a \in G$ olsun. Bir a elemanı için $a^t = 1$ sağlayan en küçük t değeri varsa bu t değerine a elemanının mertebesi denir ve $ord(a) = t$ ile gösterilir. Eğer böyle bir t değeri bulunmazsa a elemanının mertebesi ∞'dur denir.

Tanım 2.3 (Denklik Sınıfları, Çarpımsal Grup)**.** $n \in \mathbb{Z}^+$ pozitif tam sayı, $a, b \in \mathbb{Z}$ tam sayılar olmak üzere, n sayısı, $(a-b)$ farkını bölüyorsa; a, b'ye mod$n$'de kongrüdür denir ve $a \equiv b \bmod n$ şeklinde gösterilir. Bir a tam sayısının mod$n$'de kongrü olan bütün tam sayıların oluşturduğu kümeye; a'nın mod$n$'de denklik (kalan) sınıfı denir ve $\overline{a} = \{x | x \in \mathbb{Z}, a \equiv x \bmod n\}$ şeklinde temsil edilir.

mod$n$'deki tam sayılar kümesi $\mathbb{Z}_n$ ile gösterilir ve n'den küçük $\{0, 1, \cdots, n-1\}$ tam sayılarının denklik sınıflarının oluşturduğu kümeye denir. $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n |$ EBOB$(a, n) = 1\}$ şeklinde tanımlanan kümeye, $\mathbb{Z}_n$'in çarpımsal grubu denir ve n'den küçük ve n ile aralarında asal olan tam sayıların denklik sınıflarının oluşturduğu grubu temsil eder. Özel olarak grubu belirleyen n sayısı asal ise $\mathbb{Z}_n^* = \{a | 1 \leq a \leq n-1\}$ olacaktır [28].

Tanım 2.4 (Kuadratik Rezidü)**.** Bir $a \in \mathbb{Z}_n^*$ elemanı için $x^2 \equiv a \bmod n$ olacak şekilde $x \in \mathbb{Z}_n^*$ bulunabiliyorsa; bu durumda a'ya mod n'de bir kuadratik rezidü denir. Eğer böyle bir $x \in \mathbb{Z}_n^*$ yoksa; bu durumda kuadratik nanrezidü denir. $a \in \mathbb{Z}_n^*$ bir kuadratik rezidü ise bu durumda $x^2 \equiv a \bmod n$ sağlayan $x \in \mathbb{Z}_n^*$ elemanına $a \bmod n$'in karekökü denir. n modunda kuadratik rezidü olan elemanların hepsinin olduğu kümeyi KR_n, kuadratik nanrezidülerin olduğu kümeyi ise $\overline{KR_n}$ ile temsil edilecektir.

2.2 Eliptik Eğri Temelleri

Tanım 2.5 (Eliptik Eğri)**.** Sonlu bir $\mathbb{F}_p$ cismindeki $E_{a,b}(\mathbb{F}_p)$ eliptik eğrisi, $a, b \in \mathbb{F}_p$, $p \neq 2, 3$ ve $4a^3 + 27b^2 \not\equiv 0 \bmod p$ olmak üzere

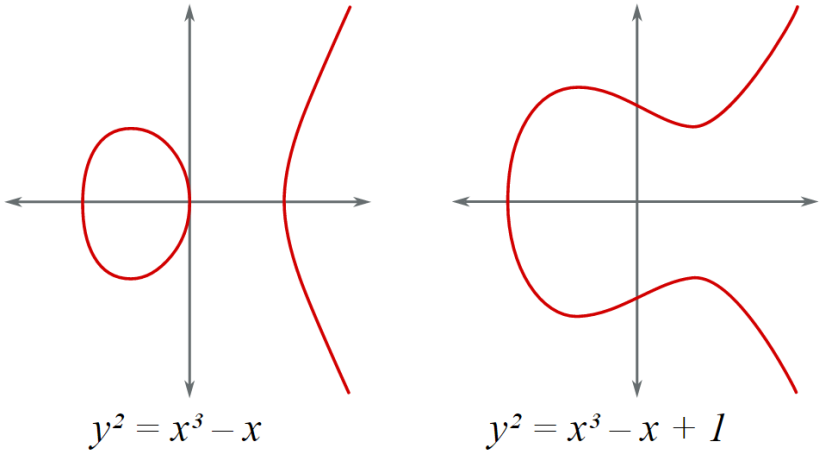
$$y^2 = x^3 + ax + b$$

eşitliğini sağlayan $P = (x, y)$ noktalarıyla sonsuzdaki O noktasının birleşiminin oluşturduğu kümedir.

Örnek olarak $E_{3,5}(F_7)$ eliptik eğrisini;

$$E_{3,5}(F_7) = \{(1,3),(1,4),(4,2),(4,5),(6,1),(6,6),O\}$$

noktalarından oluşur.



ŞEKİL 2.1: Eliptik Eğri Örnekleri
Kaynak: `https://en.wikipedia.org/wiki/Elliptic_curve`

Eliptik Eğri Grup İşlemleri :

Tanım 2.6 (Eliptik Eğri Nokta Toplama)**.** Bir eliptik eğri üzerindeki iki nokta için toplama işlemi teğetler ve kirişler kullanılarak tanımlanan bir grup operasyonu ile yapılır. Bu işleme göre bir eliptik eğri komutatif bir grup oluşturacaktır. Kullanacağımız eliptik eğriler için bu operasyon aşağıda tanımlanmıştır.

Bir $E_{a,b}(\mathbb{F}_p)$ eliptik eğrisi üzerindeki iki nokta $P_1 = (x_1, x_2)$ ve $P_2 = (x_2, y_2)$ olmak üzere $P_1 + P_2$ toplama işlemi aşağıdaki üç duruma göre şöyle tanımlanır.

1. $x_2 = x_1$ ve $y_2 = -y_1$ ise $P + Q = O$ dır.
2. Bu noktalardan herhangi biri O birim eleman ise
   $P + O = O + P = P$ dir.
3. Yukarda şartların ikisi de değilse;

$$\lambda = \begin{cases} (y_2 - y_1)/(x_2 - x_1) & P \neq Q \text{ ise} \\ (3x_1^2 + a)/2y_1 & P = Q \text{ ise} \end{cases}$$

ve

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

olma üzere; $P+Q=(x_3,y_3)$ ile belirlenmektedir. Burada bölme işlemin mod$p$'de çarpmaya göre ters ile çarpma olduğu ve diğer işlemlerin de mod p'de gerçekleşeceği gözden kaçırılmamalıdır.

Sonuc 2.1. Yukarıdaki toplama işlemiyle beraber $(E_{a,b}(\mathbb{F}_p),+)$ yapısı, komütatif bir grup oluşturur.

Tanım 2.7 (Sabitle Çarpma)**.** Eliptik eğrilerde sabitle çarpma işlemini ise şöyle tanımlayacağız, $a\in\mathbb{Z}$ ve $P\in E_{a,b}(\mathbb{F}_p)$ için;

$$aP=\underbrace{P+\cdots+P}_{\text{a tane}}$$

a tane P noktasının toplamı olarak tanımlanacaktır. Özel olarak $0\cdot P=O$ ve $1\cdot P=p$ dir.

Eliptik eğrinini nokta sayısı :

$E_{a,b}$ eliptik eğrisi üzerindeki noktaların sayısı $\#E_{a,b}$ sembolü ile temsil edilecektir. Bir $\mathbb{F}_p$ sonlu cismi üzerinde yaklaşık olarak p tane nokta vardır ve bu noktaların sayısı aşağıdaki $Hasse$ eşitsizliğini sağlar [30] .

$$p+1-2\sqrt{p}\leq\#E(\mathbb{F}_p)\leq p+1+2\sqrt{p}$$

Bu eşitsizlik kesin olarak vermese de, $E_{a,b}(\mathbb{F}_p)$ eğrisi üzerindeki nokta sayısını, "Schoof Algoritması" olarak bilinen bir algoritmada yukarıdaki eşitsizliği kullanarak polinom zamanda hesaplayabilmektedir [31]. Bu algoritma pratikte kullanılan p değerleri için verimli olmasa da, bu algoritmanın geliştirilmiş hali olan "Schoof-Elkies-Atkin" algoritması, kullanacağımız p değerleri için dakikaalar içinde hsaplama yapabilecek bir algoritmadır. [30] (syf:179)

Eliptik Eğrinin Grup Yapısı ve Üreteç Eleman :

İlerleyen bölümlerde sunulacak kleptografik ataklarda, sisteme iki eliptik eğri ve bu eğrilerin eleman sayıları, grup yapısı ve üreteç elemanları yerleştirilecektir. Bu eğriler ve elemanlar sayesinde istediğimiz sözde rastgeleliği sağlayabiliyor olacağız. Şimdi bir eğrinin üreteç elemanlarını incelemeden önce, üreteç elemanları bulmak için ihtiyaç duyacağımız grup yapısı kavramını ele alalım.

Teorem 2.2 (Grup Yapısı)**.** Her $E_{a,b}(\mathbb{F}_p)$ eliptik eğrisi için; aşağıdaki izomorfizmayı sağlayan, $n_2|n_1$ olacak şekilde n_1,n_2 tam sayıları vardır.

$$E(\mathbb{F}_p)\cong(\mathbb{Z}/n_1\mathbb{Z})\times(\mathbb{Z}/n_2\mathbb{Z})$$

(n_1, n_2) çiftine eliptik eğrinin *grup yapısı* denir [32] .

Teorem 2.3 (Üreteç Topluluğu)**.** A bir komutatif grup ve $(n_1, \ldots, n_r)$ kümesi, A grubunun grup yapısı olsun. Bu durumda $\forall X \in A$ elemanı, $0 \leq a_i < n_i$ olmak üzere $X = a_1G_1 + \cdots + a_rG_r$ şeklinde yazılabilir. Bu $(G_1, \ldots, G_r)$ elemanlarına A kümesinin üreteç topluluğu denir [32] .

Eliptik eğrilerde grup yapısı iki elemanla belirlediğinden, üreteç topluluğunu özel olarak iki elemanlı (G_1, G_2) üreteç çifti ile temsil edilebilmektedir. Bir eliptik eğrinin üreteç çiftleri birden fazladır. Bu yüzden üreteç çiftleri bulmak için, elemanlar seçilip üreteç olup olmadıkları kontrol edilerek üreteç çiftler belirlenecektir.

Grup yapısı ve üreteç çiftleri bulmak için detaylı incelemeler Kaliski'nin [32] çalışmasında bulunabilir.

Burgu Eliptik Eğriler :

Bu bölümde Kaliski'nin [32] çalışmasında sunduğu ve son iki atakta kullanacağımız sözde rassal bit üreteci (Pseudo Random Bit Generator PRBG) göreceğiz. Bu üreteci görmeden önce Kaliski'nin burgu eğriler için yaptığı bazı tanımlar ve teoremleri incelemek faydalı olacaktır. Kaliski çalışmasında burgu olma durumunu, "*aynı sonlu cisimde tanımlı iki eliptik eğrinini, tanımlı oldukları sonlu cismin bir cebirsel genişlemesinde birbirine izomorf olabilme durumu*" olarak tanımlandığını belirtmiştir. İki eğri grup yapıları aynı olması durumunda birbirine izomorfiktir ve bu durumda iki eğrinin her elemanını birbirine eşleyen bir dönüşüm tanımlanabilir.

Lemma.6.4: $E_{a,b}(\mathbb{F}_p)$ ve $E_{a',b'}(\mathbb{F}_p)$ iki eliptik eğrinin $(\mathbb{F}_p)$ cismi üzerinde birbirine izomorf olabilmesi için gerek ve yeter şart; $a = a\alpha^4$ ve $b = b\alpha^6$ olacak şekilde bir $\alpha \in \mathbb{F}_p$ olmasıdır.

Burada α değerinin, sonlu cisim içerisinde olması yerine bu cismin bir genişlemesinde olmasını sağlayarak, sonlu cisimde izomorf olmayan iki eğrinin, bu genişlemede izomorf olması sağlanabilmektedir [33]. Kaliski'nin [32] çalışmasındaki Lemma. 6.5 burada devreye giriyor:

Lemma 1. $\beta \neq 0$ sayısı, $\mathbb{F}_p$ cisminde bir kuadratik nanrezidü ve $E_{a,b}(\mathbb{F}_p)$ bir eliptik eğri olsun. Bu durumda $y = \sqrt{x^3 + ax + b}$ olmak üzere $\forall x \in \mathbb{F}_p$ için aşağıdakiler gerçeklenir;

1. y kuadratik rezidü ise $(x, \pm y)$ noktaları $E_{a,b}(\mathbb{F}_p)$ eğrisi üzerindedir.
2. y kuadratik nanrezidü ise $(\beta x, \pm\sqrt{\beta^3}y)$ noktaları $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ eğrisi üzerindedir.
3. $y = 0$ ise $(x, 0)$ noktası $E_{a,b}(\mathbb{F}_p)$ eğrisi üzerinde ve $(\beta x, 0)$ noktası $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ eğrisi üzerindedir.

Sonuç: Bu Lemma'nın bir sonucu; her x değeri için iki nokta ve iki tane birim eleman olmak üzere, iki eğri üzerinde toplam $2p+2$ tane eleman vardır.

Tanım. *(Twist-Burgu Çifti):* $E_{a,b}(\mathbb{F}_p)$, k parametresinde bir eliptik eğri olsun ve β, mod b'de bir kuadratik nanrezidü olsun. $T_{a,b,\beta}(\mathbb{F}_p)$ ile gösterilecek olan k parametresindeki bir burkulmuş bir çifti; $E_{a,b}(\mathbb{F}_p)$ ve $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ eğrilerinin birleşimi anlamına gelmektedir.

2.3 Zor Problemler

Bu bölümde asimetrik şifreleme sistemlerinin güvenliklerinin dayandığı matematiksel problemleri ele alacağız.

Tanım 2.8 (Çarpanlara Ayırma Problemi ÇAP)**.** Verilen n tam sayısı için, $p_i \in \mathbb{Z}$ asal sayılar ve $e_i \geq 1$ olmak üzere, $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ sağlayan, n tam sayısının çarpanlara ayrılışının bulunabilmesi problemine *Çarpanlara Ayırma Problemi* adı verilir.

Bu çalışmanın büyük bölümünde incelenecek olan RSA şifreleme sisteminin güvenliği bu problemin hesaplama zorluğuna dayanmaktadır

Tanım 2.9 (Ayrık Logaritma Problemi ALP)**.** G, derecesi n olan sonlu döngüsel bir grup olsun. $g \in G$ üreteç eleman ve $a \in G$ herhangi eleman olmak üzere, $0 \leq x \leq n-1$ ve $a = g^x$ sağlayan x değerine, a'nın g tabanındaki ayrık logaritması denir. G grubu, g üreteç elemanı ve $a \in G$ verilmişken, a'nın ayrık logaritmasının bulunabilmesi problemine ise *Ayrık Logaritma Problemi* adı verilir.

Asimetrik kripto sistemlerinin birçoğunun güvenliği temelde bu problemin hesaplama zorluğuna dayanmaktadır. Bunlar Diffie - Hellman anahtar değişimi, ElGamal şifreleme, DSA dijital imza, eliptik eğri kriptografisi olarak sıralanabilir.

Tanım 2.10 (Diffie - Hellman Problemi DHP)**.** [8] Verilen G sonlu, döngüsel grubu, $g \in G$ üreteç elemanı ve g^a, g^b grup elemanlarından, g^{ab} elemanının bulunabilmesi problemine *Diffie Hellman Problemi* denir.

Diffie Hellman anahtar değişimi ALP'nin bir uygulanışı olan bu problemin hesaplanması zorluğuna dayanmaktadır.

Tanım 2.11 (Eliptik Eğri Ayrık Logaritma Problemi EEALP)**.** [30]

Bir E eliptik eğrisi üzerinde n mertebeli bir P noktası ve Q noktası verilmiş olsun. $0 \leq m \leq n-1$ olmak üzere $P = mQ$ sağlayan bir m tam sayısının bulunması problemine EEALP denir.

Tanım 2.12 (Eliptik Eğri Diffie Hellman Problemi EEDHP)**.** [34] Bir $E(\mathbb{F}_p)$ eliptik eğrisi üzerinde n mertebeli bir P noktası ve $0 \leq k, \ell \leq n-1$ olmak üzere $k \cdot P$ ve $\ell \cdot P$ noktaları

verilmiş olsun. Bu şartlar altında $(k \cdot \ell \cdot P)$ sağlayan noktanın bulunabilmesi problemine EEDHP denir.

Şimdi bu probleme dayanılarak Diffie-Hellman anahtar değişiminin nasıl yapılabileceğine bakalım.

Tanım 2.13 (Eliptik Eğri Diffie Hellman Anahtar Değişimi)**.** [8, 34]

Ayşe ve Bora güvensiz bir kanalda haberleşmek için güvenli bir anahtarda uzlaşmak istemektedirler. Bunun için öncelikle bir $\mathbb{F}_p$ sonlu cisminde tanımlı $E_{a,b}(\mathbb{F}_p)$ eliptik eğrisi ve bu eğri ailesinin bir P üreteç elemanını seçerler. $E, \mathbb{F}_p$ ve P açık olarak yayımlanır.

Ayşe rastgele bir a tam sayısını seçer ve $A = a \cdot P$ değerinin Bora'ya gönderir. a değeri Ayşe'nin gizli anahtarı olarak saklanacaktır. Aynı şekilde Bora da rastgele bir b değeri seçer ve $B = b \cdot P$ değerini Ayşe'ye gönderir. Bundan sonra iki taraf da

$$a \cdot B = b \cdot A = a \cdot b \cdot P$$

hesaplayarak ortak anahtar üzerinde uzlaşırlar.

Bölüm 3

RSA Kriptosistemine Kleptografik Ataklar

RSA asimetrik şifreleme sistemi, 1978 yılında Ron Rivest, Adi Shamir ve Len Adleman tarafından geliştirilmiştir [3] . Bu bölümde RSA şifreleme sistemi için kurgulanmış kleptografik ataklar ele alınacaktır. İlk bölümde RSA şifreleme sistemi ve bu sistemin, kleptografik bir atak barındırmadığı standart durum analiz edilecek, ikinci bölümde ise Young ve Yung ikilisinin 1996 yılında ilk olarak kleptografi fikrini öne sürdükleri çalışmalarında [1] kurguladıkları *" RSA İçin İlk Kleptografik Atak "* ve bu atağın analizi incelenecektir. Devamında ise 1997 yılındaki [7] çalışmalarında sundukları atak incelenecektir. Son olarak, Coppersmith [10] kısmi bilgi atağını kullanıldığı ilk kleptografik atak olan, Crepeau ve Slakmon'un [9] çalışmalarında sundukları *" Gizli Asal Çarpan"* atağı ele alınacaktır.

3.1 RSA Şifreleme Sistemi

Kleptografi alanında, akademik çalışmalara baktığımızda, en fazla atağın kurgulandığı sistem olarak RSA şifreleme sistemini görebiliriz. Bunun sebebi RSA şifreleme sisteminin yaygın olarak kullanılmasıdır. RSA şifreleme sisteminin güvenliği *Çarpanlara Ayırma Problemine* dayanmaktadır (bkz: Bölüm 2.3) ve anahtar üretim algoritması aşağıdaki gibi olacaktır.

