Yan Kanal ve Hata Yaptırma Atak Cihazlarının, Hazırlanan Test Yazılımı ile Yönetilmesi ve Alınan Ölçümlerin Analiz Edilmesi

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

Nihan ÇAYDAŞ
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:

Prof. Dr. Ensar Gül ..... 
(Tez Danışmanı)

Dr. Ercan Ölçer ..... 
(Tez Eş-danışmanı)

Prof. Dr. Nizamettin Aydın ..... 

Dr. Öğretim Üyesi İhsan Çiçek ..... 

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

ONAY TARİHİ: 26.08.2019

MÜHÜR/İMZA: 

Yazarlık Beyanı

Ben, Nihan ÇAYDAŞ, başlığı, 'Yan Kanal ve Hata Yaptırma Atak Cihazlarının, Hazırlanan Test Yazılımı ile Yönetilmesi ve Alınan Ölçümlerin Analiz Edilmesi' 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 almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış 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: N. Çaydaş

Tarih: 26.08.2019

"Sadece hiçbir şey bilmediğimizi bilebiliriz. Ve bu, en yüksek düzeyde insan bilgeliğidir."

Tolstoy

Managing Side Channel and Fault Injection Attack Devices with Developed Test Software and Analyzing the Recorded Measurements

Nihan Çaydaş

Abstract

The security of these IT systems is important because the transactions carried out in IT products are critical and contain sensitive data. Solutions for problems such as storing and transmitting sensitive data securely, efficiency and sufficiency of these solutions stand out as critical elements in terms of security.

Cryptographic algorithms are developed and used in IT products to ensure the confidentiality and integrity of data. Mathematical crypto analysis tests of these algorithms are performed and their resistance to brute force attacks are analyzed. Reliable algorithms are widely used in information technology products.

In this thesis, it is prepared a test software for non-invasive hardware attacks. Test software is capable of controlling hardware attack devices during the attack and analyzing the data coming from the attack devices after the attack.

As a result of this thesis, it is aimed to contribute to the laboratories that make Common Criteria evaluations and ISO / IEC 19790 tests and provide control of multiple devices used for attacks with a single software to ensure that the devices work synchronously and correctly with each other so that the attacks can be performed correctly.

Keywords: Hardware Security, Non-invasive Attacks, Side Channel Attacks, Cryptographic Algorithms

Yan Kanal ve Hata Yaptırma Atak Cihazlarının, Hazırlanan Test Yazılımı ile Yönetilmesi ve Alınan Ölçümlerin Analiz Edilmesi

Nihan Çaydaş

Öz

Bilişim teknolojileri ürünlerinde yapılan işlemlerin kritik olması, hassas veriler içermesi sebebiyle bu sistemlerin güvenliği önem arz etmektedir. Hassas verilerin saklanması, güvenli olarak iletilmesi gibi problemlere yönelik çözümler, çözümlerin yeterliliği ve etkinliği güvenlik açısından kritik unsurlar olarak öne çıkmaktadır.

Verilerin gizliliği ve bütünlüğünün sağlanması için kriptografik algoritmalar geliştirilmekte ve bilişim teknolojileri ürünlerinde kullanılmaktadır. Bu algoritmaların matematiksel kripto analiz testleri yapılmakta ve kaba kuvvet (brute force) saldırılarına karşı dayanıklılıkları analiz edilmektedir. Güvenilir bulunan algoritmaların bilişim teknolojileri ürünlerinde yaygın kullanım alanları vardır.

Donanımsal saldırılar, kriptografik işlemler esnasında sistemlerden hassas verilerin elde edilmesi için kullanılan bir yöntemdir. Bu saldırı türünde kriptografik algoritmaların matematiksel olarak çok güçlü ve kırılamaz olmaları bir anlam ifade etmemektedir. Bu sebeple geliştirilen bilgi teknolojileri ürünlerinin güvenlikle ilgili aldıkları önlemlere bu tür ataklar için alınan önlemler de eklenmelidir. Eklenen bu önlemlerin de etkinliğinin ve yeterliliğinin değerlendirilmesi ve kontrol edilmesi gerekmektedir.

Bu tez çalışmasında bozucu olmayan donanımsal saldırılar için bir test yazılımı hazırlanmıştır. Hazırlanan test yazılımı, saldırı esnasında donanımsal saldırı cihazlarını kontrol etme, saldırı sonrasında ise saldırı cihazlarından gelen verileri analiz ederek sonuç verme yeteneğine sahiptir.

Bu tez çalışmasında hazırlanan yazılımla, Ortak Kriterler ve ISO/IEC 19790 değerlendirmeleri yapan laboratuvarlara katkı sağlanması hedeflenmektedir. Bir atak için kullanılan birden fazla cihazın tek bir yazılımla kontrolünün sağlanması, cihazların birbiri ile senkron ve doğru çalışması, atakların doğru bir şekilde yapılması amaçlanmıştır.

Anahtar Sözcükler: Donanım Güvenliği, Bozucu Olmayan Saldırılar, Yan Kanal Saldırıları, Kriptografik Algoritmalar

Bu tez çalışması aileme ithaf edilmiştir . . .

Teşekkür

Tez danışmanlarım Ercan ÖLÇER ve Ensar GÜL'e yardımları ve destekleri için teşekkür ederim.

Yüksek lisansıma katkısı ve Bilgi Güvenliği konusunda bana kattığı deneyimler için TÜBİTAK BİLGEM'e, motivasyon konusundaki teşvikleri ve yardımları için çalışma arkadaşlarıma teşekkür ederim.

Son olarak sevgili aileme, her zaman yanımda oldukları, sabırları ve teşvikleri için sonsuz teşekkürlerimi sunarım.

İçindekiler

Şekil Listesi

Tablo Listesi

Kısaltmalar

AES	**A**dvanced **E**ncryption **S**tandard
ALU	**A**rithmetic **L**ogic **U**nit
ANSI	**A**merican **N**ational **S**tandards **I**nstitute
CMOS	**C**omplemantary **M**etal **O**xide **S**emiconductor
CRT	**C**hinese **R**emainder **T**heorem
DES	**D**ata **E**ncryption **S**tandard
ECC	**E**lliptic **C**urve **C**ryptography
DPA	**D**ifferential **P**ower **A**nalysis
FIPS	**F**ederal **I**nformation **P**rocessing **S**tandards
HSM	**H**ardware **S**ecurity **M**odule
IC	**I**ntegrated **C**ircuit
IoT	**I**nternet **o**f **T**hings
ISO	**I**nternational **O**rganization for **S**tandardization
LFSR	**L**inear **F**eedback **S**hift **R**egister
PGP	**P**retty **G**ood **P**rivacy
PIN	**P**ersonal **I**dentification **N**umber
PRNG	**P**seudo **R**andom **N**umber **G**enerator
RSA	**R**ivest **S**hamir **A**dleman
SoC	**S**ystem **o**n **C**hip
SPA	**S**imple **P**ower **A**nalysis
TDES	**T**riple **D**ata **E**ncryption **S**tandard
TLS	**T**ransport **L**ayer **S**ecurity
TRNG	**T**rue **R**andom **N**umber **G**enerator
VPN	**V**irtual **P**rivate **N**etwork
YKA	**Y**an **K**anal **A**takları

Bölüm 1

Giriş

Bilgi Güvenliği kavramı, genel olarak, hassas bilgilerin güvenli olarak işlenmesi, saklanması, değişikliklere karşı korunması, iletilmesi gibi konuları ifade eder. Bu konuların sağlanabilmesi için içerisinde kriptolojiyi barındırır. Bilgi Güvenliği denince akla Siber Güvenlik kavramı gelmekte ve bu ikisi karıştırılabilmektedir. Bilgi Güvenliği iki alt kategoride incelenebilmektedir. Birincisi tesislerin fiziksel güvenliği, ikincisi ise bilgiye yetkisiz kişilerin elektronik olarak erişiminin engellenmesidir. İkinci kısma aynı zamanda Siber Güvenlik de denilmektedir. Bu açıdan bakıldığında Bilgi Güvenliği, Siber Güvenliği kapsamakta, Siber Güvenlik özelleştirilmiş bir alt kümeyi oluşturmaktadır. Bu sebeple, bu tez çalışmasında, Bilgi Güvenliği kavramı, Siber Güvenlik kavramı yerine de kullanılacaktır.

Bilgi Güvenliği, gelişen teknoloji ile birlikte her geçen gün daha önemli bir konu haline gelmektedir. Elektronik ortamlarda yaygın olarak kullanılan, akıllı kartlar, HSM'ler, kripto cüzdanları, IoT cihazları, kalp pilleri, otomobiller gibi birbirinden farklı her türlü alanda, güvenlik konusu en temel problem ve önlem alınması gereken birincil konu olarak öne çıkmaktadır. Bu durumun çözümü için konsorsiyumlar kurulmakta, her ürün ailesi için farklı çalışma grupları oluşturulmakta, bu konu için özel olarak çalışmalar yapılmakta, standartlar oluşturulmaya çalışılmaktadır. Türkiye'de de bu konu günden güne önem kazanmakta, ancak çalışmalar henüz başlangıç aşamasında olduğu için bazı konulara yeterince ilgi gösterilememektedir. Donanım güvenliği için de durum böyledir.

Günümüzde, güvenli varsayılan sistemlere yönelik ciddi saldırılar olmakta ve bunların bir kısmı da başarıya ulaşmaktadır. Başarıya ulaşan her saldırı, çeşitli maddi ve manevi

zararlara yol açmaktadır. Servis hırsızlıkları (IoT, elektronik sayaçlar), bilgiye erişim saldırıları (kimlik bilgilerinin ele geçirilmesi), kopyalama saldırıları, örnek saldırı türleri olarak verilebilir. Bu saldırıların başarıya ulaşmasının en büyük nedeni, saldırı teknolojilerinin sürekli gelişiyor olmasıdır. Saldırılardan korunabilmek için savunma kısmının da aynı hızda gelişebiliyor olması gerekmektedir.

Donanım güvenliği hızla önem kazanmakta ve bu konuya özel ilgi gösterilmektedir. Donanımda yer alan bir güvenlik açığının, yazılım seviyesinde düzeltilmesi, yazılımda alınan önlemler ile giderilmeye çalışılması mümkün değildir. Bu durum, güvenli bir sistem tasarımının donanım seviyesinde başladığının en önemli göstergesidir. Günümüzde, güvenlik kaygısı ile birlikte, tümdevre üzerindeki sistemlere (SoC - System on Chip) ve yeniden yapılandırılabilir donanıma doğru bir eğilim vardır. Güvenli tümdevrelere yönelik talep artmakta, bu tümdevreler, otomotiv enstitüsü, IoT cihazlarında, servis sağlayıcılarda, bankacılık uygulamaları ve askeri uygulamalarda, erişim kontrol mekanizmalarında, cep telefonlarında yaygın olarak kullanılmaktadır. Yaygın kullanım ile birlikte, belirli problemler ortaya çıkmıştır [1].

- Güvenli bir sistem nasıl tasarlanır? (donanım güvenliği mühendisliği)
- Korumanın değerlendirmesi nasıl yapılır? (kırılma maliyetini tahmin et)
- En iyi çözüm nasıl bulunabilir? (minimum zaman ve para)

Bu problemlerden ikincisine bu tez çalışmasında daha fazla değinileceği için bu konu ile ilgili bazı açıklamalar yapılması faydalı olacaktır. Güvenlik kaygısı arttıkça ve güvenli bir sistem tasarlama fikri ortaya çıktıkça, tasarlanan sistemlerin değerlendirilmesi ihtiyacı da hissedilmeye başlanmıştır. Bu konuda standartlaşma çalışmaları yapılmış, ürünlerin uluslararası kriterlere göre bazı gereksinimleri sağlaması ve belirli güvenlik seviyelerinde iddialarının olması sağlanmıştır. Ortak Kriterler (ISO/IEC 15408) [2] ve ISO/IEC 19790 [3] Standardı bu çalışmalardan ikisidir. Bu iki standart yaygın olarak kullanılmakta ve ürün güvenlik seviyeleri hakkında fikir vermektedir.

Ortak Kriterler Standardından önce, Amerika'da FC (Federal Criteria for Information Technology Security), Kanada'da CTCPEC (Canadian Trusted Computer Product Evaluation Criteria) ve Avrupa'da ITSEC (Information Technology Security Evaluation Criteria) standartları güvenlik değerlendirmelerinde kullanılmaktadır. 1996 yılında, bu üç

standart birleştirilerek Ortak Kriterler Standardı (Common Criteria for Information Technology Security) oluşturulmuştur. Başlangıçta beş ülke tarafından imzalanan anlaşma ile standart, bu beş ülkede geçerli olarak kabul edilmiştir. Şu anda aralarında Türkiye'nin de bulunduğu 18 sertifika üreticisi, 12 sertifika müşterisi ülkede bu standart geçerliliğini sürdürmektedir [4].

Ortak Kriterler Standardı, 7 değerlendirme güvence seviyesi (EAL - Evaluation Assurance Level) içerir [5]. Bu seviye arttıkça ürünün güvenlik seviyesi ve değerlendirme derinliği de artmaktadır. Değerlendirmeler, Ortak Kriterler Değerlendirme Metodolojisine göre gerçekleştirilir [6]. Bu tez çalışmasında bahsedilen ürünler için tavsiye edilen minimum güvence seviyesi EAL 4'tür. Akıllı kart işletim sistemleri ve uygulamaları için EAL 4 ve EAL 5 (artırılmış bileşen AVA_VAN.5 ile birlikte), tümdevreler için EAL 6'dır.

ISO/IEC 19790 Standardı, kriptografik modüller için gereksinimleri belirlemekte, bu standardın referans verdiği ISO/IEC 24759 [7] Standardı ise bu gereksinimler için test kriteri sağlamaktadır. Ürünler yazılımsal kripto modüller, donanımsal kripto modüller ve karma modüller olarak ayrılmaktadır. Standart, 4 güvenlik seviyesi barındırır. Bu standart, FIPS 140-2 [8] standardının ISO/IEC muadilidir.

Bu tez çalışmasında özellikle donanımsal saldırılara yönelik bir çalışma yapılması tercih edilmiştir. Bu tercihin nedenlerinden ilki, bu konuda yapılan çalışmalara katkıda bulunmak ikincisi ise, Ortak Kriterler ve ISO/IEC 19790 değerlendirmelerinde ihtiyaç duyulan donanımsal saldırılarda kullanılacak test yazılımlarını hazırlamaktır. Bu tez çalışması ile bu ihtiyacın karşılanması yolunda adım atılmış, hedeflenen çalışmalar gerçekleştirilmiştir.

Bu tez çalışmasının hedefi, bozucu olmayan donanımsal saldırılar için bir test yazılımı oluşturmaktır. Test yazılımının kapsamı, bu tez çalışması için, Basit - Farksal Güç Analizi, Basit - Farksal Elektromanyetik Analiz ve Glitch saldırıları ile sınırlandırılmıştır.

Bu tez, 6 bölüm olarak düzenlenmiştir. Bölüm 1, tezin giriş bölümüdür. Tez çalışması hakkında genel bilgiler içermektedir. Bölüm 2'de tez çalışmasına yön veren ve literatürde yer alan çalışmalara yer verilmiştir. Bölüm 3 Kriptoloji hakkında bilgiler içermektedir. Bu bölümde yer verilen bilgiler, hazırlanan test yazılımında doğrudan kullanılmıştır. Test yazılımı ile birlikte test edilebilecek kriptografik algoritmalar, doğrulama yöntemleri hakkında bilgiler bu bölümde verilmiştir. Bölüm 4, donanımsal saldırı yöntemleri hakkında bilgi içermektedir. Bu bölüm, hazırlanan test yazılımına entegre edilen ve

edilebilecek çalışmalarla ilgili bilgi sağlamaktadır. Bölüm 5, hazırlanan test yazılımının ve yazılımda kullanılan test cihazlarının anlatıldığı bölümdür. Bölüm 6, hazırlanan test yazılımı ile gerçekleştirilen bir saldırı sunmaktadır. Bölüm 7 ise, tez sonuçlarını içermektedir. Hazırlanan test yazılımının benzer yazılımlar ile karşılaştırılması ve yazılımın katkısı ve avantajları bu bölümde verilmiştir.

Bölüm 2

İlgili Çalışmalar

Bir yarı iletken tümdevrenin gerçekleştirdiği güvenlikle ilgili işlemlerin, giriş verilerinin ve gizli anahtarın değerlerine bağlı olarak tamamlanma süresi değişebilmektedir. Bu fikir ilk olarak 1996 yılında bilimsel literatürde yer almıştır [9]. Daha sonra bu yöntem, akıllı kart üzerinde, bir RSA imzalama uygulamasında başarılı bir şekilde uygulanmıştır [10].

Bu olayın yanında, farklı bir atak türü olarak, Kocher ve ark. [11] 1999 yılında güç analizi saldırılarının, akıllı kartlarda yer alan anahtar verilerini açığa çıkardığını gösterdiklerinde, kriptografik cihazların güvenliğine olan inanca büyük darbe vurmuşlardır. Elektromanyetik analiz saldırıları için çalışmalar da hız kazanmış ve başarılı saldırı uygulamaları gerçekleştirilmiştir [12], [13]. Yan kanal saldırılarının yanı sıra aktif bozucu olmayan saldırılar da literatürde yer almaktadır. Bu saldırıların amacı, cihaz üzerinde herhangi bir iz bırakmadan cihaza hata yaptırmaktır. Bu tür saldırılarla ilgili bir çalışma [14]'te yer almaktadır. Şablon saldırıları olarak adlandırılan, kriptografik cihazın karakteristiğini çıkarmaya, sonrasında atak uygulamaya odaklı saldırılar için de çalışmalar literatürde yer almıştır [15].

Aynı zamanda 1999 yılında bozucu saldırılar için de çalışmalar yapılmış [16], ancak pahalı ekipmanlar gerektirdiği için, başarı şansı çok yüksek olmasına rağmen sınırlı sayıda yayınlar literatürde yer almıştır [17], [18].

Yarı bozucu saldırılarla ilgili yayınlar 2002 yılından itibaren literatürde yerini almıştır [19], [20]. Yarı bozucu saldırılar için Skorobogatov tarafından kapsamlı bir yayın da hazırlanmıştır [18].

Zaman ilerledikçe saldırı türleri gelişmiş ve karmaşıklaşmıştır. Birden fazla saldırı türü birleştirilerek daha etkili ataklar uygulanmaya başlanmıştır. 2006 yılında yayınlanan makaledeki saldırı [21], bu tür saldırılara örnek olarak verilebilir.

Son on yılda uygulanan donanımsal saldırı türlerine [22], [23], [24], [25], [26] örnek olarak gösterilebilir.

İlgili çalışmalarıincelendiğinde, saldırı türlerinin her geçen gün değiştiği ve geliştiği görülebilmektedir. Bu durum, donanımsal saldırıların ve buna bağlı olarak sürekli gelişen donanım güvenliği konusunun ne kadar önemli olduğunu ortaya çıkarmaktadır.

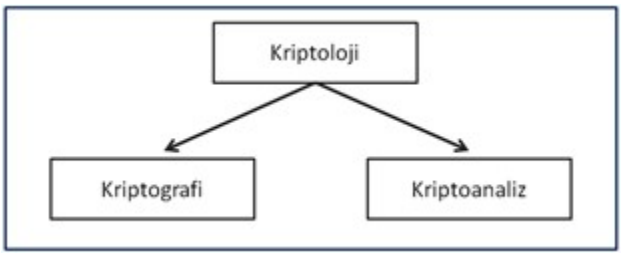
Ayrıca donanım güvenliği konusunda test yazılımları geliştiren üreticilere Riscure ve Micropross gibi firmalar örnek olarak verilebilir. Bu firmalar aynı zamanda donanımsal saldırı cihazları da üretmektedir.

Bölüm 3

Kriptoloji

3.1 Kriptoloji, Kriptografi ve Kriptoanaliz

Kriptoloji, genellikle gizli formda olan verinin, güvenli olarak iletilmesi ve depolanması ile ilgilenen bilimdir. Kriptoloji kelimesi, kryptos (gizli) ve logos'tan (kelime) türetilmiştir [27]. Genel olarak Kriptografi ve Kriptoanaliz olmak üzere ikiye ayrılmaktadır (Şekil 3.1).



Şekil 3.1: Kriptoloji, Kriptografi ve Kriptoanaliz

Kriptografi, kryptos (gizli) ve graphein (yazı) kelimelerinden türetilmiştir. Genel bir ifadeyle, bir bilginin şifrelenerek gizlenmesi ve gizli anahtarı bilen tarafından bilginin elde edilmesi konusunun araştırılmasıdır. Kriptografi başlangıçta savaş zamanlarında gizlilik sağlamakla ilgiliyken günümüzde bilgisayarlar arasındaki haberleşmenin veya bilgisayarlarda depolanan verilerin güvenli hale getirilmesi, faks, televizyon, telefon sinyallerinin şifrelenmesi, elektronik ticarette (e-ticaret) kullanıcıların kimliğinin doğrulanması ve bu işlemlerin yasal olarak kabul edilebilir kayıtlarının sağlanması gibi alanlarda da kullanılmaktadır.

Kriptoanaliz ise, kryptos (gizli) ve analyein (çözmek) kelimelerinden meydana gelmektedir. Gizli anahtara sahip olunmadan kriptografik olarak korunan bilgileri açığa çıkarma bilimidir. Kriptografinin çalışma alanı genişledikçe kripto analizin de çalışma alanı genişlemektedir. Bu iki konu yakından ilişkilidir; Bir şifreleme sistemi kurulurken güvenliğin analizi de önemli bir rol oynamaktadır [28].

3.2 Neden Şifreleme Sistemleri?

Bilişim Teknolojileri ürünlerinde yapılan işlemlerin kritik olması, hassas veriler içermesi sebebiyle bu ürünlerin güvenliği önem arz etmektedir. Hassas verilerin saklanması, güvenli olarak iletilmesi gibi problemlere yönelik çözümler; çözümlerin yeterliliği ve etkinliği kritik unsurlar olarak öne çıkmaktadır.

Bilişim Teknolojileri ürünlerindeki temel problemler aşağıdaki gibidir;

Gizlilik; Veri iletilirken, verici ve alıcı arasında gidip gelen iletilerin üçüncü bir tarafça anlaşılması istenmemektedir. Yetkisiz erişime karşı korunması gereken depolanmış veriler için de aynı durum geçerlidir.

Bütünlük; Verici ve alıcı arasında gidip gelen iletilerin üçüncü bir tarafça değişikliğe uğratılmadığının kanıtlanması gerekmektedir.

Kimlik Doğrulama; Bu özellik bir imzaya eşdeğerdir. Bir iletinin alıcısı, iletinin üçüncü bir taraftan değil gerçek vericiden geldiğine dair kanıt istemektedir. Bu kanıt, gerçek verici bu iletiyi daha sonra reddetmek istese dahi buna müsaade etmemelidir.

Reddedilemezlik; Göndericinin alıcıya belirli bir mesajı gönderdiğini ve alıcının mesajın göndericiden geldiğini doğrulayabilmesidir. Aynı zamanda göndericinin, gönderdiği mesajı inkar edememesini sağlar.

