ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

HAFİF SIKLET KRİPTOGRAFİDE ASİMETRİK TEKNİKLERİN YAZILIM UYGULAMASI VE BAŞARIM ANALİZİ

Fikriye KÜÇÜKÖMEROĞLU

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2021

ÖZET

Yüksek Lisans Tezi

HAFİF SIKLET KRİPTOGRAFİDE ASİMETRİK TEKNİKLERİN YAZILIM UYGULAMASI VE BAŞARIM ANALİZİ

Fikriye KÜÇÜKÖMEROĞLU

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik -Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Aykut KALAYCIOĞLU

Nesnelerin İnterneti hızlı bir şekilde gelişmekte ve RFID etiketler, sensör düğümleri, akıllı kartlar gibi cihazlar gün geçtikçe daha yaygın kullanılmaktadır. Bu cihazların program kodu boyutunun, RAM/ROM boyutunun, veri işleme hacminin ve işlem süresinin kısıtlı olmasından kaynaklı olarak bu cihazlarda güvenliği sağlamak zor bir iştir. Kısıtlı kapasite özelliklerine sahip ürünlere güvenlik unsuru hafif sıklet şifreleme protokolleri sayesinde katılmaktadır. Tezde ISO/IEC 29192 standardında tanımlı cryptoGPS, ELLI, ALIKE, Kimlik Tabanlı İmza yöntemlerinin C programlama dili yardımıyla yazılım uygulamalarının gerçekleştirilmesine yönelik çalışmalar yapılmıştır. Yazılım başarım analizi için kod boyutu, işlem süresi ve veri işleme hacmi yazılım metrikleri kullanılmıştır. Yöntemlerin maksimum 2 Kilobayt RAM ve ROM varlığında çalıştırılması amaçlanmıştır. ELLI algoritması RAM boyutu 2 Kilobayt olan ortamda çalıştırılmıştır. ELLI algoritmasında bellek tüketimi ile işlem süresi arasında bir ödünleşim olduğu görülmüştür. Literatürde yapılan benzer çalışmalar ile kıyaslandığında ALIKE mekanizmasının anahtar üretme aşamasında işlem süresi bakımından daha iyi sonuçlar elde edilmiştir. Sonlu cisimlerde toplama, çıkarma, çarpma, hızlı modüler çarpma, çarpımsal ters alma ve nokta çarpım işlemlerinin başarımı literatürdeki çalışmalara göre düşük kalmıştır.

Eylül 2021, 114 sayfa

Anahtar Kelimeler: hafif sıklet, asimetrik kriptografi, eliptik eğri, doğrulama, cryptoGPS, Kimlik Tabanlı İmza Yöntemi, ALIKE, ELLI, sonlu cisim, toplama, çıkarma, çarpma, çarpımsal ters alma, indirgeme, nokta çarpım

ABSTRACT

MSc Thesis

SOFTWARE IMPLEMENTATION AND PERFORMANCE ANALYSIS OF ASYMMETRIC TECHNIQUES IN LIGHTWEIGHT CRYPTOGRAPHY

Fikriye KÜÇÜKÖMEROĞLU

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assistant Professor Aykut KALAYCIOĞLU

The Internet of Things is developing rapidly and devices such as RFID tags, sensor nodes, smart cards are being used more and more. Due to the limited program code size, RAM/ROM size, throughput and execution time of these devices, it is a difficult task to ensure security on these devices. The security element is added to the products with limited capacity features thanks to the lightweight encryption protocols. In the thesis, studies have been carried out to implement software applications of cryptoGPS, ELLI, ALIKE, Identity Based Signature methods defined in ISO/IEC 29192 standard with C programming language. For software performance analysis, software metrics of code size, execution time and throughput were used. The methods are intended to be run in the presence of a maximum of 2 Kilobytes of RAM and ROM. The ELLI algorithm was run in an environment with a memory size of 2 Kilobytes. It has been observed that there is a trade-off between memory consumption and execution time in the ELLI algorithm. Compared to similar studies in the literature, better results were obtained in terms of execution time in the key generation phase of the ALIKE mechanism. The performance of addition, subtraction, multiplication, fast modular multiplication, multiplicative inverse and point multiplication operations in finite fields remained low compared to the studies in the literature.

September 2021, 114 pages

Key Words: lightweight, asymmetric cryptography, elliptic curve, authentication, cryptoGPS, Identity Based Signature, ALIKE, ELLI, finite field, addition, substraction, multiplication, multiplicative inverse, reduction, point multiplication

TEŞEKKÜR

Tez çalışmalarımın her aşamasında bilgi, öneri ve yardımlarını esirgemeyerek gelişmeme katkı sağlayan danışman hocam sayın Dr. Öğretim Üyesi Aykut KALAYCIOĞLU'na (Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı), sayın hocalarım Prof. Dr. Asım Egemen YILMAZ'a (Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı), Dr. Öğretim Üyesi Gökhan SOYSAL'a (Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı) çok teşekkür ederim.

Kriptografi ve sonlu cisim aritmetiği konularında kendisinden çok şey öğrendiğim Gazi Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği bölümünden sayın hocam Prof. Dr. Erkan Afacan'a ve hafif sıklet doğrulama yöntemlerinde uygulama geliştirme konusunda akıl danıştığım On Dokuz Mayıs Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünden sayın hocam Doç. Dr. Sedat Akleylek'e ayrıca teşekkür ederim.

117E631 numaralı TÜBİTAK projesi kapsamında gerçekleştirdiğim tez çalışmalarımda beni destekleyen TÜBİTAK kurumuna teşekkürü bir borç bilirim.

Yüksek lisans eğitimim boyunca yanımda olan, beni maddi manevi destekleyen, daima motive eden aileme içten teşekkürlerimi sunarım.

Fikriye KÜÇÜKÖMEROĞLU
Ankara, Eylül 2021

İÇİNDEKİLER

SİMGELER ve KISALTMALAR DİZİNİ

E(P, K)	Şifreleme işlemi
D(C, K)	Şifre çözme işlemi
$\phi()$	Phi fonksiyonu
F	F fonksiyonu
(x, y)	Koordinat ekseni üzerinde x ve y değerleri ile belirlenen nokta
m_i	Açık metin
c_i	Şifreli metin
k	Anahtar
k_i	Anahtar dizisi
GF(p), GF(p^n), GF(2^n)	Sonlu Galois cisimleri
#E	Eliptik eğrinin mertebesi
P, Q, R, U	Eliptik eğri üzerinde tanımlı noktalar
h(x)	Özet fonksiyonu
$\mathbb{Z}$	Tamsayılar Kümesi
$\mathbb{Z}_p^*$	Sonlu ve Periyodik Grup
Tr	Trace fonksiyonu