RSAAnahtarÜretimi(k): [3]
Girdi: k güvenlik parametresi (üretilecek asalların bit uzunluğu)
Çıktı: $(n, e) : n \in \{\{0,1\}^{2k-1}, \{0,1\}^{2k}\}, 1 < e < \varphi(n)$ açık anahtarlar
$d \equiv e^{-1} \bmod \varphi(n)$ gizli anahtar
($\varphi(n)$:Euler Phi fonksiyonu (Bkz: Bölüm 2.1)

1. $p, q \in_R \{0,1\}^k$ asal sayıları seç.
2. $\varphi(n) = (p-1)(q-1)$ hesapla.
3. $1 < e < \varphi(n)$ ve $EBOB(e, \varphi(n)) = 1$ (aralarında asal) olacak şekilde e sayısı rastgele seç.
(e açık anahtarının sabit bir sayı olması istendiğinde bu adım geçilir)
4. $d = e^{-1} \bmod \varphi(n)$ hesapla
(e açık anahtarının sabitlenmesi istendiğinde q asalı bu adımda uygun d bulana kadar rassal bir algoritma ile üretilir)
5. (n, e) açık anahtar, d gizli anahtar çıkart

ŞEKİL 3.1: RSA Şifreleme Anahtar Üretim Algoritması

Yukarıdaki algoritmayla üretilen anahtarlardan (n, e) açık anahtarları yayımlanır ve gizli olan d anahtarı saklanır. Açık anahtarlarla bir m mesajını ($m < n$) şifrelemek isteyen bir kullanıcı

$$c \equiv m^e \bmod n$$

hesaplayarak c şifreli metnini elde eder ve anahtar sahibine gönderir. Mesajın ulaşması gereken kullanıcı, d gizli anahtarıyla

$$m = c^d \bmod n$$

hesaplayarak açık metine ulaşabilir.

Bir şifreleme sisteminde kleptografik bir atak bulunup bulunmadığını anlamak için (sistemin kapalı kutu olduğunu varsayıyoruz) analiz edebileceğimiz iki parametre, sistemin verdiği çıktıların dağılımı ve çalışma zamanıdır.

3.1.1 Algoritmanın Simülasyonu ve Analizi

Kapalı kutu bir şifreleme sisteminde, kleptografik bir atak olup olmadığını test etmek için bakılabilecek ilk kriter sistemin ürettiği çıktıların, standart algoritmaların ürettiği çıktıların olasılık dağılımlarının uyum halinde olup olmadığıdır. Diğer kriter ise tersine

mühendislikle sistemde çalışan kodun görülmesi ancak bu işlem çok daha zahmetli ve bazı tekniklerle zorlaştırılabilmektedir.

Bu çalışmada incelenecek ataklarda, atak algoritmaları verildikten sonra ataklar simüle edilecek ve ürettiği çıktılar standart algoritmaların üreteceği çıktılar ile karşılaştırılacaktır. RSA şifreleme için atakları incelerken karşılaştırma kriteri olarak kullanmak üzere standart algoritma ile, ataklarda yapacağımız gibi 150 adet anahtar üretip bu anahtarların olasılık dağılımlarını ve anahtar üretimi için gerekli çalışma zamanlarını bu bölümde inceleyeceğiz.

Algoritmaların gerçeklendiği bilgisayarın fiziksel durumu, işletim sistemi ve gerçeklendiği platform gibi detaylar Bölüm 1.4.1'da yer almaktadır.

3.1.1.1 Çalışma Zamanı

İlerleyen bölümler de ele alınacak atakların, simülasyonları sonucu oluşacak çıktı değerleri ile karşılaştırmak için üretilen 150 adet standart, arka kapı barındırmayan anahtarın çalışma zamanı ortalamaları Tablo 3.1'da verilmiştir. Tablodaki değerler, 3 ayrı anahtar boyu; 256, 512, 1024 için (sırasıyla asal uzunlukları 128, 256, 512 ile temsil edilecek) -anahtar boyları güvenlik için yetersiz olsa da analiz için yeterli olacaktır- seçtiğimiz örneklerin çalışma zamanlarını göstermektedir.

TABLO 3.1: Zaman Tablosu:
150 adet anahtarın üretim zamanı ortalamaları (sn.)

k:asal bit uzunluğu	RSA
128	0,07
256	1,01
512	27,97

3.1.1.2 Açık anahtar dağılımı

Bu bölümde üretilen anahtarları analiz edebilmek için muhtemel tüm anahtarlar kümesini büyüklüklerine göre 3 parçaya böldük ve üretilen anahtarların bu 3 ayrı kümede olma istatistiklerini belirledik. Bu işlemi bitsel gösterimindeki şu analizle yapacağız.

Analiz için üretilen anahtarlarda, k-bit uzunluğunda p, q asalları ile $n = pq$ olarak üretilen açık anahtar mod değerleri, ya $2k-1$ bit uzunluğunda ya da $2k$ bit uzunluğunda olacaktır. Analiz için sadece bit uzunluğunda göre ayırmak yerine ikinci bölgeyi yani $2k$-bit uzunluğunda olanların kümesini de iki parçaya böldük ve bitsel gösterimi "11" ile başlayan $2k$-bit uzunluğundakiler ve "10" ile başlayan $2k$ bit uzunluğundakiler olarak

tasnif ettik. Bu sayede çıkarılacak anahtarları büyüklüklerine göre 3 grupta kategorize edebileceğiz.

Üretilen 150 adet dürüst RSA anahtarı için, n değerinin olasılık dağılım yüzdeleri Tablo 3.2'de verilmiştir.

TABLO 3.2: n ahatarı dağılımı tablosu:
n açık anahtar modul dağılım yüzdeleri

	RSA		
k asal bit uzunluğu	$2k-1$ bit	$2k$ bit ("10" ile)	$2k$ bit ("11" ile)
128	38	56	6
256	36	51	12
512	37	48	14

3.2 RSA İçin İlk Kleptografik Atak

Bu bölümde, Young ve Yung ikilisinin 1996 yılında [1] çalışmalarında sundukları, RSA için kleptografik atağı ele alacağız. Bu atakta saldırgan, kullanıcının şifreleme sistemine kendi (N, E) RSA açık anahtarlarını yerleştirmiş ve bunlarla kullanıcının üretilecek gizli asalı p'yi şifreleyerek, n açık anahtar değeri içerisinde yayınlanmasını sağlayacaktır.

Atakta kullanılacak sabitler ve fonksiyonların detayları aşağıdaki gibidir:

Saldırganın anahtarları : Atak algoritmasında (E, N) saldırganın RSA açık anahtarını ve D değeri gizli anahtarını temsil etmektedir. Açık anahtarlar, atak algoritması içinde yer alacak; ancak gizli anahtar sadece saldırganda bulunacaktır.

Rastgeleleştirme Fonksiyonları : Algoritmada F ve G ile temsil edilecek iki fonksiyon kullanılacaktır. Bu fonksiyonlar, manipüle edilecek olan değerlerin rastgeleliğini ve çıktıların istenilen aralığa düşmesini garantileyerek, gizli p asalını çalabilmeye uygun hale getirmeye yarayacak fonksiyonlardır. Bunlar simetrik şifreleme algoritmaları olabilir ancak tersi alınabilir olmaları gerektiğinden (anahtar geri kazanırken tersleri kullanılacaktır) kriptografik özet fonksiyonlar olamayacaklardır.

B_i sınırları : Algoritmada kullanılacak olan B_1 değeri üretilecek p asalının saldırganın N modundan küçük olmasını sağlamakta iken; B_2 değeri ise q asalının, asal olmasını sağlamaya çalışırken kullanılacak döngünün kurulabilmesinde işimize yarayacaktır. Young ve Yung bu çalışmalarında atağın implementasyonunu da yapmışlar ve B_1 ve B_2 değerlerini sırasıyla 16 ve 512 olarak seçmişlerdir.

F ve G rastgeleleştirme fonksiyonu anahtarları : i ve j değerleri, saldırgan tarafından seçilen sabit bir K değeri ile birlikte, rastgeleleştirme fonksiyonlarının anahtarlarını belirlemektedirler. Bu değerler B_i değerleri ile sınırlandırıldığından saldırgan gizli anahtarı elde etmek için i ve j değerlerini kullanmak istediğinde hangi değerlerin atak esnasında kullanıldığını bilmese de tahmin edebilecek yani muhtemel bütün i ve j değerlerini deneyerek istediğini elde edebilecektir. Algoritmada || notasyonu ile bitsel dizileri uçuca ekleyerek birleştirme işini yapacak olan birleştirme (concatenation) operatörü temsil edilmektedir.

KleptoAnahtarÜreteç(k) : [1]
Girdi: k: gizli asalların bit uzunluğu.
Çıktı: $(n, e) \rightarrow n \in \{\{0,1\}^{2k-1}, \{0,1\}^{2k}\}, e < \varphi(n)$ açık anahtarlar
$d \equiv e^{-1} \bmod \varphi(n)$ gizli anahtar.
Rastgeleleştirme fonksiyonları: F, G rastgeleleştirme için kullanılacak simetrik şifreleme fonksiyonları (ör: DES, XTEA, AES).
Gömülü Değerler:
Saldırgan RSA anahtarları: $N \in \{0,1\}^k$ ve $E < \varphi(N)$ açık anahtarları.
K anahtarı: F ve G fonksiyonlarında kullanılacak anahtar değeri.

1. $p \in_R \{0,1\}^k$ asalı seç
2. $i = 0$'dan B_1'e kadar;
   $p' \leftarrow F_{K+i}(p)$ hesapla
   $p' < N$ ise bırak değilse i'yi 1 arttır.
3. $p'' := (p')^E \bmod N$
4. $j = 0$'dan B_2'e kadar;
   $p''' \leftarrow G_{K+j}(p'')$ hesapla
   $rand \leftarrow k - bit$ uzunluğunda rastgele bit dizisi
   $X \leftarrow (p''' \parallel rand)$
   $q := X/p$
   q asal ise sonraki adıma geç, değilse j değerini 1 arttır
   Adım 1'e dön
5. $n \leftarrow p.q$; $\varphi(n) \leftarrow (p-1)(q-1)$; $e = 17$
6. $(e, \varphi(n)) = 1$ ise $d = e^{-1} \bmod \varphi(n)$ hesapla değilse e'yi 2 artır
7. (n, e, d) çıkart

ŞEKİL 3.2: Kleptografik RSA Anahtar Üretim Algoritması

Bu algoritmada; Adım 3'te sızdırılmak istenen p asalı saldırganın açık anahtarıyla şifrelenmektedir. Adım 4'deki döngüde ise p asalının rastgeleleştirilmiş ve şifrelenmiş hali rastgele bir bit dizisine ekleme operasyonuyla, kullanıcının açık anahtarı n'e dönüştürülmektedir. Bu X değeri yine Adım 4'de p asalına bölünerek q değeri elde edilir.

Adım 6'da kullanıcının açık kuvveti olan e değeri başlangıçta 17 olarak belirlenir ve daha sonra açık anahtar modülü n değerinin Euler $\varphi(n)$ fonksiyonuyla aralarında asal oluncaya kadar 2 arttırılır.

Kullanıcının açık anahtar olarak yayınladığı n değerini ele geçiren ve D gizli anahtar değerine sahip saldırgan aşağıdaki saldırı algoritmasını kullanarak $n = pq$ asallarına ayırabilecek ve buradan gizli anahtar olan d değerine ulaşabilecektir.

AnahtarEleGeçir(n): [1]
Girdi: $n \in \{\{0,1\}^{2k-1}, \{0,1\}^{2k}\}$ açık anahtar
Çıktı: $p \in \{0,1\}^k$ asalı
Operatörler: $|n|$: n sayısının bitsel uzunluğu
$n\rceil^t$: n sayısının en üst (sol) t biti

1. $U := n\rceil^{|n|-k}$.
2. $L_1 := \{p'' \leftarrow G_{K+j}^{-1}(U) : K = \text{ sabit }, j = 0, \cdots, B_2 - 1\}$
3. $L_2 := \{p' \leftarrow (p'')^D \bmod N : p'' \in L_1\}$
4. $L := \{p \leftarrow F_{K+i}^{-1}(p') : K = \text{ sabit }, i = 0, \cdots, B_1 - 1\}$
5. $c \in L$ için;
   $c|n$ ise $p = c$ çıkart
   böyle eleman bulunamazsa diğer adıma geç
6. $U = U + 1$ yap ve adım 2'ye dön

ŞEKİL 3.3: YY96 Gizli Anahtar Ele Geçirme Algoritması

AnahtarEleGecir algoritmasında, kullanıcının açık olarak yayınlanan n modül değerinden gizli asal değerlere ulaşmaya çalışılacaktır. Bunun için öncelikle n değerinin bitsel gösteriminin en düşük (sağdan) k tane biti atılacak ve kalan bitlerin tam sayı değeri U değişkenine atanacaktır. Daha sonra bu U değeri G fonksiyonunun tersi ile muhtemel bütün $K+j$ anahtarları kullanılarak p'' değerleri elde edilir. Bulunan bu değerler D kuvvetiyle şifre çözme yapıldıktan sonra F fonksiyonunun tersi ile muhtemel bütün $K + i$ anahtarları kullanılarak, muhtemel bütün p değerlerine ulaşılır. Bulunan bu değerler arasında n modülünü bölebilen bir değer varsa, gizli p asalı bulunmuş olur. Eğer böyle bir p değeri bulunamazsa KleptoAnahtarÜreteç algoritmasında X'in p asalına bölünmesi esnasında ödünç bit alınmış olabileceğinden $U \leftarrow U + 1$ olarak atanır ve 2. Adım'a dönülür

Algoritma düzgün çalışacaktır çünkü yayınlanan n açık anahtarı;

$$X = G(F(p)^E \bmod N)||rand$$

ve q asalı $q = X/p$ olmak üzere;

$$n = pq$$

$$n = \frac{G(F(p)^E \bmod N)||rand}{p} \cdot p$$

$$n = G(F(p)^E \bmod N)||rand$$

olarak belirlenecektir. $U = n\rceil^{|n|-k}$ olmak üzere;

$$G^{-1}(U) = G^{-1}(G(F(p)^E \bmod N)) = F(p)^E \bmod N$$

$$(G^{-1}(U))^D \bmod N = (F(p)^E)^D \bmod N = F(p)$$

$$F^{-1}((G^{-1}(U))^D \bmod N) = F^{-1}(F(p)) = p$$

Sonuç olarak;

$$p = F^{-1}((G^{-1}(n\rceil^{|n|-k}))^D \bmod N)$$

olacak şekilde anahtar üretim algoritmasıyla üretilen n değerinden, anahtar ele geçirme algoritmasıyla p değerine ulaşılabilecektir.

3.2.1 Atağın Simülasyonu ve Analizi

Bu bölümde atak algoritmalarının gerçeklenmesi ve üretilen çıktıların standart algoritmalar ile karşılaştırmalı analizleri yapılacaktır. Bunun için; klepto anahtar üreteç ve anahtar ele geçirme algoritmalarını gerçeklenmiş ve bir önceki bölümde standart (atak barındırmayan) RSA anahtar üretiminde olduğu gibi 150 adet anahtar üretip bu anahtarların gizli değerlerini ele geçirerek atak simüle edilmiştir. Bunun sonucunda üretilen anahtarlar için rastgele bir mesajı şifreleme/şifre çözme başarıları ve açık anahtardan, gizli anahtarı ele geçirebilme başarı oranları beklendildiği üzere %100 olarak gözlemlenmiştir. Algoritmaların gerçeklendiği bilgisayarın fiziksel durumu, işletim sistemi ve gerçeklendiği platform gibi detaylar Bölüm 1.4.1'de yer almaktadır.

3.2.1.1 Çalışma Zamanı

Simüle edilen atak için üretilen 150 adet anahtar değerinin ve bir önceki bölümde analiz edilen standart RSA anahtar üretiminin karşılaştırmalı anahtar üretim zamanları, Tablo 3.3'de verilmiştir. Bu tabloda 512-bit anahtarlar için gereken ortalama zamanın dürüst anahtar üretiminden fazla çıkması ile başarı tablosundaki şifreleme/şifre çözme arasında

zıt bir orantı oluşturulabilir. Yani anahtar üretim algoritmasında B_i sınırları azaltılarak, çalışma zamanı düzeltilebilir ancak bu işlem, üretilen anahtarlardan ele geçirebilme başarısının düşmesine sebep olabilir.

TABLO 3.3: Çalışma Zamanı:
150 adet anahtarın üretim zamanı ortalamaları (sn.)

asal.bit.uzunluğu:k	RSA	YY96
128	0,07	0,11
256	1,01	1,17
512	27,97	42,15

3.2.1.2 Açık Anahtarın Dağılımı

RSA anahtar üretiminde k-bit asallar seçerek üretilen açık anahtarın n modül değeri, ya $2k-1$ bit, ya da $2k$ bit uzunluğunda olacaktır. '$2k$-bit uzunluktaki değerleri, tekrar gruplandırmak istersek, bitsel gösterimi "10" ile başlayan $2k$ bit uzunluktaki ve "11" ile başlayan $2k$ bit uzunluktakiler olarak ayırabiliriz. Böylece üretilen anahtarları büyüklüklerine göre 3 ana grupta sınıflandırmış oluyoruz. Bu durum analiz için yeterli olacaktır. Ancak daha kesin analizler için bir adım daha ileri götürülerek 3. bitlerine göre de bir analiz yapılabilir.

Atak barındıran anahtar üretim algoritması ve dürüst RSA anahtar üretim algoritması ile üretilen anahtarların, yukarıda bahsettiğimiz kriterlere göre sınıflandırması Tablo 3.4'de bulunabilir. Tablodan da görülebileceği gibi KleptoAnahtarÜreteç algoritmasının çıktıları, dürüst anahtar üretimi algoritmasına göre ayırt edilemezliği sağlamamaktadır. Sonuç olarak atak barındıran algoritmaya sahip sistemin çıktıları, teste tabi tutulduğunda sistemin atak barındırdığı ortaya çıkacaktır.

TABLO 3.4: Açık Anahtar Dağılım Tablosu:
n açık anahtar modul dağılım tablosu

	RSA			YY96		
k asal bit uzunluğu	$2k-1$ bit	$2k$ bit "10" ile	$2k$ bit "11" ile	$2k-1$ bit	$2k$ bit "10" ile	2k bit "11" ile
128	38	56	6	31	30	38
256	36	51	12	32	32	34
512	37	48	14	34	33	32

3.3 RSA için Güçlü Kleptografik Atak

Bölüm 3.2'de ele aldığımız atağın yayınlanmasından bir sene sonra Young ve Yung [7] eserlerinde bu atağın neden geçersiz olduğunu ve daha güçlü hale getirmek için atakta nasıl değişiklikler yapılması gerektiğini açıklamışlardır.

Önceki bölümde sunduğumuz atağı ele alacak olursak. Kullanıcı bir şekilde şifreleme sisteminde arka kapı olduğundan şüpheleniyor olsun ve kontrol etmek istesin. Başka cihazlardan tersine mühendislik gibi tekniklerle elde edilen atak algoritması ve saldırganın açık anahtarları ile kullanıcı kendi sistemini test edebilecektir. Bunun için öncelikle (n, e) açık ve d gizli anahtarlarından $n = pq$ çarpanlarına ulaşacaktır. Verilen açık ve gizli anahtarlardan, p ve q çarpanlarına nasıl ulaşılabileceği Dan Boneh'nin [35] eserinde bulunabilir.

p ve q asal çarpanlarına ulaşan bir kullanıcı, p asalını aynen kleptografik sistemin üreteceği gibi kullanarak n ve q değerlerini üretecektir. Ürettiği değeri kendi n modunun en üst k bitiyle karşılaştırarak, sistemde arka kapı olup olmadığını tespit edebilecektir.