3.3 Kriptografik Algoritmalar

Yüzyıllar boyunca kripto sistemleri askeri ve diplomatik hizmetler tarafından yaygın olarak kullanılmıştır. Günümüzde bilişim sistemlerinin endüstri ve kamu tarafından

yaygın olarak kullanılması, çoğu zaman kriptografik tekniklerle verilerin özel olarak korunması ihtiyacını doğurmaktadır.

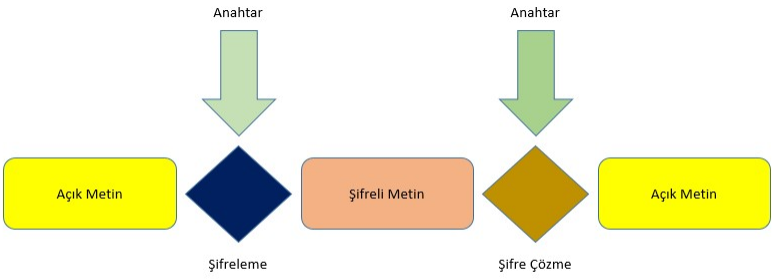
Fark edilmese dahi her gün şifreleme ile karşılaşılmaktadır. Web sitelerine yapılan bağlantıların birçoğu -https- olarak belirtilen TLS protokolü ile şifrelenmektedir. Mobil mesajlaşma uygulamaları ile gönderilen ve alınan mesajlar da şifrelenmekte ve telefonda şifreli bir klasörde saklanmaktadır. Ayrıca E-postalar OpenPGP gibi protokollerle şifrelenebilir. VPN'lerde şifreleme kullanılır ve bulutlarda saklanan tüm veriler şifrelenmektedir. Kişisel bir güvenlik önlemi olarak sabit sürücüler şifrelenebilir.

İletişim ve finans sistemlerinin büyük bir kısmı, bilgileri saldırganlardan uzak tutmak için şifreleme kullanmaktadır. Şifreleme, hızla yaygınlaşmakta olan kripto cüzdanları da güvenli hale getirmenin en önemli yönüdür. Bu gibi örnekler fazlasıyla artırılabilmektedir. Şifreleme algoritmaları bu gibi diğer birçok teknolojide kullanılmaktadır.

Aşağıdaki alt bölümlerde yaygın kullanılan algoritmalar Simetrik, Asimetrik ve Özet Alma Algoritmaları başlıkları altında anlatılmıştır.

3.3.1 Simetrik Algoritmalar

Simetrik Algoritmalar, düz metinin ve şifreli metinin aynı anahtarlar kullanılarak işleme tabii tutulduğu kriptografik işlemlerdir. Anahtarlar, gizliliği korumak için iki taraf arasında güvenli bir şekilde paylaşılmış olmalıdır. Bu problemin varlığı asimetrik algoritmalar karşısında simetrik algoritmalara dezavantaj oluşturan bir durumdur. Genel olarak dizi şifreleme ve blok şifreleme algoritmaları olarak iki alt kırınıma ayrılmaktadır. Bu tez çalışmasında yer verilecek algoritmalar blok şifreleme kırınımı altında yer almaktadır. Dizi şifreleme algoritmalarına bu çalışmada değinilmemiştir.



Şekil 3.2: Simetrik Algoritmalar

Simetrik şifreleme algoritmaları çok hızlıdır ve daha az hesaplama gücü gerektirir. Bu durum simetrik algoritmaların en büyük avantajı ve yaygın kullanılmalarının sebebidir.

3.3.1.1 3DES veya TDES (Triple DES) Algoritması

DES Algoritması 1977'de FIPS Standardı (FIPS 46-3) [29] olarak kabul edilmiştir. 2000 yılında AES Algoritması şifreleme algoritması olarak seçilmiş olsa da DES Algoritması, TDES olarak hala yaygın bir şekilde kullanılmaktadır.

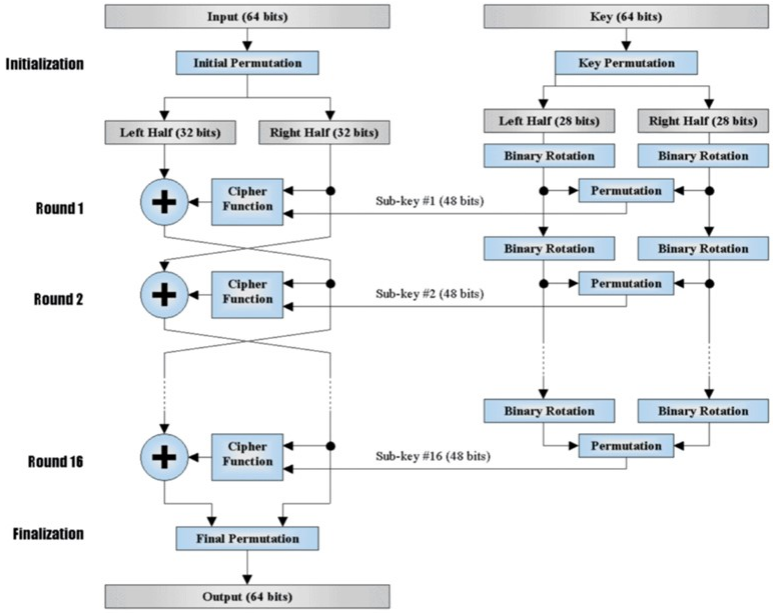
56 bitlik DES anahtarının kaba kuvvet saldırılarına karşı korunmak için yeterli olmadığı ortaya çıktığında yeni bir algoritmaya gerek kalmadan anahtar alanını genişletmenin basit bir yolu olarak TDES Algoritması seçilmiştir.

TDES Algoritması ANSI X9.52 standardında [30] modlarıyla birlikte 1998 yılında kabul edilmiştir. Modlarla ilgili açıklamalara bu tez çalışmasında yer verilmemiştir. Burada değinilecek kısım iki anahtarlı TDES ve üç anahtarlı TDES'tir. Yani 112 bit TDES ve 168 bit TDES kastedilmektedir.

TDES Algoritması, DES Algoritmasının üç defa çalıştırılmasıdır. Bu sebeple öncelikle DES Algoritması aşağıda anlatılmıştır.

DES Algoritması ile, 64 bitlik bloklar 56 bitlik bir anahtarla şifrelenmekte ve veriler permütasyon, yer değiştirme ve XOR işlemlerinden geçirilmektedir. Detayları aşağıda Şekil 3.3 [31] üzerinden anlatılmıştır.

DES Algoritmasında işlemler ikili sayı sistemi kullanılarak yapılır. Bu sebeple öncelikle giriş ve anahtar değerlerinin ikili sayı sistemine dönüştürülmesi gerekmektedir.



Şekil 3.3: DES Algoritması

İlk Permütasyon (Initial Permutation(IP)): 64 bitlik giriş verisi öncelikle ilk permütasyon olarak adlandırılan işlemden geçirilir. Bu işlem yapılırken Şekil 3.4'teki [29] değerler referans alınır.

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Şekil 3.4: DES Algoritması IP İşlemi

64 bitlik giriş verisi ikili sayı sistemine çevrilir ve IP işleminden geçirilir. Burada yapılan işlem sırasıyla ilgili bitlerin yeniden yazılmasıdır. Yani 58.bit ilk bit, 50.bit ikinci bit, 42.bit üçüncü bit, şeklinde yazılmaktadır. Son olarak ise 7.bit son biti oluşturmaktadır.Bu şekilde yeni 64 bitlik dizilim oluşturulmuş olmaktadır.

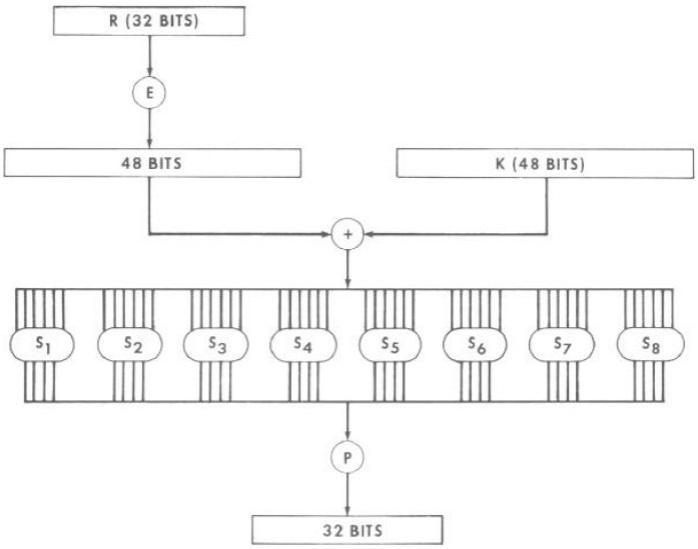
Son Permütasyon (Final Permutation (reverse IP (IP^{-1}))): Tüm işlemler tamamlandıktan sonra veri çıkış verisi olarak dışarıya verilmeden önce IP^{-1} işleminden geçirilir. Bu işlem de yine standartta yer alan Şekil 3.5 [29] referans alınarak yapılır. Yapılan işlem IP işlemi ile birebir aynıdır.

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Şekil 3.5: DES Algoritması Son IP İşlemi

Şifreleme Fonksiyonu (Cipher Function (f)): Bu aşamada anahtar fonksiyonuyla üretilen anahtar değerleri ile şifrelenecek metin işleme alınmaktadır. İçerisinde XOR, E ve S Kutusu işlemlerini barındırmaktadır (Şekil 3.6). E işlemi (Şekil 3.7) IP işlemine benzer bit değiştirme işlemidir. Bu işlemde de yine sabit bir tablo kullanılır. S Kutusu işlemi ise 48 bit giren verinin 32 bit veriye dönüştürüldüğü işlemdir. S Kutusu işlemi doğrusal olmayan bir işlemdir. Bu sebeple bu aşamadan sonra anahtar ile veri arasında bir ilişki bulunamaması beklenmektedir. Eğer bir ilişki tespit edilebiliyorsa burada bir açıklık olduğu düşünülebilir.



Şekil 3.6: DES Algoritması f Fonksiyonu

E Bit Seçme Tablosu

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Şekil 3.7: E Bit Seçme Tablosu

3.3.1.2 AES Algoritması

AES Algoritması Rijndael Algoritmasının [32] özelleştirilmesi sonucu oluşturulan FIPS 197 [33] standardı olarak yayınlanan gelişmiş bir blok şifreleme algoritmasıdır. FIPS 197 standardı, 128, 192 ve 256 bit uzunluğundaki şifreleme anahtarlarını kullanarak 128 bit veri bloklarını şifrelemeyi kapsar. Rijndael'in diğer ek bileşenleri bu standartta yer almaz. Dolayısıyla AES Algoritması diğer bileşenleri kapsamamaktadır.

AES algoritmasının işlemleri Durum (State) adı verilen iki boyutlu bir bayt dizisi üzerinde gerçekleştirilir. Bu bayt dizisi oluşturulurken aşağıdaki gibi hareket edilir:

g_0	g_4	g_8	g_{12}
g_1	g_5	g_9	g_{13}
g_2	g_6	g_{10}	g_{14}
g_3	g_7	g_{11}	g_{15}

$\Rightarrow$

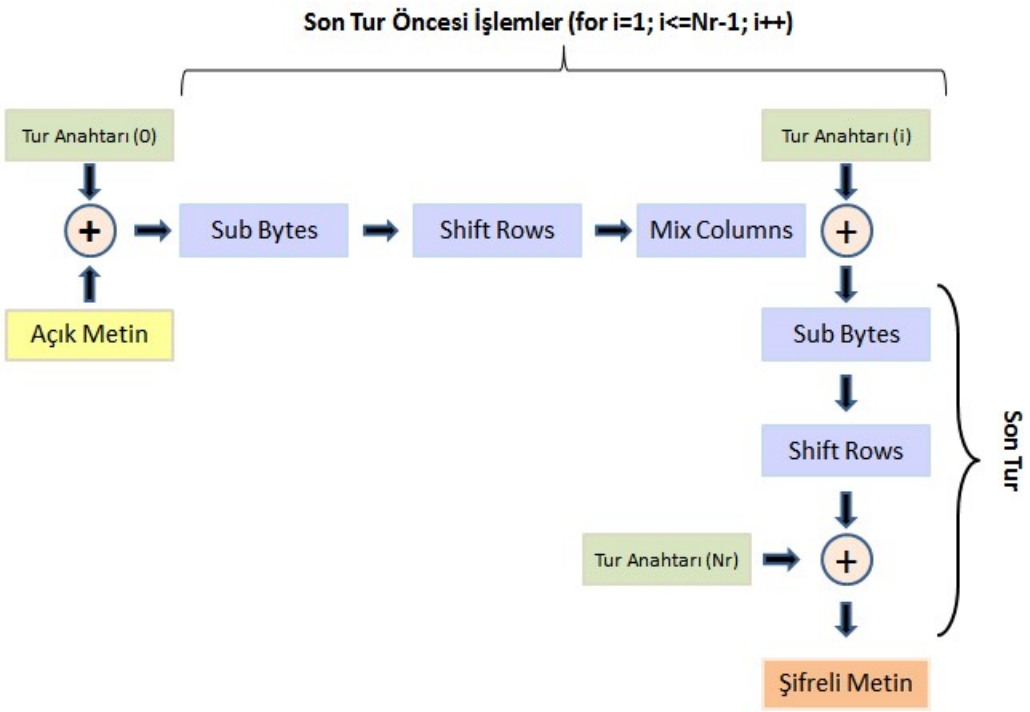
$D_{0,0}$	$D_{0,1}$	$D_{0,2}$	$D_{0,3}$
$D_{1,0}$	$D_{1,1}$	$D_{1,2}$	$D_{1,3}$
$D_{2,0}$	$D_{2,1}$	$D_{2,2}$	$D_{2,3}$
$D_{3,0}$	$D_{3,1}$	$D_{3,2}$	$D_{3,3}$

$\Rightarrow$

c_0	c_4	c_8	c_{12}
c_1	c_5	c_9	c_{13}
c_2	c_6	c_{10}	c_{14}
c_3	c_7	c_{11}	c_{15}

g: giriş verisi, **D:** Durum, **c:** çıkış verisi

AES Algoritması, Şekil 3.8'de verilen akışa göre çalışmaktadır. Buradaki tur sayısı anahtar boyuna göre değişiklik göstermektedir. Tur sayısı (Nr) anahtar boyu 128 bit için 10, 192 bit için 12 ve 256 bit için 14 turdur.



Şekil 3.8: AES Algoritması

Şekilde verilen işlemlerin içeriği şu şekildedir:

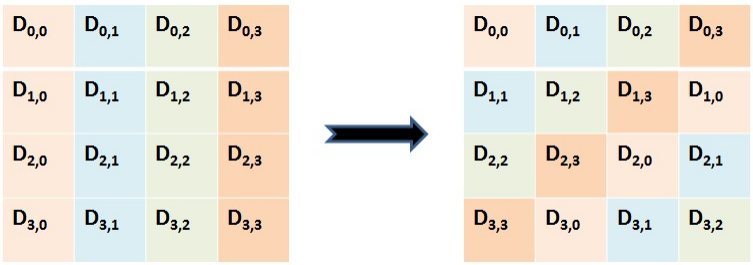
Sub Bytes (S Kutusu Dönüşümü): Doğrusal olmayan bir S Kutusu kullanılarak her bir bayt başka bir bayt değerine dönüştürülür. S Kutusu Dönüşümü oluşturulan Durum üzerindeki her bir bayt üzerinde bağımsız olarak çalışır ve tersine döndürülemeyen bir işlemdir.

Örnek olarak Durum'daki $D_{2,1}$'in F3 değerine sahip olduğu varsayılırsa, aşağıdaki S Kutusu Tablosu (Şekil 3.9) [33] kullanılarak F3 değerinin yeni değeri olan 0D elde edilecektir.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Şekil 3.9: AES S Kutusu İşlemi

Shift Rows (Satır Kaydırma): Satır Kaydırma İşleminde, Durum'un son üç satırındaki baytlar döngüsel olarak kaydırılır. İlk satır kaydırılmaz. Satır Kaydırma İşlemi aşağıdaki şekilde verilmiştir:



Şekil 3.10: AES Satır Kaydırma İşlemi

Mix Columns (Sütun Karıştırma): Sütun Karıştırma işleminde Satır Kaydırma İşleminden sonra oluşan yeni Durum'daki her bir sütun aşağıdaki özel matrisle çarpılır:

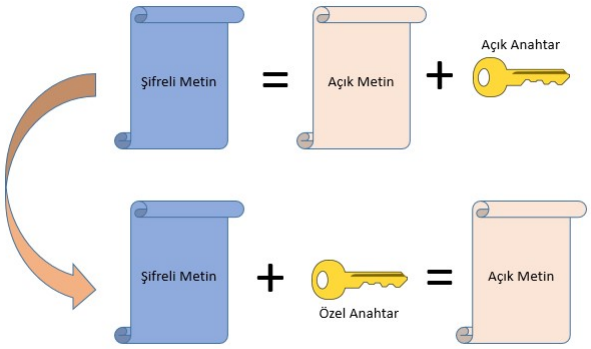
$$\begin{vmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{vmatrix}$$

AES Algoritmasında tüm bu işlemler son tura kadar tekrar tekrar yapılır. Son tura gelindiğinde S Kutusu ve Satır Kaydırma İşlemi yapılırken Sütun Karıştırma İşlemi yapılmaz. Satır Kaydırma İşleminden sonra elde edilen veri doğrudan son tur anahtarı ile XOR

işleminden geçirilir ve dışarıya çıkış verisi yani Şifreli Veri olarak verilir.

3.3.2 Asimetrik Algoritmalar

Asimetrik algoritmalar; özel ve açık anahtar (public, private) olmak üzere iki anahtarla işlem gerçekleştiren kriptografik algoritmalardır. Açık anahtardan özel anahtar elde edilemez ve özel anahtarın güvenliğinin sağlanması güvenlik açısından yeterli olmaktadır.



Şekil 3.11: Asimetrik Algoritmalar

Asimetrik algoritmalar simetrik anahtarların paylaşılmasında, kimlik doğrulama işlemlerinde, PGP, TLS gibi haberleşme protokollerinde, dijital imzalamada yaygın olarak kullanılmaktadır.

3.3.2.1 RSA Algoritması

1978 yılında açıklanan RSA Algoritması [34], Ron Rivest, Adi Shamir ve Leonard Adleman tarafından geliştirilmiştir. Algoritma isimlendirilirken geliştiricilerin isimlerinin baş harfleri bir araya getirilmiştir. Açık anahtarlı veya asimetrik şifreleme algoritmaları diye sınıflandırılan algoritmaların içerisinde yer almaktadır.

Algoritmanın matematiksel güvenliği çok büyük sayıların çarpanlarına ayrılması problemine dayanmaktadır. Bu sebeple algoritmanın yapı taşını oluşturan iki büyük asal sayının gerçekten asal sayı olduğundan emin olunması gerekmektedir.

Algoritmanın işleyişi esnasında kullanılan parametreler aşağıda verildiği şekilde oluşturulur:

- İki büyük asal sayı seçilir (p ve q), Örneğin 2048 bit RSA işlemi için p ve q'nun 1024 bit olması gerekmektedir. p ve q değerleri gizli kalması gereken değerlerdir.
- Modülüs (n) değeri hesaplanır (n = p x q). Modülüs değeri aynı zamanda herkese açıktır. Modülüsün açık olmasının zafiyet oluşturmamasının dayanağı çarpanlara ayırma işleminin zorluğu problemidir.
- $\phi(n)$ değeri hesaplanır ($\phi(n)$ = (p-1) x (q-1)), $\phi(n)$ değeri de gizli değerdir.
- $1 < e < \phi(n)$ olacak şekilde e değeri belirlenir. Algoritmanın verimli çalışması için genelde 65537 değeri kullanılmaktadır. Bu değer herkese açıktır.
- $d \equiv e^{-1} (\text{mod } \phi(n))$ değeri hesaplanır. Bu değer özel (gizli) anahtar değeridir. Bu anahtarın açığa çıkması tüm güvenliğin yıkılmasına yol açar.

Yukarıdaki işlemler gerçekleştirildiğinde açık ve özel anahtar değerleri ortaya çıkmış olacaktır. e ve n değeri açık; d değeri de gizli anahtar değerlerini oluşturmaktadır. p, q ve $\phi(n)$ değerlerinin de açığa çıkmaması ve özel anahtar kadar ciddi bir şekilde korunması gerekmektedir. Çünkü bu değerler özel anahtarın hesaplanmasında kullanılan değerlerdir.

RSA Şifreleme: RSA şifreleme işlemi aşağıda verilmiştir:

c şifrelenmiş, m şifrelenecek mesaj olmak üzere;

$$c \equiv m^e \text{ (mod n)}$$

RSA Şifre Çözme: RSA şifre çözme işlemi aşağıda verilmiştir:

m elde edilecek mesaj olmak üzere;

$$m \equiv c^d \text{ (mod n)}$$

RSA Algoritması asimetrik algoritma olması ve alt yapısındaki işlemlerden dolayı simetrik algoritmalara göre çok yavaş çalışmaktadır. Bu sebeple RSA-CRT olarak bilinen bir yöntem geliştirilmiş ve RSA Algoritmasının performansı kısmen de olsa artırılmıştır. Özellikle akıllı kartlarda RSA-CRT kullanılmaktadır.

3.3.2.2 Eliptik Eğri (EC) Algoritması

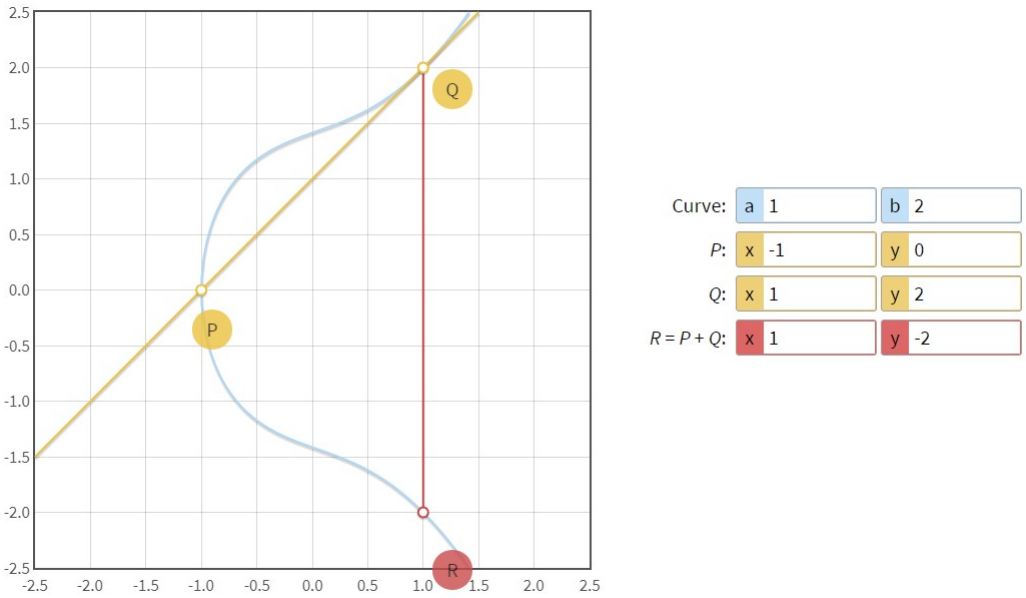
Literatürde ECC (Elliptic Curve Cryptography) olarak bilinen algoritma asimetrik algoritmalar sınıfında yer almaktadır. Eliptik eğri temelli protokollerde, herkese açık bilinen bir temel noktaya göre rastgele bir eliptik eğri elemanının ayrık logaritmasını bulmanın mümkün olmadığı varsayılmaktadır (Ayrık Logaritma Problemi). Eliptik eğri şifrelemesinin güvenliği, bir nokta çarpımını hesaplama kabiliyetine ve orijinal ve çarpım noktaları verilen çarpımın yeniden hesaplanamamasına bağlıdır. Eliptik eğrinin büyüklüğü problemin zorluğunu belirler.