Kısaltmalar

AES	Advanced Encryption Standard
ALIKE	Authenticated Lightweight Key Exchange
ASIC	Application Specific Integrated Circuit
C	Şifreli metin (Ciphertext)
CMOS	Complementary Metal Oxide Semiconductor
D	Şifre Çözme (Decryption)
DES	Data Encryption Standard
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
E	Şifreleme (Encryption)
EBOB	En Büyük Ortak Bölen
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EEPROM	Electronically Erasable Programmable ROM
ELLI	Elliptic Light
FIPS	Federal Information Processing Standard
FPGA	Field Programmable Gate Array
GF	Galois Field
GPS	Girault Poupard Stern

IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
K	Anahtar (Key)
L	Left
LFSR	Linear Feedback Shift Register
NIST	National Institute of Standards and Technology
P	Açık metin (Plaintext)
R	Right
RAM	Random Access Memory
RFID	Radio Frequency Identification
ROM	Read-Only Memory
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm
SP	Substitution-Permutation
SPN	Substitution-Permutation Network
SRAM	Static RAM
SSL	Secure Socket Layer

ŞEKİLLER DİZİNİ

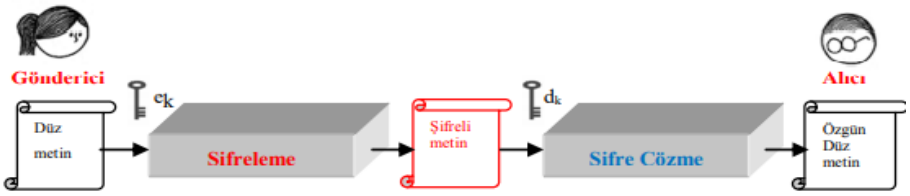
ÇİZELGELER DİZİNİ

1. GİRİŞ

Kriptoloji, iki veya daha fazla tarafın birbiri ile haberleşmesi sırasında paylaşılan bilgilerin gizliliği, bütünlüğü, bilgilerin kaynağının doğrulanması, tarafların kimliğinin doğrulanması gibi bilgi güvenliğini sağlamaya yönelik çalışmalar yapan, matematiksel problemleri temel alan yöntem ve uygulamaları kapsayan bir disiplindir (Menezes vd. 1996, Çetin 2006, Sarıtaş 2010, Paar ve Pelzl 2009).

Kriptoloji, kriptografi ve kriptoanaliz olmak üzere iki kola ayrılmaktadır. Kriptografi, yetkili olmayan kişilerin veya kurumların önem taşıyan anlamlı mesajlara ulaşmasını engellemek için bu mesajların anlaşılmaz forma dönüştürülmesi ve şifreli mesajlardan anahtar aracılığıyla orijinal mesajların elde edilmesi ile ilgilenmektedir. Kriptoanaliz, anlaşılır olmayan şifreli mesajların anahtar bilinmeden şifresinin çözülerek orijinal mesajlara ulaşılmasına yönelik çalışmalar yapmaktadır (Çetin 2006, Paar ve Pelzl 2009).

Şekil 1.1'de temel kriptografi işleyişi görülmektedir. Gönderici, alıcıya bir şifreleme anahtarı ve algoritması kullanarak şifrelediği metni iletmektedir. Alıcı, bir şifre çözme anahtarı ve algoritması ile şifreyi çözerek orijinal metne erişmektedir (Kurt 2012).



Şekil 1.1 Temel kriptografi işleyişi (Kurt 2012)

Kriptografi biliminin geçmişi oldukça eskidir. Kriptografiyi ilk Mısırlılar olmak üzere yazı kültürüne sahip pek çok medeniyet kullanmıştır. Birinci Dünya Savaşı sona erdikten sonra Amerika Birleşik Devletleri ve Deniz Kuvvetleri temel seviyede kriptografik çalışmalar yapmıştır. O dönemlerde kriptografi milli sır ve stratejilerin korunması için faydalanılan bir araçtır. 1975'te Martin Hellman ve Bruce Schneier çözülmesi zor olan matematiksel problemleri temel alan açık anahtarlı kripto sistemleri

önermiştir. 1977'de Feistel tarafından Veri Şifreleme Standardı (*Data Encryption Standard*) tasarlanmıştır. 1978 yılında Rivest, Shamir, Adleman RSA adlı açık anahtarlı şifreleme ve imzalama şemalarını ileri sürmüştür. Hem güçlü hem pratik olan açık anahtarlı ElGamal algoritmasının geliştirildiği yıl 1985'tir. 1986'da Victor Miller ve 1987 yılında Neal Koblitz birbirinden bağımsız olarak açık anahtarlı şifreleme sistemlerini eliptik eğriler ile tasarlamıştır. 1990'ların sonlarında eliptik eğriler ile şifreleme ticari alanda yer almaya başlamıştır. 1991'de RSA kripto sisteminin kullanıldığı ilk uluslararası imza standardı, 1994 yılında ElGamal şemasına dayanan Dijital İmza Standardı kabul edilmiştir (Schneier 1996, Menezes vd. 1996, Özcan 2006, Paar ve Pelzl 2009).

1.1 Kriptografinin Başlıca Kavramları

Açık Metin (*Plaintext*) : Şifrelenmek istenen orijinal metindir.

Şifreli Metin (Ciphertext) : Orijinal metnin şifrelenmiş halidir.

Gizli Anahtar (*Secret Key*) : Göndericinin açık metni şifrelemek, alıcının şifreli metni çözmek için kullandığı; yalnızca haberleşen tarafların sahip olması gereken bilgidir.

Açık Anahtar (*Public Key*) : Gönderici ve alıcının yanı sıra haberleşme ağında bulunan herkesin erişimine açık olan bilgidir.

Şifreleme (*Encryption*) : Açık metnin bir anahtar kullanılarak şifrelenmesi işlemidir.

Şifre Çözme (*Decryption*) : Şifreli metinden bir anahtar yardımıyla açık metnin elde edilmesi işlemidir.

Şifreleme Algoritması (*Encryption Algorithm*) : Açık metni şifrelemek için faydalanılan algoritmadır.

Şifre Çözme Algoritması (*Decryption Algorithm*) : Şifreli metnin şifresini çözmeyi sağlayan algoritmadır (Çetin 2006, Koyutürk 2020).

1.2 Kriptografinin Hedefleri

Gizlilik (*Confidentiality/Privacy*) : Kişisel bilgilerin yetkisi olmayan herkesten saklanmasıdır.

Bütünlük (*Integrity*) : Alıcıya gönderilen bilginin yetki olmadan değiştirilmemiş, üzerine ekleme yapılmamış ve yeniden düzenlenmemiş olmasıdır; bir başka deyişle gönderilen bilginin orijinal bilgi ile birebir aynı olmasıdır.

İnkar Edilememe (*Non-Repudiation*) : Göndericinin bilgiyi gönderdiğinin ve alıcının bilgiyi aldığının ispat edilebilmesidir.

Bilginin Kaynağının Doğrulanması (*Data Origin Authentication*) : Bilgiyi gönderen kaynağın güvenilir olup olmadığının tespit edilmesidir.

Kimlik Doğrulaması (*Entity Authentication*) : Bilgiyi gönderen kişinin doğru kişi olup olmadığının belirlenmesidir.