Young ve Yung aşağıdaki bölümde verilecek atak mekanizmasında bu sorunu gidermişlerdir, bununla beraber seçilen asalın saldırganın N açık mod değerinden küçük olması gerekliliği sonucu, asalların belli bir bölgeye sıkışabilecek olması sorununu gidermek adına ikili çalışmalarında PBRM (Olasılıksal Eğilim Kaldırma Yöntemi) adını verdikleri bir metod önermişlerdir. Ataktan önce bu metodu görelim.

3.3.1 Olasılıksal Eğilim Kaldırma Yöntemi (Probabilistic Bias Removal Method PBRM)

Bir önceki bölümde ele aldığımız atak için aşağıda anlatacağımız gibi, değerlerde bir yönlenme sorunu oluşmaktadır. Young ve Yung [7] çalışmalarında bu sorunu şu şekilde açıklamışlardır. İlk adımda seçtiğimiz p asalı, saldırganın N açık anahtarından küçük olması gerekmekteydi. Ancak bu durumda saldırgan ya N değerini olabildiğince büyük seçecek, ya da seçilen p asalları belli bir bölgede (istenilen aralıkta küçük değerler alacak şekilde) birikecektir. Dolayısıyla sonuçta oluşan n açık anahtar değeri de olması gereken dağılımı sergileyemeyecektir.

Herhangi bir aralıkta verilmiş rastgele bir değer için, bu değeri girdi olarak alıp, daha geniş bir kümeye aynı dağılımı koruyarak transfer etmek istediğimizi düşünelim. Örneğin; $[1, R]$ aralığında düzgün dağılıma sahip verilmiş x değeri için, x' değerini $2R > S$ olmak üzere $[1, S]$ aralığında düzgün dağılıma sahip olacak şekilde elde etmek istiyoruz. Bu işlem, aşağıda tanımlanan PBRM fonksiyonu kullanılarak gerçekleştirilebilir.

PBRM(R,S,x): [7]
Girdi: $S, R \in \mathbb{Z}, R \in (S/2, S), x \in \{0, \ldots, R-1\}$
Çıktı: $x' \in \{0, \ldots, S-1\}$

1. $b \in_R \{0,1\}$ seç
2. $x < S - R$ ve $b = 1$ ise:
   $x' = x$
3. $x < S - R$ ve $b = 0$ ise :
   $x' = S - x$
4. $x \geq S - R$ ve $b = 1$ ise :
   $x' = x$
5. $x \geq S - R$ ve $b = 0$ ise :
   başa dön
6. x' çıkart

ŞEKİL 3.4: Olasılıksal Eğilim Kaldırma Algoritması

Klepto anahtar üretim algoritması p değerini rastgeleleştirdikten sonra PBRM fonksiyonundan geçireceği için, gizli anahtarı ele geçirmek için PBRM fonksiyonunun tersini kullanmamız gerekecektir. Bu fonksiyon aşağıdaki gibi basitçe hesaplayabiliriz.

$$\mathsf{PBRM}^{-1}(R, S, x') = \begin{cases} x = x' & x' < R \\ x_{1,2} = x', S - x' & x' \geq R \end{cases}$$

3.3.2 Klepto RSA Anahtar Üretimi

Atağın yer aldığı [7] çalışmada yazarlar, güvenliği Ayrık Logaritma Problemi'nin zorluğuna dayanan Diffie - Hellman anahtar değişimi için de, kleptografik bir atak sunmuşlar ve daha sonra bu ataktaki stratejiyi RSA için Kleptografik anahtar üretim algoritmasında kullanmışlardır. Bu atağın detaylarına bu çalışmada girmeyeceğiz ancak [7]'de detaylı açıklama bulunabilir.

Atak algoritması saldırganın, $P \in \{0,1\}^k$ asal ve $g \in \mathbb{Z}_P^*$ üreteç eleman olmak üzere, Elgamal (Y, g, P) açık anahtarı ve $Y = g^X \bmod P$ sağlayan X gizli anahtarı kullanmaktadır. ElGamal şifreleme sistemi ile ilgili detaylar Bölüm 4.2'de sunulacaktır. P değeri ile kurbanın üretilecek p asalının bitsel gösterimleri eşit uzunluktadır. Yani $|P| = |p| = k$ olacaktır.

$G_K(a)$ fonksiyonu, K anahtarı ile a değerini rastgeleleştirmek için kullanacağımız bir G fonksiyonunu temsil edecektir. Bu fonksiyon simetrik şifreleme algoritmaları olabilir. Ancak; daha önce de belirttiğimiz gibi, tersi alınabilir olması gerektiğinden özet fonksiyonu kullanılamaz.

KleptoAnahtarÜreteç(k): [7]
Girdi: k, gizli asalların bit uzunluğu
Çıktı: $(n, e) \rightarrow n \in \{\{0,1\}^{2k-1}, \{0,1\}^{2k}\}, e < \varphi(n)$ açık anahtarlar
$d \equiv e^{-1} \bmod \varphi(n)$: gizli anahtar
Rastgeleleştirme ve Özet fonksiyonları:
G : simetrik şifreleme fonksiyonları.(örn.: AES)
H : Kriptografik özet fonksiyonu
Gömülü Değerler:
Saldırgan ElGamal anahtarları: (Y, g, P) açık anahtar, X gizli anahtar
$P \in \{0,1\}^k$ asal, $g \in \mathbb{Z}_P^*$ üreteç, $X \in_R \mathbb{Z}_P^*$, $Y = g^X \bmod P$
K anahtarı: G fonksiyonunda kullanılacak anahtar değeri.
$1 < W, a, b < P$ sabit tam sayılar.

1. $c_1 \in_R \{0, ..., N-1\}$ seç.
2. $z \leftarrow g^{c_1 - Wt} Y^{-ac_1 - b} \bmod P$
3. $z' \leftarrow PBRM(P, 2^k, z)$
4. $z'' = H(z')$
5. z' çift ise: $z' = z' + 1$
   z' tek ise: geç.
6. $i = 0$'dan B_1'e kadar:
   $p \leftarrow z'' + 2i$ (sadece tek olan değerleri deneniyor)
   p asal ise Adım 7'e geç.
   Değilse i'yi 1 arttır.
   Adım1'e dön.
7. $v \leftarrow \text{PBRM}(P, 2^k, g^{c_1} \ modN)$

8. $j = 0$'dan B_2'e kadar:

   $U \leftarrow G_{K+j}(v)$

   $RND \in_R \{0,1\}^k$ seç

   $n' \leftarrow (U \parallel RND)$

   $n' = pq + r$ den q'yu hesapla.

   q asalsa $n \leftarrow n' - r$ olarak belirle ve Adım 10'a geç.

   Değilse j'yi 1 arttır.

   Adım 1'e dön.

9. e ve d, RSA kuvvetlerini hesapla.

ŞEKİL 3.5: Kleptografik RSA Anahtar Üretim Algoritması

Adım 2'de aslında bir z mesajının ElGamal şifrelemeyle (r, s) şifreli metni elde edilmektedir. Daha sonra bu r ve s değerlerini birbirine eşitlenip z mesajı çekilmekte ve bu değerler atak algoritmasında kullanılmaktadır. Bu adımda yapılan atak Young ve Yung aynı çalışmada sundukları Ayrık Logaritma Atağı'nın temelini teşkil etmektedir. Burada ise Çarpanlara Ayırma Problemi'nin zorluğuna dayanan RSA şifreleme sisteminin gizli değerlerini çalmak için Ayrık Logaritma Problemi'nin zorluğunu kullanılmaktadır.

Kullanıcının açık anahtarlarını ele geçiren saldırgan, aşağıdaki algoritmayla gizli anahtar değerine ulaşabilecektir.

AnahtarEleGeçirme (n, e, k): [7]
Girdi: (n, e) kurbanın açık anahtarları, k : gizli asalların bit uzunluğu
Saldırgan ElGamal anahtarları : (Y, g, P) açık anahtar, X gizli anahtar
$P \in \{0,1\}^k$ saldırganın ElGamal asalı, $g \in \mathbb{Z}_P^*$ üreteç, $X \in_R \mathbb{Z}_P^*$, $Y = g^X \bmod P$
K anahtarı: G fonksiyonunda kullanılan anahtar değeri.
$W, a, b \in \{0, \cdots, P\}$: saldırganın belirlediği sabit değerler,
$S = 2^k$: PBRM için üst sınır
Çıktı: d gizli anahtar

1. $U \leftarrow n\rceil^{|n|-k}$

2. $L_1 = \{v = G_{K+i}^{-1}(U) : \ i = 0, ..., B_2; \}$

3. $L_2 = \{\mathsf{PBRM}^{-1}(P, S, v) : v \in L_1\}$ ($g^{c_1} \bmod N$ için aday değerler)

4. $L_3 = \{z = hg^{-Wt}h^{-aX}Y^{-b} \bmod P : h \in L_2;\ t = 0, 1\}$

5. $L_4 = \{z' : \forall z \in L_3$ için $z' = z$ ve $z' = S - z\}$

6. $L_5 = \{z'' : \forall z' \in L_4$ için $z'' = H(z')$ ve z'''nün en düşük biti 1 yapılır$\}$

7. $\forall z'' \in L_5$ için:
   $i \in \{0, ..., B_1\}$ için:
   $p' = z'' + 2i$
   $p'|n$ ise:
   $p = p'$ yap ve bırak
   değilse $i = i + 1$
   p bulunamazsa başa dön ve $U = U + 1$ yap

8. $q = n/p$, $\varphi(n) = (p - 1)(q - 1)$

9. $d \equiv e^{-1} \bmod \varphi(n)$ çıkart

ŞEKİL 3.6: Anahtar Ele Geçirme Algoritması [7]

Kullanıcının yayınladığı açık anahtarından gizli asalları elde etmek için yukarıdaki saldırı algoritmasını çalıştıran saldırgan, Adım 1'de p asalının bit sayısı olan k tane biti n modunun en düşük bitlerinden atmaktadır böylece rastgele eklenmiş bitleri çıkarır.

Adım 3'te PBRM metodunun tersini uygulayarak ($g^{c_1} \bmod P$) değerine ulaşır, sonraki adımda atak mekanizmasının şifrelediği bu değeri gizli anahtarını kullanarak ulaşır (burada şifreleyip/çözme işlemi yerine ortak z değerini bulabilme söz konusudur. Bu yüzden çalışmada ayrık logaritma atağı olarak nitelendirilmiştir). z değerini elde eden saldırgan bundan sonraki adımları aynen atak algoritmasının yaptığı gibi hesaplayarak p ve q değerlerine ulaşabilir.

3.3.3 Atağın Simülasyonu ve Analizi

Bölüm 3.2'de olduğu gibi bu atak için de, KleptoAnahtarÜreteç ve AnahtarEleGeçirme algoritmalarının gerçeklenmesi ve analizleri yapılmış ve sonuçlar bu bölümde ele alınacaktır. Bu bölümdeki atakla ilgili üretilen 150 adet anahtar için rastgele bir mesajı şifreleme/-şifre çözme başarıları ve açık anahtardan, gizli anahtarı ele geçirebilme başarı oranları

test edilmiş ve her iki durumun da %100 başarı ile sonuçlandığı belirlenmiştir. Algoritmaların gerçeklendiği bilgisayarın fiziksel durumu, işletim sistemi ve gerçeklendiği platform gibi detaylar Bölüm 1.4.1'da yer almaktadır.

3.3.3.1 Çalışma Zamanı

Simüle ettiğimiz atak için üretilen 150 adet anahtar değerinin ve Bölüm 3.1'de standart RSA anahtarları ile yapılan analizlerle karşılaştırmalı olarak, anahtar üretim zamanları, Tablo 3.5'de verilmiştir. Bu tabloda 512-bit anahtarlar için gereken çalışma zamanının yüksek çıkmasının sebebi, olması gerekenden (asallık testindeki işlemlerden) fazla olan moduler kuvvet alma işlemleri olarak görülebilir. Bu işlemler anahtar ele geçirirken de çalışma zamanlarını fazlaca artmasına (5-10 dk gibi) sebep olduğu görülmüştür.

Tablo 3.5: Çalışma Zamanı
150 adet anahtarın üretim zamanı ortalamaları (sn.)

asal.bit.uzunluğu:k	RSA	YY97
128	0,07	0,71
256	1,01	6,46
512	27,97	132,52

3.3.3.2 Açık Anahtarların Dağılımı

Atak barındıran algoritmayla ürettiğimiz anahtarlar için, n açık anahtar değerinin, büyüklüğüne göre olasılık dağılımı, önceki atakta olduğu gibi standart RSA anahtarları ile karşılaştırmalı olarak Tablo 3.6'de verilmiştir. Bu tablodan da görülebileceği gibi, standart RSA anahtarları ile karşılaştırıldığında, atak barındıran sistemin ürettiği anahtar değerleri olması gereken dağılıma sahip değildir. Yani sistem, üretilen anahtarlarının dağılıma göre teste tabi tutulduğunda, sistemin atak barındırdığı ortaya çıkacaktır.

Tablo 3.6: Açık Anahtar Dağılım Tablosu:
Üretilen 150 adet anahtarın büyüklüklerine göre dağılım tablosu

	RSA			YY97		
k asal bit uzunluğu	$2k-1$ bit	$2k$ bit "10" ile	$2k$ bit "11" ile	$2k-1$ bit	$2k$ bit "10" ile	$2k$ bit "11" ile
128	38	56	6	28	36	34
256	36	51	12	28	38	33
512	37	48	14	39	28	32

Sonuç olarak atak, Bölüm 3.2'deki atağa göre güvenliği arttırmış olsa da hala sistemin üreteceği çıktılara göre test edilmesi durumunda, standart algoritma ile ayrıt edilemezliği sağlayamamaktadır ve tespit edilebilir bir ataktır.

3.4 Gizli Asal Çarpan

Crepeau ve Slakmon [9] çalışmalarında, RSA şifreleme sistemi için toplamda 4 adet arka kapı kurgulamışlardır. Bu atakların 3'ünde p ve q asalları rastgele seçilmekte ve atak mekanizması açık ve gizli (e, d) kuvvet değerlerinin, bilinen bazı ataklara karşı zafiyet barındırabilecek şekilde üretilmektedir. Bu arka kapılardan sızdırılmak istenen değeri, sızdırmak için bir asimetrik şifreleme kullanılmamaktadır. Bu nedenle bu ataklar simetrik ataklar olarak nitelendirebiliriz ve kleptografi kapsamına girememektedir.

İkilinin çalışmalarındaki son ataklarında ise diğerlerinden farklı olarak kuvvet değerlerinin manipülasyonu ile değil ancak asallardan birinin bitsel gösteriminin üst yarısının (bitlerinin sol yarısı) sızdırılması şeklindedir. Çalışmadaki diğer 3 atak gibi bu atakta da sızdırılmak istenilen bilgiye asimetrik bir şifreleme uygulanmamakta ancak simetrik şifreleme olarak sınıflandırabileceğimiz bir şekilde permutasyona (rastgeleleştirme) tabi tutulmaktadır. Bu yüzden bu atağı da simetrik bir atak olarak nitelendiriyoruz. Atağın bu özelliğinden dolayı Kleptografi kapsamına girmese de *"Coppersmith Kısmi Bilgi Atağını"* kullanan ilk arka kapı çalışması olduğu için bu çalışmada yer almaktadır.

3.4.1 Coppersmith Kısmi Bilgi Atağı

Don Coppersmith [10] çalışmasında, aşağıdaki teoremi ispatlamıştır.

Teorem 3.1 (Coppersmith). [10] *$N = PQ$ çarpımı ve çarpanlardan birinin en üst $(1/4 + \epsilon)(\log_2 N)$ tane biti bilinirse, $\log N$ ve $1/\epsilon$ a bağlı polinom zamanda N çarpımını, P ve Q çarpanlarına ayrılabilir.*

Bu teoreme göre RSA açık anahtarı N değeri, gizli asallardan birinin üst yarı bitleri bilindiği taktirde çarpanlarına ayrılabilmektedir. Bu durum kleptografik bir atakta, gizli bilgiyi sızdırmak için, asallardan birinin hepsini sızdırmak yerine üst yarı bitlerini sızdırmanın yeterli olacağı anlamına gelmektedir.

3.4.2 Permutasyon Fonksiyonu

Önceki ataklarda rastgeleleştirme ve döngüler oluşturabilmek için simetrik şifreleme fonksiyonlarını ve özet fonksiyonlarını kullanmıştık. Crepeau ve Salkmon [9] çalışmalarında

permütasyon için kullandıkları fonksiyonları ele alırken, arka kapıların donanımsal uygulamasında, simetrik şifreleme fonksiyonlarının kullanılması durumunda, implementasyon alanının kapasitesinde problem oluşturabileceği için RSA ile aynı aritmetiğe sahip fonksiyonların kullanılmasının daha uygun olacağını belirtmişlerdir.

İkilinin bu atak için önerdiği iki permutasyon, $\mu, \beta \in \mathbb{Z}^+$ sabit tamsayılar olmak üzere ve $|x|$ ifadesi x'in bitsel uzunluğunu ve $\lfloor \mu \rfloor_n$ notasyonu ise μ'nun en düşük n tane bitini göstermek üzere, aşağıdaki gibidir;

$$\pi_{\beta,\mu}(x) = (x \oplus (2\mu)\rfloor_{|x|})^{-1} \bmod \beta$$

ve

$$\pi_{\beta,\mu}(x) = (x^{-1} \bmod \beta) \oplus (2\mu)\rfloor_{|\beta|}$$

Atağın implementasyonunu yaparken ikinci fonksiyonu kullanmayı tercih ettik. Bu fonksiyonun tersi aşağıdaki gibidir.

$$\pi^{-1}_{\beta,\mu}(x) = (x \oplus (2\mu)\rfloor_{|\beta|})^{-1} \bmod \beta$$

Bu fonksiyon düzgün çalışacaktır çünkü;

$$\begin{aligned}
\pi^{-1}(\pi(x)) &= \pi^{-1}((x^{-1} \bmod \beta) \oplus (2\mu)\rfloor_{|\beta|}) \\
&= ((x^{-1} \bmod \beta) \oplus (2\mu)\rfloor_{|\beta|} \oplus (2\mu)\rfloor_{|\beta|})^{-1} \bmod \beta \\
&= ((x^{-1} \bmod \beta) \oplus 0)^{-1} \bmod \beta && (a \oplus a = 0) \\
&= ((x^{-1} \bmod \beta))^{-1} \bmod \beta && (a \oplus 0 = 1) \\
\pi^{-1}(\pi(x)) &= x
\end{aligned}$$

olacaktır.

3.4.3 Atak

Arka kapı barındıran şifreleme sistemi aşağıdaki algoritmayla anahtar üretecektir. Algoritma sabitlenmiş e kuvvet değerine göre gerekli anahtarları üretebilecek şekilde sunulmuştur. $k - bit$ asal sayılar üreterek, çıktı olarak $2k$ veya $2k - 1$ bit n açık anahtarını üretir.

KleptoAnahtarÜreteç(k, e) : [9]
Girdi: k: asal bit uzunluğu, e açık anahtar kuvvet değeri.
Çıktı: (e, n, d) RSA anahtarları.
Gömülü Değerler:
$\beta, \mu \in_R \{0,1\}^{k/2}$ permütasyon fonksiyonunda kullanılacak sabit değerler.
Rastgelelştirme sonksiyonu:
$\pi_{\beta,\mu}(x) = (x^{-1} \bmod \beta) \oplus 2\mu\rfloor_{|\beta|}$
Operatörler : $||$: bitsel dizileri uç uca ekleme operatörü.
$|x|$: x'in bitsel uzunluğu.
$\mu\rceil_n$: μ'nun en üst (sol) n tane biti.
$\mu\rfloor_n$: μ'nun en alt (sağ) n tane biti.