Eliptik eğri şifrelemesinin en büyük faydası, depolama ve iletişimdeki güvenlik gereksinimlerini daha küçük bir anahtar boyu ile sağlıyor olmasıdır. RSA algoritması ile karşılaştırıldığında, RSA tabanlı bir şifrelemede kullanılan büyük bir modülüs ve buna bağlı olarak daha büyük bir anahtarla sağlanan aynı güvenlik seviyesinin Eliptik Eğri ile çok daha küçük bir anahtar boyu ile sağlandığı değerlendirilmektedir. 256-bit eliptik eğrinin, 3072-bit RSA ile eşdeğer güvenlik sağladığı iddia edilmektedir [35].

Eliptik Eğri aşağıdaki formülle ifade edilir:

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

Şekil 3.12'de bir eğri örneği [36] verilmiştir.



Şekil 3.12: Eliptik Eğri Örneği

Bir eliptik eğriyi oluşturan etki alanı parametreleri ve açıklamaları aşağıda verilmiştir:

- p, sonlu alanın boyutunu belirten asal sayı,
- Eliptik eğri denkleminin a ve b katsayıları,
- Alt grubu oluşturan taban noktası G,
- Alt grubun sırası n,
- Alt grubun kofaktörü h.

Etki alanı parametreleri sonlu alandaki eğriyi tanımlayan değişkenlerdir. Güvenlik açısından bu değişkenler belirlenirken belirli şartları sağlaması gerekmektedir. Belirlenen ve testlerden geçirilmiş, kullanılması tavsiye edilen eğrilere NIST [37], SEC [38] ve Brainpool [39] eğrileri örnek olarak gösterilebilir.

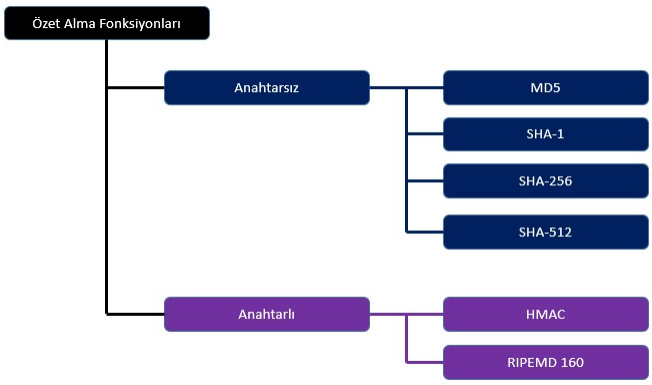
Etki alanı parametreleri belirlendikten sonraki aşama özel ve açık anahtarların oluşturulmasıdır.

- Özel anahtar d, $\{1, \ldots, n-1\}$ aralığında rassal bir sayı,
- Açık anahtar H, $H = d \times G$ işlemi sonucunda elde edilen noktadır.

3.3.3 Özet Alma Algoritmaları

Özet alma algoritmalarının ortaya çıkış amacı dijital imzalama algoritmalarının işini kolaylaştırmak olarak belirtilebilir. Çünkü güçlü bilgisayarlar bile dijital imzaları hesaplamak için çok zaman harcamaktadır. Ayrıca, büyük belgelerin çok sayıda imzaya ihtiyacı olacaktır, çünkü imza oluşturma için giriş boyutunda bir sınır bulunmaktadır. Bu sorunu çözmek için özel bir teknik kullanılmaya başlanmıştır. Belge önce çok daha kısa bir sabit uzunlukta sıkıştırılmakta ve daha sonra sıkıştırılmış verinin dijital imzası hesaplanmaktadır. Burada, sıkıştırmanın geri dönüştürülebilir olması önemli değildir. Çünkü özet her zaman orijinal dokümandan yeniden üretilebilir. Bu tür hesaplamalar için kullanılan fonksiyonlara tek yönlü fonksiyonlar (özet fonksiyonları) denir.

Özet fonksiyonları genel olarak iki alt sınıfa ayrılmaktadır [40]:



Şekil 3.13: Özet Alma Algoritmaları

Literatürde özet alma fonksiyonlarına yönelik çok sayıda anlamlı atak bulunmamaktadır. Bunun sebebi özet alma fonksiyonlarının kullanım amacıdır.

3.4 Rasgele Sayı Üretimi

Kriptografik algoritmalarla ilişkili olarak rasgele sayılara tekrar tekrar ihtiyaç duyulmaktadır. Özellikle akıllı kartlarda, kart okuyucu ile oluşturulan her bir oturumun benzersizliğini sağlamak için rasgele sayılar gerekmektedir. Ayrıca kimlik doğrulama işlemlerinde, veri şifrelemelerinde dolgu olarak kullanılmaktadır. Haberleşme esnasında gönderilen paketlerin sırasını belirten sayaçların başlangıç değerleri rasgele sayılar ile oluşturulmaktadır. Bu kritik işlevler için ihtiyaç duyulan rasgele sayıların uzunluğu genellikle iki ila sekiz bayt arasında değişmektedir.

Kullanılan bu yöntemlerin güvenliği, dışarıdan tahmin edilemeyen veya dış etkenlerden etkilenmeyen rastgele sayılara dayanır. Akıllı kartlar için ideal çözüm, akıllı kart mikro denetleyicisindeki donanım tabanlı rasgele sayı üreteci olacaktır. Bununla birlikte üretecin, sıcaklık, besleme gerilimi ve radyasyon gibi dış etkilere karşı tam olarak korunması gerekmektedir. Aksi halde güvenliği rasgele sayıların rastsallığına dayanan algoritmaların çalışması tehlikeye düşmüş olacaktır. Akıllı kart mikro denetleyicisindeki rastgele sayı üreteçleri, genellikle voltaj kontrollü osilatörler tarafından sağlanan doğrusal geri besleme kaydırma yazmaçlarına (LFSR'ler) dayanmaktadır [41].

Son teknoloji ile bile, mikro denetleyici kalıbının silikonunda dış etkilere karşı korumalı (rastgele bir sayı üreteci veya TRNG) iyi bir rastgele sayı üreteci üretmek zordur. Bu

sebeple, geliştiriciler üretilen rasgele sayıları iyileştirmek için yazılım kullanmaktadırlar. Bu yaklaşımla, rastgele sayı üretecinden çıkan sayılar, rastgele sayılar için gerekli kriterleri yerine getiren bir sözde rasgele sayı üretecine girdi olarak kullanılmaktadır [42].

Sözde rasgele sayı üreteçleri (PRNG'ler), rastgele sayılar üretmek için kesin bir deterministik algoritma kullanmaktadır. Algoritma ve giriş değerleri biliniyorsa bu sayılar tahmin edilebilmekte, ancak dizilerin rastsallığı genellikle oldukça iyi olmaktadır [40].

Akıllı kartlar için bir üretim grubundaki tüm akıllı kartların farklı rastgele sayı dizileri üretmesini sağlamak çok önemlidir. Bu durum sağlandığında, bir kart tarafından üretilen rastgele sayılar, aynı gruptaki başka bir kart tarafından üretilemeyecektir. Bu durum, akıllı kart işletim sisteminin karta kazınması sırasında, rastgele sayı üretecinin tohum numarası (başlangıç değeri) olarak kullanılacak rastgele bir sayıyı depolayarak elde edilmektedir.

3.5 Kimlik Doğrulama ve Tanımlama

Kimlik doğrulama, bir varlığın kimliğini kanıtlama işlemidir. Bir kimlik doğrulama işleminde, bir varlığın gerçekten iddia ettiği varlık olup olmadığını belirlemek için belirli yöntemler kullanılmaktadır. Kimlik doğrulama, sahip olunan bir şey (örneğin kimlik kartı), bilinen bir bilgi (PIN veya şifre) veya biyometrik bir özelliğe dayanabilmektedir. "İki faktörlü kimlik doğrulama" ve "üç faktörlü kimlik doğrulama" terimleri de bu özelliklere bağlı olarak kullanılmaktadır. İki faktörlü kimlik doğrulama, üç farklı kimlik doğrulama özelliğinden ikisini birleştirir (sahiplik, bilgi veya biyometrik özellik ile doğrulama). Bu, başarılı bir kimlik doğrulama için iki farklı kimlik doğrulama özelliğinin pozitif sonuçlarla gerçekleştirilmesi gerektiği anlamına gelmektedir. Üç faktörlü kimlik doğrulamasında, üç kimlik doğrulama özelliğinin tümü birleştirilmektedir. Bu, başarılı kimlik doğrulama için üç özelliğin de pozitif sonuçlarla gerçekleştirilmesi gerektiği anlamına gelmektedir [40], [43].

Örnek olarak; Bir kişinin kimlik kartını (sahiplik) sunması, PIN bilgisini (bilgi) bilmesi ve bir parmak izi testinden geçmesi (biyometrik özellik) durumunda, kişinin üç faktörlü kimlik doğrulaması yapılmış olacaktır. Üç kimlik doğrulama özelliğinin tamamı, kimliğin doğrulanması için olumlu sonuçlar vermek zorundadır.

Bu arada, kimlik doğrulama, bir varlığın veya bir mesajın orijinal ve değişmemiş olup olmadığını belirleme işlemidir, tanımlama işlemi ise bir varlığın sistem tarafından tanınmasını sağlayan süreçtir. Kimlik doğrulama ile kimlik tespiti yapılabilir. Yani bir doğrulama işlemi sonucunda bu kişi A kişisidir denilebilir.

Kimlik doğrulama işlemi sonrasında izin verilen işlemler için yetkilendirme kavramı kullanılmaktadır. Yani yetkilendirme basitleştirilmiş terimle, bir varlığa belirli bir işlemi gerçekleştirme izni verme yetkisidir.

Doğrulamanın amacı bir iletişim tarafının kimliğini ve gerçekliğini doğrulamaktır. Akıllı kartlar için kartın veya kart okuyucunun, karşı tarafın sırasıyla gerçek bir kart okuyucu mu yoksa gerçek bir akıllı kart mı olduğunu belirlemesi anlamına gelmektedir.

Kimlik doğrulama genel olarak statik kimlik doğrulama ve dinamik kimlik doğrulama olarak sınıflandırılmaktadır. Statik kimlik doğrulamada, aynı (statik) veri her zaman kimlik doğrulama için kullanılmaktadır (örneğin, bir şifre kullanılması). Buna karşılık, dinamik kimlik doğrulama, eski oturumlar sırasında kaydedilen verilerin tekrar kullanılmasına dayalı saldırılara karşı koruma sağlamayı amaçlamaktadır. Bu sebeple her kimlik doğrulamada farklı veriler kullanılmaktadır.

3.6 Dijital İmzalama

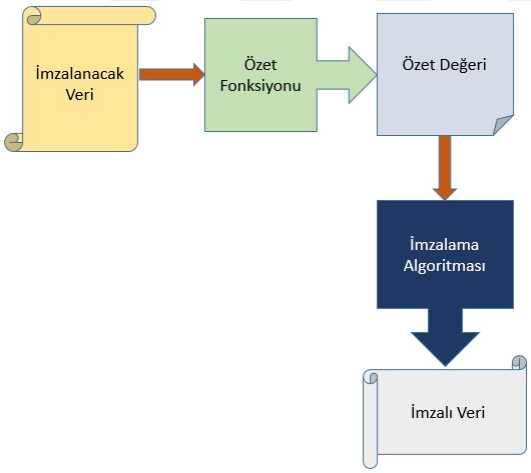
Elektronik imza olarak da adlandırılan dijital imzalar, elektronik olarak iletilen mesajların veya elektronik belgelerin doğruluğunu tespit etmek için kullanılmaktadır. Bir mesajın veya belgenin değiştirilip değiştirilmediğini belirlemek için imzanın doğrulama işlemi kullanılabilmektedir.

Bir imza, sadece imza sahibi tarafından doğru bir şekilde üretilebilme özelliğine sahiptir. Herhangi bir mesaj alıcısı tarafından da doğrulanabilmektedir. Bu dijital imzanın temel özelliğidir. Örneğin; sadece bir akıllı kart bir belgeyi imzalayabilir, ancak tüm kartlar imzanın orijinal olup olmadığını kontrol edebilir. Özellikleri sebebiyle, asimetrik algoritmalar dijital imzalar için ideal bir yöntem sağlamaktadır.

Dijital imza terimi, normalde yalnızca asimetrik şifreleme algoritmalarıyla bağlantılı olarak kullanılmaktadır. Çünkü açık ve özel anahtarların kombinasyonu bu algoritmaları dijital imzalarla kullanım için çok uygun hale getirmektedir. Bununla birlikte, simetrik

algoritmalara dayanan imzalar pratikte sıklıkla kullanılmaktadır. Bu tür imzalarda, imza oluşturmak için kullanılan gizli anahtar biliniyorsa, yalnızca belgenin doğruluğunu kontrol etmek mümkündür. Bu nedenle, böyle bir imza aslında teknik olarak bir imza değildir, ancak bu şekilde ifade edilmektedir. "Dijital" terimi, bu durumdan dolayı kullanılan algoritmanın türünü belirtmek için kullanılamamaktadır [40].

İmzalanacak mesaj veya belgeler genellikle bir kaç GB uzunluğunda olmaktadır. Bu durum, belgelerin tamamının imzalanmasını imkansız kılmaktadır. Bu sebeple, öncelikle verilerden bir özet değer hesaplanmaktadır. Bu özet alma işlevi geri alınamaz, yani orijinal veriler sıkıştırılmış verilerden kurtarılamaz. Özet değerleri çok hızlı bir şekilde hesaplanabilir ve bu durum, özet alma işlemini dijital imzalar için ideal bir yardımcı işlev yapmaktadır.



Şekil 3.14: İmzalama İşlemi

3.7 Sertifikalar

Dijital imzaların kullanımıyla birlikte ortaya çıkan bir problem vardır. Bir mesajın dijital imzasını kontrol etmek isteyen herkes ilgili açık anahtara ihtiyaç duymaktadır. Bununla birlikte, açık anahtar hiçbir koruma olmadan gönderilemez, çünkü alıcı anahtarın doğruluğunu kontrol etmek zorundadır. Açık anahtar, bu nedenle güvenilirliği doğrulanmış güvenilir bir kuruluş tarafından imzalanmaktadır. Bu varlığa sertifika yetkilisi (veya sertifika otoritesi) (CA - Certification Authority) denir. Sertifika yetkilisi tarafından imzalanmış olan ortak anahtarın, beraberindeki dijital imza ve bazı ek parametrelerin

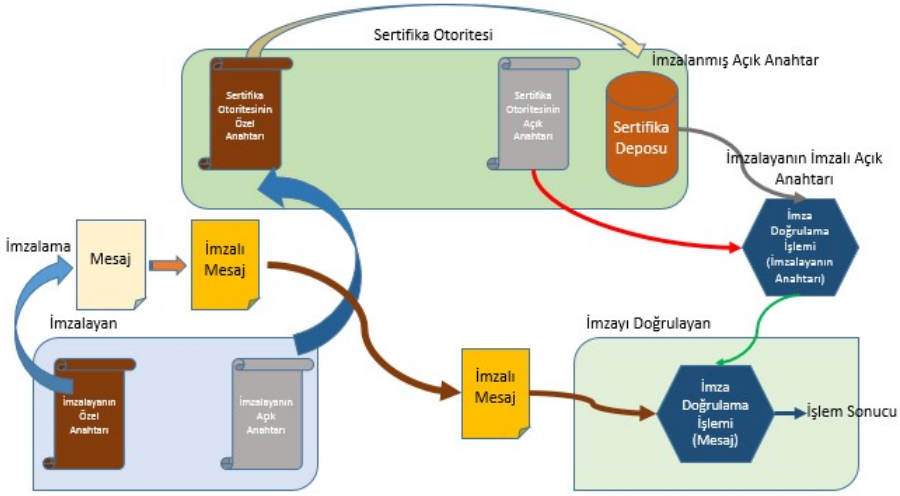
birleşimi sertifika olarak adlandırılmaktadır. Sertifika yardımıyla imza oluşturma ve imza doğrulama süreci gerçekleştirilmektedir.

Bu süreçte, güven merkezi (TC) olarak adlandırılan başka bir varlık daha kullanılmaktadır. Güven merkezi sertifikaları ve ilgili kara listeleri oluşturan ve yöneten merkezdir. Ayrıca isteğe bağlı olarak dijital imza kartları için anahtarlar oluşturmaktadır. Kural olarak, bir güven merkezi de ortak bir sertifika dizini tutar; böylece imzalı bir mesajı kontrol etmek isteyen herkes ilgili imzalı ortak anahtarı merkezden internet üzerinden talep edebilmektedir.

Bir sertifika yalnızca imzalı ortak anahtarı değil, aynı zamanda çok sayıda ek parametre ve seçenek de içermektedir. Bu bilgiler başka bir bilgiyi kullanmadan bir sertifikanın genel anahtarını doğrulamayı mümkün kılmaktadır. Aynı zamanda, özet değeri ve imzayı oluşturmak için kullanılan algoritmaların açıkça belirtilmesi gerekmektedir.

Sertifikaların yapısının standartlaştırılması amacı ile standartlar üretilmiştir. İlgili standartların en iyi bilineni sertifikaların yapısını ve kodlamasını belirleyen X.509 [44]'dur. Ayrıca bu standart, ISO/IEC 9594-8 [45] olarak ISO/IEC standartları ailesine katılmıştır.

Geniş kapsamlı X.509 standardı, birçok dijital imza uygulamasının temelini oluşturmaktadır. Burada belirtilebilecek bazı örnekler, İnternet Güvenli Soket Katmanı (SSL) [46] ve Gizliliği Geliştirilmiş Posta (PEM) [47], Güvenli Çok Amaçlı İnternet Posta Uzantıları (S/MIME) [48] ve Güvenli Elektronik İşlem (SET) [49] gibi uygulamalardır.



Şekil 3.15: Sertifikaların İmza Doğrulamada Kullanılması

Bölüm 4

Donanımsal Saldırılar

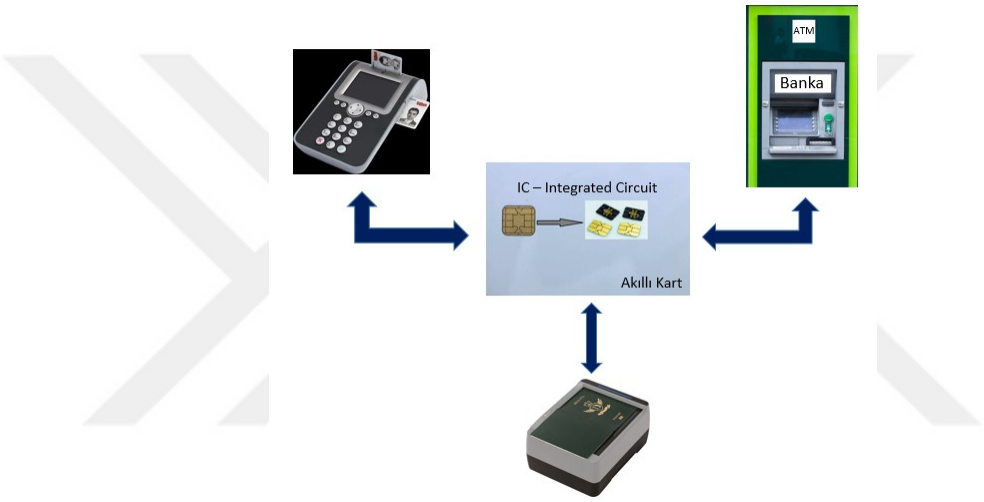
Kriptografi alanındaki bilimsel araştırmaların bir çoğu, kriptografik algoritmaların, şifrelerin ve protokollerin matematiksel alt yapısını incelemektedir. İletişim güvenliği konusu daha kritik görüldüğünden, uç noktadaki donanımından ziyade bir kanal üzerinden akan bilgiye yapılan saldırılara daha fazla önem verilmiştir. Bununla birlikte, kriptografik işlemleri gerçekleştiren donanım tüm sistemin önemli bir parçasıdır ve bu donanımların da incelenmesi gerekmektedir.

Bilgisayar donanımı teknolojisindeki hızlı gelişmeler, emtia haline gelen daha küçük, daha hızlı ve daha ucuz cihazların ortaya çıkmasına sebep olmuştur. Sonuç olarak, kriptografik donanım, artık evlerde kullanılan IoT cihazlarından cep telefonlarına; kredi kartlarından kimlik kartlarına (akıllı kartlar) kadar her şeyde yaygın şekilde kullanılır hale gelmiştir. Bu cihazlar, genellikle özel anahtarları veya diğer hassas verileri depolamaktadır. Bu nedenle, bu özel verilerin açığa çıkmasına yol açacak donanımsal zayıflıkların oluşturacağı kayıpların yıkıcı etkilere neden olabileceği aşikardır.

Kriptografik donanımlara kolayca erişim sağlanabildiğinden, saldırganlar uygulamalar hakkında özel detayları öğrenmek için donanımın iç yapısını (Örneğin Tersine Mühendislik) inceleyebilirler. İnceleme sonucunda elde edilen bilgiler, algoritmaların matematiğine doğrudan saldırmadan donanıma yapılacak saldırıları gerçekleştirmek için kullanılabilir. Yani, tamamen güvenli algoritmalar ve protokoller kullanılıyor olsa bile, saldırgan donanımın yapısındaki zafiyetlerden dolayı hassas bilgileri (özel anahtar, kişisel veriler vs.) öğrenebilecektir. Ayrıca hassas bilgiler elde edilemese bile, saldırganlar donanımı bozabilecek, güvenlik sisteminde hatalara sebebiyet verebilecektir.

Kriptografik işlemler gerçekleştiren birçok donanım türü olsa da bu bölüm altında spesifik olarak akıllı kartlara odaklanılacaktır.

Akıllı Kartlar: Tümdevrelere sahip, kartlardır. Kartlar, kendi güç kaynaklarını barındırmamaktadır. Kart okuyucular (ATM'ler, TC Kimlik Kartları için KİOSK'lar, Pasaport Okuyucu Sistemleri, vs.), hem kartın güç kaynağı hem de kartla iletişim kuran cihazlardır. Kartlar, kalıcı bellek alanlarında özel anahtarlar, kişisel veriler gibi hassas bilgilere sahiptir. Özel anahtar verisi, standart olan şifreleme ve doğrulama protokollerinin gerçekleştirilmesinde kullanılır. Örnek olarak, kartın, kart okuyucuya, kendisini tanıtmak için karttaki özel anahtarla imzalanmış bir veriyi göndermesi verilebilir.