Erişilebilirlik (*Availability*) : Bilgiye yetkisi olan kişilerin ulaşmasıdır (Menezes vd. 1996, Özcan 2006, Sarıtaş 2010, Stallings 2017).

1.3 Hafif Sıklet Kriptografi

Nesnelerin İnterneti hızlı bir şekilde gelişmekte ve RFID etiketler, sensör düğümleri, akıllı kartlar gibi cihazlar gün geçtikçe daha yaygın kullanılmaktadır. Bu cihazların program kodu boyutunun, RAM/ROM boyutunun, veri işleme hacminin ve işlem süresinin kısıtlı olmasından kaynaklı olarak bu cihazlarda güvenliği sağlamak zor bir iştir. Hafif sıklet kriptografi, kaynak kısıtlı ortamlarda bulunan cihazlarda güvenliği sağlamayı hedefleyen bir kriptografi alt alanıdır (Ertan vd. 2018).

Teknolojinin gelişmesiyle birlikte bilgisayarların ve internetin kullanımı yaygınlaşmıştır. Bilgisayar ve internet tabanlı sistemlerde bilgi iletiminin güvenli bir şekilde yapılması çok büyük öneme sahiptir. Geçmişten bugüne kadar bilgi güvenliği probleminin çözülmesine yönelik birçok şifreleme yöntemi önerilmiştir. Simetrik şifreleme algoritmaları hızlıdır, bilgi güvenliğini sağlamaktadır ve bu algoritmaların donanımda uygulanması kolaydır ancak bu şifreleme sistemlerinde alıcı ile gönderici arasında anahtar güvenli bir şekilde dağıtılmamaktadır, kimlik doğrulaması ve veri bütünlüğünün sağlanması hem zor hem de maliyetli olmaktadır. Asimetrik şifreleme metotlarının geliştirilmesi sonucunda anahtar güvenliği sorunu ortadan kaldırılmıştır. Asimetrik şifreleme yöntemlerinin daha çok güvenlik fonksiyonuna sahip olması ve sayısal imzalar ile doğrulama yapabilmesi avantajından dolayı bu yöntemler yazılım ile donanım uygulamalarında daha fazla tercih edilmektedir. Son dönemlerde Nesnelerin İnterneti kavramı araştırmaların odak noktası olmuştur. Nesnelerin İnternetinde internet ağı, mobil ağlar ve sensör ağları kullanılarak genişletilmiştir. Ağda enerji tüketimi, işlem yapma gücü veya bellek tüketimi kısıtlı olmayan sunucu, masaüstü bilgisayar, akıllı telefon, tablet ve benzeri cihazlar ile düşük kapasitedeki cihazlar internete bağlıdır ve bu cihazlar birbiriyle iletişim halindedir. Bu sebeple Nesnelerin İnterneti uygulamalarında verilerin gizliliğini, bütünlüğünü ve güvenliğini sağlamada mevcut şifreleme algoritmaları yetersiz kalmaktadır. Heterojen yapıdaki cihazların bir arada bulunduğu Nesnelerin İnterneti ortamında güvenliği sağlamak amacıyla hafif sıklet kripto sistemlerin tasarlanmasına gerek duyulmuştur (Çetin 2006, Özcan 2006, Kumar vd. 2007, Sarıtaş 2010, Kurt 2012, Suo vd. 2012, Ertan vd. 2018).

Tez çalışmasında gömülü cihazlarda kullanılabilecek, doğrulama yapmayı hedefleyen ISO/IEC 29192 standardında tanımlı hafif sıklet asimetrik yöntemlerin yazılım yoluyla uygulanması, başarım analizinin yapılması ve bu yöntemlerin kaynak kısıtları ile ilişkisinin belirlenmesi önemli bir çalışma konusu olarak değerlendirilmektedir. Tez çalışmasının, kriptografi alanında çalışacaklar için farklı algoritmaların başarımını sunan bir kaynak olacağı düşünülmektedir. Tez kapsamında, literatüre katkıda bulunmak için hafif sıklet asimetrik doğrulama algoritmalarının kısıtlı kaynaklar altında yazılımda uygulanması ve başarımının analiz edilmesi hedeflenmiştir. Yazılım uygulamaları için ISO/IEC 29192 standardında tanımlanan cryptoGPS, kimlik tabanlı

imza yöntemi, ALIKE ve ELLI hafif sıklet asimetrik doğrulama algoritmaları seçilmiştir.

CryptoGPS, kimlik tabanlı imza, ALIKE ve ELLI kaynakları kısıtlı olan donanım uygulamaları için tasarlanmış asimetrik doğrulama yöntemleridir. Tezde bu asimetrik doğrulama yöntemlerinin yazılım uygulamalarında hafif sıklet olma durumu araştırılmıştır. Tez çalışmalarında ELLI doğrulama tekniği bellek kullanımının kısıtlı olduğu ortamda başarılı bir şekilde çalıştırılmıştır. ELLI doğrulama tekniğinde bellek tüketimi minimum iken işlem süresinde bir artış olduğu farkedilmiştir. ALIKE mekanizmasında RSA algoritmasına oranla daha kısa sürede anahtar üretilmiştir. Tezde cryptoGPS ve kimlik tabanlı imza algoritmalarının gerçekleştirilmesine yönelik sonlu asal cisimlerde tanımlı toplama, çıkarma, çarpma, hızlı modüler çarpma, bölme, çarpımsal ters alma ile nokta toplama, nokta aynılama, nokta çarpma işlemlerinin yazılım uygulamalarında literatürde yapılmış benzer çalışmalara göre daha yüksek işlem süreleri elde edilmiştir.

Bir algoritmanın yazılım uygulamasının hafif sıklet olarak kabul edilmesi için altı adet performans ölçütü göz önüne alınmaktadır:

Kod Boyutu (*Code Size*): Yazılım performansını ölçmede faydalanılan bir parametredir. Kod boyutu ile kod satır sayısı kastedilmektedir. Kod satır sayısı, boş satırların ve yorum satırlarının dâhil edilmediği toplam satır sayısıdır (McKay vd. 2017).

Basitlik (*Simplicity*): Bir yazılım metriği olan basitlik, tasarım süresinin bir ölçüsüdür. Bir tasarımın basitliği kod satır sayısı yardımıyla betimlenmektedir (Good ve Benissa 2008).

Yazmaç Sayısı (*Register*): Bellek yerine yazmaçların kullanılması algoritmanın performansına olumlu yönde etki etmektedir. Az sayıda yazmaç kullanan fonksiyonların çağrılma maliyeti daha az olmaktadır. Bu parametre yazılım performansını değerlendirmeye yardımcı olan bir parametredir (McKay vd. 2017).

RAM/ROM boyutu (*RAM/ROM size*): Yazılım başarımını ölçmede kullanılan bir metriktir. Kod boyutunun artması, kodun tamamının yürütüldüğü zamanki saat döngü sayısını, kodun işlendiği süreyi, dolaylı olarak kodun ROM'da tutulduğu bayt sayısını ve RAM' de tutulan geçici değişkenlerin bayt sayısını arttıracaktır. Program kodu ROM'da, hesaplama boyunca kullanılan geçici değişkenler RAM' de saklanmaktadır. RAM ve ROM kullanımı algoritmanın işlendiği saat döngü sayısını ve algoritmanın işlem süresini etkilemektedir (McKay vd. 2017).