1. $p \in_R \{0,1\}^k$ ve $(e, p-1) = 1$ olacak şekilde p asalı seç
2. $q' \in_R \{0,1\}^k$ tek tamsayısını seç ve $n' = pq'$ hesapla
3. $n \leftarrow n'\rceil^{k/4} \;||\; \pi_{\beta,\mu}(p\rceil^{k/2}) \;||\; n'\rfloor_{5k/4}$
4. $q \leftarrow \lfloor n/p \rfloor$
5. q çift ise:
   $q = q + 1$
6. $(e, q-1) > 1$ veya q asal değil iken:
   $m \in \{0,1\}^{k/4}$ ve çift m seç.
   $q \leftarrow q \oplus m$
7. $n \leftarrow pq$ ve $d \leftarrow e^{-1} \bmod \varphi(n)$ hesapla
8. (n, e) açık anahtarlar, (d) gizli anahtarları çıkart

ŞEKİL 3.7: Kleptografik RSA Anahtar Üretim Algoritması

Yayınlanan n mod değerini elde eden saldırgan, aşağıdaki algoritma ile p ve q asallarına ulaşabilecektir.

Klepto anahtar üretim ve anahtar ele geçirme algoritmaları düzgün çalışacaktır. Çünkü; anahtar üretme algoritmasıyla üretilen n açık anahtar modül değeri

$$n \leftarrow n'\rceil^{k/4} || \pi_{\beta,\mu}(p\rceil^{k/2}) || n'\rfloor_{5k/4}$$

AnahatarEleGeçirme(n, β, μ) : [9]
Girdi: (n, e) açık anahtarlar
$\beta, \mu \in \{0,1\}^{k/2}$ permütasyon fonksiyonunda kullanılan sabit değerler.
Çıktı: d gizli anahtar
Rastgeleleştirme fonksiyonu:
$\pi_{\beta,\mu}(x) = (x^{-1} \bmod \beta) \oplus 2\mu\rfloor_{|\beta|}$

1. $p_0 = \pi^{-1}_{\beta,\mu}(n\rceil^{3k/4}\rfloor_{k/2})$ hesapla $(p_0 = p\rceil^{k/2})$
2. $p, q = Coppersmith(n, p_0)$
3. $\varphi(n) = (p-1)(q-1)$
4. $d = e^{-1} \bmod \varphi(n)$ çıkart.

ŞEKİL 3.8: Anahtar Ele Geçirme Algoritması [9]

şeklinde belirlenmekte ve bu değerin sağ ve solunda birleştirilen parçalar gizli anahtarın ele geçirilmesiyle ilgili bir önem teşkil etmemektedir. Saldırgan anahtarı ele geçirmek istediğinde bu parçaları atacak ve geriye kalan değer ise;

$$\pi^{-1}_{\beta,\mu}(\pi_{\beta,\mu}(p\rceil^{k/2})) = p\rceil^{k/2}$$

π^{-1} ters fonksiyonu ile beraber, gizli p asalının bitlerinin üst yarısını verecektir. Bundan sonra ise Coppersmith atağı ile

$$p, q = \mathsf{Coppersmith}(n, p\rceil^{k/2})$$

hesaplayarak gizli asallara ulaşabilecektir.

3.4.4 Atağın Simülasyonu ve Analizi

Bölüm 3.2 ve 3.3'de olduğu gibi bu atak için de, KleptoAnahtarÜreteç ve AnahtarEleGeçirme algoritmalarının gerçeklenmesi ve analizleri yapılmış ve sonuçlar bu bölümde ele alınacaktır. Algoritmaların gerçeklendiği bilgisayarın fiziksel durumu, işletim sistemi ve gerçeklendiği platform gibi detaylar Bölüm 1.4.1'da yer almaktadır.

Diğer ataklarda olduğu gibi 150 adet anahtar değeri ile bu analizleri yapacağız. Bu bölümde ele aldığımız "Gizli Asal Çarpan" atağıyla ilgili üretilen 150 adet anahtar için rastgele bir mesajı şifreleme/şifre çözme başarıları ve açık anahtardan, gizli anahtarı ele geçirebilme başarı oranları test edilmiş. Ancak kurbanın açık anahtarından, gizli asalları ele geçirirken Coppersmith Kısmi Bilgi atağını kullanmak yerine, açık anahtardan ele

geçirerek Coppersmith algoritmasına girdi olarak verilmesi gereken, p asalının üst yarı bitlerini ele geçirebilme başarıları test edilmiştir. Yani Coppersmith Kısmi Bilgi atağının doğru çalışıyor olduğu kabulü altında gizli asallar ele geçirilebilir durumda olacaktır.

Bu şartlar altında şifreleme/şifre çözme başarıları ve açık anahtardan, gizli anahtarı ele geçirebilme başarı oranları %100 olarak gözlemlenmiştir.

3.4.4.1 Çalışma Zamanı

Simüle ettiğimiz atak için üretilen 150 adet anahtar değerinin ve Bölüm 3.1'de standart RSA anahtaları ile yapılan analizlerle karşılaştırmalı olarak, anahtar üretim zamanları, Tablo 3.7'de verilmiştir.

TABLO 3.7: Çalışma Zamanı:
150 adet anahtarın üretim zamanı ortalamaları (sn.)

asal bit uzunluğu:k	RSA	CS03
128	0,07	0,20
256	1,01	3,35
512	27,97	48,03

3.4.4.2 Açık Anahtarın Dağılımı

Atak barındıran algoritmayla ürettiğimiz anahtarlar için, n açık anahtar değerinin, büyüklüğüne göre olasılık dağılımı, önceki atakta olduğu gibi standart RSA anahtarları ile karşılaştırmalı olarak Tablo 3.8'de verilmiştir. Önceki bölümlerde ele aldığımız atakların aksime, bu atakda standart RSA anahtarları ile dağılım yüzdeleri uyuşmaktadır. Bu da atağın anahtarların dağılımını ölçerek tespit edilmesini engelleyecektir.

TABLO 3.8: Açık Anahtar Dağılım Tablosu:
n açık anahtarı dağılım yüzdeleri

	RSA			CS03		
asal.uzunluğu k	2k-1 bit	2k bit "10" ile	2k bit "11" ile	2k-1 bit	2k bit "10" ile	2k bit "11" ile
128	42	48	9	37	52	10
256	34	51	14	26	58	14
512	38	46	15	34	60	6

Sonuç olarak bu atak açık anahtarların ve anahtar üretimi çalışma zamanlarının standart RSA algoritmasıyla ile ayırt edilemezliği sağlamaktadır. Ancak daha önceden de bahsedildiği gibi atağın simetrik oluşu, atağı güvensiz hale getirmektedir.

Bölüm 4

Ayrık Logaritma Tabanlı Sistemlere Ataklar

Önceki bölümde Çarpanlara Ayırma Probleminin zorluğuna dayanan RSA şifreleme sistemine yapılan atakları incelemiştik. Kleptografik ataklar genel olarak RSA şifreleme sistemine karşı yoğunlaşmış olsa da diğer asimetrik şifreleme ve imzalama sistemlerine de karşı kurgulanan arka kapı çalışmaları mevcuttur. Bu bölümde Ayrık Logaritma Problemi'nin (detaylı açıklama için bkz. Bölüm 2.3) zorluğuna dayanan sistemlere karşı kurgulanan kleptografik atakları inceleyeceğiz.

Öncelikle Diffie-Hellman anahtar değişim protokolüne karşı kurgulanmış bir atağı inceleyecek sonrasında ise ElGamal şifreleme sisteminde, şifreleme esnasında şifreli metni arka kapı barındıracak şekilde nasıl hazırlanabileceğini göreceğiz. Son olarak da DSA imzalama sistemine karşı kurgulanmış kleptografik bir atağı inceleyeceğiz.

4.1 Diffie-Hellman Atağı

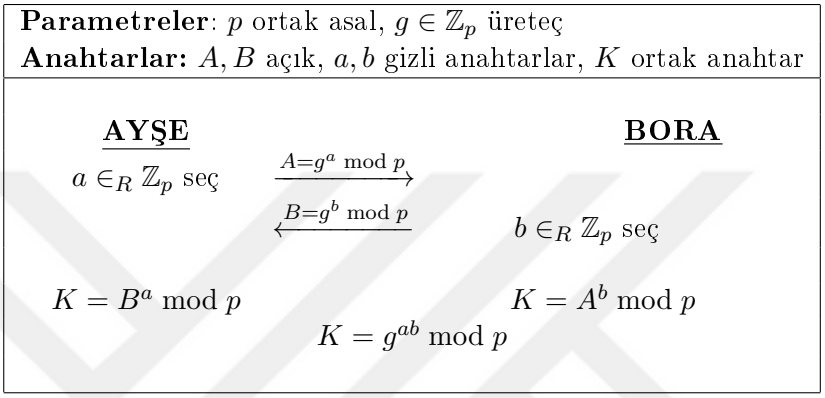
Diffie-Hellman anahtar değişim protokolü, şifreli haberleşebilmek için anahtarların dağıtılması problemine 1976 yılında Diffie ve Helllman tarafından [8] çalışmalarında getirdikleri ilk pratik çözümdür.

Bu protokolle birlikte, güvensiz bir kanalda şifreli haberleşmek isteyen tarafların birbirlerini görmeden, ortak anahtarı paylaşabilmeleri mümkün hale gelmiştir. Ayrık Logaritma Problemi'nin zorluğuna dayanan sistemlere karşı kurgulanmış kleptografik atakları ele alacağımız bu bölümde ilk olarak bu protokole karşı, Young ve Yung tarafından [11] kitaplarında sundukları kleptografik atağı inceleyeceğiz. Bunun için öncelikle standart Diffie-Hellman anahtar değişim protokolünü özet halinde vereceğiz.

4.1.1 Diffie-Hellman Anahtar Değişimi

Diffie-Hellman anahtar değişim protokolünün güvenliği, *Ayrık Logaritma Problemi'nin* (bkz Bölüm 2) zorluğuna dayanmaktadır ve aşağıdaki gibi çalışır.

Ayşe ve Bora güvenli olmayan bir kanalda güvenli haberleşmek istemektedirler ve bu sebeple ortak bir anahtarda uzlaşmaları gerekmektedir. Bunun için öncelikle bir p asalı ve bu asalın oluşturduğu $\mathbb{Z}_p$ grubunun bir g üreteç elemanı üzerinde açık olarak uzlaşırlar. Daha sonra aşağıdaki şemada olduğu gibi K ortak anahtarı üzerinde uzlaşırlar.



ŞEKİL 4.1: Diffie-Hellman Anahtar Değişimi [8]

Bu esnada hattı dinleyen bir saldırgan, (A, B) açık anahtar değerlerini ele geçirse bile bu değerlerin ayrık logaritmaları olan (a, b) gizli anahtarlarına ve dolayısıyla K ortak anahtar değerine ulaşamayacaktır.

Anahtar değişiminde gizli ve açık parametreler aşağıdaki algoritma ile üretilecektir.

DHParUret(p, g): [8]
Girdi: p asal sayı, $g \in \mathbb{Z}_p$ üreteç eleman
Çıktı: A açık, a gizli anahtar değerleri

1. $a \in_R \mathbb{Z}_p$ seç
2. $A = g^a \bmod p$ hesapla
3. a, A çıkart

ŞEKİL 4.2: Diffie-Hellman Anahtar Değişimi Parametre Üretim Algoritması

4.1.2 Klepto Diffie-Hellman

Şimdi Diffi-Hellman anahtar değişimi için Young ve Yung tarafından [11] kitabında sundukları kleptografik atağı inceleyelim. Bu atağa göre anahtar değişimi yapan taraflardan birinin cihazının atak algoritmasına göre anahtar üretmesi yeterli olacaktır.

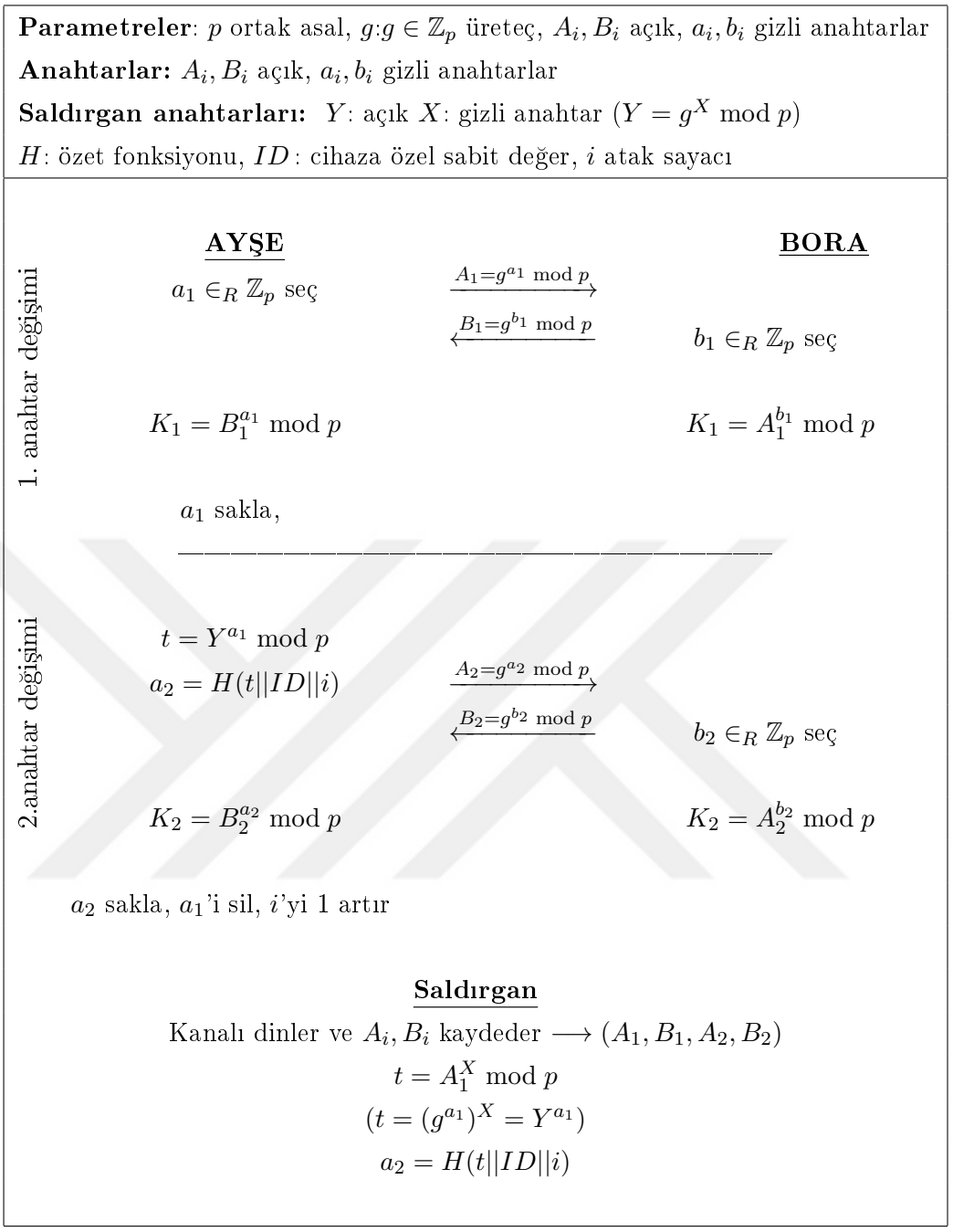
Arka kapı barındıran cihaz, açık ve gizli parametrelerin üretilmesi esnasında, üretilecek değerleri manipüle ederek saldırganın çalabilceği hale getirecektir ve aşağıdaki algoritmaya göre çalışacaktır.

KleptoDHParUret$(p, g, a_{i-1}, Y, ID, i)$ [11]
Girdi: p: asal sayı, $g : g \in \mathbb{Z}_p$ üreteç
Gömülü Değerler: Y saldırganın açık anahtar değeri
$(Y = g^X \bmod p)$ (X saldırganın gizli anahtarı)
ID : $|p| - bit$ uzunlukta cihaza özel sabit değer, i sayaç değeri (0'dan başlar her değişimden sonra 1 arttırılır)
a_{i-1} bir önceki anahtar değişiminin gizli anahtar değeri (her değişimden sonra eski değer silinir yeni değer saklanır)
θ : i sayaçını sınırlandırmak için sabit değer
Çıktı: A açık anahtar, a gizli anahtar

1. $i = 0$ veya $i > \theta$ ise:
   DHParUret(p, g) çıkart, a'tı sakla, $i = i + 1$
2. $t = Y^{a_{i-1}} \bmod p$
3. $a = H(t||ID||i)$
4. $A = g^a \bmod p$
5. A, a çıkart, a sakla, $i = i + 1$

ŞEKİL 4.3: Klepto Diffie-Hellman Parametre Üretim Algoritması

Bu algoritmayla üretilen anahtarların nasıl kullanılacağı ve nasıl çalınacağı aşağıdaki şema ile daha kolay anlaşılabilir.



ŞEKİL 4.4: Klepto Diffie-Hellman Anahtar Değişimi

Anahtar değişimleri esnasında hattı pasif olarak dinleyen saldırgan, (A_i, B_i) değerlerini kaydeder ve aşağıdaki ele geçirme algoritmasıyla ortak anahtar K_i değerine ulaşabilecektir.

AnahtarEleGeçir($A_{i-1}, B_{i-1}, A_i, B_i, p, g, X, ID$): [11]
Girdi: $A_{i-1}, B_{i-1}, A_i, B_i$: K_i anahtarını çalmak için gerekli açık anahtarlar
p, g kullanıcıların ortak asal sayı ve üreteç değeri
X : saldırganın gizli anahtarı
ID : cihaza özel p ile aynı uzunlukta sabit değer
θ : i sayaçını sınırlandırmak için sabit değer
Çıktı : K_i anahtarı

1. $t = A_{i-1}^X \bmod p$
2. $i = 0$
3. $i < \theta$ iken :
   $a_i = H(t||ID||i)$
   $A_i = g^{a_i} \bmod p$ ise:
   bırak.
   $A_i \neq g^{a_i} \bmod p$ ise:
   $i = i + 1$ yap.
4. $K_i = B_i^{a_i} \bmod p$ çıkart.

ŞEKİL 4.5: Klepto Diffie-Hellman Anahtar Ele Geçirme Algoritması

Saldırganın ele geçirme algoritmasının ilk adımında yaptığı hesaplama, çalınmak istenen gizli anahtar değerini verecektir çünkü:

$$t = A_{i-1}^X \bmod p = (g^{a_{i-1}})^X = (g^X)^{a_{i-1}} = Y^{a_{i-1}} \bmod p = t$$

Anahtar üretimi esnasında oluşturulan t değerine eşit olacaktır. Ele geçirme algoritmasını geri kalanında i değeri, θ ile sınırlandırıldığından tahmin edilebilir durumdadır ve ancak polinomsal bir zaman gerektirecektir. Geri kalan işlemler ise anahtar üretim algoritmasıyla aynı şekildedir.

4.1.3 Atağın Simülasyonu ve Analizi

Önceki bölümde incelediğimiz RSA için kleptografik ataklarda olduğu gibi, Diffie - Hellman atağı için de algoritmaların simülasyonu yapılmış ve standart algoritma ile karşılaştırılmalı analizlerini bu bölümde ele alınacaktır.

Diğer ataklarda olduğu gibi, algoritmaların gerçeklendiği bilgisayarın fiziksel durumu, işletim sistemi ve gerçeklendiği platform gibi detayları Bölüm 1.4.1'de yer almaktadır.