Şekil 4.1: Akıllı Kart ve Kart Okuyucu Örnekleri

Güvenliğin akıllı kartlar ile ilgili önemli bir yönü, donanımın kullanıldığı ortamın kontrolsüz olmasıdır. Diğer pek çok donanımda varsayılabilecek fiziksel güvenlikle ilgili varsayımlar, akıllı kartlar için geçerli olmamaktadır. Akıllı kartın saldırganın eline geçme tehlikesi olduğu kadar, akıllı karta bağlanan diğer donanımlar da saldırganın kontrolü altında olabilir. Örneğin, kredi kartı olarak kullanılan bir akıllı kart, bir saldırganın kontrolü altındaki kart okuyucuya (ATM, POS makinesi, vs.) bağlanabilir. Çalışma ortamı, potansiyel olarak sayısız saldırgan ve tehdit bulundurduğu için analizin iyi yapılıp tehditlerin doğru bir şekilde belirlenmesi büyük önem arz etmektedir. Örneğin finansal işlemler için akıllı kartların kullanıldığı durumlarda, kart sahibine ve kart okuyucuya banka tarafından güvenilemez.

Akıllı kartların kullanım alanlarına bağlı olarak bir diğer istenmeyen durum da geriye dönük uyum sorunlarından kaynaklanan kısıtlamalardır. Kartlara uygun protokolleri

destekleyen kart okuyucuların sahada kullanılıyor olması, yeni kartların da bu protokollere göre geliştirilmesi zorunluluğunu barındırmaktadır. Kriptografik algoritmalar ve protokoller her geçen gün gelişiyorken, bunun sahaya hızlı bir şekilde yansıtılamıyor olması, güvenliğin en iyi şekilde sağlanmasına engel olarak karşımıza çıkmaktadır. Örneğin DES, SHA-1 ve RSA-1024 bit, dünyadaki tüm büyük otoriteler tarafından matematiksel olarak güvensiz kabul edilmesine rağmen, sahada kullanımının önüne geçilememektedir.

Bu bölümün altında yer alan saldırı tipleri, matematiksel algoritmaların kırılmasını (yani kriptanalizini) içermemektedir. Algoritmalardaki zayıflıklar var olsa da bu bölümde anlatılacak saldırılar, algoritma zayıflıklarına değil, algoritmaların işlenmesine ve donanımın fiziksel güvenliğine yönelik saldırılardır.

Akıllı kartlar gibi kriptografik donanımlar kurcalamaya karşı dayanıklı olarak üretilirken, asla tam bir koruma sağlamamaktadır. Yeterli kaynaklar sağlandığında, akıllı karta erişimi olan bir saldırgan, cihazda depolanan özel bilgileri doğrudan gözlemleyebilmektedir (Örneğin FIB - Focused Ion Beam). Böyle bir saldırının maliyeti, milyonlarca doları bulmaktadır. Bu durum da karta yapılabilecek saldırıları ciddi anlamda azaltmaktadır. Bu nedenle daha az maliyetli olan, yarı-bozucu ve bozucu olmayan saldırıların gerçekleştirilmesi pratikte daha fazla tehdit oluşturmaktadır. Ancak akıllı kartların askeri teknolojiler gibi çok kritik ortamlarda da kullanılıyor olması, maliyetli olan saldırıların da göz ardı edilmemesi gerektiğini gözler önüne sermektedir.

Donanımsal Saldırı Yöntemleri

Kriptografik cihazların fiziksel olarak saldırıya uğramasının birkaç yolu vardır [50]:

Yan kanal saldırıları: Cihaz normal çalışırken cihazın üzerinden ve/veya ara yüzlerinden güç ölçümünün ve elektromanyetik yayılımının izlenmesi yoluyla cihaza yönelik saldırılara verilen isimdir.

Yazılım saldırıları: Cihazın normal iletişim ara yüzü kullanılarak protokollerde, şifreleme algoritmalarında veya cihazın uygulamalarında bulunan güvenlik açıklıklarından yararlanılarak yapılan saldırılara verilen isimdir.

Hata oluşturma: Anormal çevresel koşullar oluşturularak cihazda hata oluşturularak erişim sağlamaya yönelik saldırılara verilen isimdir.

Problama: Daha çok akıllı kartlar benzeri ürünlere yapılan bu tür saldırılarda tümdevre yüzeyine doğrudan erişmek için problama denilen yöntem kullanılabilir. Böylece tümdevrenin iç yüzeyi incelenebilir, içeride değişiklikler ve müdahaleler yapılabilir.

Tersine mühendislik: Cihazın iç yapısını anlamak ve işlevselliğini öğrenmek amacıyla kullanılır.

Tüm problama ve tersine mühendislik teknikleri, bozucu olarak adlandırılan türden ataklardır. Özel laboratuvarlarda, uzun çalışma sürelerini gerektirirler ve işlem sırasında, saldırı altındaki cihazlar zarar görür. Diğer üç saldırı yöntemi, bozucu olmayan türden ataklardır. Bu saldırılar sırasında, saldırıya uğrayan cihaz fiziksel olarak zarar görmez. Ancak bazı durumlarda, hata oluşturma saldırılarında, cihaz zarar görebildiğinden bu saldırı türü yarı bozucu saldırılar olarak değerlendirilmektedir. Hata yaptırma saldırılarında cihaza fiziksel olarak erişim gerekmektedir. Lazer atışı, elektro manyetik enjeksiyon, ısıtma gibi saldırı türleri yarı bozucu saldırılardır.

Bozucu olmayan saldırılar, iki nedenden dolayı bazı uygulamalarda özellikle tehlike oluşturmaktadır. İlk olarak, cihazın sahibi gizli anahtarların veya verilerin çalındığını fark etmeyebilir; bu nedenle, ele geçirilen anahtarların geçerliliği kötüye kullanılmadan önce anahtarların iptal edilme olasılığı düşüktür. İkinci olarak, bozucu olmayan saldırılarda kullanılan cihazlar düşük maliyetli cihazlardır.

Bozucu olmayan saldırılar, donanımsal tasarım ve yazılım hakkında ayrıntılı bilgiye sahip olmayı gerektirmektedir. Bozucu saldırılar içinse donanım hakkında temel bir bilgi yeterli olmaktadır. Bozucu saldırılar, genellikle geniş bir ürün yelpazesinde benzer tekniklerle çalışmaktadır. Bu sebeple, saldırılar çoğu zaman tersine mühendislik saldırıları ile başlamaktadır. Yarı bozucu ataklar ise cihazın işlevselliğini öğrenmek ve güvenlik devrelerini test etmek için kullanılmaktadır. Bu saldırılar, iç katmanlara herhangi bir fiziksel temas gerektirmediğinden, pahalı ekipmanlara ihtiyaç duymamaktadır.

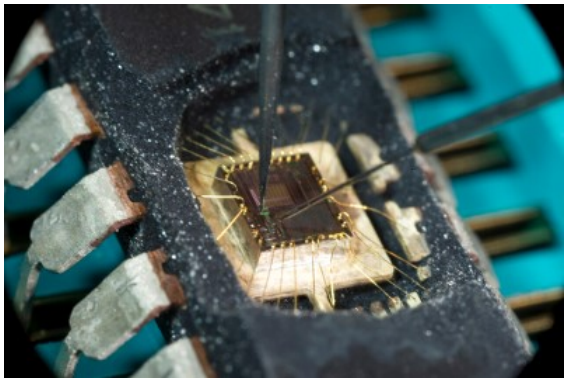
4.1 Bozucu (Invasive) Saldırılar

Bu saldırılar, cihazın iç bileşenlerine doğrudan erişim gerektirmektedir. Bu cihaz bir güvenlik modülü veya bir USB ise, belleklerine erişim için cihazların açılması gerekir. Bir akıllı kart veya bir mikro denetleyici durumunda, tümdevrenin pasivasyon tabakasının

altında gömülü olan iç yollara erişim sağlamak için, paketlemenin odaklanmış iyon ışını (FIB) veya lazer ile etkisiz hale getirilmesi gerekir. Bu tür saldırıların başarılı bir şekilde gerçekleştirilmesi için iyi donanımlı ve bilgili bir kişi tarafından yapılıyor olması gerekir. Ayrıca saldırılacak cihaz boyutu küçüldükçe ve cihaz karmaşıklığı arttıkça saldırılar daha da zorlu ve pahalı hale gelmektedir.

Paketten çıkarma ve kimyasal aşındırma gibi bazı işlemler, küçük bir yatırım ve minimum bilgiye sahip olan herkes tarafından gerçekleştirilebilir. Saldırıların karmaşıklığının artmasına rağmen, bazı saldırılar pahalı laboratuvar ekipmanı olmadan da yapılabilmektedir. Örneğin salt okunur bir belleğin optik olarak okunması.

Akıllı kartlar için bozucu saldırılar, tümdevre paketinin kaldırılmasıyla başlar. tümdevre açıldıktan sonra, problama veya modifikasyon ataklarını gerçekleştirmek mümkün olmaktadır. Bu tür saldırılar için en önemli araç problama aracıdır. Problama aracının ana bileşeni, uzun çalışma mesafeli görüntü mercekli özel bir optik mikroskoptur. Problama aracı sabit bir platform üzerinde kuruludur ve prob kollarının mikron altı hassasiyette tümdevre yüzeyinde hareket etmesini sağlamaktadır. Her bir kol üzerinde elastik sondalama iğnesi bulunmaktadır ve tümdevre üzerindeki veri yolu hatlarına zarar vermeden elektriksel temas sağlamaktadır [51].



Şekil 4.2: Problama

Paketten çıkarılmış tümdevrede, üst katman, tümdevreyi çevreleyen alüminyum bağlantı hatları ve iyon saldırısından koruyan bir pasivasyon katmanı (genellikle silikon oksit veya nitrür) ile kaplıdır. Bu pasivasyon katmanı, probların temasından önce çıkarılmalıdır. Bu işlem için en uygun teknik bir lazer kesicinin kullanılmasıdır. Dikkatlice ayarlanan lazer ışınları pasivasyon tabakasını temizler. Pasivasyon katmanında ortaya çıkan delik, sadece tek bir veri yolunun ortaya çıkacağı kadar küçük yapılabilir. Bu durum, komşu

hatlarla yanlışlıkla oluşabilecek teması önler ve delik ayrıca probun konumunu dengeler; titreşim ve sıcaklık değişikliklerine karşı daha az hassasiyet oluşmuş olur. Güvenli bir işlemcide depolanan bilgileri doğrudan her bir bellek hücresinden okumak pratik olmamaktadır. Bunun yerine saklanan verilere, bellek veri yolu üzerinden erişilmesi anlamlı olmaktadır. Problama, tüm veri yolunu gözlemlemek ve saklanan verilere erişilirken verileri ele geçirmek için kullanılmaktadır.

Tüm bellek hücrelerinin cihaz yazılımının yardımı olmadan okunabilmesi için, belleğe erişmek üzere adres sayacı gibi bir merkezi işlemci birimi (CPU) bileşeninin kötüye kullanılması gerekmektedir. Program sayacı, her komut döngüsü sırasında otomatik olarak artırılmakta ve bir sonraki adresi okumak için kullanılmaktadır. İşlemcinin sadece program sayacını normal okuma sırasını bozacak şekilde atlama, çağrı veya geri dönüş komutları çalıştırmasının önlenmesi gerekmektedir.

Bir cihazın nasıl çalıştığını anlamak için bir başka yaklaşım tersine mühendisliktir. Tersine mühendislik için ilk adım, tümdevrenin bir haritasını oluşturmaktır. tümdevre yüzeyinin dijital kameralı bir optik mikroskop kullanılarak yüksek çözünürlüklü fotoğrafları çekilir ve birkaç metre büyüklüğünde mozaikleri üretilir. Veri ve adres yolu hatları gibi temel mimari yapılar, bağlantı kalıpları incelenerek ROM, SRAM, EEPROM, aritmetik mantık birimi (ALU) hızlı bir şekilde tanımlanabilir. ve talimat kod çözücüsü. Tüm modüller genellikle kolayca tanınabilen mandallar ve veri yolu sürücüleri aracılığıyla ana veri yoluna bağlanır. Saldırıyı gerçekleştiren kişinin tamamlayıcı metal-oksit-yarı iletken (CMOS) tümdevre tasarım teknikleri ve mikro denetleyici mimarileri hakkında bilgi sahibi olması gerekmektedir. Ancak bu gerekli bilgiler çok sayıda ders kitabından kolayca elde edilebilmektedir.

Bozucu saldırılar ve tümdevre yapısına herhangi bir değişiklik uygulamak için kullanılan en yaygın araç FIB (Focused Ion Beam)'dir. FIB ile metal ve polisilikon ara bağlantılar kesilebilir ve mikron altı hassasiyetlerde yenileri eklenebilir. Lazer interferometre (girişimölçer) aşamaları kullanılarak, bir tümdevre yüzeyinde kör bir şekilde hareket edilebilir.

4.2 Bozucu Olmayan (Non-Invasive) Saldırılar

Bozucu olmayan saldırılar, test edilen cihazın test öncesi hazırlanmasını gerektirmemektedir. Saldırıyı gerçekleştirecek kişi, test edilecek cihaza kablolarla temas edebilir veya cihazı analiz için bir test donanımına bağlayabilir. Bir saldırı yolu bulunduğunda, bu saldırılar kolayca ölçeklendirilebilir ve saldırının çoğaltılması fazla bir maliyet gerektirmez. Ek olarak, saldırı gerçekleştirildikten sonra cihazda herhangi bir kanıt kalmaz. Bu nedenle bozucu olmayan saldırılar, herhangi bir cihazın güvenliğine karşı en ciddi saldırı yöntemi olarak kabul edilir.

Bozucu olmayan saldırılar *pasif* veya *aktif* saldırılar olarak iki kısımda incelenebilir [52]. Yan kanal saldırıları olarak da adlandırılan pasif saldırılarda, saldırgan cihazla herhangi bir etkileşime girmez, genellikle cihazın sinyallerini ve elektromanyetik yayılımlarını gözlemler. Güç analizi ve zamanlama saldırıları bu tür saldırılara örnek olarak verilebilir. Kaba kuvvet ve hata (glitch) saldırıları gibi aktif saldırılar, güç kaynağı hattı dahil cihaza uygulanan sinyallerle oynamayı içerir. Bozucu olmayan saldırılar alt bölümlerde detaylandırılmıştır.

4.2.1 Yan Kanal Saldırıları

Özellikle bozucu olmayan pasif saldırılar son yıllarda çok dikkat çekmiştir. Pasif bozucu olmayan saldırılar genellikle yan kanal saldırıları olarak isimlendirilirler. En önemli yan kanal saldırısı türleri aşağıda verilmiştir:

- Zaman Analizi Saldırıları
- Güç Analizi Saldırıları
- Elektromanyetik Analiz Saldırıları

Bu saldırılar aşağıdaki alt bölümlerde detaylandırılmıştır. Bu saldırılara ek olarak ses, ısı gibi yan kanal saldırıları da literatürde mevcuttur. Ancak bu tez çalışmasında bu tür yan kanal saldırılarına değinilmemiştir.

4.2.1.1 Zaman Analizi Saldırıları

Bir akıllı kart tümdevresinde gerçekleştirilen güvenlikle ilgili işlemler, giriş verisi ve özel anahtara bağlı olarak farklı zamanlarda tamamlanmaktadır. Dikkatli bir zaman ölçümü ve bu zaman ölçümünün analizi sonucunda gizli anahtar elde edilebilmektedir. Bu konu ilk olarak Kocher tarafından 1996 yılında ortaya atılmış ve literatürde yerini almıştır [9]. Bu konudan yola çıkılarak 1998 yılında RSA algoritmasına bu yöntem uygulanmış ve RSA anahtarı başarılı bir şekilde elde edilmiştir [10]. Atak basit bir şekilde açıklanacak olursa, RSA algoritmasının akıllı karta implemente edilmesinde, hiçbir önlem olmadan basitçe aşağıdaki yöntem kullanılmış ve bu da RSA imzalama esnasında yapılan işlemleri doğrudan anahtara bağımlı hale getirmiştir. Yani anahtar değerindeki 1 ve 0'ların sayısına göre işlem süresi değişiklik göstermektedir.

```
m: mesaj
x = m
for i = n - 2 down to 0
        x = x^2
        if (k_i == 1) then
                x = x × m
end for
return
```

Çoğu şifreleme algoritması zamanlama saldırılarına karşı savunmasızdır. Bunun en başlı sebebi, algoritmaların yazılımsal implemantasyonlarından kaynaklanmaktadır. Gereksiz birçok dallanma ve koşul, önbellek kullanımını ve performans optimizasyonunu gerektirmektedir. Sonuç olarak, performans özellikleri tipik olarak hem şifreleme anahtarına hem de giriş verilerine bağlı olmaktadır.

Zamanlama saldırıları, güvenlik koruması şifrelere veya pinlere dayanan akıllı kartlara veya anahtarları koruyan kontrol sistemlerine uygulanmaktadır. Bölüm 6'te gerçekleştirilen saldırıda zamanlama saldırısından da faydalanılmıştır.

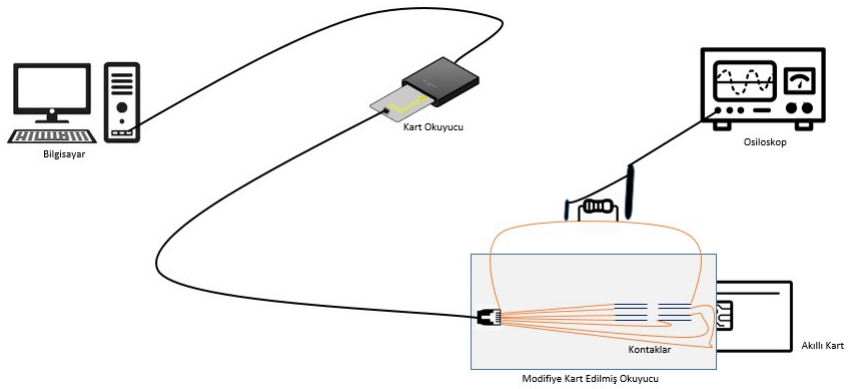
4.2.1.2 Güç Analizi Saldırıları

Güç analizi saldırıları, gizli bilgilerin kriptografik cihazlardan çıkarılmasını sağlayan saldırılardır. Diğer saldırıların aksine, kriptografik algoritmaların matematiksel özellikleri yerine, cihazların güç tüketimi özelliklerine odaklanırlar. Güç analizi saldırıları, kullanımı

kolay ve maliyeti düşük test ekipmanları ile yapılabilecek bozucu olmayan saldırılardır. Bu nedenle bu saldırılar, akıllı kartlar gibi kriptografik cihazların güvenliği için ciddi tehdit oluşturmaktadır. Güç analizi saldırılarının anlaşılabilmesi ve uygulanabilmesi için, kriptoloji, istatistik, ölçüm teknolojisi ve mikro elektronik gibi farklı çalışma alanlarında bilgi sahibi olunması önemli bir husustur.

Güç analizi saldırılarının temel amacı, güç tüketimini analiz ederek kriptografik bir cihazın anahtarını ortaya çıkarmaktır. Bu saldırı gerçekleştirilirken temel olarak, güç tüketiminin iki bağımlılığından yararlanılır. Bunlar, *veri bağımlılığı* ve *işlem bağımlılığı*dır. Güç analizine bağlı zayıflıklar, kriptografik bir cihazın anlık güç tüketiminin işlediği verilere ve gerçekleştirdiği işleme bağlı olmasından kaynaklanmaktadır.

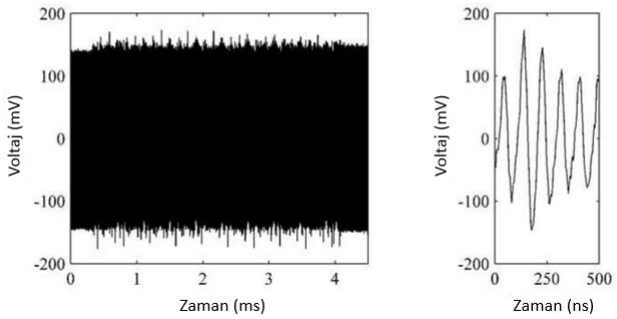
Güç analizi saldırıları aşağıdaki örnek görsel üzerinden anlatılacak olunursa;



Şekil 4.3: Güç Analizi Düzeneği

Mevcut örnekte, akıllı kartın bir AES şifrelemesi yapacak şekilde programlanmış olduğunu varsayalım. Akıllı kart, kartın takılı olduğu modifiye edilmiş okuyucu ile RJ-45 birimi üzerinden bir bilgisayara bağlı olsun. Kart, bilgisayardan bir düz metin almakta, aldığı metni şifrelemekte ve sonucu bilgisayara geri göndermekte olsun. Güç analizi saldırılarında şifreleme yapılırken, akıllı kartın güç tüketimi ölçülür. Bu amaçla, kartın güç kaynağının topraklama kablosuna 1 ohm'luk bir direnç yerleştirilmiştir. Bu direnç boyunca voltaj düşüşü dijital bir osiloskop kullanılarak ölçülür ve kaydedilir.

Aşağıdaki şekil örnek bir voltaj düşüşünü göstermektedir. Bu kaydedilen voltaj düşüşü, akıllı kartın güç tüketimi ile orantılıdır. Bu nedenle, voltaj düşüşü güç tüketimi olarak ve buna karşılık gelen iz, güç izi olarak adlandırılır. Bir güç izinin şekli, net bir şekilde cihaz tarafından yürütülen işlemlere ve cihazın işlediği verilere bağlıdır.



Şekil 4.4: Örnek Güç Tüketimi

Güç analizi saldırıları genel olarak *Basit Güç Analizi* ve *Farksal Güç Analizi* olarak iki kısımda incelenmektedir.

Basit Güç Analizi

Basit güç analizi (SPA) saldırıları, kriptografik işlemler sırasında toplanan güç tüketimi ölçümlerinin doğrudan yorumlanmasını içeren bir tekniktir. Başka bir deyişle, saldırıyı gerçekleştiren kişi anahtarı doğrudan belirli bir güç izinden türetmeye çalışır. Bu, SPA saldırılarını pratikte oldukça zor hale getirebilir. Bu saldırının uygulanabilmesi için çoğu zaman, saldırı altındaki cihaz tarafından yürütülen şifreleme algoritmasının implementasyonu hakkında ayrıntılı bilgi sahibi olunması gerekir. Ayrıca, yalnızca bir güç izi mevcutsa, bu izin düzgün bir şekilde alınabilmesi için genellikle karmaşık istatistiksel yöntemler kullanılmalıdır.