İşlem Süresi (*Execution Time*): İşlemin başlamasından ilk çıktı üretilmesine kadar geçen süredir. Hem donanım hem yazılım uygulamalarının performansını değerlendirmede rol oynamaktadır (McKay vd. 2017).

Veri İşleme Hacmi (*Throughput*): Saniye başına bit olarak ifade edilen, yeni çıktıların zamana bağlı olarak üretilme hızıdır. Genel anlamda saat frekansı ile döngü başına üretilen bit sayısının çarpımıdır. Hem yazılım hem donanım ortamında gerçekleştirilmiş olan uygulamaların başarımını değerlendirmede bu ölçüt kullanılabilmektedir (Good ve Benissa 2008).

Tez beş bölümden oluşmaktadır. Birinci bölümde kriptoloji kavramından kısaca bahsedilmekte, kriptografinin başlıca kavramları ve hedeflerine değinilmektedir. Hafif sıklet kriptografi tanıtılmakta ve neden hafif sıklet kriptografiye ihtiyaç duyulduğu belirtilmektedir. Ayrıca bu bölümde tezin amacı, önemi ve kapsamı hakkında bilgi verilmektedir. İkinci bölümde literatürde konu ile ilgili yapılmış olan çalışmalar özetlenmekte, tez çalışmasında yazılımda gerçeklenmesi planlanan hafif sıklet doğrulama mekanizmalarının anlaşılmasına yardımcı olan kuramsal temeller yer almaktadır. Üçüncü bölümde cryptoGPS, kimlik tabanlı imza, ALIKE ve ELLI algoritmaları hakkında genel bilgilendirme yapılmakta, her bir algoritma için kullanılan parametreler, güvenlik gereksinimleri ve uygulama adımları açıklanmaktadır. Dördüncü bölümde tez çalışmalarında gerçeklenmiş yazılım uygulamalarına ait başarım analizi yapılmaktadır. Beşinci bölümde elde edilen yazılım uygulamalarına ait sonuçlar değerlendirilmekte ve literatürde yer alan çalışmalar ile karşılaştırılmaktadır.

2. KURAMSAL TEMELLER ve KAYNAK ÖZETLERİ

Bu bölümde kriptografik yöntemlerin anlaşılmasına katkıda bulunan genel bilgiler, sonlu cisim ve eliptik eğri aritmetiği yer almakta, literatür kaynakları özetlenmektedir.

Tez çalışması kapsamındaki asimetrik şifreleme algoritmalarının anlaşılması, bu algoritmaların temelini oluşturan matematiksel temellerin bilinmesini gerekli kılmaktadır. Tezde uygulanması hedeflenen asimetrik şifreleme teknikleri sonlu cisim (*finite field*) uzayında tanımlıdır. Bu nedenle sonlu cisim aritmetiğinin kavranması önemlidir.

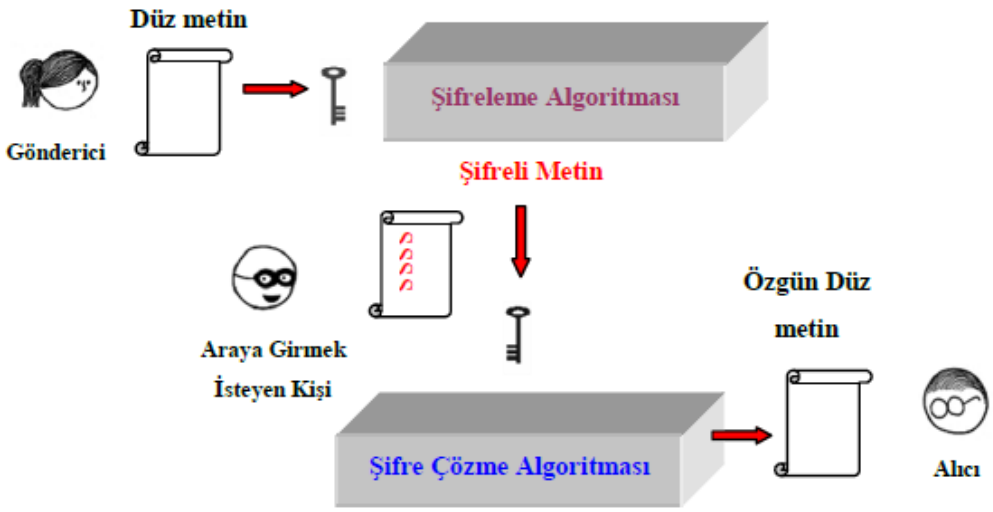
Sonlu cisimlerin anlaşılabilmesi için gruplar (*group*), değişmeli gruplar (*abelian group*), halkalar (*ring*) ve cisimler (*field*) ile ilgili bir takım tanımlamalara ve teoremlere ihtiyaç duyulmaktadır. Grup, halka ve cisim yapıları anlatılırken (Kurt 2012, Engin 2017, Afacan 2017, Koyutürk 2020) kaynakları birebir takip edilmektedir. Grup, halka ve cisim tanımlamaları (Kurt 2012, Engin 2017, Afacan 2017, Koyutürk 2020) kaynaklarının notasyonu kullanılarak hiçbir katkı yapılmadan sunulmaktadır.

2.1 Kriptografi Sistemleri

İhtiyaçlar doğrultusunda geliştirilmiş olan ve kullanılan anahtar bakımından farklılık gösteren iki adet şifreleme sistemi vardır. Bunlardan birincisi simetrik şifreleme sistemi, ikincisi asimetrik şifreleme sistemidir (Kurt 2012, Soysaldı 2018).

2.1.1 Simetrik şifreleme sistemi

Simetrik şifrelemede gönderici ve alıcı ortak bir anahtar belirlemektedir. Bu anahtarın taraflar arasında gizli kalması gerekmektedir. Simetrik şifreleme geleneksel şifreleme, simetrik anahtarlı şifreleme, tek anahtarlı şifreleme veya gizli anahtarlı şifreleme olarak da anılmaktadır. Simetrik kripto sistemlerin yapısı şekil 2.1'de yer almaktadır (Çetin 2006, Kurt 2012).