Atağın gerçeklenmesi ve kıyas yapabilmek için, öncelikle standart Diffie - Hellman anahtar paylaşım protokolünü 100 adet anahtar değişecek şekilde gerçeklenmiştir, ataklar içinse aynı şekilde ancak özel olarak birbirini takip edecek şekilde i değerleri ile (anahtar ele geçirmenin başarılı olabilmesi için, anahtarların takip eden anahtar değişiminden olması gerekmekte) 100 adet anahtar değişimini simüle edilmiştir.

Simülasyonlar, p asal sayı bitsel uzunluğu 128, 256 ve 512 bit olacak şekilde 3 ayrı anahtar kümesinde gerçeklenmiştir.

4.1.3.1 Anahtar Değişim ve Ele Geçirme Başarıları

Standart ve klepto algoritma için üretilen 100 adet (A_i, B_i) anahtar çifti için anahtar değişimi başarıları test edilmiş ve hepsinin başarılı olarak gerçekleştiği tespit edilmiştir.

Anahatar ele geçirme ise 100 adet anahtarın ilk K_0 ortak anahtarı hariç (zaten atağın gerçekleşebilmesi için önceden paylaşılmış anahtarlar gerekmektedir) hepsi için yani %99 başarı oranıyla gerçekleştiğini söyleyebiliriz.

4.1.3.2 Çalışma Zamanı

Kapalı kutu bir Bir şifreleme sisteminde kleptografik bir atak olup olmadığını anlamak için teste tabi tutulacak iki parametreden biri çalışma zamanı, diğeri ise açık anahtarların dağılımıdır.

Bu bölümde incelediğimiz atakta anahtar üretim algoritması standart anahtar üretim algoritmasına göre, fazladan sadece bir kuvvet alma ve özet fonksiyondan geçirme işlemi yapmaktadır. Özet alma işlemi modüler kuvvet alma gibi işlemlerin olduğu bir algoritmada ihmal edilebilecek kadar kısa zamanda gerçekleşmektedir, diğer fazla işlem olan kuvvet alma ise tabloda da görülebileceği gibi ayırt edilemezliği bozmamaktadır.

Simüle edilen 100 adet anahtar değişimi içi, anahtarların üretilme süreleri Tablo 4.1'da yer almaktadır. Tabloda da görülebileceği gibi çalışma zamanları arasında ayırt edilebilir bir fark bulunmamaktadır. Sonuç olarak Diffie-Hellman anahtar değişimi için bu bölümde ele aldığımız kleptografik atak çalışma zamanı olarak ayırt edilemezliği sağlamaktadır.

TABLO 4.1: Çalışma Zamanı:
150 adet anahtarın üretim zamanı ortalamaları (sn.)

asal.bit.uzunluğu:k	Standart DH	Klepto DH
128	0,0007	0,001
256	0,005	0,012
512	0,07	0,15

4.1.3.3 Açık Anahtar Dağılımı

Kapalı kutu bir sistemde arka kapı olup olmadığını anlamak için teste tabi tutulması halinde kontrol edilen parametrelerden birinin açık anahtarların dağılımı olduğunu bir önceki bölümde belirtmiştik.

Diffie-Hellman anahtar değişim protokolünde üretilen açık anahtarlar çıktı olarak verilmeden önce tabi tutulduğu son işlem modüler kuvvet alma işlemi olduğu için, anahtarların $|p|$ asal bit uzunluğu ile $|p| - 10$ bit uzunluğa kadar yayılabilmektedir. Bu sebeple çıktıları analiz ederken, RSA şifreleme sisteminde olduğu gibi bit uzunluğu ve ayrıca büyüklüklerini kontrol etmek yerine, sadece bit uzunluklarına bakmamız gerekecektir.

Bu bölümde ele aldığımız atak için, açık anahtarların ve standart Diffie-Hellman anahtar değişiminde üretilen açık anahtarların karşılaştırılması Tablo 4.2'da yer almaktadır.

Tabloda da görülebileceği gibi arka kapı barındıran açık anahtarlar ile standart algoritma ile üretilen anahtarlar arasında dağılım olarak ayırt edici bir fark bulunmamaktadır. Sonuç olarak Diffie-Hellman atağının açık anahtarların dağılımı olarak ayırt edilemezliği sağladığını söyleyebiliriz.

TABLO 4.2: Açık Anahtar Dağılım Tablosu:
A_i açık anahtarları dağılım yüzdeleri

	$\|p\| = 128$		$\|p\| = 256$		$\|p\| = 512$	
$\|p\| = k$ bit	Standart	Klepto	Standart	Klepto	Standart	Klepto
k bit	52	50	41	33	38	46
$k-1$ bit	21	28	27	36	27	31
$k-2$ bit	16	9	17	16	17	12
$k-3$ bit	10	4	6	4	11	7
$< k-3$ bit	1	10	9	12	7	5

4.2 ElGamal Şifreleme Atağı

Güvenliği *Ayrık Logaritma Problemi'nin* nin zorluğuna dayanan asimetrik şifreleme sistemlerinden en çok bilineni, ElGamal [4] şifreleme sistemidir diyebiliriz. Bu bölümde,

Yung ve Youn tarafından [11] eserlerinde sunulan, ElGamal şifreleme sistemine karşı kleptografik bir atağı inceleyeceğiz. Atak mekanizmasına geçmeden önce şifreleme sistemini kısaca hatırlatmak faydalı olacaktır.

4.2.1 ElGamal Şifreleme Sistemi

Şifreleme için kullanılmak üzere gizli ve açık anahtarlar üretmek isteyen Ayşe, öncelikle $k - bit$ uzunluğunda bir p asal sayısını rastgele seçer ve bu asalın oluşturduğu $\mathbb{Z}_p$ kümesinde üreteç bir eleman olacak şekilde $g \in \mathbb{Z}_p$ belirler. Sonrasında rastgele bir $x \in_R \mathbb{Z}_p$ seçer ve $y = g^x \bmod p$ hesaplar. Bu anahtarları (p, g, y) açık anahtar ve x gizli anahtar olarak yayımlar.

Ayşe'ye bu anahtarlarla şifreli mesaj göndermek isteyen Bora rastgele bir $r \in \mathbb{Z}_p$ seçerek;

$$c_1 = g^r \bmod p \text{ ve } c_2 = y^r m \bmod p$$

değerlerini hesaplar. Son olarak $c = (c_1, c_2)$ ikilisini şifreli mesaj olarak gönderir.

c şifreli mesajını çözmek isteyen Ayşe m mesajına;

$$m = c_2 / c_1^x \bmod p$$

hesaplayarak ulaşabilecektir [4].

4.2.2 Klepto ElGamal Şifreleme Sistemi

Önceden de belirttiğimiz gibi atak algoritması şifreleme esnasında çalışmaktadır ve üretilen şifreli mesaj değerleri arka kapı barındıracak şekilde üretilmektedir.

Atak bulunan cihazda saldırganın ElGamal açık anahtarı $Y = g^X \bmod p$ bulunmaktadır ve asal sayı ve üreteç değeri olarak kurbanla saldırgan aynı değerleri kullanmaktadır. Saldırganın gizli anahtar X değeridir ve atak barındıran sistemde bulunmaz, saldırgan mesajları ele geçirmek için bu değeri kullanacaktır.

Sistemde ayrıca ID ve i değerleri de yer alacaktır. Bu değerlerden ID, p asalı ile aynı $k - bit$ uzunluğunda ve her cihaz için (saldırganın ürettiği her cihaz özel yerleştirdiği değerler) özel olan bir değerdir. i değeri ise atak algoritmasıyla üretilen şifreli metinlerin sayısını belirtir ve ilk kullanımda 0'dan başlatılır ve her kullanımdan sonra 1 arttırılır.

Ayşe'nin ElGamal şifreleme yapacak cihazının atak algoritmasıyla çalıştığını ve Bora'ya bir şifreli mesaj göndermek istediğini varsayalım. Bora'nın ElGamal açık anahtarları (p, g, y) olsun. Atak barındıran cihaz aşağıdaki algoritmayla şifreleme yapacaktır.

KleptoElGamalŞifrele(p, g, y, m): [11]
Girdi: p $k-bit$ asalsayı, $g \in \mathbb{Z}_p$ üreteç, m mesaj
Gömülü Değerler: Y: saldırganın açık anahtarı, $ID \in \{0,1\}^k$ cihaza özel sabit, i: atak sayaç değeri, θ sayaç durdurmak için sabit değer H: kriptografik özet fonksiyonu
Çıktı: $c = (c_1, c_2)$ şifreli metin

1. $i = 0$ veya $i > \theta$ ise:
   - 1.1 $r \in_R \mathbb{Z}_p$ seç
   - 1.2 $c_1 = g^r \bmod p$
   - 1.3 $c_2 = y^r m \bmod p$
   - 1.4 $c = (c_1, c_2)$ çıkart, $i = i + 1$ yap, r sakla
2. $i < \theta$ ise:
   - **2**.1 $t = Y^r \bmod p$
   - **2**.2 $r = H(t||ID||i)$
   - **2**.3 $c_1 = g^r \bmod p$
   - **2**.4 $c_2 = y^r m \bmod p$
   - **2**.5 $c = (c_1, c_2)$ çıkart, $i = i + 1$ yap, r sakla

ŞEKİL 4.6: Klepto ElGamal Şifreleme Algoritması

İletişim kanalının pasif olarak dinleyen saldırgan i. şifreli metnin orijinal açık metnini ele geçirmek için (c_{i-1}, c_i) değerlerini kaydedecek ve aşağıdaki algoritmayla m_i mesajına ulaşabilecektir.

MesajEleGeçir(c_i, c_{i-1}, X, Y, ID): [11]
Girdi: c_i: Çalınmak istenen mesajın şifreli metni, c_{i-1}: Bir önceki şifreli metin.
$Y(= g^X \bmod p), X$: Saldırganın açık ve gizli anahtarları.
ID: Cihaza özel sabit değer.
Sabit Parametreler: θ: Sayaç sıfırlamak için kullanılmış sabit değer.
H: kriptografik özet fonksiyonu.
Çıktı: m_i: i. açık metin.

1. $c_{11}, c_{12} = c_{i-1}$
   $c_{21}, c_{22} = c_i$
2. $t = c_{11}^X \bmod p$
3. $i \in (1, \theta)$ için:
   $r = H(t||ID||i)$
   $g^r = c_{21}$ ise:
   bırak.
   değilse:
   $i = i + 1$
4. $m_i = c_{22}/Y^r \bmod p$ hesapla ve m_i çıkart

ŞEKİL 4.7: Klepto ElGamal Mesaj Ele Geçirme Algoritması

KleptoElGamalŞifrele ve MesajEleGeçir algoritmaları doğru çalışacaktır çünkü c_{i-1} ve c_i şifreleme sırasına göre takip eden şifreli metinler olmak üzere;

$$c_{11}, c_{12} = c_{i-1} \text{ ve } c_{21}, c_{22} = c_i$$

ve $c_{11} = g^k \bmod p$ şeklinde üretilmiş olsun. Bu durumda c_{21} değeri,

$$c_{21} = Y^k \bmod p$$

ile üretildiğinden ve $Y = g^X \bmod p$ olduğundan;

$$Y^k \bmod p = (g^X)^k \bmod p = (g^k)^X \bmod p = c_{11}^X \bmod p$$

olacaktır. Dolayısıyla KleptoElGamalŞifrele algoritmasında c_i şifreli metninin birinci kısmını üretirken g üretecinin kuvvetine koyduğumuz t değeri ile MesajEleGeçir algoritmasında, mesaja ulaşmak için kullandığımız t değeri birbirine eşittir. Dolayısıyla KleptoElGamalŞifrele ve MesajEleGeçir fonksiyonları için aşağıdaki eşitlik gerçeklenecektir.

$$\text{MesajEleGeçir}(\text{KleptoElGamalŞifrele}(m)) = m$$

4.2.3 Atağın Simülasyonu ve Analizi

Bu çalışmada incelediğimiz diğer kleptografik ataklarda olduğu gibi, ElGamal şifreleme sistemi için kurgulanmış bu atağınn da simülasyonunu ve standart sistem ile karşılaştırmalı analizleri yapılmış ve sonuçlar bu bölümde sunulacaktır.

Simülasyon işlemi, standart ve klepto algoritmalarla 100'er adet metni şifreleyerek gerçekleştirilmiş ve her iki versiyon için de beklenildiği gibi şifreleme/şifre çözme işlemleri %100 başarı ile sonuçlandığı görülmüştür. Diğer yandan, şifreli mesajın gönderildiği tarafın, (anahtar sahibi) x gizli anahtar değerini bilmiyorken, şifrelenmiş m mesajlarını ele geçirme başarısı ise %99 başarı ile (ilk şifreli mesaj standart algoritma ile üretildiğinden) sonuçlandığı, yapılan testler sonucunda belirlenmiştir.

Atağın ayırt edilemezliğini incelemek için diğer ataklarda, üretilen açık anahtarların, üretilme zamanlarına ve olasılık dağılımlarını inceliyorduk ancak bu atak için tek kriterimiz çalışma zamanları olacaktır. Bunun sebebi ise atak mekanizmasının şifreleme esnasında gerçekleşmesi ve c_i şifreli metinlerin dağılımlarını test etmenin ayırt edici bir kriter olarak düşünemeyeceğimizdir.

Ancak çalışma zamanlarını incelersek, atağın standart algoritmayla önemsiz farktaki bir zamanda çalıştığını söyleyebiliriz. Standart ve kleptografik versiyonlar için üretilen 100'er adet şifreli metin için, üretim zamanları Tablo 4.3'da yer almaktadır.

Tablo 4.3: Çalışma Zamanı
100 adet şifreli metin için şifreleme zamanı ortalamaları (sn.)

asal.bit.uzunluğu :k	Standart ElGamal	Klepto ElGamal
128	0,001	0,002
256	0,011	0,017
512	0,15	0,22

4.3 DSA İmza İçin Kleptografik Atak

Güvenliği Ayrık Logaritma Problemi'nin zorluğuna dayanan sistemler için kurgulana kleptografik atakları incelediğimiz bu bölümde Diffie-Hellman anahtar değişimi, ElGamal şifreleme sisteminden sonra, DSA [5] dijital imzala algoritmasına karşı kurgulanmış kleptografik bir atağı bu bölümde inceleyeceğiz.

Young ve Yung [11] çalışmalarında, ElGamal [4], Pointcheva-Stern [36], DSA [5] ve Schnorr dijital imza [37, 38] algoritmaları için, bu bölümde incelediğimiz ayrık logaritma atağının nasıl uygulanabileceğini göstermişlerdir. Ataklarında hepsinde temel mantık, DH ve ElGamal ataklarındaki gibi rastgele seçilmesi gereken değer yerine, bir bakıma saldırganla kurban arasında bir ortak anahtar oluşacak şekilde manipüle ederek, imzalamada kullanılan gizli anahtar değerini saldırganın ele geçirebileceği şekilde sızdırmaktır.

Öncelikle DSA dijital imza algoritmasını kısaca hatırlamak faydalı olacaktır.

4.3.1 DSA Dijital İmza Algoritması

DSA imzalama 1991 yılında Amerika, Ulusal Standartlar ve Teknoloji Enstitüsü NIST tarafından önerilmiş ve daha sonra 1993 yılında FIPS-186 dökümanında standart olarak yayınlanmıştır [5].

Kısaca algoritma şöyle çalışmaktadır: Öncelikle L ve N bit uzunluklarında p ve q asal sayıları, q asalı $p-1$ değerini bölecek iekilde seçilir. Bit uzunluklarıL ve N, FIPS-186'da, imza güvenliğinin sağlaması beklenen zaman süresine göre anahtar boyu artacak şekilde ((1024,160),(2048,224),(2048,256)) olarak önerilmiştir.

Asalların seçilmesinden sonra $\mathbb{Z}_p$'de derecesi q olacak şekilde bir $g \in \mathbb{Z}_p$ elemanı seçilir. Bu (y, q, g) değerleri açık parametreler olacaktır.

Kullanıcı, imzalamada kullanacağı anahtarları bu parametrelere göre seçecektir, bunun için öncelikle $x \in_R \mathbb{Z}_p^*$ gizli anahtarı ve $y = g^x \bmod p$ olacak şekilde açık anahtarını belirler. (p, q, g, y) değerleri açık anahtar, x ise gizli anahtar değeri olarak kullanılacaktır.

Bu anahtarlarla bir m mesajını imzalamak isteyen kullanıcı, öncelikle $k \in_R \mathbb{Z}_q$ değerini seçer ve

$$r = (g^k) \bmod p \text{ ile } s = k^{-1}(H(m) + xr) \bmod q$$

hesaplayarak imza değerini oluşturur.

$$Imza_x(m) = (r, s)$$

çifti olacaktır.

İmzanın doğruluğunu kontrol etmek isteyen bir kullanıcı açık anahtarları kullanarak;

$$r \stackrel{?}{=} (g^{s^{-1}H(m)} y^{rs^{-1}} \bmod p) \bmod q$$

şeklinde imzayı kontrol edebilecektir [5].

4.3.2 Klepto Dijital İmza Algoritması (DSA)

Bu bölümde inceleyeceğimiz kleptografik DSA atağında, atak barındıran cihaz saldırgan, kurbanla aynı (p, q, g) parametreleriyle ve kurbanın anahtarları ile aynı şekilde üretilmiş,saldırganın Y açık anahtarının barındırmaktadır ve atak algoritması bu açık anahtarın kuvvetinde, kurbanın x gizli anahtar değerini $Y^x \bmod p$ şeklinde hesaplayıp, bu değeri saldırganın ele geçirebileceği şekilde sızdırarak ele geçirebilmesine imkan sağlayacaktır.

Şifreleme için kullanılmak üzere (p, q, g, y) kurbanın açık anahtarı, x gizli anahtarı ve Y ve X değerleri sırasıyla saldırganın, kurbanla aynı parametrelerde ürettiği açık ve gizli anahtar değerleri olsun.

Atak barındıran cihaz, bir m mesajını aşağıdaki algoritmaya göre imzalayacaktır.

KleptoİmzalaA(p, q, g, x, m) : [11]
Girdi : $p \in \{0,1\}^L$ asal sayı, $q \in \{0,1\}^N$ ve $q|p-1$ asal sayı
g : $ord(g) = q$ olacak şekilde $g \in \mathbb{Z}_p$; $x \in \mathbb{Z}_q$ kullanıcı gizli anahtarı
Çıktı : $Imza_x(m) = (r, s)$ dijital imza
Gömülü Dğerler :
Y : Saldırganın açık anahtarı, ($Y = g^X \bmod p$, X : gizli anahtar),
ID : $|ID| = |p|$ Cihaza özel sabit değer, i Atak sayaç değeri (her ataktan sonra 1 arttırılır), H: Kriptografik özet fonksiyonu

1. $t = Y^x \bmod p$
2. $k = H(t||ID||i)$
3. $r = (g^k \bmod p) \bmod q$
4. $s = k^{-1}(H(m) + xr) \bmod q$
5. (r, s) çıkart

ŞEKİL 4.8: Klepto DSA İmza Algoritması [11]

Atak algoritmasındaki, i sayaç değeri, her imzada aynı x değeri alacağından gereksiz gibi görünse de k sözde rastgele değerini bulurken, H özet fonksiyonuna soktuğumuz x ve ID değerleri değişmezken, değişecek tek değer i değeri olduğundan, k değerinin rastgele görünmesi bu i değerine bağlıdır.

İmzalanmış bir metnin onaylanması standart DSA algoritmasındaki gibi gerçekleşecekken, imzalanmış bir metinden, kurbanın gizli anahtarını ele geçirmek isteyen saldırgan aşağıdaki algoritmayla bunu gerçekleştirebilecektir.