SPA saldırıları, belirli bir girdi seti için yalnızca bir veya çok az güç izi mevcutsa pratikte kullanışlıdır. Örneğin, bir tüketicinin markette alışveriş tutarını kredi kartı (akıllı kart) ile ödediği bir senaryo düşünelim. Tüketicinin belirli dönemlerde (Örneğin ayda bir) markete gelerek alışveriş yaptığını ve genel olarak benzer tutarlarda harcama yaptığını varsayalım. Saldırı amacıyla modifiye edilmiş bir akıllı kart okuyucusu, kartın güç tüketimini kaydedebilir. Bu şekilde, saldırgan benzer açık metinler için birkaç iz toplayabilmiş olacaktır. Bu durumda SPA saldırısı anlamlı olmaktadır.

SPA saldırılarının amacı, yalnızca az sayıda güç izi verildiğinde (az sayıda açık metin için) anahtarı açığa çıkarmaktır. Uç durumlar için, bu saldırı, saldırganın tek bir güç izine dayanarak anahtarı ortaya çıkarmaya çalıştığı anlamına gelir. Bu sebeple SPA saldırıları Tek Atış SPA (single-shot SPA) saldırıları ve Çoklu Atış SPA (multiple-shot

SPA) saldırıları olarak ayrılmaktadır [52]. Tek Atış SPA saldırılarında, sadece bir güç izi kaydedilebilir. Çoklu Atış SPA saldırılarında, birden fazla güç izleri kaydedilebilir. Çoklu Atış SPA saldırılarında, aynı açık metin için güç tüketimi birçok kez ölçülebilir veya farklı açık metinler ile ölçüm alınabilir. Bir açık metin için birkaç güç izine sahip olmanın avantajı, izlerin ortalamasının hesaplanabilmesi, bu şekilde gürültünün azaltılabilmesidir. Tek bir ölçüm alma veya çoklu ölçüm almadaki farklılıklara rağmen, SPA saldırılarının temeli her zaman aynıdır. Yani saldırganın, saldırı altındaki cihazın güç tüketimini izleyebilmesi, saldırıya uğramış cihazda, anahtarın (doğrudan veya dolaylı olarak) güç tüketimi üzerinde önemli bir etkisi olması gerekir.

Farksal Güç Analizi:

Farksal güç analizi (DPA) saldırıları en popüler güç analizi saldırıları türüdür. Bunun nedeni DPA saldırılarının saldırı cihazı hakkında ayrıntılı bilgi gerektirmemesidir. Cihaz tarafından yürütülen şifreleme algoritmasını bilmek genellikle yeterlidir. Ayrıca, kaydedilen güç izleri aşırı derecede gürültülü olsa bile cihazın gizli anahtarını ortaya çıkarabilirler. SPA saldırılarının aksine, DPA saldırıları çok sayıda güç izi gerektirir. Bu nedenle, bir DPA saldırısı yapmak için bir süre fiziksel olarak şifreleme cihazına sahip olmak gerekir.

İki tür saldırı arasındaki diğer önemli fark, kaydedilen güç izlerinin farklı bir şekilde analiz ediliyor olmasıdır. SPA saldırılarında, bir cihazın güç tüketimi temel olarak zaman ekseni boyunca analiz edilir. Saldırgan belirli desenleri bulmaya çalışır veya belirli güç şablonları ile tek bir güç ölçümü ile eşleştirmeye çalışır. DPA saldırıları durumunda, zaman ekseni boyunca güç izlerinin şekli o kadar önemli değildir. DPA saldırıları, belli zamanlardaki güç tüketiminin işlenmiş verilere nasıl bağlı olduğunu analiz eder. Bu nedenle, DPA saldırıları yalnızca güç izlerinin veri bağımlılığına odaklanır.

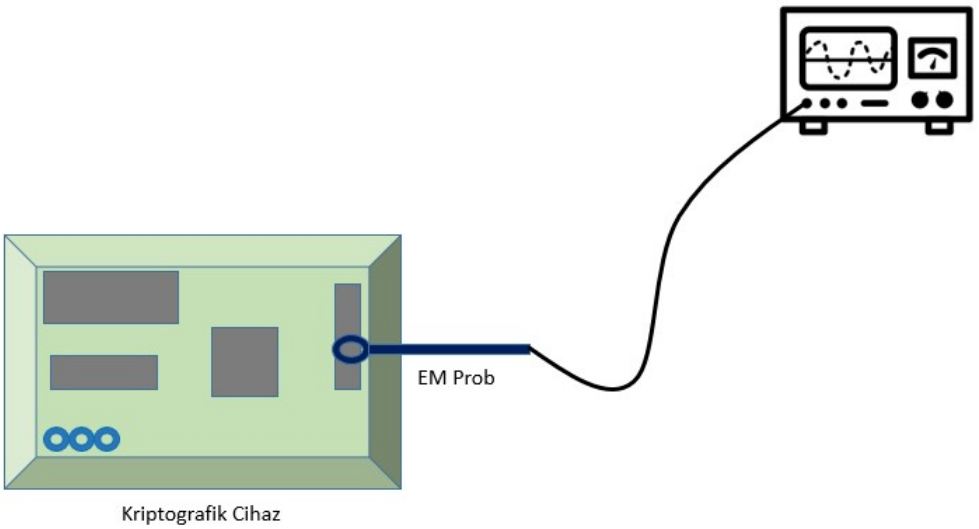
Özetle, modern güvenlik sistemlerinde güvenlik büyük ölçüde kriptografi kullanımına bağlıdır. Şifreleme algoritmaları, verimli hesaplamaları gerektiren oldukça karmaşık fonksiyonlardır. Şifreleme cihazı, bir şifreleme algoritması uygulayan ve karşılık gelen anahtarı depolayan bir cihazdır. Algoritma herkese açıkken, anahtar gizli tutulur. Bu nedenle, şifreleme algoritmasının hesaplanması sırasında anahtar hakkında hiçbir bilgi sızıntısı olmaması önemlidir. Güç analizi saldırıları, kriptografik bir cihazın anlık güç tüketiminin işlediği verilere ve gerçekleştirdiği işleme bağlı olmasından faydalanmaktadır. Bu bağımlılığa dayanarak, bir şifreleme cihazının gizli anahtarını açığa çıkarmak

mümkündür. Güç analizi saldırıları yeterli önlemlerle engellenebilir. Bu önlemler, ara değerler, şifreleme algoritmasının işlemleri ile algoritmayı çalıştıran cihazın güç tüketimi arasındaki ilişkiyi ortadan kaldırır.

4.2.1.3 Elektromanyetik Analiz Saldırıları

Bir iletken boyunca akan herhangi bir elektrik akımı elektromanyetik (EM) yayılmalara neden olmaktadır. Kriptografik bir işlem gerçekleştiren cihazın güç tüketimi veriler işlenirken değiştiğinden, bu değişiklik EM alanı için de geçerli olmaktadır. Bu sebeple cihazdan yayılan EM analiz edilerek gizli bilgiler elde edilebilmektedir.

Güç analizinde bazı durumlarda hiçbir bilgi elde edilememektedir. Güç izinde değişimler olmamakta, güç izleri gizli verilerle ilgisiz gibi görünüyor olabilmektedir. Aynı zamanda güç tüketimi cihazın tüm harcadığı gücü temsil etmektedir. Eğer güç tüketimindeki odaklanılan kısım dışındaki tüketimler tespit edilebilirse bir izolasyon yapılabilecek ve daha net sonuçlar elde edilebilecektir. Bu durum güç analizi ile pek mümkün olmazken, EM analizi bu avantajı sağlamaktadır. İlgili yerler EM probu ile dinlenerek gereksiz tüketimler analize dahil edilmemektedir. Ancak burada da odak noktalarının doğru seçilmesi ve probların tasarımı önem arz etmektedir.



Şekil 4.5: Örnek EM Ölçüm Ortamı

Elektromanyetik analiz saldırıları *Basit Elektromanyetik Analiz (SEMA)* ve *Farksal Elektromanyetik Analiz (DEMA)* saldırıları olarak iki ana başlık altında incelenebilir. Bu

açıdan güç analizi için geçerli olan açıklamalar elektromanyetik analiz için de geçerlidir. Elektromanyetik analiz ile güç analizini ayıran en temel fark kriptografik cihazdan ölçüm alma yöntemidir. Ölçüm alındıktan sonra benzer analiz çalışmaları yapılmaktadır.

4.2.2 Hata Oluşturma (Glitch) Saldırıları

Bozucu olmayan saldırılar arasında voltaj ve saat sinyaline müdahale saldırıları en yaygın kullanılan saldırılardandır. Saldırılan sisteme düşük voltaj ve/veya aşırı voltaj verilerek bir koruma devre dışı bırakılmaya veya bir işlemci yanlış işlem yapmaya zorlanabilmektedir (Glitch Saldırıları). Bu nedenle güvenlik tümdevreleri (Secure IC) voltaj ve saat darbelerindeki değişiklikleri algılamak üzere sensörlere sahiptir.

Bu saldırının bozucu veya yarı bozucu kategorisinde olmamasının sebebi, cihaz üzerinde herhangi bir soyma, aşındırma gibi cihaza fiziksel zarar veren işlemlerin yapılmıyor olmasıdır. Örneğin akıllı kartlar için bu saldırı öncesinde paketten çıkarma işlemi yapılmaz. Ancak bozucu ve yarı bozucu bir saldırının uygulanması için paketten çıkarma işleminin yapılması zorunludur.

Elektromanyetik enjeksiyon, ısı artırma gibi farklı türden hata yaptırma atakları da mevcuttur. Ancak bu tez çalışmasında bu hata türlerine değinilmemiştir.

Bu tez çalışmasında geliştirilen yazılımla bir akıllı karta Glitch saldırısı yapılmış ve sonuçları analiz edilmiştir. Detaylar Bölüm 6 altında yer almaktadır.

4.2.3 Kaba Kuvvet Saldırıları

Kaba kuvvet saldırıları, kullanıcı şifresi veya kişisel kimlik numarası (PIN) gibi bilgileri elde etmek için kullanılan bir deneme yanılma yöntemidir. Kaba kuvvet saldırısında, istenen verinin değeriyle ilgili çok sayıda ardışık tahminler üretilir. Kaba kuvvet saldırılarında zaman ve kaynak en önemli iki faktördür. Yani birim zamanda denenebilecek kombinasyon sayısı kaba kuvvet saldırılarının başarısını sağlayan etkendir. Bu sebeple, olabilecek tahmin değerlerini içeren sözlükler kullanılarak hızlı bir şekilde bu ataklar gerçekleştirilebilmektedir.

Kaba kuvvet saldırıları için aşağıdaki önlemler alındığında bu saldırılar genellikle uygulanamaz olmaktadır.

- Karmaşık şifreler veya PIN'ler oluşturularak kombinasyon uzayının genişletilmesi,
- Başarısız giriş sayısının sınırlandırılması (Örneğin 3 hatalı PIN girişi sonrasında PUK kodunun doğrulanmasının beklenmesi)
- Belirli bir başarısız giriş sayısı sonrasında geciktirici bir aksiyon alınması (Örneğin hesabı geçici süre kilitleme)

Bu sebeple kaba kuvvet saldırıları diğer saldırı türlerine destek olarak daha fazla kullanılmaktadır. Örneğin bir şekilde PIN sayacının atlatılması sağlanarak, sınırsız deneme hakkı kazanılması sonrasında kaba kuvvet saldırısının denenmesi.

4.2.4 Yazılımsal Saldırılar

Bir cihaza saldırmanın diğer bir yolu, ara yüz sinyalleri ve erişim protokollerinin analizi ve buralarda açıklık aranmasıdır. Bir güvenlik protokolünün yanlış uygulanması, saldırganın yararlanabileceği bir açık kapı bırakabilmektedir. Bazı mikro denetleyiciler ve akıllı kartlar, tümdevre üzerinde belleğe erişim sağlayan ve üreticinin cihazı test etmesini sağlayan fabrika test ara yüzüne sahiptir. Bir saldırgan bu ara yüzden yararlanmanın bir yolunu bulabildiğinde, tümdevre içerisinde depolanan hassas bilgileri kolayca elde etmiş olacaktır. Test devreleriyle ilgili bilgiler gizli tutulmaktadır. Ancak saldırgan, test moduna geçebilme hedefiyle tümdevre üzerinde çeşitli işlemler deneyebilir. Mikro denetleyiciler için bu tür saldırılar işe yarayabiliyorken akıllı kartlarda genellikle bu tür test devreleri testlerden sonra imha edilmektedir. Ayrıca, gömülü işletim sistemi geliştiricileri tarafından da kimlik doğrulaması olmadan kodlara erişimi önleyecek şekilde kontroller eklenmektedir.

4.3 Yarı Bozucu (Semi-Invasive) Saldırılar

Yarı bozucu saldırılarda, kriptografik cihaz bozucu saldırılarda olduğu gibi paketinden ayrılırken pasivasyon katmanı sağlam kalır. Bununla birlikte, bozucu saldırıların aksine tümdevre yüzeyine doğrudan elektriksel temas yapılmaz. Yarı bozucu saldırılar *aktif* ve *pasif* olarak ikiye ayrılmaktadır.

Pasif yarı bozucu saldırıların amacı normal okuma devreleri kullanılmadan veya problama yapılmadan hafıza hücrelerinin içeriğinin okunmasıdır. Bu tür bir başarılı saldırı [19]'da yayınlanmıştır.

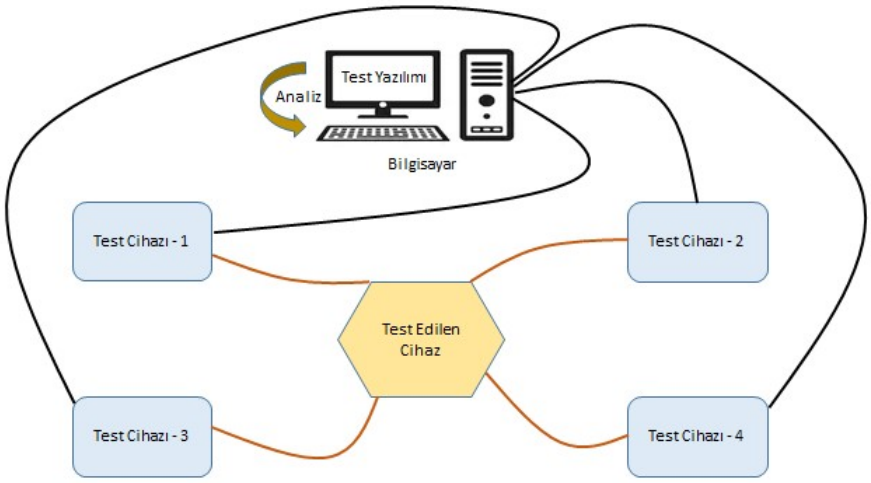
Aktif yarı bozucu saldırıların amacı, test edilen cihazdaki hataları tetiklemektir. Bu tetikleme işlemi, X ışınları, yoğun elektromanyetik alan gibi yöntemlerle yapılabilir. Örneğin, [20]'de bir optik hata örneği yayınlanmıştır.

Yarı bozucu saldırılar, bozucu saldırılar kadar pahalı ekipman gerektirmez. Ancak, yarı bozucu bir saldırı yapmak için gereken çaba hala yüksek sayılabilir. Özellikle, bir modern tümdevrenin yüzeyinde yapılacak bir saldırı için doğru pozisyonun belirlenmesi süreci biraz zaman ve uzmanlık gerektirmektedir. Yarı bozucu saldırılarda en kapsamlı yayın Skorobogatov'un doktora tezi [18] sayılabilir.

Bölüm 5

Geliştirilen Test Yazılımı

Geliştirilen yazılım, Java tabanlı bir test ve analiz yazılımıdır. Yazılım, bozucu olmayan donanımsal saldırılarda kullanılan test cihazlarını yönetme, test cihazlarından alınan sonuçları analiz etme yeteneğine sahiptir. Aynı zamanda geliştirilmeye açık ve literatürde yer alan çeşitli saldırı yöntemleri yazılıma entegre edilebilecek durumdadır. Yazılım geliştirilirken, yazılım kullanılarak saldırı gerçekleştirilecek ürünlerde herhangi bir kısıtlamaya gidilmemiştir. Kriptografik işlemler gerçekleştiren donanımsal ürünler kapsam içerisine alınmıştır. Akıllı kartlar, FPGA ve SASEBO GII gibi çeşitli donanımlar üzerinde yazılım ile bozucu olmayan saldırı türleri uygulanmış ve başarılı sonuçlar alınmıştır.



Şekil 5.1: Test Yazılımı ve Çalışma Ortamı

Geliştirilen test yazılımı, test cihazları ile DLL dosyaları aracılığıyla haberleşmektedir. Test yazılımının olduğu bilgisayar ve test cihazları aynı ağ içerisindedir, IP ve belirli portlar üzerinden birbirine bağlanmış durumdadır. Dışarıdan herhangi bir müdahale olmaması için bu ağ izole bir ağ olarak oluşturulmuştur. Ağda yalnızca test yazılımının olduğu bilgisayar ve test cihazları vardır.

Yazılım hazırlanırken öncelikle yazılımdan beklenen ihtiyaçlar belirlenmiş, sonrasında ihtiyaç doğrultusunda uygun yazılım dili seçilmiştir. Seçilen yazılım dili ile kodlama gerçekleştirilmiş, yazılımın istenen ihtiyaçları karşılayıp karşılamadığı ürün testi ile kontrol edilmiştir. Süreç aşağıdaki alt başlıklarda detaylı olarak verilmiştir.

5.1 Test Yazılımından Beklenen Gereksinimlerin Belirlenmesi

Bu tez çalışması kapsamında hazırlanan test yazılımı için, öncelikle ihtiyaç analizi yapılmıştır. Yani test yazılımının yapması beklenen işlemlerin neler olduğu, bu işlemlerin hangi ihtiyaçlar için çözüm sunacağı belirlenmiştir.

Yazılımdan beklenen isterler özet olarak aşağıdaki gibidir:

- Test cihazlarının kontrolü bu yazılım üzerinden sağlanmalı, aynı anda birden fazla test cihazının kullanılabileceği durumlarda, cihazların birbiri ile uyumlu ve doğru çalışmasından bu yazılım sorumlu olmalı,
- Test edilecek cihaza gönderilecek veriler, yazılım ile cihaza gönderilmeli, ekstra veri gönderimine gerek kalmamalı,
- Test cihazlarından ve test edilen cihazdan alınacak sonuçlar, bu yazılım tarafından alınmalı ve sonuçlar üzerinde yapılacak işlemler hazırlanan yazılım ile yapılabilmeli,
- Test cihazlarından alınan sonuçlar istenirse dışarıya istenilen formatta aktarılabilmeli,
- Test cihazlarından alınan sonuçlar yazılım tarafından analiz edilebilmeli,
- Yazılıma literatürde yer alan analiz yöntemleri entegre edilebilmeli,

- Yazılım, üzerinde koşturulduğu bilgisayarın işletim sisteminden bağımsız olarak çalışabiliyor olmalıdır.

Yukarıda verilen isterler belirlenirken, güvenlik testleri gerçekleştiren laboratuvarların temel ihtiyaç ve gereksinimleri göz önünde bulundurulmuş ve geleceğe yönelik olası ihtiyaçları doğrultusunda analiz çalışmaları yapılmıştır. Buradaki en önemli hususlardan birisi yazılımın sürekli geliştirilebilir olmasıdır. Yeni bir donanımsal saldırı cihazı veya atak yöntemi yazılımın içerisine eklenebilir olmalıdır. Bu test yazılımı hazırlanırken hem yazılımın kullanıcısı hem de geliştiricisi olarak hareket edildiği için, isterleri belirleyen ve onları uygulamaya döken taraf aynı olmuştur. Bu sebeple yazılımın geliştirilebilir olmasının gelecekteki ister ve tespitler için de önemli bir etken olduğu düşünülebilir.

Hazırlanan test yazılımı aynı zamanda test cihazlarından gelen verileri de işlediği ve gerektiğinde analiz ettiği için, üzerinde koştuğu bilgisayara yönelik de bazı isterler ortaya çıkmıştır. Yazılımın başarılı ve hızlı çalışması, sonuçları doğru ve seri bir şekilde işleyebiliyor olması önemli bir unsurdur. Bu sebeple geliştirilen yazılımın kullanıldığı bilgisayarın donanımsal özellikleri de aşağıda verilmiştir. Aynı zamanda bu özellikler minimum sistem gereksinimleri olarak da düşünülebilir:

- 32 GB DDR3 RAM,
- 3.0 TB SATA 7200 rpm, 6 Gb/s 3.5" HDD, mümkünse SSD tavsiye edilmektedir. Çünkü cihazlardan alınan verinin boyutu büyüdükçe, bu süre açısından ciddi bir negatif etki oluşturmaktadır.
- Intel Z820 Xeon E5-2620 2.00 GHz iki işlemci,
- NVIDIA Quadro 410, 4GB ekran kartı,

Burada sistem kaynaklarını kullanan en önemli kısım yazılımın test sonuçlarını analiz ettiği kısımdır. Özellikle Donanımsal Saldırılar bölümünde anlatılan Farksal Güç Analizi ve Farksal Elektromanyetik Analiz saldırıları ciddi sistem kaynağı kullanan, içerisinde birçok istatiksel, matematiksel ve karşılaştırma işlemleri barındıran saldırı türleridir. Diğer saldırı türleri için de analiz türlerine göre sistem kaynakları önemli bir konu olmaktadır. Bölüm 6'da gerçekleştirilen saldırı için de durum böyledir.

5.2 Yazılım Dilinin ve Geliştirme Ortamının Belirlenmesi

Bu bölümün giriş kısmında da belirtildiği gibi yazılım dili olarak Java seçilmiştir. Yazılımda Java 8, Enterprise Edition, JDK (Java Development Kit) 8 kullanılmıştır. Java dilinin tercih edilmesindeki en büyük sebep, dilin çok yaygın olarak kullanılması, geliştirme esnasında bir sorun ile karşılaşıldığında çözümün kolayca bulunabilecek olmasıdır. Ayrıca Java dilinin yaygın olarak kullanılmasının, bu yazılımın geliştirilmesine, yaygınlaştırılmasına ve sürdürülebilir olmasına katkı sağlayacağı düşünülmüştür. Aynı zamanda Java, farklı işletim sistemleri ile kullanımda uyum problemi yaşanmayacak yazılım dilleri arasındadır.

IDE olarak (Integrated Development Environment - Tümleşik Geliştirme Ortamı) IntelliJ IDEA kullanılmıştır. Bu programın seçilmesinde, program geliştiricileri tarafından programın sürekli geliştiriliyor olması, yazılım geliştiricileri için kullanışlı bir arayüz sunması etkili olmuştur. Test Yazılımı, programın 2018.1 versiyonu ile geliştirilmeye başlanmış, 2018.2 versiyonu ile geliştirilmeye devam edilmiştir. Bu versiyondan sonraki versiyon güncellemeleri takip edilmiş, yazılımı etkileyen bir değişiklik olmadığı sürece programı yükseltme işlemi gerçekleştirilmemiştir. Test yazılımının kullanıldığı ağ, bir iç ağ olduğu ve yazılım güvenli bir ortamda çalıştırıldığı için bu durumla ve Java, JDK versiyonu ile ilgili herhangi bir tehdit öngörülmemiştir.