Şekil 2.1 Simetrik şifreleme yapısı (Kurt 2012)

Şifrelemede gönderici, açık metni şifreleme anahtarı yardımıyla şifreleyerek alıcıya göndermektedir. Şifre çözmede alıcı, aynı anahtarı kullanarak şifrelenmiş metni çözerek açık metni elde etmektedir. Denklem 2.1 ve denklem 2.2 ile sırasıyla şifreleme ve şifre çözme eşitlikleri verilmektedir. Burada P ile açık metin, C ile şifreli metin, K ile anahtar, E ile şifreleme işlemi ve D ile şifre çözme işlemi temsil edilmektedir (Çetin 2006, Kurt 2012).

$$C = E(P, K) \tag{2.1}$$

$$P = D(C, K) \tag{2.2}$$

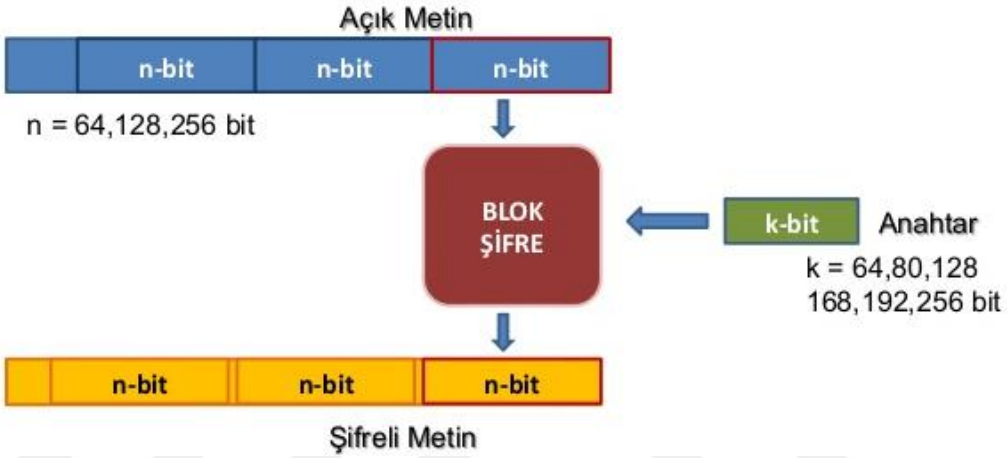
Simetrik şifreleme yöntemleri hızlı olması açısından avantajlıdır. Fakat bu yöntemlerde anahtar dağıtımı ve anahtar yönetiminde zorluk yaşanmaktadır (Çetin 2006, Kurt 2012).

(Çetin 2006, Sarıtaş 2010) tarafından belirtildiği gibi simetrik anahtarlı şifreleme algoritmaları iki çeşittir:

1. Blok Şifreleme Algoritmaları
2. Dizi Şifreleme Algoritmaları

2.1.1.1 Blok şifreleme

Şekil 2.2'de görüldüğü gibi blok şifrelemede, bloklara bölünmüş açık metnin her seferinde bir bloğu aynı anahtar kullanılarak şifrelenmektedir. Bilgi, bloklar halinde şifrelendiği için bu şifrelemeye blok şifreleme adı verilmektedir. Blok ile bit grupları kastedilmektedir. Bir blok yaygın olarak 64 ya da 128 bit uzunluğunda olmaktadır (Çetin 2006, Paar ve Pelzl 2009, Sarıtaş 2010).



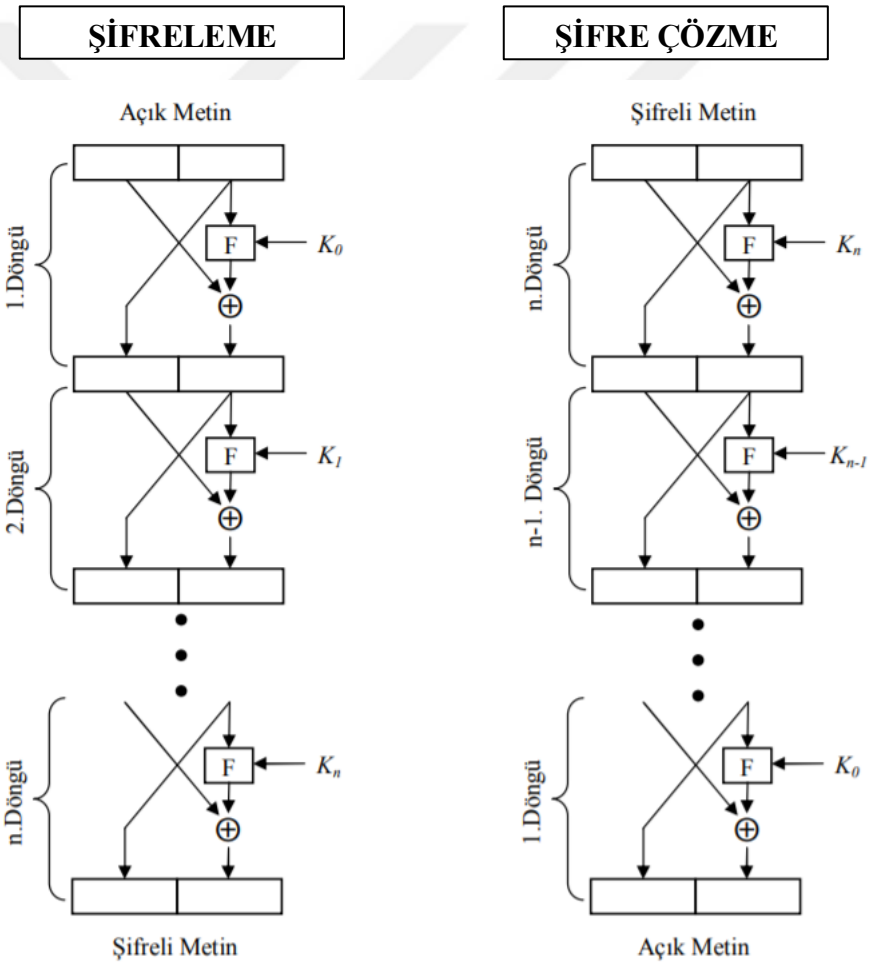
Şekil 2.2 Blok şifreleme yapısı (Anonymous 2008)

Claude Shannon'a göre güçlü blok şifreleme yöntemleri hem karıştırma (*confusion*) hem yayılma (*diffusion*) işlemleri temel alınarak oluşturulmaktadır. Karıştırma, anahtarın şifreli metin ile bağlantısını gizleyerek kriptoanaliz yapılmasını zorlaştırmaktadır. Yayılma, açık metin ile şifreli metnin istatistiksel ilişkisinin anlaşılmamasını sağlamaktadır (Altan vd. 2004, Paar ve Pelzl 2009, Sarıtaş 2010, Stallings 2017).

Blok şifrelerin tasarımında Feistel ağı (*Feistel Network*) veya Yer değiştirme-Permütasyon ağı (*Substitution-Permutation Network*) mimarisi kullanılmaktadır. Feistel ağı (Şekil 2.3), Horst Feistel'in tasarlamış olduğu bir yapıdır (Altan vd. 2004, Sarıtaş 2010).