AnahtarEleGecir $(p, q, g, y, X, Y, ID, r, s, m)$: [11]
Girdi : $p \in \{0,1\}^L$ asal sayı, $q \in \{0,1\}^N$ ve $q|p-1$ asal sayı.
g : $ord(g) = q$ olacak şekilde $g \in \mathbb{Z}_p$; $y \in \mathbb{Z}_q$: Kullanıcı açık anahtarı.
$X \in \mathbb{Z}_q$: Saldırgan gizli anahtarı.
$|ID| = |p|$: Cihaza özel sabit değer.
(r, s) : İmza değerleri, m : İmzalanan metin.
Çıktı : $x \in \mathbb{Z}_q$: kullanıcı gizli anahtarı.

1. $t = y^X \bmod p$
2. $i = 0$
3. $r = (g^k \bmod p) \bmod q$ oluncaya kadar:
   $k = H(t||ID||i)$
   $r = (g^k \bmod p) \bmod q$ ise bırak
   değilse $i = i + 1$
4. $x = (sk - H(m))r^{-1} \bmod p$
5. x çıkart

ŞEKİL 4.9: Klepto DSA Anahtar Ele Geçirme Algoritması

KleptoİmzalaAnahtarEleGeçir algoritmaları

$$\text{AnahtarEleGeçir}_X(\text{KleptoİmzalaA}_x(m)) = x$$

olacak şekilde düzgün çalışacaktır çünkü klepto imzala algoritmasında: $k = H(t||ID||i)$ üretirken kullanılan t değeri için;

$$t = Y^x \bmod p = (g^X)^x \bmod p = g^{xX} \bmod p$$

olacak şekilde saldırgan anahtar ele geçirme aşamasında aynı t değerine ve k değerine ulaşacaktır. Bundan sonra s değeri, $s = k^{-1}(H(m) + xr) \bmod q$ ile üretildiğinden bu eşitlikten x'i çözersek:

$$x = (sk - H(m))r^{-1} \bmod q$$

şeklinde x değeri ele geçirilecektir.

4.3.3 Atağın Simülasyonu ve Analizi

Diğer ataklarda olduğu gibi bu atak için de, standart ve klepto algoritmaların gerçeklenmesini ve karşılaştırmalı analizlerini yaptık. Analizler, p asalının bit uzunluğu 512 ve 1024 bit olacak şekilde iki farklı anahtar kümesinde, standart ve klepto algoritmalar için aynı anahtarlar ve parametreler kullanılacak şekilde gerçekleştirildi. İki anahtar takımında da q asalının bit uzunluğu 160 olarak seçildi.

Her iki algoritma için de, imzalama/onaylama ve klepto algoritma için gizli anahtar ele geçirebilme başarısını beklendiği gibi % 100 olarak sağlandığı belirlenmiştir.

Algoritmalarda özet fonksiyonu olarak SHA-1 fonksiyonunu kullandık. Algoritmaların gerçeklendiği bilgisayarın fiziksel durumu, işletim sistemi ve gerçeklendiği platform gibi detayları Bölüm 1.4.1'de yer almaktadır.

4.3.3.1 Rastgele Değer Dağılımları

Atağın imzalama esnasında rastgele seçilen değerin manipüle ediliyor olmasından dolayı, imza algoritmasında çıktı olarak (r, s) imza değerleri ile birlikte, rastgele değer olan k'yı da aldık ve bu değerin kullanıcıya çıktı olarak veriliyor olması varsayımı altında, k değerlerin dağılımını bit uzunluğu olarak inceledik.

Standart ve klepto algoritmalar için, imzalamada $r = (g^k \bmod p) \bmod q$ eşitliğindeki k değerinin dağılımları Tablo 4.4'da bulunmaktadır. Bu dağılımlar, q asalının bit uzunluğuna bağlı olarak belirlenecektir ancak burada standart ve klepto algoritmalarının kullandığı rastgele değerlerin $|q|$ değerinin çevresinde nasıl dağıldığı incelenebilir.

Tablo 4.4: Rastgele Değer Dağılım Tablosu:
k rastgele değer dağılım yüzdeleri

	$\|p\| = 512$, $\|q\| = 160$		$\|p\| = 1024$, $\|q\| = 160$	
$k \in \mathbb{Z}_q$	StandartDSA	KleptoDSA	StandartDSA	KleptoDSA
160 bit	43	47	28	49
159 bit	32	20	37	27
158 bit	15	20	22	11
157 bit	4	3	6	7
$<$ 157 bit	6	10	7	6

Tablo 4.4'deki değerleri inceleyecek olursak; dağılımların, atağın ayırt edilemezliğini bozacak seviyede farklılaşmadığını görebiliriz. Aynı bit uzunlukları için farklılıklar olsa da, bu farklılık seçilen her örneklem kümesi için, klepto dağılımlarından az veya çok olabilecek şekilde büyük (%10 seviyeleri) değişiklikler gösterebilecektir.

4.3.3.2 Çalışma Zamanı

Atağın varlığının test edilmesi halinde, gözlemlenecek kriterlerden birinin çalışma zamanı olduğunu daha önceki bölümlerde söylemiştik. Bu atak için de imzalanan 100 mesajın imzalama sürelerinin hesapladık. Her iki algoritma için çalışma zamanları Tablo 4.5'te yer almaktadır.

Tablo 4.5: Çalışma Zamanı:
100 adet metin için imzalama zamanı ortalamaları (sn.)

asal.bit.uzunluğu:k	Standart DSA	Klepto DSA
512	0,008	0,017
1024	0,030	0,058

Tablodan görülebileceği üzere, kullandığımız asal bit uzunluklarında, çalışma zamanları atağın ayırt edilemezliğini bozmamaktadır.

Bölüm 5

Eliptik Eğrilerle Ataklar

Bölüm 3'te sunduğumuz RSA ataklarında, üretilecek anahtarın yarısı kadar uzunluktaki saldırgan anahtarıyla, çalınmak istenen gizli değer şifrelenerek sızdırılmaya çalışılmaktaydı. Bu durum atak mekanizmasından haberdar olan başka bir saldırgana karşı, kullanıcının güvenlik seviyesini yarıya indirmektedir. Daha kısa anahtar boyları ile daha yüksek güvenlik seviyesini amaçladığımız böyle durumlarda eliptik eğri kriptolojisi öne çıkmaktadır.

Eliptik eğrilerin kriptografide kullanılmasıyla daha kısa anahtar boylarında daha yükse güvenlik seviyelerine ulaşılabilmektedir. Şekil 5.1'de yer alan liste `www.keylength.com` sitesinden alınmıştır ve NIST'in (Amerikan Ulusal Standartlar ve Teknoloji Enstitüsü) önerdiği anahtar boylarını kıyaslamaktadır.

Böylece Young ve Yung ikilisi 2006, 2007 ve 2010 yılındaki 3 çalışmalarında eliptik eğrileri kleptografik ataklarda kullanılışlarına dair farklı yaklaşımları ele almışlardır.

Bu çalışmaların hepsinde, saldırgan ile kurban cihazı arasında bir gizli anahtar değişimi gerçekleştirilmekte ve bu anahtar değişiminin ortak anahtarı ile kurban sistemin üreteceği gizli anahtar değerini belirleyecek şekilde, kleptografik ataklar sunmuşlardır.

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve
2010 (Legacy)	80	2TDEA*	1024	160	1024	160
2011 - 2030	112	3TDEA	2048	224	2048	224
> 2030	128	AES-128	3072	256	3072	256
>> 2030	192	AES-192	7680	384	7680	384

ŞEKİL 5.1: NIST Önerilen Anahtar Boyları

5.1 Eliptik Eğriler

Bu bölümde eliptik eğriler ve kriptografi de kullanılışları hakkında gerekli altyapı hakkında özet olarak ele alacağız. Detaylı açıklamalar Bölüm 2'de yer almaktadır.

Eliptik Eğriler:

Ataklarda kullanacağımız eliptik eğrileri aşağıdaki gibi tanımlayabiliriz.

Tanım 5.1. Sonlu bir $\mathbb{F}_p$ cismindeki $E_{a,b}(\mathbb{F}_p)$ eliptik eğrisi, $a, b \in \mathbb{F}_p$ ve $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ olmak üzere

$$y^2 = x^3 + ax + b$$

eşitliğini sağlayan $P = (x, y)$ noktalarına sonsuzdaki $\mathcal{O}$ noktasının eklenmesiyle oluşan kümedir.

Eliptik Eğrilerde Grup Operasyonları:

Bir eliptik eğri üzerindeki iki nokta için toplama işlemi "teğet ve kiriş kuralı " olarak bilinen grup operasyonu ile yapılır. Bu işleme göre bir eliptik eğri komutatif bir grup oluşturacaktır [30].

Eliptik eğrilerde sabitle çarpma işlemini ise şöyle tanımlayacağız, $a \in \mathbb{Z}$ ve $P \in E_{a,b}(\mathbb{F}_p)$ için; $a \cdot P$ işlemi, a tane P noktasının toplamı olarak tanımlanacaktır. Özel olarak $0 \cdot P = \mathcal{O}_E$ ve $1 \cdot P = P$ dir.

Grup operasyonları için detaylı algoritmalar ve eliptik eğrilerle ilgili detaylı bilgiler Bölüm 2'de bulunabilir.

5.2 Gizli Anahtar Değişimi

Bu bölümde ele alınacak kleptografik ataklarda, saldırganla kurban cihazı arasında eliptik eğriler kullanılarak gizli bir anahtar değişimi gerçekleştirilecek ve paylaşılan ortak anahtar, cihazın kurban için üreteceği gizli bilgileri (ör: RSA için p asalı, SSL/TLS için PreMaster anahtarı) belirlemede kullanılacaktır. Bu saldırıyı Bölüm ele aldığımız rastgele bit üreteci sayesinde gerçekleştireceğiz.

Rastgele bit üreteci, rastgele değerler üretiyor görünecek ancak aslında saldırganla cihaz arasındaki gizli anahtar değişimi için gerekli gizli ve açık anahtarları üretiyor olacaktır. Bu durumu biraz daha netleştirmek istersek RSA için nasıl kullanılacağını şöyle özetleyebiliriz. Bir sonraki bölümde sunulacak olan $m_A, m_K = \mathsf{AnahtarÇiftiÜreteç}(Y_0, Y_1)$ fonksiyonu ile üretilen değerlerden m_K'yı paylaşılacak gizli anahtar olarak, RSA asallarından p'nin üst bitlerini, sonuna m_K'nın bit uzunluğu kadar rastgele $rast$ bit dizisi eklenerek , p asalını belirleyecektir.

$$p = m_K||rast$$

Aynı oturum için üretilecek RSA açık modulu n değeri ise, m_A değerinin bitleri arasına sadece saldırganın ulaşabileceği şekilde gizlenecektir. Uygulamada bu durum, $\alpha \in \{0,1\}^{|n|/4}$ ve $\gamma \in \{0,1\}^{3|n|/4-|m_A|}$ olmak üzere

$$n = \alpha||m_A||\gamma$$

formatında olacaktır.

Rastgele bit üreteci gibi davranacak, gizli anahtar değişim protokolünü; sistem parametrelerinin seçildiği *hazırlık aşaması*, cihaz tarafından gizli bilginini sızdırılmaya müsait hale getirildiği *saldırı aşaması* (veya anahtar değişimi) ve saldırganın yayınlanan açık değerleri kullanarak, mahrem bilgilere nasıl ulaşabileceğini gösteren *Anahtar ele geçirme aşaması* olarak 3 adımda bu protokolü inceleyeceğiz. Bu protokolde Bora ürettiği şifreleme sistemine atak mekanizmasını yerleştiren saldırgan, Ayşe ise mahrem bilgileri çalınan kullanıcı (kurban) pozisyonundadır.

Gizli Anahtar Değişimi : [13]

Adım 1: Bora, $x_0 \in [0, r_0 - 1]$ ve $x_1 \in [0, r_1 - 1]$ tam sayılarını rastgele seçer ve $(Y_0, Y_1) = (x_0 \cdot G_0, x_1 \cdot G_1)$ çiftini, diğer gerekli parametreler ile birlikte (detaylı bilgiler uygulama bölümünde bulunmaktadır.) üretim esnasında kurbanın cihazına yerleştirir.

Adım 2: Ayşe (cihaz), $(m_A, m_K) = \mathsf{GizliAnahtarÜreteç}(Y_0, Y_1)$ üretir ve m_A değerini, sadece Bora'nın ulaşabileceği şekilde, yayınlanacak açık parametreler içinde saklar.

Adım 3: Bora, $m_K = \mathsf{KleptoAnahtarEleGeçir}(m_A, x_0, x_1)$ hesaplar ve gizli değeri elde eder.

ŞEKİL 5.2: Saldırgan ile Cihaz Arasındaki Gizli Anahtar Değişim Protookolü

Şimdi bu üç adımı inceleyelim. Bundan sonraki bölümlerde yerine göre $E_{a,\,b}(\mathbb{F}_p)$ eğrisi, E_0 ile; $E_{a\beta^2,\,b\beta^3}(\mathbb{F}_p)$ eğrisi ise E_1 ile gösterilebilecektir.

5.2.1 Adım 1: Hazırlık Adımı

Bu adımda saldırgan, atak barındıracak olan cihazı üretirken, kullanıcının gizli bilgilerini sızdırabilecek şekilde çıktı üretecek olan algoritmanın kullanacağı parametreleri seçecektir. Öncelikle $E_{a,b}(\mathbb{F}_p)$ ve $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ burgu eğrileri belirleyecek olan $a, b \in \mathbb{F}_p$, $\beta \in \mathbb{F}_p$ ve β kuadratik nanrezidü bir sayı değerlerini seçecektir. Bu değerleri rastgele seçecek ancak $r_0 = \#E_0$ ve $r_1 = \#E_1$ nokta sayılarının, asal sayı olmasını sağlayacaktır. Böylece eğrilerin grup yapıları $(r_0, 1)$ ve $(r_1, 1)$ formunda olarak eğrilerin tanımladığı gruplar, döngüsel olacak ve her iki eğri için de o eğriden rastgele seçiecek $G_0 \in E_0$ ve $G_1 \in E_1$ elemanları, (sonsuz haric) üreteç olacaktır.

Saldırgan yukarıdaki parametreleri belirledikten sonra i değeri 0 ve 1 olmak üzere rastgele $x_i \in \mathbb{Z}_{r_i}$ değerleri seçer ve bu değerleri, eğrilerin üreteçleri ile çarparak

$$Y_0 = x_0 \cdot G_0 \quad \textbf{ve} \quad Y_1 = x_1 \cdot G_1$$

açık anahtarlarını elde eder.

Bütün parametreleri belirledikten sonra saldırgan $a, b, \beta, p, r_0, r_1, G_0, G_1, Y_0, Y_1$ değerlerini cihaza sabit değerler olacak şekilde yerleştirir. x_0, x_1 gizli anahtar değerleri ise gizli anahtarı ele geçirmek için kullanmak üzere saklar.

5.2.2 Adım 2: Anahtar değişimi

Bir önceki adımda (Y_0, Y_1) açık anahtarlarıyla Ayşe (uygulamada Ayşe'den bağımsız olarak cihaz) bu adımda gizli anahtar değişimi için gizli ve açık birer anahtar değeri üretecektir. Bunlardan açık olanı Bora'ya gönderecek (yayınlanacak açık parametrelerin içine saklayacak şekilde uygulamaya göre değişecek bir yöntemle) ve bir sonraki adımda Bora gizli değeri hesaplayacaktır. Gizli değer cihaz ile Bora'nın paylaştığı gizli ortak anahtar olacak ve cihaz Ayşe'nin mahrem bilgilerini bu ortak anahtar ile belirleyecektir.

Algoritmanın ilk adımında kullanılacak *EgriSeç* fonksiyonu, eğrilerin nokta sayısı olasılığında iki eğriden birini seçen bir fonksiyondur. Fonksiyon $r_0/2p+2$ olasılıkla 0, $r_1/2p+2$ olasılıkla 1 çıktısını vererek belirtilen olasılıklarda E_0 veya E_1 eğrilerinden birini seçecektir.

Aşağıdaki AnahtarçiftiüretecAnahtarçiftiüretec ve sonraki bölümde ele alınacak OrtakAnahBulOrtakAnahBul algoritmalarında kullanılacak olan, EncodeEncode, DecodeDecode fonksiyonları ve Kaliski'nin [32] çalışmasında sunduğu ECPRBGECPRBG Bölüm Ek.A'da yer almaktadır.

AnahhtarCiftiUretec(Y_0, Y_1) [13]
Girdi: $T_{a,b,\beta}(\mathbb{F}_p)$: $p \in \mathbb{Z}$ asal, $a, b \in \mathbb{Z}_p$, $\beta \in \mathbb{Z}_p$ kuadratik nanrezidü (Burgu eğri çifti parametreleri).
$r_0 = \#E_0$, $r_1 = \#E_1$: Eğrilerin nokta sayıları (asal olacak şekilde eğriler belirlenecek)
$G_0 \in_R E_0$, $G_1 \in_R E_1$: Üreteç elemanlar
$(Y_0, Y_1) = (x_0 \cdot G_0, x_1 \cdot G_1) \in (E_0, E_1)$: Saldırgan açık anahtarları
Çıktı:$m_A \in \{0,1\}^{|p|+1}$: Açık anahtar , $m_k \in \{0,1\}^j$: Gizli anahtar (j: bit uzunluğu belirleyecek herhangi tam sayı)

1. $u =$ EğriSeç(1^k) ve $v =$ EğriSeç(1^k) hesapla
2. $u = 0$ ise:
   $w \in_R \{0, 1, \ldots, r-1\}$ seç ve $m_A =$ Encode$(T_{a,b,\beta}(\mathbb{F}_p), wG_0)$
3. $u = 1$ ise:
   $w \in_R \{0, 1, \ldots, r'-1\}$ seç ve $m_A =$ Encode$(T_{a,b,\beta}(\mathbb{F}_p), wG_1)$
4. $u \neq v$ ise:
   $v = 0$ ise $z \in_R \{0, 1, \ldots, r-1\}$ seç ve $P = z \cdot G_0$
   $v = 1$ ise $z \in_R \{0, 1, \ldots, r-1\}$ seç ve $P = z \cdot G_1$

5. $u = v$ ise:
   $v = 0$ ise $P = w \cdot Y_0$
   $v = 1$ ise $P = w \cdot Y_1$
6. $(m_k, P_l, c_l) = \mathsf{ECPRBG}(T_{a,b,\beta}(\mathbb{F}_p), G_0, G_1, P, l)$
7. (m_A, m_K) çıkart

ŞEKİL 5.3: Gizli Anahtar Değişimi Anahtar Çifti Üretim Algoritması (Cihaz)

5.2.3 Adım 3: Geri Kazanım

Bu adımda atak barındıran sistemin yayınladığı değerlerden bir şekilde (uygulanan sisteme göre değişecektir) m_A anahtar değişimi parametresini elde eden saldırganın gizli değeri nasıl elde edeceği ele alınacaktır. Bora, Ayşe'den aldığı m_A değerini aşağıdaki algoritmaya sokarak ortak anahtarları m_k değerini elde edecektir.