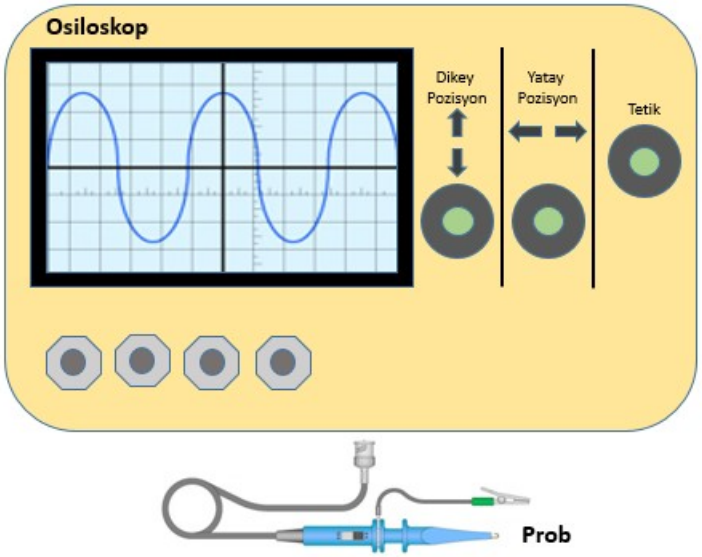
5.3 Test Yazılımında Kullanılan Cihazlar

Donanımsal saldırılar gerçekleştirilirken birçok gelişmiş cihaz ve ölçüm sistemleri kullanılmaktadır. Bu cihazlar gelişen teknoloji ile birlikte her geçen gün gelişmekte, farklı ve daha gelişmiş saldırılar uygulanmasını mümkün hale getirmektedir. Cihaz geliştiricileri arttıkça rekabetçi bir ortam oluşmakta, cihazlar hızla yaygınlaşmaktadır. Ayrıca bu cihazlar yaygınlaştıkça bu tür saldırılar, önceden laboratuvar ortamlarında araştırma amacıyla gerçekleştirilirken her ortamda yapılabilir hale gelmektedir. Bu sebeple gelişen teknoloji daima takip edilerek ürün geliştirme aşamasında bu hususa dikkat edilmesi önem arz etmektedir.

Geliştirilen test yazılımına entegre edilen donanımsal saldırılarda kullanılan cihazlar aşağıda ilgili başlıklar altında genel olarak belirtilmiştir. Ancak bu çalışma içerisinde özellikle hiçbir marka ve model hakkında bilgi verilmemiştir.

5.3.1 Osiloskop

Osiloskoplar, elektronik geliştirme, üretim, test ve servis hizmetleri için çok yaygın kullanılan, en kullanışlı test araçlarından biridir. Osiloskop, test edilen cihazdaki gerilimin zamana bağlı olan değişimlerini iki eksenli olarak gösteren test cihazıdır. Genel olarak analog ve dijital osiloskop olarak iki kısma ayrılmakta ancak dijital osiloskoplar yaygın olarak kullanılmaktadır. Bir osiloskop temelde, ekran, dikey kontroller, yatay kontroller ve tetik kontrollerinden oluşmaktadır.



Şekil 5.2: Temel Bir Osiloskop Görseli

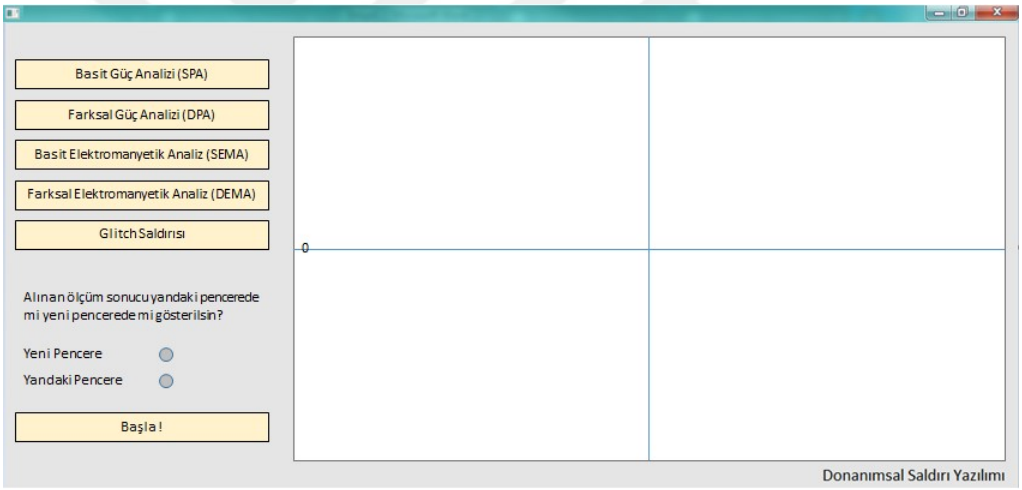
Ekran, alınan ölçüm sonucunun gösterildiği ara yüzdür. Dikey kontroller görüntülenen sinyalin genliğini kontrol etmeyi sağlamaktadır. Yatay kontroller zaman ekseninin kontrolünü sağlar. Tetik ise ölçüm alınması için şartı belirtir. Kriptografik cihazlardan alınan ölçümler osiloskopa prob aracılığıyla aktarılır. Osiloskop bu tez kapsamında, kriptografik cihazlardan alınan güç ve elektromanyetik sinyallerin ölçümü için kullanılmıştır.

5.3.2 Voltaj/Saat Glitch Cihazı

Bu tür cihazlar, akıllı kartlar gibi kriptografik donanımların gerçekleştirdikleri kritik işlemler esnasında cihaza müdahalede bulunmayı sağlarlar. Saat ve voltaj sinyallerinde pozitif ve negatif bir etki oluşturabilirler. Bu etki çok küçük süreler için (örneğin nano saniye) yapılabilmektedir.

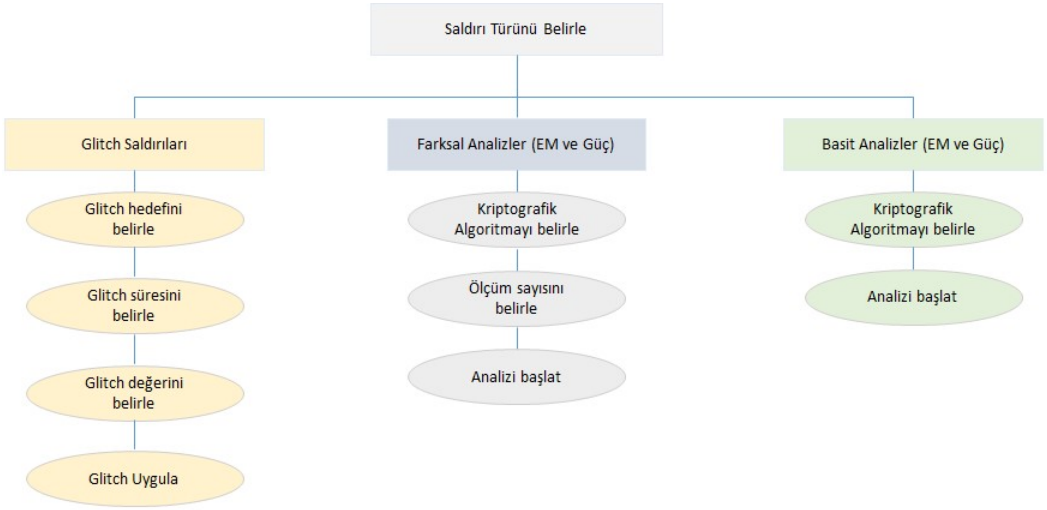
5.4 Hazırlanan Test Yazılımının Tasarımı ve İşlevleri

Test yazılımı hazırlanırken, asıl hedef olarak bozucu olmayan saldırıların test yazılımı ile yapılması seçilmiş ve bu noktaya odaklanılmış, ara yüz tasarımlarının üzerinde çok durulmamıştır. Bu sebeple karmaşık olmayan, bilgilendirici ve yönlendirici bir ara yüz tasarlanmıştır.



Şekil 5.3: Kullanıcı Ara Yüzü

Test yazılımı, bozucu olmayan saldırılardan Güç Analizi, Elektromanyetik Analiz ve Glitch Saldırılarını hedeflemektedir. Bu sebeple öncelikle hangi saldırının yapılacağı belirlenmelidir. Saldırı belirlendikten sonra saldırı türüne göre işlemler farklılık göstermektedir. Arka tarafta işleyen süreç, genel hatlarıyla aşağıdaki şekil üzerinden açıklanmıştır:



Şekil 5.4: Yazılımın Akışı

Ölçüm alındıktan sonra benzer işlemler gerçekleştirildiği için yukarıdaki şekil üzerinde, Farksal Güç Analizi ile Farksal Elektromanyetik Analiz, Basit Güç Analizi ile Basit Elektromanyetik Analiz birlikte ele alınmıştır. Glitch saldırısı tamamen farklı bir saldırı türü olduğu için ayrı tutulmuştur. Yukarıdaki şekil üzerinden anlatıma devam edilirse, öncelikli hedef, saldırı türünün belirlenmesidir.

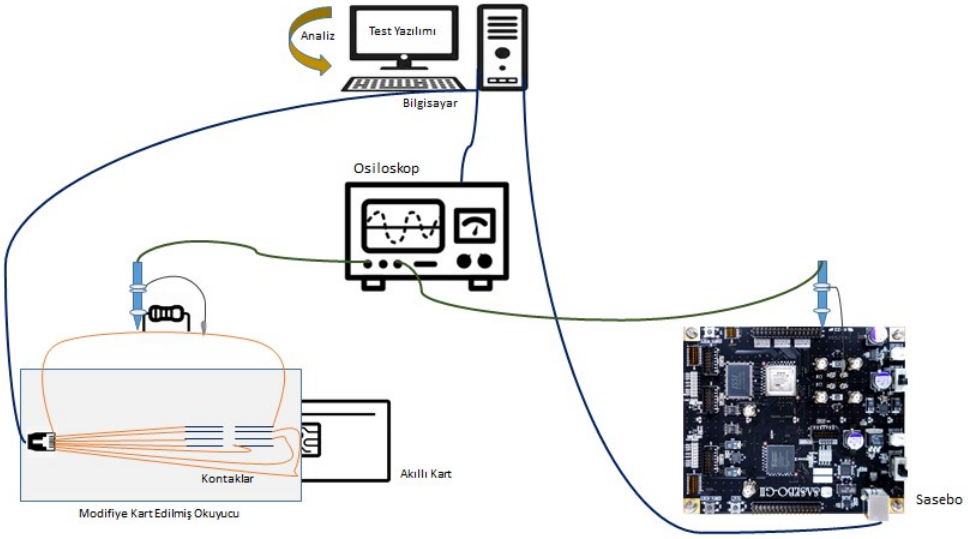
Eğer saldırı türü **Glitch Saldırısı** olarak seçilmişse, mevcut test cihazlarından dolayı bu saldırı sadece akıllı kartlara uygulanabilecektir. Saldırıda Osiloskop, Glitch cihazı ve akıllı kart okuyucular kullanılacaktır. Burada kullanılacak akıllı kart okuyucular pasif görevde, yazılım ile etkileşim içerisinde olmayacaktır. Bu kart okuyucuların kullanım sebebi, osiloskopla ölçüm almayı kolaylaştırmaktır. Glitch cihazı üzerinden osiloskop ile ölçüm almak mümkün olmadığı için böyle bir yöntem seçilmiştir. Akıllı kart ile haberleşme Glitch cihazı üzerinden sağlanmaktadır. Çünkü kullanılan Glitch cihazı aynı zamanda bir akıllı kart okuyucusu görevi gerçekleştirmektedir. Ekstra akıllı kart okuyucu kullanılmasının sebebi yukarıda da bahsedildiği gibi, atak esnasında aynı zamanda ölçümde alınarak atağın kontrollü olarak gerçekleştirilmesidir. Burada test yazılımının, iki cihaza bağlanıp iki cihazı aynı anda kontrol etmesi bu atak için yeterli olacaktır. Yazılımın yaptığı işlemler açıklayıcı olması açısından aşağıda sözde kod ile verilmiştir. **G-C** - Glitch Cihazını, **Os** - Osiloskopu temsil etmektedir:

```
Bağlan_G-C (ip-adresi):
    Bağlantıyı aç
        Eğer başarısız ise hata dön
    Kart takılı mı (ATR alınabildi mi)?
        Eğer başarısız ise hata dön
    Başarılı cevap dön

Bağlan_Os (ip-adresi):
    Bağlantıyı aç
        Eğer başarısız ise hata dön
    Başarılı cevap dön

Glitch_Hazirla (tür, süre, değer):
    tür == Voltaj Glitch
        Voltaj_Glitch-Uygula(süre,
        değer)
    tür == Saat Glitch
        Saat_Glitch-Uygula(süre,
        değer)

Voltaj_Glitch-Uygula (süre, değer):
    Os_Ölçüm-İçin-Hazırla
        Eğer başarısız ise hata dön
    komut = «Glitch Uygulanacak
    Komut»
    Glitch Gönder (Vcc, komut, süre,
    değer)
        Eğer başarısız ise hata dön
    Başarılı cevap dön

Saat_Glitch-Uygula (süre, değer):
    Os_Ölçüm-İçin-Hazırla
        Eğer başarısız ise hata dön
    komut = «Glitch Uygulanacak
    Komut»
    Glitch Gönder (Clk, komut, süre,
    değer)
        Eğer başarısız ise hata dön
    Başarılı cevap dön
```

Şekil 5.5: Glitch Saldırısı Sözde Kod

Verilen sözde kod, Glitch uygulaması için yapılan işlemleri anlatmaktadır. Bu işlemlerden sonra osiloskoptan ve Glitch cihazından alınan sonuç değerleri yazılıma geri gelmekte ve yazılım içerisinde işlenmektedir. Bu tez çalışmasında bir Glitch saldırısı uygulaması Bölüm 6'da verdildiği için, alınan cevapların yorumlanması ve analizine ilgili bölümde detaylı olarak yer verilmiştir.

Eğer saldırı türü **Farksal Güç Analizi** veya **Farksal Elektromanyetik Analiz** olarak seçilmişse, bu saldırılar akıllı kartlar, FPGA gibi kriptografik işlem gerçekleştiren donanım ürünleri üzerinde uygulanabilir olacaktır. Saldırıda osiloskop ve saldırı akıllı kart üzerinde yapılıyor ise kart okuyucu kullanılacaktır. Eğer akıllı kart kullanılıyorsa, kart okuyucu, test yazılımı ile akıllı kartın haberleşmesini sağlayacaktır. SASEBO gibi FPGA tabanlı kriptografik bir cihaz test edilmek istendiğinde, test yazılımı SASEBO ile doğrudan DLL dosyası ile haberleşmesini yapacak, cihazı test ortamında kullanılabilir hale getirecektir. Aşağıdaki şekil, güç analizi ölçümü almak için örnek bir test ortamını göstermektedir. Elektromanyetik analiz için sadece ölçümü alma yöntemi farklılık göstermektedir. Yani aşağıdaki şekilde problar elektromanyetik prob ile değiştirilmiş olacaktır. Ayrıca aşağıdaki şekilde iki test edilecek cihaz aynı anda yazılıma ve osiloskopa bağlı görünmektedir. Ancak osiloskop ve test yazılımı ile aynı anda sadece bir cihaz test edilebilir. Şeklin bu şekilde verilmesinin sebebi, bağlantıların bir örnek resim üzerinden ifade edilmesinin istenmesidir.



Şekil 5.6: Güç ve Elektromanyetik Analiz için Test Ortamı

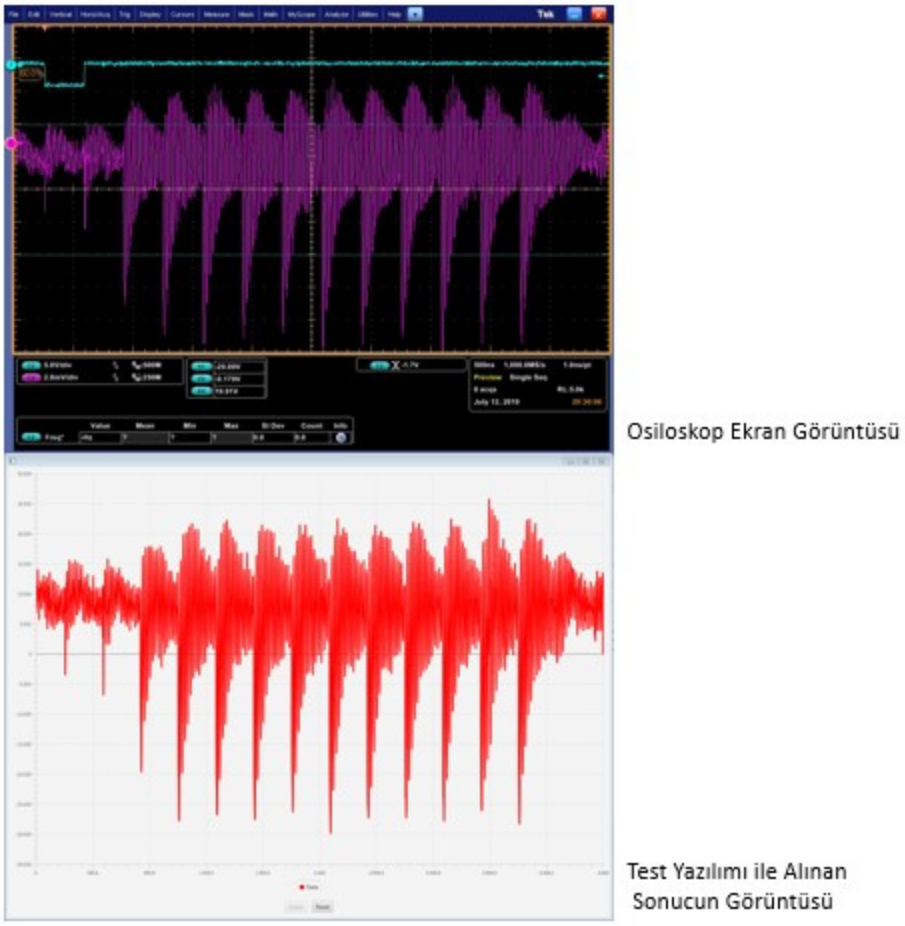
Farksal Güç ve Elektromanyetik Analizlerde çok sayıda ölçüm alınmakta ve bu ölçümler analiz işlemine sokularak hassas veriler elde edilmeye çalışılmaktadır. Test yazılımında yapılan işlemler aşağıda sözde kod olarak verilmiştir.

```
Ölçüm Al(ölçüm sayısı):
        İ = 0;
        İ < ölçüm sayısı:
                Os-Ölçüm Al
                Alınan Ölçüm dosyaya kaydet
        Başarılı cevap dön

Hamming Weight Hesapla (ölçüm dosyaları):
        Saldırılacak nokta aralığını belirle
        1 değerine sahip bit sayılarını bul
        Başarılı cevap dön

Hamming Distance Hesapla (ölçüm dosyaları):
        Saldırılacak nokta aralığını belirle
        Değişen bit sayılarını bul
        Başarılı cevap dön

Anahtar Uzayı Oluştur ():
        Algoritma türünü belirle
        Anahtarın bir bayt veya bir bit değeri için olasılıklarını oluştur.
        Başarılı cevap dön

Tahmin Matrisleri Oluştur ():
        Her bir anahtar değeri için az güç tüketen ve çok güç tüketen olarak iki grup oluştur (Hamming Distance veya Hamming Weight sonucuna göre), alınan ölçümleri bu gruplara girdi kabul et

Çıkarma İşlemi ()
        Her bir anahtar değeri için oluşturulan tahmin matrislerini birbirinden çıkar. Pik görülen sonuçtaki anahtar değeri, anahtarın gerçek değeridir.
```

Şekil 5.7: DPA - DEMA Sözde Kod

Eğer saldırı türü **Basit Güç Analizi** veya **Basit Elektromanyetik Analiz** olarak seçilmişse, Farksal Analiz de olduğu gibi, bu saldırılar akıllı kartlar, FPGA gibi kriptografik işlem gerçekleştiren donanım ürünleri üzerinde uygulanabilir olacaktır. Saldırıda kullanılan cihazlar Farksal Analiz ile aynı cihazlardır; yani osiloskop ve saldırı akıllı kart üzerinde yapılıyor ise kart okuyucu kullanılacaktır. Haberleşme de Farksal Analiz ile benzer şekilde olmaktadır.

Basit Güç ve Elektromanyetik Analizde ölçüm sayısı genellikle bir olarak kullanıldığı için test yazılımı, kriptografik işlem esnasında ölçülen güç tüketimi veya elektromanyetik yayılımı doğrudan çıktı olarak dışarıya vermektedir. Aşağıda AES 128 bit şifreleme işlemi esnasında SASEBO GII'den aldığımız güç tüketim ölçüm sonucu verilmiştir.



Şekil 5.8: SASEBO GII AES 128 Bit Güç Tüketimi

5.5 Geliştirilen Yazılımın Kontrol ve Test Edilmesi

Yazılımın geliştirme aşamasında çeşitli problemlerle karşı karşıya kalınmıştır. Osiloskop ekranında görünen ölçüm ile osiloskoptan yazılıma gelen ölçüm değerinin (ki bu değer kimi zaman 50 milyon nokta ifade etmektedir) yazılımda işlenerek ekrana basılması bile önemli bir problemin çözümü olarak karşımıza çıkmıştır. Çeşitli sebeplerle bellek taşmaları sonucunda yazılımın hata vererek sonlanması gibi durumların çözülmesi ve tekrarlanmaması için alınan önlemlerin işe yarıyor olması, yazılımda yapılan analiz işlemlerinin doğru yapılıyor olması kritik bir öneme sahiptir. Tüm bu sebeplerle geliştirilen yazılımın

doğru ve istikrarlı çalışması ile ilgili alınan önlemlerin işe yaradığı test edilerek kontrol edilmiştir.

Yazılımın yaptığı farksal analizin (Elektromanyetik ve Güç) doğru yapılıyor olduğu, daha önceden alınan ve başka ortamlarda analiz edilen test verileri [53] ile kontrol edilmiştir. Örnek bir kontrol (Farksal Güç Analizi) aşağıda verilmiştir. Kontrol için kullanılan örnek test verisinin özellikleri aşağıdaki gibidir:

- Alınan ölçümler, FPGA üzerinden alınan güç ölçümleridir.
- 200 farklı veri ile AES 128 bit şifreleme işlemi yapılmaktadır, anahtar değeri sabittir ve bilinmemektedir. Her bir ölçüm 5000 nokta ile ifade edilmektedir.
- Ölçümler, AES şifreleme işleminin ilk S kutusu dönüşümünün hemen sonrasına yoğunlaşmaktadır. Bilinmeyen AES anahtarının ilk baytına atak gerçekleştirilecektir.