Şifreleme modunda Feistel ağı girdileri açık metin ve anahtardır. Açık metin L ve R olarak iki yarıya ayrılmıştır. Her i. döngünün girdileri bir önceki döngüde oluşturulmuş

olan L_i, R_i girişleri ve K_i alt anahtarıdır. Her döngünün alt anahtarı farklıdır. Açık metnin sağ yarısına ve alt anahtara bir F fonksiyonu uygulanmaktadır. F fonksiyonunun görevi blokların yerini değiştirmektir. F fonksiyonunun çıktısı ile açık metnin sol yarısı xor'lanmaktadır. Ardından metnin sağ ve sol yarısı yer değiştirmektedir. Döngünün n defa tekrarlanması sonucunda şifreli metin elde edilmektedir. Tüm döngülerde aynı işlemler gerçekleştirilmektedir. Şifre çözmede Feistel ağının girdileri şifreli metin ve anahtar olmaktadır. Alt anahtarları kullanma sırası şifrelemedekinin tam tersidir. Şifre çözmek için şifreleme ile temel olarak aynı işlemler yapılmaktadır (Yavuzer Aslan 2012, Stallings 2017).



Şekil 2.3 Feistel ağı yapısı (Yavuzer Aslan 2012)

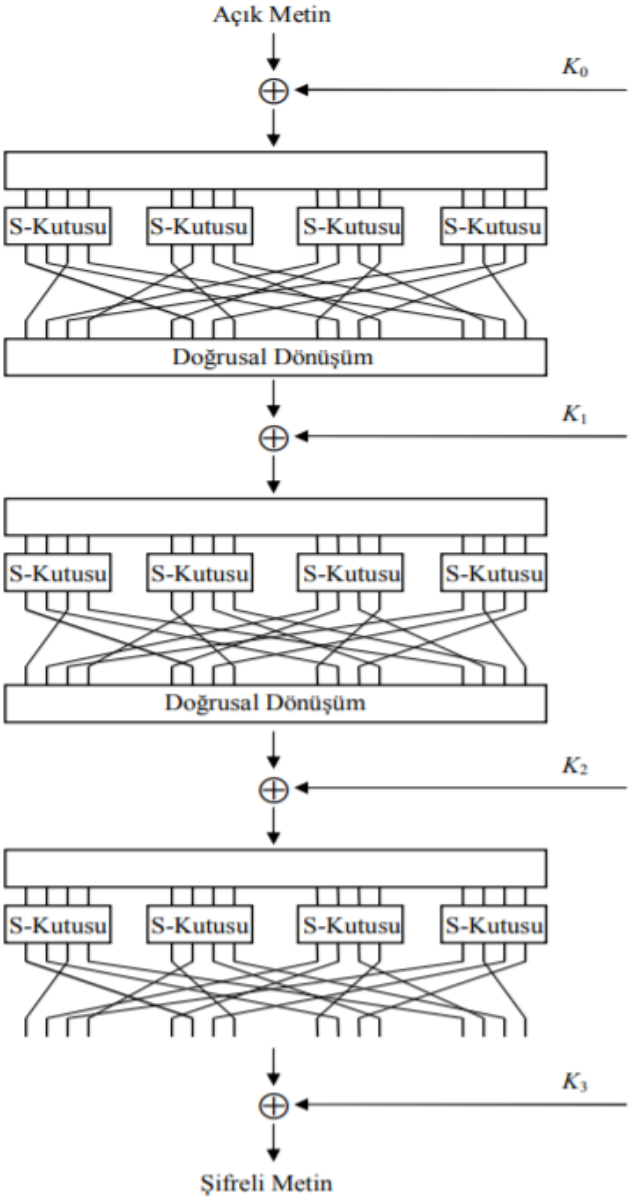
Yer değiştirme-Permütasyon ağı mimarisinin elemanları açık metin, anahtar, doğrusal olmayan yer değiştirme kutuları (S-kutuları), doğrusal dönüşüm kutuları (P-kutuları) ve

şifreli metindir. Şekil 2.4' te girdi ve çıktıların 16-bit, döngü sayısının 3 olduğu SPN mimarisi örneği verilmektedir. Yer değiştirme-Permütasyon ağında, Feistel ağının aksine bir döngüde bir bloğun tamamı işlenmektedir. İki mimaride de her döngünün alt anahtarı birbirinden farklıdır (Yavuzer Aslan 2012).

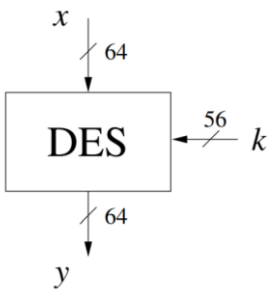
Yer değiştirme-Permütasyon ağında S-kutularının görevi karıştırma işlemi ile blok şifrenin karmaşıklığını arttırmaktır. Karıştırma, blokların yeri değiştirilerek yapılmaktadır ve doğrusal değildir. Yer değiştirme-Permütasyon ağı mimarisindeki P-kutuları yayılma işlemi yapmaktadır. Yayılma, doğrusal dönüşümler yardımıyla sağlanmaktadır. Doğrusal dönüşümler sayesinde bir girdi bloğu doğrusal olarak karıştırılmaktadır ve bir çıktı bloğu türetilmektedir. Girdi ve çıktı blokları eşit uzunluktadır (Yavuzer Aslan 2012).

Veri Şifreleme Standardı (*DES: Data Encryption Standard*) ve Gelişmiş Şifreleme Standardı (*AES: Advanced Encryption Standard*) en çok bilinen blok şifreleme algoritmalarıdır (Sarıtaş 2010).

DES algoritması, 1970'li yıllarda NIST (*National Institute of Standards and Technology*) tarafından geliştirilmiş ve standart haline getirilmiştir. Bu algoritmada, 64 bitlik bir açık metin 56 bitlik bir anahtarla şifrelenmektedir ve elde edilen şifreli metin 64 bit uzunluğunda olmaktadır (Şekil 2.5). Bu algoritmanın tasarımında kullanılan mimari, Feistel ağı mimarisidir (Sarıtaş 2010, Yavuzer Aslan 2012).

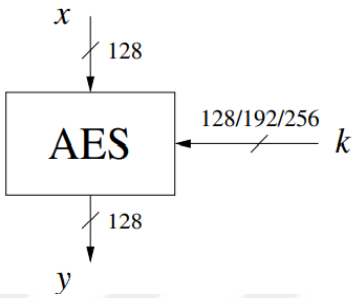


Şekil 2.4 Yer değiştirme-Permütasyon (SP) ağı (Yavuzer Aslan 2012)



Şekil 2.5 DES blok şifresi (Paar ve Pelzl 2009)

AES algoritmasında 128 bit uzunluğundaki bir açık metni şifrelemek için 128, 192 ya da 256 bitlik bir anahtar kullanılmaktadır. Şifreli metin açık metin ile aynı uzunlukta olup 128 bittir (Şekil 2.6). Bu algoritma Yer değiştirme-Permütasyon ağı mimarisi temel alınarak tasarlanmıştır (Sarıtaş 2010, Yavuzer Aslan 2012).