$\mathsf{OrtakAnahBul}(m_A, x_0, x_1)$: [13]
Girdi: $T_{a,b,\beta}(\mathbb{F}_p)$: $p \in \mathbb{Z}$ asal, $a, b \in \mathbb{Z}_p$, $\beta \in \mathbb{Z}_p$ kuadratik nanrezidü (Burgu eğri çifti parametreleri).
$r_0 = \#E_0$, $r_1 = \#E_1$: Eğrilerin nokta sayıları
$G_0 \in E_0$, $G_1 \in E_1$: Üreteç elemanlar
$x_0 \in [0, r_0 - 1]$ ve $x_1 \in [0, r_1 - 1]$: Saldırgan gizli anahtarları
$m_A \in \{0,1\}^{|p|+1}$
Çıktı:$m_k \in \{0,1\}^j$: Ortak anahtar (j: bit uzunluğu belirleyecek tam sayı)

1. $U = Decode(T_{a,b,\beta}(\mathbb{F}_p), m_A)$ hesapla
2. $U \in E_{a,b}(\mathbb{F}_p)$ ise $P = x_0 \cdot U$
3. $U \in E_{a\beta^2, b\beta^3}(\mathbb{F}_p)$ ise $P = x' \cdot U$
4. $(m_k, P) = ECPRBG(T_{a,b,\beta}(\mathbb{F}_p), G_0, G_1, U)$ hesapla ve m_k çıkart

ŞEKİL 5.4: Gizli Anahtar Değişimi Ortak Anahtar Bulma Algoritması

5.3 Eliptik Eğrilerin Kullanışdığı Diğer Ataklar

Adam Young ve Moti Yung; 2006, 2007 ve 2010 yıllarında eliptik eğrilerin kullanıldığı kleptografik ataklar sunmuşlardır. Bu çalışmada neden 2007 yılındakini ele aldığımız ve diğer ataklardan kısaca bu bölümde ele alalım.

2006:
"A Space Efficient Backdoor in RSA and Its Applications" başlığıyla yayınlanan çalışmada, eğrilerin tanımlandığı cismi belirleyen p asalı yerine 2^m ile tanımlı bitsel eğrileri kullanmışlardır. Yazarların 2010 yılındaki çalışmalarında, [12] için yaptıkları eleştiride, 2006 yılındaki atakta anahtar üretimi esnasında kriptografik özet fonksiyonları kullanılmakta olduğunu ve özet fonksiyonun güvenli olup olmamasının atağın güvenliğini etkileyeceğini belirtmişlerdir.

2007:
"Space-Efficient Kleptography Without Random Oracles" başlığıyla yayınlanan çalışmada bu bölümde ele aldığımız çalışmadır. Atakta p asal olmak üzere; $\mathbb{F}_p$ cisminde tanımlı bir burgu eğri çifti kullanılmaktadır. Bu atak hakkında yazarların [14] daki eleştirisine göre; saldırgan 1/2'ye çok yakın bir olasılıkla anahtarı ele geçirmede başarısızlık yaşamaktadır. Bunun sebebi AnahtarÇiftiÜretec fonksiyonunda seçilen eğrilerin farklı olması durumunda, aynı fonksiyonun 4. adımında rastgele bir değerle anahtar değişim parametresi üretilmesidir. Bu adım yazarların güvenlik ispatı için koydukları bir adımdır. Biz atağın implementasyonu esnasında bu adımı kaldırarak, atağı gerçekledik.

2010:
"Kleptography from Standard Assumptions and Applications" başlığıyla yayınladıkları çalışmada yazarlar, diğer ataklar hakkında yukarıda bahsedilen eleştirileri yapmışlar ve bu sorunları çözdüklerini iddia etmişlerdir [14]. Bizim bu atağı tercih etmememizin sebebi ise; bir önceki bölümdeki AnahtarÇiftiÜretec algoritmasında saldırganın açık anahtarı olarak kullandığımız (Y_0, Y_1) değerleri yerine, [14]'da M sistemin üretmesini istediğimiz gizli değerin bit uzunluğunu göstermek üzere $2M$ tane açık anahtar noktası kullanılıyor olmasıdır. Sonuç olarak güvenlik ispatından vazgeçerek, yüzlerce bit uzunluğunca bir sayı adedinde nokta belirlemekten kaçınmış olduk.

5.4 Atağın RSA Şifreleme Sistemine Uygulanması

Bir önceki bölümde sıraladığımız, eliptik eğrileri kullanan kleptografik atakların hepsinde yazarlar, atağın RSA şifreleme sistemine uygulanabileceğinden bahsetmişlerdir. 2006 yılındaki çalışmada RSA şifreleme ve imza için ayrı ayrı iki uygulama senaryosunu

algoritmalarıyla verirken, bu çalışmada incelediğimiz atağın olduğu makalede RSA'e uygulanabileceğinden sadece bahsetmişlerdir. 2010 yılındaki çalışmada ise, algoritmaları vermeden atağın RSA'e nasıl uygulanabileceği hakkında fikirlerini belirtmişlerdir. Bu bölümde yazarların 2006 yılında verdikleri uygulama algoritmasına benzeyecek şekilde, atağı Bölüm 3.4'te yer alan *"Gizli Asal Çarpan"* atağıyla birleştirerek incelenecektir.

KleptoAnahtarÜreteç(k, e):
Girdi: k: asal bit uzunluğu, e açık anahtar kuvvet değeri
Çıktı: (e, n, d) RSA anahtarları
Gömülü Değerler: $T_{a,b,\beta}(\mathbb{F}_p)$: $p \in \mathbb{Z}$ asal, $a, b \in \mathbb{Z}_p$, $\beta \in \mathbb{Z}_p$ kuadratik nanrezidü. (Burgu eğri çifti parametreleri)
$r_0 = \#E_0$, $r_1 = \#E_1$: Eğrilerin nokta sayıları (asal olacak şekilde eğriler belirlenecek)
$G_0 \in_R E_0$, $G_1 \in_R E_1$: Üreteç elemanlar
$(Y_0, Y_1) = (x_0 G_0, x_1 G_1) \in (E_0, E_1)$: Saldırgan açık anahtarları; $K = |P| + 1$

1. $m_A, m_K = \text{AnahtarÇiftiÜreteç}(Y_0, Y_1)$
2. p asal ve $(e, p-1) = 1$ oluncaya kadar:
   $rast \in_R \{0,1\}^{k-K}$ seç
   $p = m_K || rast$
   ($|p| = k$ olacak şekilde m_K'nın yanına rastgele bit dizisi ekleniyor)
3. $q' \in_R \{0,1\}^k$ tek tamsayısını seç ve $n' = pq'$ hesapla
4. $n \leftarrow n'\rceil^{k/4} \;||\; m_A \;||\; n'\lfloor_{|n'|-(k+K)}$
   (n' 'nün ortasındaki parça çıkarılıp yerine m_A konuluyor)
5. $q \leftarrow \lfloor n/p \rfloor$
6. q çift ise:
   $q = q + 1$
7. $(e, q-1) > 1$ veya q asal değil iken:
   $m \in \{0,1\}^{k/4}$ ve çift m seç.
   $q \leftarrow q \oplus m$
8. $n \leftarrow pq$ ve $d \leftarrow e^{-1} \bmod \varphi(n)$ hesapla
9. (n, e) açık anahtarlar, d gizli anahtarları çıkart

ŞEKİL 5.5: Klepto RSA Anahtar Üretim Algoritması

Yayınlanan n mod değerini elde eden saldırgan, aşağıdaki algoritma ile p ve q asallarına ulaşabilecektir.

AnahatarEleGeçirme(n, β, μ):
Girdi: (n, e) açık anahtarlar
Çıktı: d gizli anahtar

1. $m_A = n\rceil^{k+K}\lfloor_K$ hesapla $(p_0 \tilde{=} p\rceil^{k/2})$
2. $m_K = \mathsf{OrtakAnahBul}(m_A, x_0, x_1)$
3. $p, q = \mathsf{Coppersmith}(n, m_A)$
4. $\varphi(n) = (p-1)(q-1)$
5. $d = e^{-1} \bmod \varphi(n)$ çıkart

ŞEKİL 5.6: Anahtar Ele Geçirme Algoritması

5.4.1 Atağın Simulasyonu ve Analizi

Bu bölümde standart ve klepto algoritmaların gerçeklenmesi ve karşılaştırmalı analizlerini ele alacağız. Bu atakta, diğer RSA ataklarında olduğu gibi üç ayrı anahtar boyunda analiz yapmak yerine sadece 256 ve 512 bit asallarla üretilen 512 ve 1024 bit RSA açık anahtarları için analizler yapılmıştır. Bunun sebebi [13] çalışmasında verilen burgu çifti parametrelerini direk kullanılmasıdır. Bu parametrelerle üretilen anahtar değerlerinde 128 bit asal ile oluşturulacak açık anahtarın ele geçirilebilmesi algoritmaya uygun değildir. Buna uygun parametreler belirlemek yerine, direk verilen parametreleri kullanmayı tercih ettik. Kullanılan parametreler aşağıda verilmiştir.

Atak için üretilen 150 adet anahtar değeri ile bu analizler yapılmıştır ve rastgele bir mesajı şifreleme/şifre çözme başarıları ve açık anahtardan, gizli anahtarı ele geçirebilme başarı oranları %100 olarak belirlenmiştir.

Algoritmaların gerçeklendiği bilgisayarın fiziksel durumu, işletim sistemi ve gerçeklendiği platform gibi detaylar Bölüm 1.4.1'da yer almaktadır.

$p = 7FFFFFFFFFFFFFFFFFFFFFFFFFFFFEB827AAD8FF16901B27758B57A11F$
$a = 66B8D3AFB14D309911554443EAF593E6CDC0431376AD682FE0EDF029$
$b = 3AC5CF725D8207054CC3BEC8D0CEEB2D569B03D467F21133DA080DE0$
$\beta = 5D0E74ABC6D516767E80F78C50A4D8BF8C0854D247BFFBFAA4837582$

ŞEKİL 5.7: Atak Algoritmasında Kullanılan Burgu Çifti Parametreleri

5.4.1.1 Çalışma Zamanı

Simule ettiğimiz atak için üretilen 150 adet anahtar değerinin ve Bölüm 3.1'de standart RSA anahtarları ile yapılan analizlerle karşılaştırmalı olarak, anahtar üretim zamanları, Tablo 5.1'de verilmiştir.

TABLO 5.1: Çalışma Zamanı:
150 adet anahtarın üretim zamanı ortalamaları (sn.)

asal bit uzunluğu :k	RSA	YY07
256	1,01	14,17
512	27,07	67,29

5.4.1.2 Açık Anahtarın Dağılımı

Atak barındıran algoritmayla ürettiğimiz anahtarlar için, n açık anahtar değerinin, büyüklüğüne göre olasılık dağılımı, önceki atakta olduğu gibi standart RSA anahtarları ile karşılaştırmalı olarak Tablo 5.2'de verilmiştir. Önceki bölümlerde ele aldığımız atakların aksime, bu atakda standart RSA anahtarları ile dağılım yüzdeleri uyuşmaktadır. Bu da atağın anahtarların dağılımını ölçerek tespit edilmesini engelleyecektir.

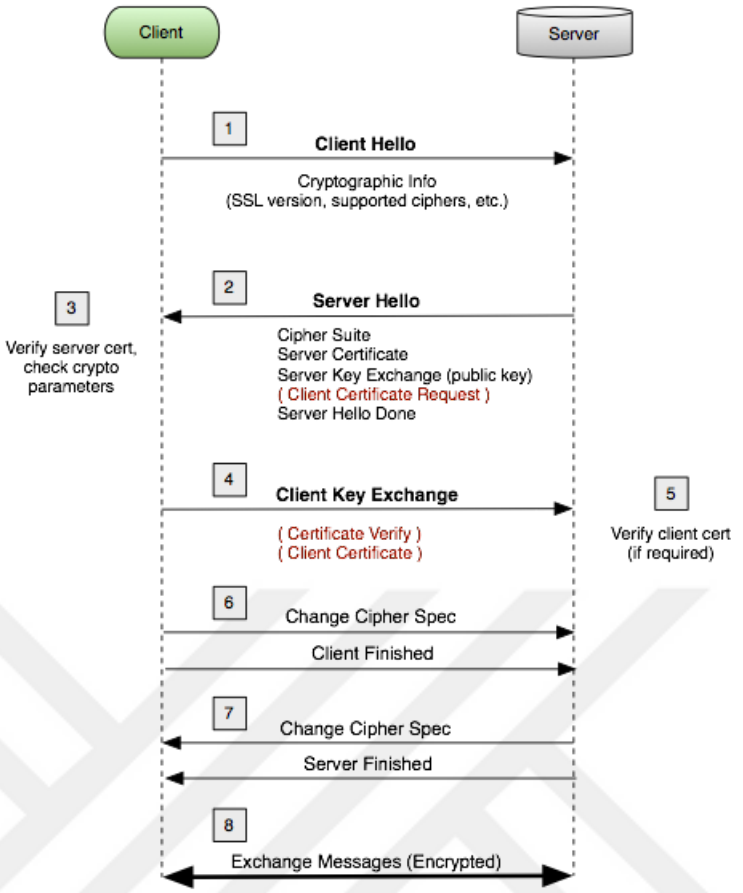
TABLO 5.2: Açık Anahtar Dağılım Tablosu:
n açık anahtarı dağılım yüzdeleri

	RSA			YY07		
asal.uzunluğu k	2k-1 bit	2k bit "10" ile	2k bit "11" ile	2k-1 bit	2k bit "10" ile	2k bit "11" ile
256	34	51	14	32	48	18
512	38	46	15	37	46	16

Sonuç olarak atak; açık anahtar dağılımında ayırt edilemezliği sağlamaktadır ancak teste tabi tutulduğu zaman, çalışma zamanı atak barındırıyor olduğu şüphesini doğurabilecektir. Bölüm 1.1'de çalışma zamanı sorununun Dual_EC_PRBG rastgele bit üretecinde de olduğunu vurgulayan çalışmalar belirtilmişti.

5.5 Atağın SSL Protokolüne Uygulanması

SSL/TLS kullanıcı ve sunucu arasında güvenli iletişimi sağlamak amamcıyla kullanılan kriptografik bir protokoldür. SSL el-sıkışma protokolüyle başlar, el-sıkışma tamamlandıktan sonra kullanıcı ve sunucu güvenli haberleşmek için kullanacakları kanalı kurmuş olurlar. Protokol adımları Şekil 5.8'de yer almaktadır.



ŞEKİL 5.8: SSL/TLS Protokolü

Protokolün 1. Adımında *Client Hello* mesajının içinde, kullanıcının rastgele seçerek sunucuya gönderdiği *Hello Nonce* değeri yer alır. Açık halde giden bu değer, trafiği dinleyen saldırgan tarafından yakalanabilir. Sonraki adımda sunucunun, kullanıcıya gönderdiği *Server Hello* mesajının içinde sunucunun rastgele belirlediği *Hello Nonce* değeri bulunur ve saldırgan trafiği dinleyerek bu değeri de elde edebilir. Sonraki adımda seçilen şifreleme yöntemine göre iki durum söz konusudur. Bunlar;

1. RSA: İstemci ürettiği ve sunucunun açık anahtarıyla şifrelediği preMS değerini sunucuya gmnderir.
2. DH: İstemci ve sunucu arasında bir Diffie-Hellman anahtar değişimi protokolü koşulur. Uzlaşılan anahtar preMS olarak kullanılır.

Sonrasında simetrik şifreleme için kullanılacak ortak anahtar kullanıcı ve sunucu *Hello Nonce* değerleri ve preMS anahtarı kullanılarak belirlenir. Sonuç olarak ortak anahtarı ele geçirmek isteyen saldırganın trafiği pasif olarak dinleyip Nonce değerlerini ve preMS anahtarını ele geçirmesi yeterli olacaktır.

[13]'de atağın SSL protokolüne uygulanması senaryosu özet olarak aşağıdaki gibidir. Atak barındıran sistemin sahibi kullanıcı Bölüm 5.2.2'deki *AnahtarÇiftiÜretme* algoritmasıyla $m_A, m_K = AnahtarÇiftiÜreteç(Y_0, Y_1)$ değerlerini belirler ve bu değerleri $HelloNonce := m_A$ ve $preMS := m_K$ olarak atar.

Pasif olarak hattı dinleyen saldırgan, kullanıcı ve sunucu'nun $HelloNonce$ değerini m_A değerine atayarak $m_K = OrtakAnahBul(m_A, x_0, X_1)$ algoritmasıyla m_K preMS anahtarını elde edecektir. Elde ettiği $HelloNonce$ değerleri ve preMS anahtarı, kullanıcı ile sunucu arasındaki şifreleme için kullanılacak olan ortak anahtarı elde etmek için yeterli olacaktır.

Sonuç olarak SSL/TLS protokolünün kapalı kutu olarak çalıştığı uygulamalarda arka kapı olup olmadığını anlayabilmek (çalışan kodları görmeden) imkansızdır diyebiliriz.

Bölüm 6

Sonuç

Bu çalışmada *"Kleptografi"* çalışma alanı altında, sunulmuş belli başlı kriptografik sistemler için arka kapı çalışmalarını ve bu çalışmaların uygulamalarını analizleri ile beraber ele aldık. Yapılan analizlerde, bir kripto sistemin tespit edilemeyecek şekilde arka kapı barındırmasının mümkün olabildiğini gördük. Bu nedenle, satın alınacak ve kritik altyapılarda kullanılacak kriptografik sistemlerin yazılım kodlarının incelenmesi gerekmektedir. Aksi taktirde tam güvenlik sağlandığı iddia edilemez.

Sonuç olarak çalışan kodların görülemediği kripto sistemlerin kritik alanlarda kullanılmasının hem milli, hem ticari, hem de şahsi güvenlik ve mahremiyet adına riskli olabileceğini söyleyebiliriz.

Ek A

Burgu Eğriler ve Kaliski'nin Rastgele Bit Üreteci

Bölüm 5'te eliptik eğrilerin kullanıldığı ve rastgele bit üreteci gibi çalışarak saldırgana mahrem bilgiyi sızdırabilecek atak mekanizması ele alınmıştı. Aynı bölümde eliptik eğrilerin kullanıldığı böyle bir atağın çıktılarının dağılımını, düzgün dağılıma sahip hale geirebilmek için burgu eğrilerin kullanılması gerektiğinden de bahsedilmişti. Bu bölümde burgu eğriler ve bu eğrilerin rastgele bit üretmek için nasıl kullanılabileceği ele alınacaktır. Kaliski'nin [32] çalışmasında sunduğu rastgele bit üreteci Bölüm 5'te incelenen [13] atağında direk olarak ve yine aynı bölümde bahsettiğimiz [14] çalışmasındaki atakta dolaylı olarak kullanılmıştır.

A.1 Burgu Eğriler

Bölüm 5'te incelenen kleptografik ataklarda tek bir eğri yerine, birbirine burgu olarak tabir edilecek iki eliptik eğri kullanılmıştı. Kaliski'nin [32] çalışmasında "*aynı sonlu cisimde tanımlı iki eliptik eğrinini, tanımlı oldukları sonlu cismin bir cebirsel genişlemesinde birbirine izomorf olabilme durumu*"olarak tanımladığı burgu eğrilerle ilgili gerekli özellikleri detaylıca incelemiş bu eğrileri kullanarak kriptografide kullanılacak bir "Rastgele Bit Üreteci" (PRBG) nasıl oluşturulabileceğini ayrıntılarıyla ele almıştır. Young ve Yung ikilisinin 2007 yılındaki [13] çalışmalarındaki kleptografik atakta da bu PRBG rastgele bit üreteci olarak kullanılarak temel bir gizli anahtar değişimi mekanizması oluşturulacak ve SSL, RSA gibi sistemlere bu gizli anahtar değişimi mekanizması yerleştirilerek ataklar elde edilecektir.

Tanım A.1 (Burgu Çifti)**.** $E_{a,b}(\mathbb{F}_p)$, k parametresinde bir eliptik eğri olsun ve β, mod p'de bir kuadratik nanrezidü olsun. $T_{a,b,\beta}(\mathbb{F}_p)$ ile gösterilecek olan k parametresindeki bir burgu çifti; $E_{a,b}(\mathbb{F}_p)$ ve $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ eğrilerinin birleşimi anlamına gelmektedir.