Yazılımda yapılan farksal analiz aşağıda adım adım verilmiştir:

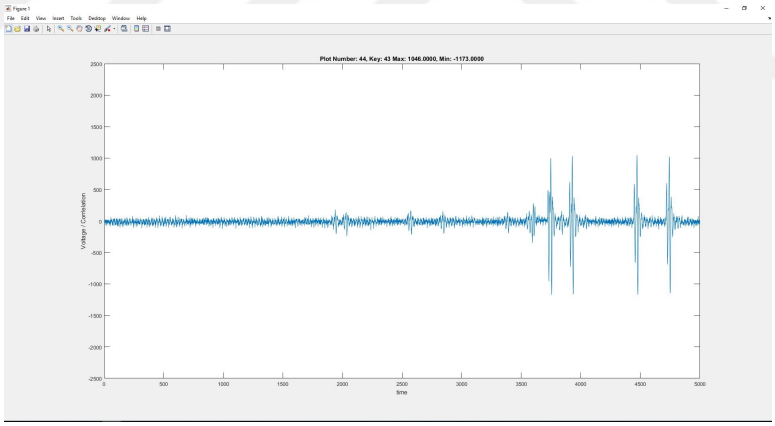
- Öncelikle anahtarın ilk baytı için olası tüm anahtar değerleri oluşturulmakta ve bu değerler bir diziye yazılmaktadır (0, ..., 255) (dizinin boyutu 1×256).
- 200 farklı açık verinin, AES anahtarının ilk baytı ile işleme giren kısmı, yani verinin ilk baytı bir diziye atılmaktadır (dizinin boyutu 200×1)
- Giriş verileri ve tahmini anahtar değerleri XOR'lanmakta ve çıkan sonuçlar, S kutusu dönüşümüne sokulmaktadır. Bu sonuçlar da bir diziye aktarılmaktadır. Bu dizi, 200 verinin ilk baytının olası tüm AES değerleri ile işlem sonucunu vermektedir. Bu sebeple dizinin boyutu 200×256'dır. Örneğin; 1. verinin, olası AES anahtarları için (256 farklı durum) işlem sonucu bu dizide yer almaktadır, 1.satırdaki tüm değerler, 1.verinin 256 farklı AES anahtarı ile işlem sonucunu içermektedir.
- S kutusu dönüşümü sonucunda elde edilen 200×256'lık dizideki her veri için Hamming Weight hesabı yapılmaktadır. Bunun için öncelikle veriler ikili sayı tabanına dönüştürülmekte, ikili sayılardaki 1'lerin sayısı yeni bir diziye aktarılmaktadır. 1 bayt için en yüksek Hamming Weight değeri 8 (1111 1111), en düşük Hamming

Weight değeri 0 (0000 0000)'dır. Bu sebeple Hamming Weight değeri 4'ten küçük olanlar az güç tüketen, 4'ten büyük olanlar çok güç tüketen gruba alınmakta, 4 çıkanlar analize dahil edilmemektedir. Bu sonuçlar da bir diziye atılmaktadır (200×256).

- Son aşama olarak, gerçek ölçüm sonuçları yukarıdaki tahminlere göre gruplandırılmakta ve az güç tüketimine sahip ölçümler, çok güç tüketimine sahip ölçümlerden çıkarılmaktadır.

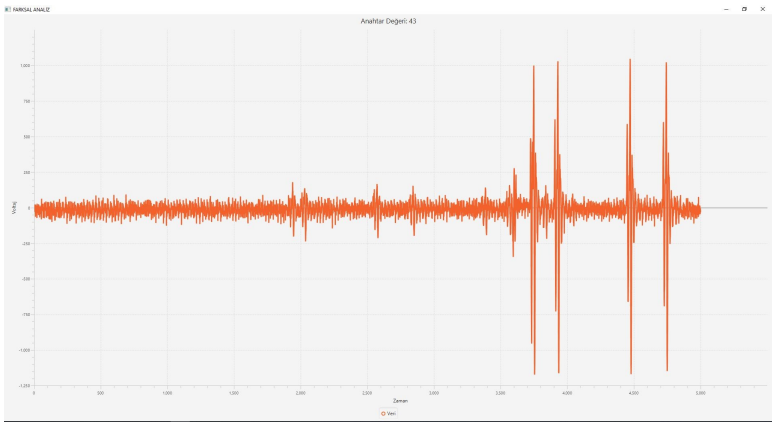
Tüm bu işlemler gerçekleştiğinde sonuç olarak 256×5000'lik bir dizi elde edilmektedir. Bu dizideki satırlar tahmini AES anahtar değerlerini, sütunlar ise Hamming Weight sonrası birbirinden çıkarılan noktaları ifade etmektedir.

Yukarıda verilen analiz aşamaları hem MATLAB hem de test yazılımında gerçekleştirilmiş ve anahtarın ilk baytının değeri 43 olarak bulunmuştur. Aşağıdaki görsellerde hem MATLAB analiz sonucu hem de test yazılımının analiz sonucu verilmiştir. Anahtar değeri doğru bayt için MATLAB analiz sonucu:



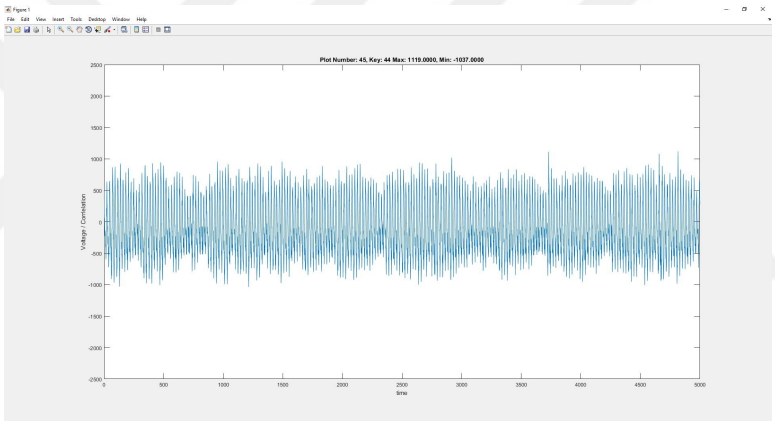
Şekil 5.9: MATLAB Doğru Anahtar için Farksal Analiz Sonucu

Anahtar değeri doğru bayt için hazırlanan yazılımın analiz sonucu:

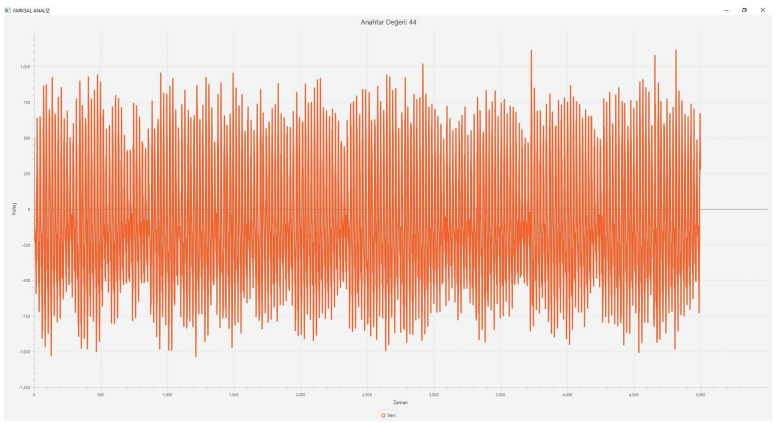


Şekil 5.10: Hazırlanan Yazılımın Doğru Anahtar için Farksal Analiz Sonucu

Aşağıda anahtar tahmin değeri yanlış olan örnek bir bayt değeri (44) için MATLAB ve hazırlanan test yazılımının analiz sonuçları peş peşe verilmiştir:



Şekil 5.11: MATLAB Yanlış Anahtar Değeri (44) için Farksal Analiz Sonucu

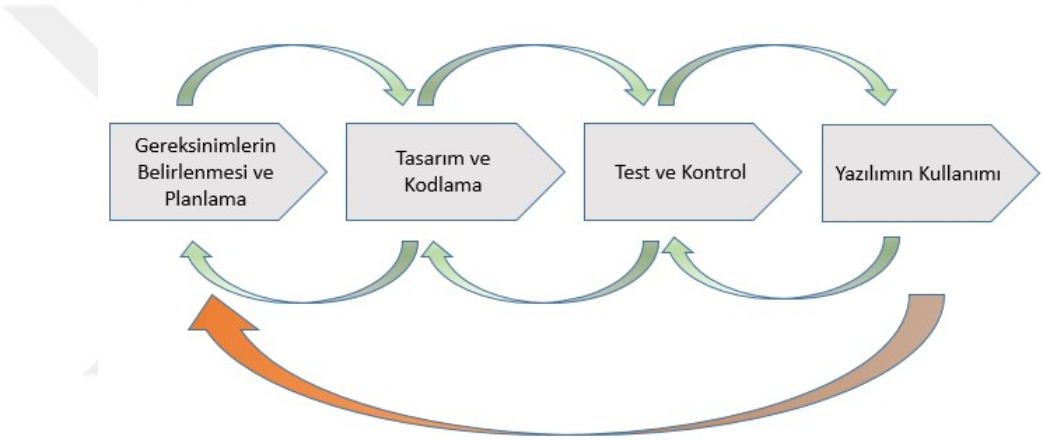


Şekil 5.12: Hazırlanan Yazılımın Yanlış Anahtar Değeri (44) için Farksal Analiz Sonucu

5.6 Yazılımın Aktif Kullanımı ve Yaşam Döngüsü

Geliştirilen test yazılımı, ihtiyaçlar doğrultusunda şu anda aktif olarak güvenlik değerlendirmelerinde kullanılmaktadır. Bu durumun, yazılımın geliştirilmesine ve iyileştirilmesine katkı sağlayacağı düşünülmektedir. Yazılımın kullanılması gereken koşullar çoğaldıkça yazılımın da sürekli değişeceği ve her geçen gün daha iyiye gideceği muhakkaktır. Geliştirilen yazılım için ihtiyaçlar değiştikçe ve arttıkça yazılımda güncellemeler yapılması gerekebileceği, bu güncellemelerin seri bir şekilde gerçekleştirilebilmesi için dinamik bir yaşam döngüsü modeline ihtiyaç bulunmaktadır.

Yazılımı geliştirirken, aşağıdaki döngü takip edilmiştir.



Şekil 5.13: Yazılım Yaşam Döngüsü

Döngüden de görülebileceği gibi, yazılımın sürekli geliştirilebilir olması ve isteklere seri cevaplar verebilmesi gerekmektedir. Her aşamadan bir diğer aşamaya geçilebilmektedir. Örneğin testler esnasında bir sorun tespit edildiğinde, kodlamaya geri dönülerek sorunun çözümlenmesi ve testlerin tekrar edilmesi.

Bölüm 6

Geliştirilen Yazılım ile Glitch ve Zaman Saldırısı Uygulaması

Bu bölümde, hazırlanan test yazılımı ile bir akıllı kart uygulamasına gerçekleştirilen Voltaj Glitch saldırısı detaylı olarak incelenecektir. Glitch saldırısı PIN Verify [54] işlemi-ne gerçekleştirilmiştir. Amaç, PIN bilgisi bilinmeyen bir kartın PIN değerini ele geçirmek olacaktır. Akıllı kart tümdevreleri genellikle güvenlik kaygıları ile geliştirildiği için, tümdevreler, Glitch saldırısı esnasında, Glitch sensörleri ile saldırıyı tespit etmekte ve kart reset almaktadır. Hatalı işlem yapılamaması amacıyla işlem yarıda kesilerek süreç sonlandırılmaktadır. Saldırı uygulanacak akıllı kartın tümdevresi de yüksek seviyede bir Ortak Kriterler Değerlendirmesine tabi tutulmuş ve bu standarttan sertifikalandırılmış bir tümdevredir. Üzerinde koşan akıllı kart işletim sistemi için aynı durum söz konusu değildir.

Bu saldırıda, akıllı kart tümdevresinin aldığı güvenlik önleminin avantaja dönüştürülmesi ve PIN sayacının devre dışı bırakılması hedeflenmiştir.

Yapılan işlemler alt bölümlerde detaylı olarak verilmiştir.

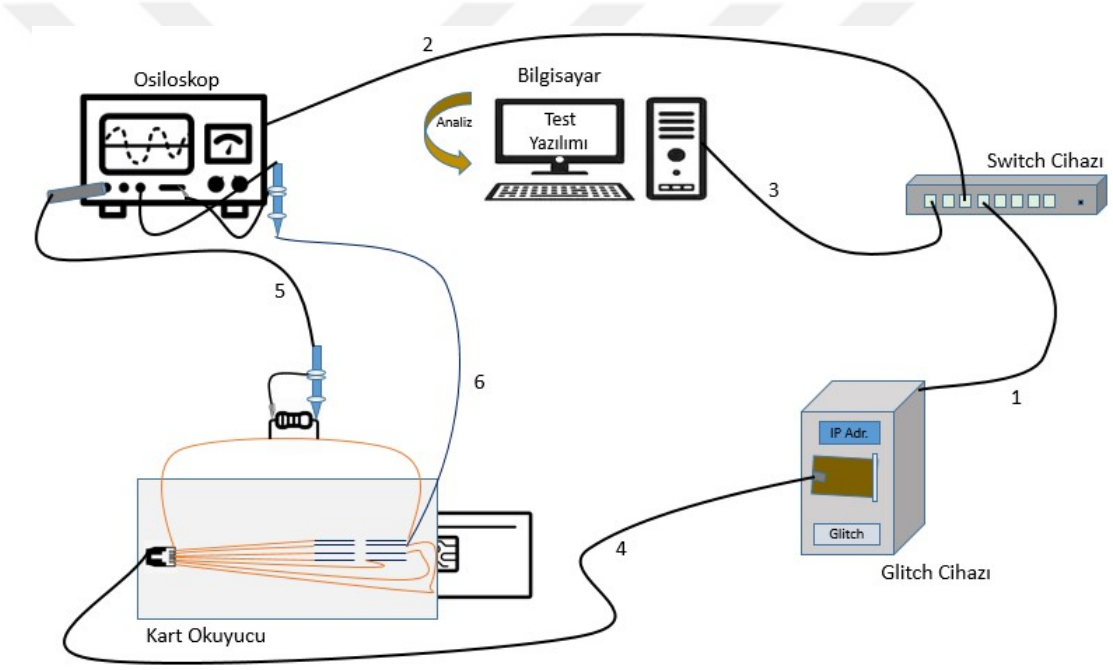
6.1 Test Ortamının Hazırlanması

Saldırı için kullanılacak test ortamının hazırlanması, saldırının ilk aşamasıdır. Saldırı esnasında Glitch cihazı, osiloskop, voltaj probu (500 MHz, 10 M-Ohm), ölçümü iyileştirmek

için filtre (50 Ohm, DC-23 MHz, Low Pass) kullanılmıştır . Glitch cihazı ve osiloskop test yazılımına bağlı olacak ve işlemler yazılımın kontrolü altında gerçekleştirilecektir.

Akıllı karta PIN Verify komutu Glitch cihazı aracılığıyla gönderilecektir. Test yazılımına Glitch cihazına gönderilecek komut ve voltaj hattına koyulacak kesintilerle ilgili bilgiler verilecektir. Komutu göndermeden önce osiloskop cihazının ölçüm almaya hazır hale getirilmesi işini test yazılımı gerçekleştirecektir. Sonrasında komut gönderildikten sonra alınan sonuçlar, cihazlardan test yazılımı ile toplanacak ve yazılım tarafından bu veriler analiz edilecektir.

Test ortamı Şekil 6.1 ile verilmiştir. Şekilde verilen numaraların detaylı açıklamaları şeklin altında yer almaktadır.



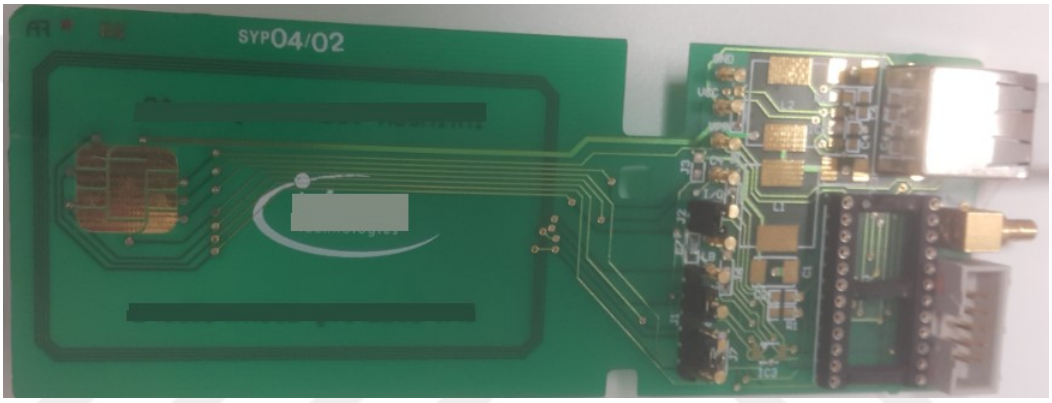
Şekil 6.1: Glitch Test Ortamının Hazırlanması

(1) ile belirtilen bağlantı, Glitch cihazını RJ45 konektörü üzerinden Switch ile bağlar. Bu bağlantıdaki amaç, Glitch cihazı ile test yazılımının yüklü olduğu bilgisayarı aynı ağa dahil etmektir. Çünkü her iki cihaz birbiri ile IP üzerinden haberleşmekte, bu durum da aynı anda istediğimiz kadar cihazı kullanabilmeyi sağlamaktadır. Glitch cihazı ile test yazılımının haberleşmesi buradan sağlanmaktadır. Detaylar Bölüm 6.2 altında verilmiştir.

(2) ile belirtilen bağlantı, (1) ile benzer şekilde, osiloskopu, test yazılımının yüklü olduğu bilgisayar ile aynı ağa dahil eder. Test yazılımı osiloskop ile de IP üzerinden haberleşmektedir. Gelen giden veriler bu hat üzerinden aktarılmaktadır.

(3) ile belirtilen bağlantı, bilgisayarı test cihazları ile aynı ağa dahil eder.

(4) ile belirtilen bağlantı, saldırı uygulanacak kart üzerinden güç ölçümü alabilmek için yapılmıştır. Glitch cihazı üzerinden güç ölçümü almak mümkün olmayacağı için, Glitch cihazına akıllı kart yerine Şekil 6.2 verilen aparat takılmıştır. Bu aparatın ucuna modifiye edilerek hazırlanan akıllı kartın takılı olduğu kart okuyucu takılmıştır. Böylece osiloskop probları ile ölçüm almak mümkün hale gelmiştir.



Şekil 6.2: Akıllı Kart Okuyucu ile Glitch Cihazı Bağlantı Aparatı

(5) ile verilen kısımda, kartın toprak ucuna bağlanan direnç üzerinden prob ile ölçüm alınmakta, alınan ölçüm filtreden geçirilerek osiloskopa gitmektedir.

(6) ile verilen kısımda, kartın I/O ucuna prob bağlanmıştır. Osiloskop üzerinde, tetik noktası olarak, I/O ucuna bağlanan probun takılı olduğu kanal seçilmiştir. Bu bağlantıdaki amaç I/O hattında değişiklik olduğunda osiloskopun tetik almasını sağlamaktır. Yani PIN Verify komutu karta gönderildiğinde, I/O hattında oluşacak dalgalanmadan dolayı osiloskop tetiklenecek ve PIN Verify için kartın güç tüketim değerini, toprak hattına bağlanan prob üzerinden alacaktır.

Test ortamının hazırlanması sürecinde, test yazılımında da aşağıdaki ayarlamaların yapılması gerekmektedir:

1. Toprak hattına bağlanan prob, osiloskopun hangi kanalına bağlanacaksa, yazılım üzerinde de ölçüm alınacak kanal olarak o kanal seçilecek,

2. Glitch koyulacak komut, yani PIN Verify komutu, yazılıma verilecek, Örneğin, PIN-1 1234 denemesi için 002000010431323234

3. Glitch için voltajın kaça indirileceği ve bu işlemin ne kadar süre için yapılacağı bilgisi verilecek, Örneğin; 1000 mV, 20 ns

4. Glitch'in işlem başladıktan ne kadar süre sonra koyulacağı bilgisi verilecektir.

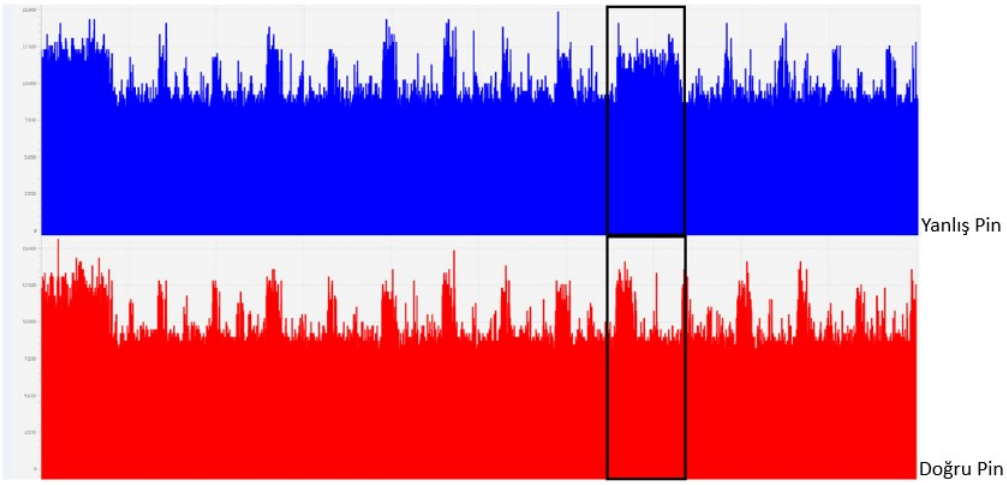
Tüm bu hazırlık aşamaları tamamlandıktan sonra Bölüm 6.2'daki işlemlerle devam edilecektir.

6.2 Saldırının Gerçekleştirilmesi

Saldırı gerçekleştirilen tümdevre güvenli bir tümdevre olduğu için Glitch esnasında reset almakta ve hatalı işlem yapmaya izin vermemektedir. Bu sebeple saldırı aşağıdaki şekilde gerçekleştirilmiştir:

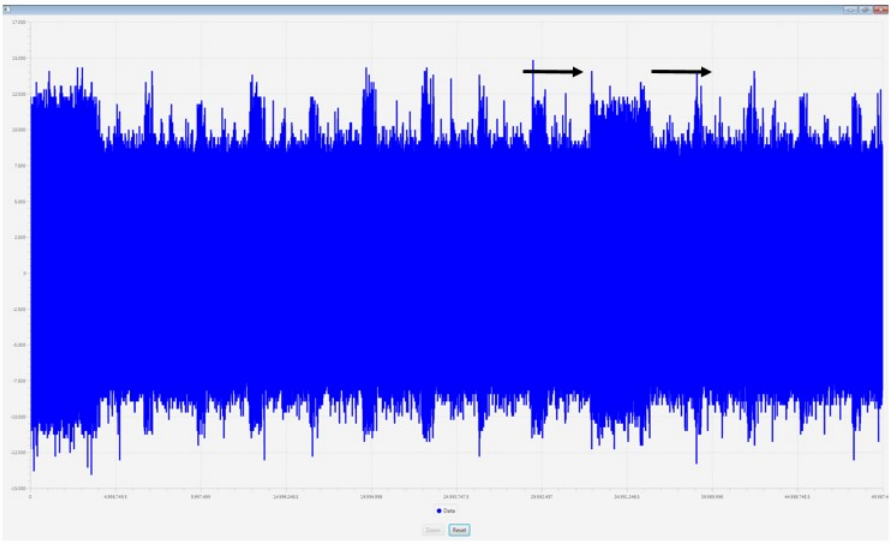
1. Öncelikle PIN bilgisini bilinen tamamen aynı özelliklerde olan farklı bir kart ile doğru PIN Verify ve yanlış PIN Verify işlemi yapılmış, sonuçlar ve işlemlerin tamamlanma süreleri kontrol edilmiştir. Doğru PIN için işlemin daha uzun sürdüğü görülmüştür (Doğru PIN $\approx$ 280 ms, Yanlış PIN $\approx$ 260 ms). PIN doğrulandıktan sonra, yetki verme vs. gibi ekstra işlemler yapılacağı için, yanlış PIN Verify'a göre daha fazla işlem gerçekleştirilmiş olacaktır. Bu beklenen ve normal bir sonuçtur.