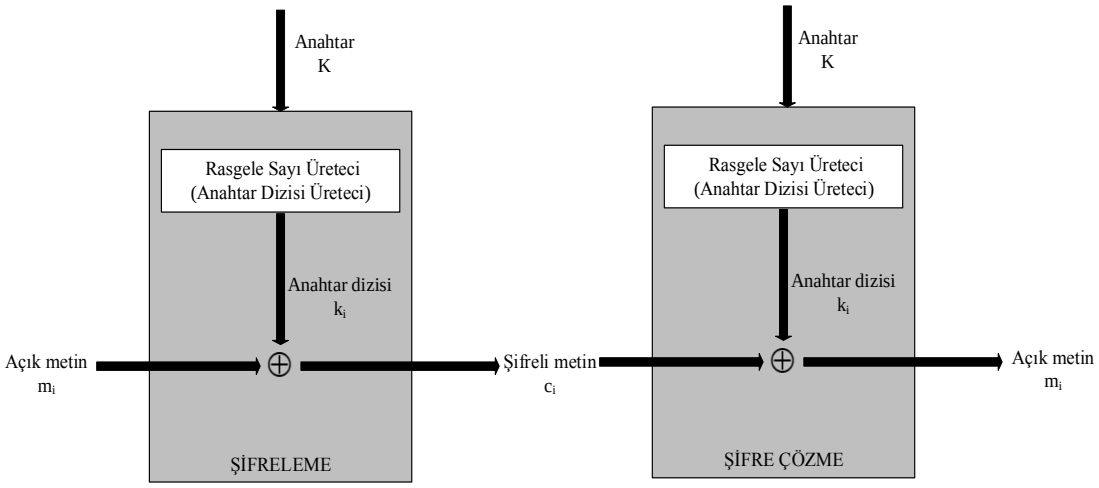
Şekil 2.6 AES blok şifresi (Paar ve Pelzl 2009)

2.1.1.2 Dizi şifreleme

Dizi şifreleri bir açık metnin her bir bitini/karakterini bir anahtar dizisi kullanarak tek tek şifrelemektedir. Şifreleme anahtarı rasgele sayı üreteci vasıtasıyla bir anahtar dizisine dönüştürülmektedir. Böylelikle açık metin ile anahtarın bit sayısı eşit olmaktadır. Açık metin ve anahtar bit bit toplanarak şifreli metin üretilmektedir. Şifre çözmede ise şifrelemenin tam tersi işlemler yapılmaktadır. Şifreli metin ile anahtarın toplanması sonucunda açık metin elde edilmektedir. Denklem 2.3 şifreleme ve denklem 2.4 şifre çözme işlemine ait eşitliklerdir. Şekil 2.7 ile dizi şifreleme ve şifre çözme yapısı ifade edilmektedir (Çetin 2006, Paar ve Pelzl 2009, Kurt 2012, Yavuzer Aslan 2012, Stallings 2017).

$$c_i = m_i + k_i \pmod 2 \tag{2.3}$$

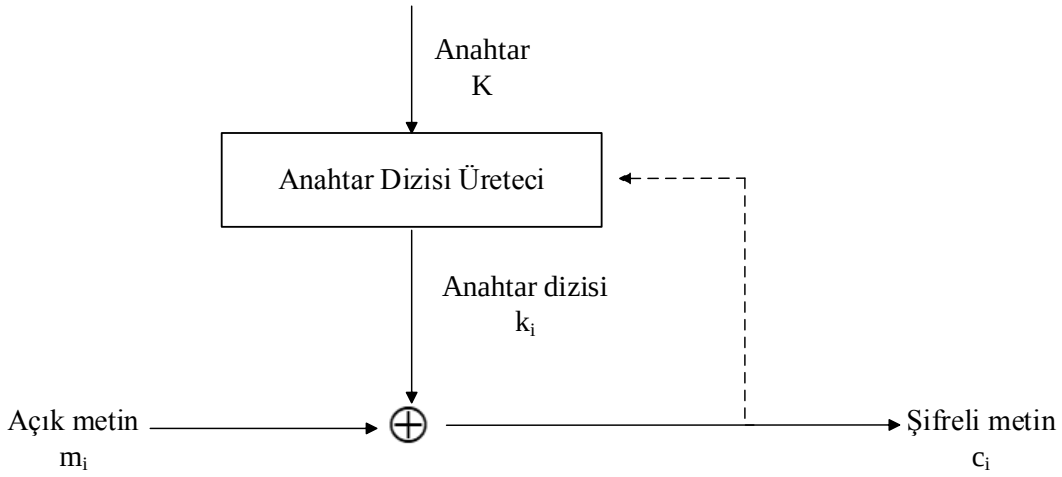
$$m_i = c_i + k_i \pmod 2 \tag{2.4}$$



Şekil 2.7 Dizi şifreleme diyagramı (Stallings 2017)

İki çeşit anahtar dizisi oluşturulması mümkündür: 1. Tam Rasgele Anahtar Dizisi (*True Random Key Stream*), 2. Sözde Rasgele Anahtar Dizisi (*Pseudo Random Key Stream*). Anahtarın tüm bitleri birbirinden bağımsız olarak üretilmişse anahtar dizisi tam rasgeledir. Sözde rasgele anahtar dizisinde ise anahtarın her bir biti bir önceki bitle ilişkilendirilmektedir (Altan vd. 2004).

Dizi şifreleme algoritmaları senkron ya da asenkron yapıdadır (Şekil 2.8). Anahtar dizisi açık metinden bağımsız olduğunda şifreleme algoritması senkron, şifreli metin ile bağlantılı olduğu durumda asenkrondur (Paar ve Pelzl 2009, Sarıtaş 2010).

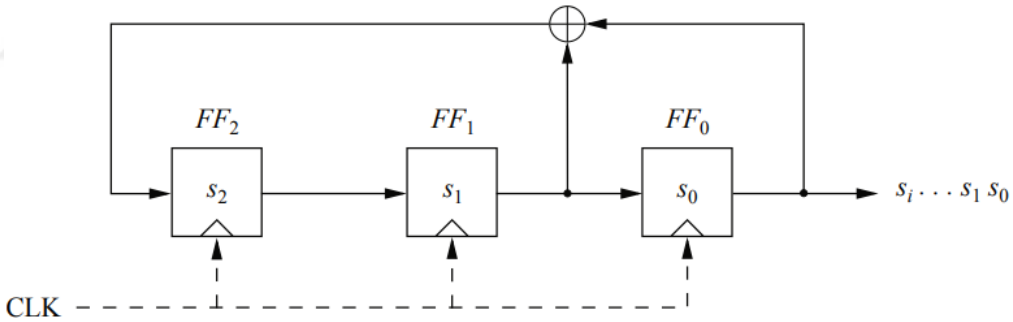


Şekil 2.8 Senkron ve asenkron dizi şifresi (Paar ve Pelzl 2009)

Bir dizi şifresinin anahtar dizisi tam rasgele sayı üreteci ile üretilmekte, anahtar dizisi sadece yetkililer tarafından bilinmekte ve her bir anahtar dizisi bir kez kullanılmakta ise bu dizi şifresi tek kullanımlık şifre (*one time pad*) olarak adlandırılmaktadır. Tek kullanımlık şifreler koşulsuz güvenlik (*unconditionally security*) sağlamaktadır (Paar ve Pelzl 2009).