Kaliski, çalışmasında p asal olmak üzere, 0'dan $2p+2$'ye kadar olan tam sayılar ile $\mathbb{F}_p$ cisminde tanımlı bir burgu çifti noktaları arasında bir dönüşüm tanımlamıştır. Bunun için aşağıdaki Lemma'dan faydalanmaktadır.

Lemma A.1. *[32] $\beta \neq 0$ tam sayısı, $\mathbb{F}_p$ cisminde bir kuadratik nanrezidü ve $E_{a,b}(\mathbb{F}_p)$ bir eliptik eğri olsun. Bu durumda $y = \sqrt{x^3 + ax + b}$ olmak üzere $\forall x$ için aşağıdakiler gerçeklenir;*

1. *y kuadratik rezidü ise $(x, \pm y)$ noktaları $E_{a,b}(\mathbb{F}_p)$ eğrisi üzerindedir.*
2. *y kuadratik nanrezidü ise $(\beta x, \pm\sqrt{\beta^3} y)$ noktaları $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ eğrisi üzerindedir.*
3. *$y = 0$ ise $(x, 0)$ noktası $E_{a,b}(\mathbb{F}_p)$ eğrisi üzerinde ve $(\beta x, 0)$ noktası $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ eğrisi üzerindedir.*

Sonuc A.1. Bu Lemma'nın bir sonucu; her $x \in \mathbb{F}_p$ değeri için iki nokta ve iki tane birim eleman olmak üzere, iki eğri üzerinde toplam $2p+2$ tane eleman vardır.

Lemma A.2. *[32] $T_{a,b,\beta}(\mathbb{F}_p)$ bir burgu çifti olsun.Aşağıda verilen $X_T[T_{a,b,\beta}(\mathbb{F}_p)](P, i)$ fonksiyonu, $T_{a,b,\beta}(\mathbb{F}_p)$ burgu çifti ile $\{0, 1, \ldots, 2p+1\}$ kümesi arasında, polinom zaman hesaplanabilir, olasılıksal polinom zaman tersinebilir bir fonksiyondur.*

$$sgn : \mathbb{F}_p \to \{0,1\} = \begin{cases} 0 & (p-1)/2 \geq y > 0 \\ 1 & y > (p-1)/2 \end{cases}$$

olmak üzere;

$\mathsf{X_T[T_{a,b,\beta}(\mathbb{F}_p)](P)}$ Fonksiyonu *[32]*
Girdi: $T_{a,b,\beta}(\mathbb{F}_p)$ *Burgu parametreleri:* (a, b, β, p); $P = (x, y) \in T_{a,b,\beta}(\mathbb{F}_p)$ *noktası*
Çıktı: $m \in \{0, 1, \ldots, 2p+1\}$

1. $P \in E_{a,b}(\mathbb{F}_p)$ *ise*
   $y \neq 0$ *ise:* $m = x + sgn(y)$
   $y = 0$ *ise:* $m = 2x$
   $P = O$ *ise:* $m = 2p$

2. $P \in E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ ise:
   $y \neq 0$ ise: $m = 2x/\beta + sgn(y)$
   $y = 0$ ise: $m = 2x/\beta + 1$
   $P = O$ ise: $m = 2p + 1$

3. m çıkart ve bitir

ŞEKİL A.1: $T_{a,b,\beta}(\mathbb{F}_p)$ burgu eğri çiftindeki noktaları $m \in \{0, 1, \ldots, 2p+1\}$ tam sayılarına resmeden $\mathsf{X_T[T_{a,b,\beta}(\mathbb{F}_p)](P) = m}$ fonksiyonu [32]

Bu fonksiyonun tersi aşağıdaki gibi bulunabilir. Ters fonksiyonu burgu eğri çiftinden rastgele bir nokta seçmek için ataklarda kullanacağız. Ters fonksiyon girilen değerin tek veya çift olması durumuna göre iki parçadan oluşmaktadır ve fonksiyonda kullanılacak olan w değeri $w = x^3 + ax + b$ ile hesaplanmaktadır.

$\mathsf{X_T^{-1}[T_{a,b,\beta}(\mathbb{F}_p)](m)}$ Ters Fonksiyonu
Girdi: $T_{a,b,\beta}(\mathbb{F}_p)$ Burgu parametreleri: (a, b, β, p) , $m \in \{0, 1, \ldots, 2p+1\}$
Çıktı: $P = (x, y) \in T_{a,b,\beta}(\mathbb{F}_p)$ noktası

1. m çift ise:
   $x = m/2$ ve $w = x^3 + ax + b$
   w kuadratik rezidü ise: $P = (x, \sqrt{w})$
   w kuadratik nanrezidü ise: $P = (\beta x, \sqrt{\beta^3 w})$
   $w = 0$ ise: $P = (x, 0)$
   $x = p$ ise: $P = O_{E_{a,b}(\mathbb{F}_p)}$

2. m tek ise:
   $x = (m-1)/2$ ve $w = x^3 + ax + b$
   w kuadratik rezidü ise: $P = (x, -\sqrt{w})$
   w kuadratik nanrezidü ise: $P = (\beta x, -\sqrt{\beta^3 w})$
   $w = 0$ ise: $P = (x\beta, 0)$
   $x = p$ ise: $P = O_{E_{a\beta^2,b\beta^3}(\mathbb{F}_p)}$

3. P çıkart ve bitir.

ŞEKİL A.2: $m \in \{0, 1, \ldots, 2p+1\}$ tam sayılarını $T_{a,b,\beta}(\mathbb{F}_p)$ burgu eğri çiftindeki noktalara resmeden $\mathsf{X_T^{-1}[T_{a,b,\beta}(\mathbb{F}_p)](m)}$ fonksiyonu [32]

Aşağıdaki Encode fonksiyonu, sabitleyebileceğimiz bit uzunluğundaki sayılarla çalışabilmek için $\mathsf{X_T}$ fonksiyonunun çıktılarını düzenleyecek olan fonksiyondur. Girdi olarak bir $P = (x, y)$ noktasını alacak ve bu noktanın $\mathsf{X_T}$ fonksiyonundaki görüntüsünü sabit bir uzunluktaki bit dizisine dönüştürecektir.

Encode($T_{a,b,\beta}(\mathbb{F}_p), P$): [13]
Girdi: $T_{a,b,\beta}(\mathbb{F}_p)$ Burgu parametreleri:(a, b, β, p); $P = (x, y)$ noktası
Çıktı: $P_s \in \{0,1\}^{k+1}$

1. $t = X_T[T_{a,b,\beta}(\mathbb{F}_p)](P)$ hesapla
2. t'nin bitsel gösterimi t_s olsun
3. $|t_s| > k + 1$ ise 0^k çıkart
4. $P_s = 0^{k+1-|t_s|}||t_s$ çıkart

ŞEKİL A.3: $T_{a,b,\beta}(\mathbb{F}_p)$ burgu eğri çiftindeki noktaları sabit uzunluktaki bit dizisine resmeden Encode fonksiyonu

Bu fonksiyonun tersi aşağıdaki *Decode* fonksiyonu ile hsaplanabilir.

Decode($[T_{a,b,\beta}(\mathbb{F}_p), P_s$): [13]
Girdi: $T_{a,b,\beta}(\mathbb{F}_p)$ Burgu parametreleri:(a, b, β, p); $P_s \in \{0,1\}^{k+1}$
Çıktı: $P = (x, y)$ noktası

1. P_s'e karşılık gelen tamsayı α olsun
2. $P = X_T^{-1}[T_{a,b,\beta}(\mathbb{F}_p)](\alpha)$ çıkart

ŞEKİL A.4: $X_T[T_{a,b,\beta}(\mathbb{F}_p)](P) = m$ fonksiyonunun sonucu olan $m \in \{0, 1, \ldots, 2p+1\}$ kümesindeki tam sayıları sabit uzunluktaki bit dizisine dönüştüren fonksiiyon[13]

Sonuç.2: $T_{a,b,\beta}(\mathbb{F}_p)$ bir burgu çifti olsun. Encodefonksiyonu, $T_{a,b,\beta}(\mathbb{F}_p)$ burgu çifti ile $\{0, 1, \ldots, 2p + 1\}$ kümesindeki tam sayıların $(k + 1)$-bit gösterimleri arasında, polinom zaman hesaplanabilir, olasılıksal polinom zaman tersinebilir bir fonksiyondur. Kolayca görülebileceği gibi Encode fonksiyonunun tersi Decode fonksiyonudur.

A.2 Kaliski'nin Rastgele Bit Üreteci

Bu bölümde Kaliski'nin [32] çalışmasında sunduğu "sözde rastsal" bit üretecini ele alacağız. [13] çalışmasında bu üreteç direk olarak kullanılmış, [14] çalışmasında ise bu üretecin basitleştirilmiş bir hali kullanılmıştır. Bu yüzden bu üreteci burada kısaca görmek, atakları anlamada faydalı olacaktır.

Tanım A.2 (Burgu çifti örneklemleri)**.** k parametresindeki bir burgu çifti örneklem kümesi şunlardan oluşur:

- $T_{a,b,\beta}(\mathbb{F}_p)$ burgu çifti parametreleri: $p \in \{0,1\}^k$ asal sayı; $a, b \in \mathbb{Z}_p$; $\beta \in \mathbb{Z}_p$ kuadratik nanrezidü eleman.
- $E_{a,b}(\mathbb{F}_p)$ nin: n nokta sayısı, (n_1, n_2) grup yapısı
  $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ nin: n' nokta sayısı, (n'_1, n'_2) grup yapısı
- $E_{a,b}(\mathbb{F}_p)$ eğrisinin, (G_1, G_2) üreteçleri
- $E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ eğrisinin $G'_1, G'_2)$ üreteçleri
- Burgu çiftinde bir P noktası

Bir burgu çifti örneklem kümesi, $\langle T_{a,b,\beta}(\mathbb{F}_p), (G_1, G_2), (G'_1, G'_2), P\rangle$ ile temsil edilecek.

Tanım A.3 (ECPRBG)**.** n ve (n_1, n_2) sırasıyla E_0 eğrisinin nokta sayısını ve grup yapısını; n' ve (n'_1, n'_2) de sırasıyla E_1 eğrisinin nokta sayısını ve grup yapısını göstersin. ECPRBG bit üreteci; burgu çifti örneklemleri üzerinde ve j üretilmek istenen bit uzunluğunu temsil etmek üzere;

$$v(x_{j-1}) \dots v(x_0)$$

dizisini üreten bir fonksiyondur. Burada x_0; k parametresinde düzgün rastgele seçilmiş bir örneklem ve $x_{i+1} = f(x_i)$ ile belirlenecektir. f fonksiyonu:

$$f(\langle T_{a,b,\beta}(\mathbb{F}_p), (G_1, G_2), (G'_1, G'_2), P_i\rangle) = \langle T_{a,b,\beta}(\mathbb{F}_p), (G_1, G_2), (G'_1, G'_2), P_{i+1}\rangle$$

şeklinde tanımlanan f fonksiyonunda P_{i+1} noktası şöyle hesaplanacaktır;

$$P_{i+1} = \begin{cases} (x \bmod n_1)G_1 + \lceil x/n_1 \rceil G_2 & 0 < x0n \text{ ise} \\ ((x-n) \bmod n'_1)G'_1 + \lceil (x-n)/n'_1 \rceil G'_2 & n \le x < 2p+2 \text{ ise} \end{cases}$$

Bu denklemde x değeri $X = X_T[T_{a,b,\beta}(\mathbb{F}_p)](P)$ ile belirlenir ve son olarak $v(x)$ fonksiyonu;

$$v(\langle T_{a,b,\beta}(\mathbb{F}_p), (G_1, G_2), (G'_1, G'_2), P_i\rangle) = \begin{cases} sgn(x, n_1) & 0 \le x < n \text{ ise} \\ sgn(x-n, n'_1) & n \le x < 2p+2 \text{ ise} \end{cases}$$

PRBG fonksiyonu rastgele bit üreteci gibi çıktılar verecektir ancak girdi olarak kullanılan parametrelerin bilinmesi halinde aynı çıktılar sonradan da elde edilebilecektir. Bu fonksiyonu aşağıdaki algoritmada okuması daha kolay olacak şekilde bulunabilir.

ECPRBG(k) : [32]

Girdi: $T_{a,b,\beta}(\mathbb{F}_p)$: $p \in \mathbb{Z}$ asal, $a, b \in \mathbb{Z}_p$, $\beta \in \mathbb{Z}_p$ kuadratik nanrezidü. (Burgu eğri çifti parametreleri)

$E_{a,b}(\mathbb{F}_p)$ nin: n nokta sayısı, (n_1, n_2) grup yapısı, (G_1, G_2) üreteçleri

$E_{a\beta^2,b\beta^3}(\mathbb{F}_p)$ nin: n' nokta sayısı, (n'_1, n'_2) grup yapısı, (G_1, G_2) üreteçleri

j: Üretilmek istenen bit uzunluğu.

Çıktı: $m_k \in \{0,1\}^j$: Sözderassal bit dizisi

1. $k_0 \in_R \{0, \cdots, 2p+2\}$ seç
2. $P_0 = X_T^{-1}[T_{a,b,\beta}(\mathbb{F}_p)](k_0)$
3. $i = 0$'dan $j-1$'e kadar:
   $x_i = X_T[T_{a,b,\beta}(\mathbb{F}_p)](P_i)$
   $0 \leq x_i < n$ ise:
   $v(x_i) = sgn(x_i, n_1)$
   $P_{i+1} = (x \bmod n_1) \cdot G_1 + \lfloor x/n_1 \rfloor \cdot G_2$
   $n < x_i \leq 2p+2$ ise:
   $v(x_i) = sgn(x_i - n, n'_1)$
   $P_{i+1} = ((x-n) \bmod n'_1) \cdot G'_1 + \lceil (x-n)/n'_1 \rceil G'_2$
4. $v(x_{j-1})|| \cdots ||v(x,0)$ çıkart

ŞEKİL A.5: Eliptik Eğri Sözde Rassal Bit Üreteci

Kaynaklar

[1] A. Young and M. Yung. The dark side of "black-box" cryptography or: Should we trust capstone? In *Advances in Cryptology—CRYPTO'96*, pages 89–103. Springer, 1996.

[2] G. J Simmons. The subliminal channel and digital signatures. In *Advances in Cryptology*, pages 364–378. Springer, 1984.

[3] R. L Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[4] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in cryptology*, pages 10–18. Springer, 1984.

[5] PUB FIPS. 186-2. Digital Signature Standard (DSS). *National Institute of Standards and Technology (NIST)*, 2000.

[6] B C. Neuman and T. Ts' O. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, 1994.

[7] A. Young and M. Yung. Kleptography: Using cryptography against cryptography. In *Advances in Cryptology—Eurocrypt'97*, pages 62–74. Springer, 1997.

[8] W. Diffie and M. E Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[9] C. Crépeau and A. Slakmon. Simple backdoors for RSA key generation. In *Topics in Cryptology—CT-RSA 2003*, pages 403–416. Springer, 2003.

[10] D. Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In *Advances in cryptology—EUROCRYPT'96*, pages 178–189. Springer, 1996.

[11] A. Young and M. Yung. *Malicious cryptography: Exposing cryptovirology*. John Wiley & Sons, 2004.

[12] A. Young and M. Yung. A space efficient backdoor in RSA and its applications. In *Selected Areas in Cryptography*, pages 128–143. Springer, 2006.

[13] A. L Young and M. Yung. Space-efficient kleptography without random oracles. In *Information Hiding*, pages 112–129. Springer, 2007.

[14] A. Young and M. Yung. Kleptography from standard assumptions and applications. In *Security and Cryptography for Networks*, pages 271–290. Springer, 2010.

[15] Z. Golebiewski, M. Kutyłowski, and F. Zagórski. Stealing secrets with ssl/tls and ssh–kleptographic attacks. In *Cryptology and Network Security*, pages 191–202. Springer, 2006.

[16] E. J. Goh, D. Boneh, B. Pinkas, and P. Golle. The design and implementation of protocol-based hidden key recovery. In *Information Security*, pages 165–179. Springer, 2003.

[17] M. Gogolewski, M. Klonowski, P. Kubiak, M. Kutyłowski, A. Lauks, and F. Zagórski. Kleptographic attacks on e-voting schemes. In *Emerging Trends in Information and Communication Security*, pages 494–508. Springer, 2006.

[18] M. Gogolewski, M. Gomu^{a}kiewicz, J Kubiak, and M. Lauks. Kleptographic attaks on e-auction schemes. *Tatra Mt. Math. Publ*, 41(47):47–64, 2008.

[19] N. Perlroth, J. Larson, and S. Shane. NSA able to foil basic safeguards of privacy on web. *The New York Times*, 5, 2013.

[20] J. Ball, J. Borger, and G. Greenwald. Revealed: how US and UK spy agencies defeat internet privacy and security. *The Guardian*, 6, 2013.

[21] E. B Barker and J. M. Kelsey. *Recommendation for random number generation using deterministic random bit generators (revised)*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory, 2007.

[22] S. Checkoway, R. Niederhagen, A. Everspaugh, M. Green, T. Lange, T. Ristenpart, D. J Bernstein, J. Maskiewicz, H. Shacham, and M. Fredrikson. On the practical exploitability of dual ec in tls implementations. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 319–335, 2014.

[23] Kristian G. Dual-EC-PRBG Comments. `http://www.math.ntnu.no/~kristiag/drafts/dual-ec-drbg-comments.pdf`, 2006. Son Erişim: Şubat 2016.

[24] Dan S. and Niels F. On the possibility of a back door in the nist sp800-90 dual ec prng. crypto 2007 rump session„ 2007. Son Erişim: Şubat 2016.

[25] B. Schoenmakers and A. Sidorenko. Cryptanalysis of the dual elliptic curve pseudorandom generator. *IACR Cryptology ePrint Archive*, 2006:190, 2006.

[26] I. Mironov and N. Stephens-Davidowitz. Cryptographic reverse firewalls. In *Advances in Cryptology-EUROCRYPT 2015*, pages 657–686. Springer, 2015.

[27] A.r Russell, Q. Tang, M. Yung, and H. S. Zhou. Cliptography: Clipping the power of kleptographic attacks. Technical report, Cryptology ePrint Archive, Report 2015/695, 2015. http://eprint. iacr. org, 2015.

[28] K. Ruohonen. Mathematical cryptology. *Lecture Notes*, 2010.

[29] J. Hoffstein, J. Pipher, J. H. Silverman, and J. H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[30] D. Hankerson, A. J Menezes, and S. Vanstone. *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[31] R. Schoof. Counting points on elliptic curves over finite fields. *Journal de théorie des nombres de Bordeaux*, 7(1):219–254, 1995.

[32] B. S. Kaliski. *Elliptic curves and cryptography: A pseudorandom bit generator and other tools*. PhD thesis, Massachusetts Institute of Technology, 1988.

[33] I. F Blake, G. Seroussi, and N. Smart. *Elliptic curves in cryptography*, volume 265. Cambridge university press, 1999.

[34] V. S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology—CRYPTO'85 Proceedings*, pages 417–426. Springer, 1985.

[35] D. Boneh et al. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2):203–213, 1999.

[36] D. Pointcheval and J. Stern. Security proofs for signature schemes. In *Advances in Cryptology—EUROCRYPT'96*, pages 387–398. Springer, 1996.

[37] C.P. Schnorr. Efficient identification and signatures for smart cards. In *Advances in cryptology—CRYPTO'89 proceedings*, pages 239–252. Springer, 1989.

[38] C.P. Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.