2. Aynı ölçüm değerleri için, işlemlerin farklılaştığı, yani güç tüketimlerinin değiştiği yerin tespiti ölçüm üzerinde yapılmıştır (Şekil 6.3).



Şekil 6.3: Doğru PIN ve Yanlış PIN Karşılaştırma

3. Sonraki aşama alınan ölçümler doğrultusunda, güç tüketimlerinin farklılaştığı yere yakın bir konumdan Glitch koyulması, tüketimlerin değiştiği noktanın yakın bir ilerisine zaman ekseninde ilerlenmesidir. Geliştirilen test yazılımında bir döngü içerisinde bu işlem yapılmaktadır. Döngünün başlangıç değeri, tüketimlerin değiştiği konumun 30 ms öncesi, bitiş değeri tüketimlerin değiştiği konumun 30 ms sonrasıdır (Şekil 6.4). Bu ilerleme çok küçük aralıklarla, yani hassas olarak gerçekleştirilmiştir.



Şekil 6.4: Zaman Ekseninde Yakınlaşma

4. İlerleme gerçekleştirilirken atağın başarılı bir şekilde uygulanabilmesi için beklenti, farklılık sonrasındaki herhangi bir zaman diliminde sayaç değerinin hala değişmemiş olmasıdır. Saldırı gerçekleştirilen kartta, döngünün son noktası olan, farklılaşma

sonrası 30 ms'de, sayaç ile ilgili hala bir işlem yapılmamıştır. Bu durum da Glitch'i farklılaşma sonrası 30 ms de dahil, 30 ms içerisinde herhangi bir noktaya koyabilmeyi sağlamaktadır. Bu sebeple Glitch 30 ms'ye koyularak sınırsız deneme hakkı elde edilmiştir.

5. Oluşturulan bu şablon, aynı özellikteki başka bir kartta uygulanmış, bilinmeyen PIN değeri elde edilmiştir (Bulunan PIN = 0x38373433, 8743).

Bölüm 7

Sonuç ve Tartışma

Bu tez çalışmasında, bozucu olmayan donanımsal saldırı türleri için bir test yazılımı geliştirilmiş ve bu yazılımla, bozucu olmayan saldırıların başarılı bir şekilde gerçekleştirilebildiği gösterilmiştir.

Bu tez çalışmasında hazırlanan test yazılımı ile birlikte, analiz ve ölçüm alma işlemleri tek bir yazılımda toplanmıştır. Benzer yazılımlar incelendiğinde, analiz yazılımları ölçüm alma kısmı ile ilgilenmemekte, ölçüm alma yazılımları da analiz işlemlerini gerçekleştirmemektedir. Bu yönüyle hazırlanan test yazılımı, benzer işlemleri yapan diğer yazılımlardan daha üstün olmaktadır. Ayrıca test yazılımı, isteğe bağlı olarak farklı ortamlarda analiz yapmaya da imkan tanımaktadır. Test yazılımı ile alınan ölçümler istenilen formatta dışarı verilebilmektedir. Bu özelliği ile test yazılımı, ölçüm alma işlemlerini yöneten benzer yazılımların yaptığı aynı işlemleri gerçekleştirmiş olmakta, test yazılımının, benzer yazılımlardan eksik bir yönü bulunmamaktadır.

Hazırlanan test yazılımının Glitch saldırısı yeteneği bulunmaktadır. Test yazılımı bu özelliği ile benzer yan kanal analiz yazılımlarından farklı olarak ekstra bir yeteneğe de sahip olmaktadır. Çünkü Glitch saldırısı, hem özel bir saldırı cihazı gerektirmekte hem de bu cihazdan gelen veriler doğrultusunda saldırıyı yönlendirmeyi gerektirmektedir. Glitch Saldırısı için hazırlanan yazılımlar sadece Glitch saldırılarını uygulamaktadır. Hazırlanan test yazılımı ise hem yan kanal saldırılarının analizini hem de Glitch saldırısını yapabilme yeteneğine sahiptir.

Hazırlanan test yazılımının bir diğer önemli özelliği, modüler yapısı sayesinde yazılıma yeni saldırı cihazları ve analiz yöntemleri eklemekteki kolaylıktır. Ortalama düzeyde

Java programlama bilgisine sahip bir kişi tarafından kolaylıkla yazılıma yeni cihaz ve analizler eklenebilmektedir. Test yazılımı ile benzer işleri yapan yazılımlar incelendiğinde, benzer yazılımlar için bu durumun söz konusu olmadığı görülmüştür. Benzer yazılımların kaynak kodlarına erişim bulunmamaktadır. Bu sebeple yazılımlar üzerinde herhangi bir değişiklik yapılamamaktadır. Bir değişiklik talebinde bulunulması gerektiğinde geliştirici firma ile iletişime geçilmesi ve yeni özelliğin firmadan talep edilmesi gerekmektedir. Bu durum ekstra maliyet ve zaman kaybına yol açmaktadır.

Hazırlanan Test Yazılımı benzer işleri yapan üç farklı yazılımla karşılaştırılmış ve bu karşılaştırma sonucu Tablo 7.1'de verilmiştir.

Tablo 7.1: Hazırlanan Test Yazılımının Benzer Yazılımlar ile Karşılaştırılması

	HTY	**Y1**	**Y2**	**Y3**
Ölçüm Alma	Var	Var	Yok	Yok*
Yan Kanal Analiz Yeteneği	Var	Yok	Yok	Var
Yeni Saldırı Cihazı Ekleyebilme	Var	Yok	Yok	Yok
Yeni Bir Saldırı Türü Ekleme	Var	Uygulanabilir Değil	Yok	Yok
Glitch Saldırısı Yeteneği	Var	Yok	Var	Yok
Maliyet	Var Olan Cihazlar İçin Ekstra Maliyet Yok Yeni Cihazlar İçin Cihaz Maliyeti Kadar	Cihaz ve Yazılım Lisans Maliyeti	Cihaz ve Yazılım Lisans Maliyeti	Yazılım Lisans Maliyeti

HTY: Hazırlanan Test Yazılımı, **Y1:** PicoScope 6 [55], **Y2:** MPManager v1.6.0 [56], **Y3:** Inspector v4.4 [57]

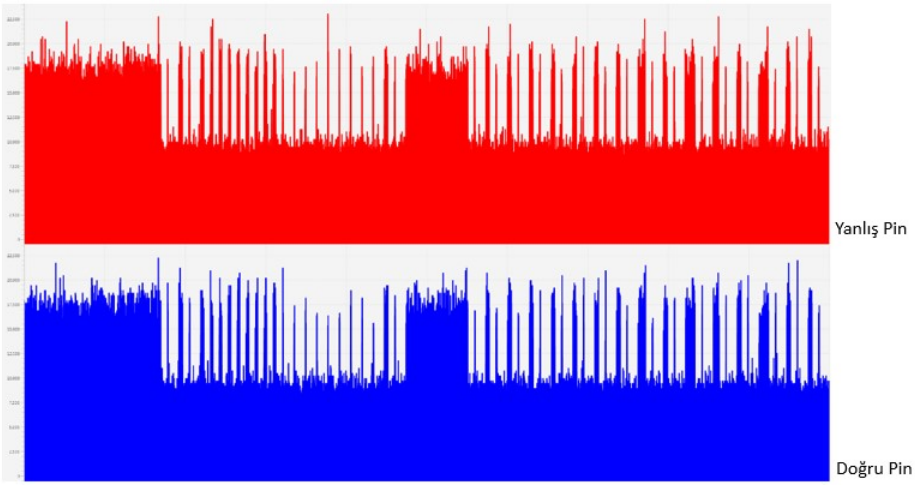
*Y3'te kısmi bir ölçüm alma işlemi olmakla birlikte, bu tez çalışmasında bahsedilen çoklu ölçüm alma işlemleri bu yazılımla uygulanamamaktadır. Bu sebeple bu yazılımın ölçüm alma özelliği "Yok" olarak belirtilmiştir.

Ayrıca bu tez çalışmasında, Bölüm 6'da gerçek bir akıllı kart uygulamasına, hazırlanan test yazılımı ile başarılı bir Glitch saldırısı gerçekleştirilmiştir. Bu saldırı, çeşitli sensörleri

ile güvenli bir kullanım sağlayan bir tümdevre üzerine geliştirilen ürünün, güvenli olarak geliştirilmediğinde, tümdevrenin sağladığı güvenlik özelliğinin bir anda yeni bir saldırı kapısı aralayabildiğini göstermiştir.

Bu saldırı sonucunda, güvenli bir akıllı kart uygulamasının geliştirilebilmesi için aşağıdaki hususlar öne çıkmıştır:

1. Herhangi bir doğrulama ve yetkilendirme işlemi için kontrol yapılırken, kaba kuvvet saldırılarına karşı önlem olarak belirli deneme sayaçları koyulması ve bu sayaçların değerlerinin çok büyük değerler olmaması,
2. PIN, şifre gibi doğrulama verilerinin karmaşıklığının yüksek olması,
3. Herhangi bir doğrulama ve yetkilendirme işlemleri için, yaygın saldırı türlerinden olan donanımsal saldırı yöntemleri de göz önünde bulundurularak geliştirme yapılmalı, sayaç gibi kritik değerlerden önce PIN veya şifre verisi ile doğrudan bağlantılı hiçbir işlem gerçekleştirilmemelidir. Örnek olarak, donanımsal saldırılar göz önünde bulundurularak geliştirilmiş bir üründen alınan, sayaç düşmeden önce gerçekleştirilen işlemlere ait güç tüketimleri Şekil 7.1'te verilmiştir. Şekilde de görüleceği üzere, güç tüketimleri doğru ve yanlış PIN değeri için herhangi bir ipucu içermemektedir.



Şekil 7.1: Güvenlik Önlemli PIN Verify İşlemi

Kısıtlı zaman sebebiyle, hazırlanan test yazılımının kapsamı, bozucu olmayan saldırılar içerisinde yer alan, Basit Güç-Elektromanyetik Analiz, Farksal Güç-Elektromanyetik

Analiz ve Glitch saldırıları ile sınırlandırılmıştır. Gelişmiş ve karmaşık saldırı türlerinin yazılıma eklenmesi gelecek çalışmalar olarak planlanmıştır. Ayrıca aşağıdaki çalışmaların, gelecekte gerçekleştirilmesi planlanmaktadır:

- Test yazılımına yeni donanımsal saldırı cihazları eklenerek, yeni ve daha karmaşık saldırıların yazılımla gerçekleştirilmesi,
- Yarı bozucu ve bozucu saldırı türlerinin, test yazılımına eklenmesiyle ilgili çalışmaların yapılması,
- Test yazılımının sürekli güncel tutulması ve literatüre eklenen yeni saldırı türlerinin yazılıma eklenebilmesi için dinamik bir çalışmanın yapılması,
- Test yazılımı aracılığıyla saldırı gerçekleştirilebilecek ürün gruplarının sayısının artırılması için çalışmaların yapılması, önceliğin kritik ürün gruplarına verilmesi,
- Hazırlanan test yazılımının ürün değerlendirmelerinde yaygın olarak kullanımının sağlanması.

Matematiksel analizlerde yaygın olarak kullanılan MATLAB programının, R2016 ile birlikte, Java tabanlı uygulamalarda MATLAB fonksiyonlarının kullanımını sağlayan, Java için MATLAB Engine API [58]'si bulunmaktadır. Bu API'nin avantajlarının incelenmesi ve uygun bulunursa test yazılımına eklenmesi de gelecek çalışma olarak düşünülebilir.

Kaynaklar

[1] S. Skorobogatov. Hardware Security: Present Challenges and Future Directions, July 2018. URL https://www.cl.cam.ac.uk/~sps32/Slides_NTU2018.pdf.

[2] ISO/IEC 15408-1,2,3:2009. Information Technology – Security Techniques – Evaluation Criteria for IT Security. Standard, International Organization for Standardization, Geneva, Switzerland, 2009.

[3] ISO/IEC 19790:2012. Information technology – Security Techniques – Security Requirements for Cryptographic Modules. Standard, International Organization for Standardization, Geneva, Switzerland, 2012.

[4] Common Criteria CCRA. CCRA Members, 01-07-2019. URL https://www.commoncriteriaportal.org/ccra/members/.

[5] Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance Components, v3.1, r5, April 2017. URL https://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf.

[6] ISO/IEC 18045:2008. Information Technology – Security Techniques – Methodology for IT Security Evaluation. Standard, International Organization for Standardization, Geneva, Switzerland, 2008.

[7] ISO/IEC 24759:2017. Information Technology – Security Techniques – Test Requirements for Cryptographic Modules. Standard, International Organization for Standardization, Geneva, Switzerland, 2017.

[8] NIST. Cryptographic Module Validation Program, June 2019. URL https://csrc.nist.gov/projects/cryptographic-module-validation-program/standards.

[9] P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, pages 104–113, London, UK, August 1996. Springer-Verlag. ISBN 3-540-61512-1. URL `http://dl.acm.org/citation.cfm?id=646761.706156`.

[10] J. F. Dhem, F. Koeune, P. A. Leroux, P. Mestré, J. J. Quisquater, and J. L. Willems. A practical implementation of the timing attack. In *Proceedings of the The International Conference on Smart Card Research and Applications*, CARDIS '98, pages 167–182, Berlin, Heidelberg, September 1998. Springer-Verlag. ISBN 3-540-67923-5. URL `http://dl.acm.org/citation.cfm?id=646692.703439`.

[11] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397, Berlin, Heidelberg, August 1999. Springer-Verlag. ISBN 3-540-66347-9. URL `http://dl.acm.org/citation.cfm?id=646764.703989`.

[12] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '01, pages 251–261, London, UK, May 2001. Springer-Verlag. ISBN 3-540-42521-7. URL `http://dl.acm.org/citation.cfm?id=648254.752700`.

[13] J. J. Quisquater and D. Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Proceedings of the International Conference on Research in Smart Cards: Smart Card Programming and Security*, E-SMART '01, pages 200–210, London, UK, UK, September 2001. Springer-Verlag. ISBN 3-540-42610-8. URL `http://dl.acm.org/citation.cfm?id=646803.705980`.

[14] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The sorcerer's apprentice guide to fault attacks. *Proceedings of the IEEE*, 94(2):370–382, Feb 2006. ISSN 0018-9219. doi: 10.1109/JPROC.2005.862424.

[15] S. Chari, J. R. Rao, and P. Rohatgi. Template attacks. In *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, pages 13–28, London, UK, August 2002. Springer-Verlag. ISBN 3-540-00409-2. URL `http://dl.acm.org/citation.cfm?id=648255.752740`.

[16] O. Kömmerling and M. G. Kuhn. Design principles for tamper-resistant smartcard processors. In *Proceedings of the USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology*, WOST'99, pages 2–2, Berkeley, CA, USA, May 1999. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=1267115.1267117`.

[17] R. J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley Publishing, 2 edition, 2008. ISBN 9780470068526.

[18] S. Skorobogatov. Semi-invasive attacks-a new approach to hardware security analysis. 01 2005.

[19] D. Samyde, S. Skorobogatov, R. Anderson, and J. J. Quisquater. On a new way to read data from memory. In *Proceedings of the First International IEEE Security in Storage Workshop*, SISW '02, pages 65–, Washington, DC, USA, December 2002. IEEE Computer Society. ISBN 0-7695-1888-5. URL `http://dl.acm.org/citation.cfm?id=829507.830220`.

[20] S. Skorobogatov and R. J. Anderson. Optical fault induction attacks. In *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, pages 2–12, London, UK, UK, August 2002. Springer-Verlag. ISBN 3-540-00409-2. URL `http://dl.acm.org/citation.cfm?id=648255.752727`.

[21] S. Skorobogatov. Optically enhanced position-locked power analysis. volume 4249, pages 61–75, 10 2006. doi: 10.1007/11894063_6.

[22] F. Standaert, T. G. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In A. Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, pages 443–461, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-01001-9.

[23] Y. Yarom and K. Falkner. Flush+reload: A high resolution, low noise, l3 cache side-channel attack. pages 719–732, San Diego, CA, August 2014. USENIX Association. ISBN 978-1-931971-15-7. URL `https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/yarom`.

[24] T. Zhang, Y. Zhang, and R. B. Lee. Cloudradar: A real-time side-channel attack detection system in clouds. In F. Monrose, M. Dacier, G. Blanc, and J. Garcia-Alfaro,

editors, *Research in Attacks, Intrusions, and Defenses*, pages 118–140, Cham, 2016. Springer International Publishing. ISBN 978-3-319-45719-2.

[25] R. Hund, C. Willems, and T. Holz. Practical timing side channel attacks against kernel space aslr. In *2013 IEEE Symposium on Security and Privacy*, pages 191–205, May 2013. doi: 10.1109/SP.2013.23.

[26] S. Skorobogatov. How microprobing can attack encrypted memory. In *2017 Euromicro Conference on Digital System Design (DSD)*, pages 244–251, Aug 2017. doi: 10.1109/DSD.2017.69.

[27] J. S. Gustavus. Cryptology, 2016. URL `https://www.britannica.com/topic/cryptology`.

[28] H. C. A. van Tilborg. *Fundamentals of Cryptology: A Professional Reference and Interactive Tutorial*. Kluwer Academic Publishers, Norwell, MA, USA, 1st edition, 1999. ISBN 0792386752.

[29] National Institute of Standards and Technology. Data Encryption Standard (DES). FIPS Publication 46-3, 1999. URL `http://csrc.nist.gov/publications/fips/fips46-3/`.

[30] American National Standards Institute (ANSI). Triple Data Encryption Algorithm Modes of Operation. ANSI X9.52, 1998.

[31] D. Rudolf, 2000. URL `http://homepage.usask.ca/~dtr467/400/`.

[32] J. Daemen and V. Rijmen. *The design of Rijndael: AES — the Advanced Encryption Standard*. Springer-Verlag, 2002. ISBN 3-540-42580-2.

[33] National Institute of Standards and Technology. Advanced Encryption Standard (AES). *NIST FIPS PUB 197*, 2001.

[34] R Rivest, A Shamir, and L Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of The ACM - CACM*, 1978.

[35] E. Barker. SP 800-57. Recommendation for Key Management, Part 1: General, Rev. 4 National Institute of Standards & Technology (NIST). Technical report, Gaithersburg, MD, 20899-8900, United States, 2016.

[36] A. Corbellini, 2015. URL `https://andrea.corbellini.name/ecc/interactive/reals-add.html`.

[37] FIPS PUB 186-4. Digital Signature Standard (DSS) National Institute of Standards & Technology (NIST). Technical report, Gaithersburg, MD, 20899-8900, United States, 2013.

[38] SEC 2: Recommended Elliptic Curve Domain Parameters Standards for Efficient Cryptography (SEC), 2010.

[39] J. Merkle and M. Lochter. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation RFC 5639. ISSN 10.17487/RFC5639, 2010.

[40] W. Rankl and W. Effing. *Smart Card Handbook*. Wiley Publishing, 4th edition, 2010. ISBN 0470743670, 9780470743676.

[41] T. Tashev. The Period of the LFSR Based Generalized Shrinking-multiplexing Generator. In *Proceedings of the 2007 International Conference on Computer Systems and Technologies*, pages 55:1–55:6. ACM, 2007. ISBN 978-954-9641-50-9. doi: 10.1145/1330598.1330657. URL `http://doi.acm.org/10.1145/1330598.1330657`.

[42] M. Dichtl. How to Predict the Output of a Hardware Random Number Generator. In C. D. Walter, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, pages 181–188. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-45238-6.

[43] D. Todorov. *Mechanics of User Identification and Authentication*. Auerbach Publications, Boston, MA, USA, 2007. ISBN 1420052195.

[44] R. Housley, T. Polk, Dr. W. S. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459, 1999. URL `https://rfc-editor.org/rfc/rfc2459.txt`.

[45] ISO/IEC 9594-8:2017. Information Technology – Open Systems Interconnection – The Directory – Part 8: Public-key and attribute certificate frameworks. Standard, International Organization for Standardization, Geneva, Switzerland, 2017.

[46] P. Karlton A. Freier and P. Kocher. The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101, 2011. URL `https://rfc-editor.org/rfc/rfc6101.txt`.

[47] T. Pedersen. *PEM, Privacy-Enhanced Mail*, pages 919–920. Springer US, Boston, MA, 2011. ISBN 978-1-4419-5906-5. doi: 10.1007/978-1-4419-5906-5_297.

[48] B. Ramsdell J. Schaad and S. Turner. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification. RFC 8551, 04-2019. URL `https://rfc-editor.org/rfc/rfc8551.txt`.

[49] Y. Li and Y. Wang. Secure Electronic Transaction (SET Protocol). 2014.

[50] S. Skorobogatov. *Physical Attacks and Tamper Resistance*, pages 143–173. Springer New York, New York, NY, 2012. ISBN 978-1-4419-8080-9. doi: 10.1007/978-1-4419-8080-9_7. URL `https://doi.org/10.1007/978-1-4419-8080-9_7`.

[51] M. Weiner, 10-06-2019. URL `https://www.sec.ei.tum.de/en/research/invasive-attacks/`.

[52] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. 01 2007. ISBN 978-0-387-30857-9. doi: 10.1007/978-0-387-38162-6.

[53] E. Oswald. Guided Analysis of WS1. Technical Report, IAIK, Graz, Austria, 2007.

[54] ISO/IEC 7816-4:2013. Identification Cards – Integrated Circuit Cards – Part 4: Organization, Security and Commands for Interchange. Standard, International Organization for Standardization, Geneva, Switzerland, 2013.

[55] PicoScope, 15-08-2019. URL `https://www.picotech.com/library/newsletter/september-2007`.

[56] Micropross, 15-08-2019. URL `https://www.micropross.com/index.php`.

[57] Riscure, 15-08-2019. URL `https://www.riscure.com/`.

[58] MATLAB Support, 15-08-2019. URL `https://www.mathworks.com/help/matlab/matlab-engine-api-for-java.html`.