Dizi şifreleme algoritmalarının çoğunda donanımda uygulanması kolay olan doğrusal geri beslemeli kaydırmalı yazmaçlar (*Linear Feedback Shift Registers*-LFSRs) kullanılmaktadır. Doğrusal geri beslemeli kaydırmalı yazmacın bileşenleri flip-flop grupları ve bir geri besleme yoludur. Doğrusal geri beslemeli kaydırmalı yazmaçlarda giriş biti önceki durum bitlerinin bazılarının XOR'lanması ve sonucun geri beslenmesi, diğer bitler bir önceki durum bitlerinin birer bit sağa kaydırılması ile oluşturulmaktadır. Bir LFSR için derece içerdiği flip-flop sayısıdır (Paar ve Pelzl 2009, Panda vd. 2012).

Şekil 2.9' da derecesi 3 olan bir LFSR örneği görülmektedir.



Şekil 2.9 Doğrusal geri beslemeli kaydırmalı yazmaç örneği (Paar ve Pelzl 2009)

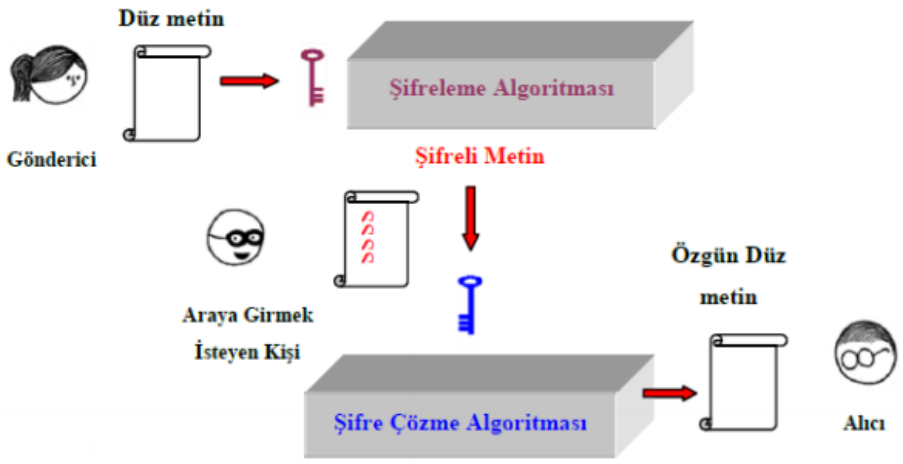
Modern simetrik kriptografi yöntemlerinin güvenlik düzeyi yüksektir, bu yöntemler hızlıdır ve yaygın olarak kullanılmaktadır. Fakat simetrik anahtarlı algoritmaların bir takım eksik yönleri bulunmaktadır. Anahtarın güvenli bir haberleşme kanalı üzerinden gönderilmesi mümkün olmamaktadır. Çok sayıda anahtar çiftinin oluşturulması ve iletilmesi gerekmektedir. Simetrik kriptografi ile inkar edilememe (*non-repudiation*) hedefi karşılanamamaktadır. Pek çok asimetrik şifreleme sisteminin öne sürülmesinin nedeni simetrik şifreleme sistemlerinin bu dezavantajlarını telafi etmektir (Paar ve Pelzl 2009).

2.1.2 Asimetrik şifreleme sistemi

Gönderici ve alıcının kullanacağı anahtar konusunda anlaşmaları simetrik şifreleme algoritmalarının en temel problemlerinden birisidir. Anahtar dağıtımı problemini çözmek için 1977 yılında Whitfield Diffie ve Martin Hellman tarafından asimetrik şifreleme sistemi öne sürülmüştür. Asimetrik şifreleme sistemi sayesinde gizli anahtarın bir anahtar dağıtım merkezinden edinilmesine gerek kalmamıştır (Çetin 2006, Kurt 2012).

Asimetrik şifreleme yöntemlerinde herkes gizli ve açık anahtarlardan oluşan birer anahtar çiftine sahiptir. Herkes birbirinin açık anahtarını bilmektedir. Gizli anahtar ise kişisel olup yalnızca kullanıcısı tarafından bilinmektedir. Gönderici göndermek istediği metni alıcının açık anahtarı ile şifrelemekte ve alıcıya göndermektedir. Alıcı bu şifreli metni kendi gizli anahtarını kullanarak çözmekte ve orijinal metni bulmaktadır. Şifreleme ve şifre çözmede farklı anahtarların kullanıldığı asimetrik kripto sisteminde şifreleme açık anahtar ile gerçekleştirilmektedir. Bu nedenle asimetrik şifreleme sistemi açık anahtarlı şifreleme sistemi olarak da adlandırılmaktadır (Çetin 2006, Sarıtaş 2010, Kurt 2012).

Şekil 2.10'da asimetrik kripto sisteminin genel yapısı verilmektedir.



Şekil 2.10 Asimetrik kripto sistemi (Kurt 2012)

Simetrik anahtarlı kriptografide yer değiştirme (*substitution*) ve permütasyon (*permutation*) işlemleri kullanılmakta iken açık anahtarlı kriptografinin temeli matematiksel problemlere dayanmaktadır (Çetin 2006, Stallings 2017).

Anahtar oluşturma (*key establishment*), inkar edilememe (*non-repudiation*), kimliklendirme (*identification*) ve şifreleme (*encryption*) işlevleri açık anahtarlı algoritmalarda güvenliği sağlamaktadır. RSA ve Diffie-Hellman Anahtar Değişimi protokolleri güvenli olmayan bir ortamda iletişime geçen gönderici ve alıcının gizli anahtarlar oluşturmasına olanak sağlamaktadır. İnkar edilememe ve mesajın bütünlüğünün korunması için Sayısal İmza (*Digital Signature*) ve Eliptik Eğrilerle Sayısal İmza (*Elliptic Curve Digital Signature*) algoritmalarından faydalanılmaktadır. Akıllı kartların kullanıldığı uygulamalarda meydan okuma-ve-karşılık verme mekanizmaları (*challenge-and-response mechanisms*) vasıtasıyla kimliklendirme yapılmaktadır. RSA ve ElGamal şifreleme yöntemleriyle mesajlar şifrelenmektedir (Paar ve Pelzl 2009).

(Bozkurt 2005, Özcan 2006, Yavuz 2008, Paar ve Pelzl 2009, Kurt 2012) asimetrik kripto sistemlerin güvenliğinin temel alınan matematiksel problemlere bağlı olduğunu belirtmektedir. Kullanılan matematiksel problemler göz önüne alınarak sınıflandırılmış başlıca üç adet asimetrik şifreleme ailesi bulunmaktadır:

1. Bir Tamsayının Çarpanlarına Ayrılması Problemi Temelli Algoritmalar (*Algorithms Based on Integer-Factorization Problem*)
2. Ayrık Logaritma Problemi Temelli Algoritmalar (*Algorithms Based on Discrete Logarithm Problem*)
3. Eliptik Eğri Temelli Algoritmalar (*Algorithms Based on Elliptic Curve*)

Temeli bir tamsayının çarpanlarına ayrılması problemine dayanan açık anahtarlı kriptografik yöntemlerin güvenlik seviyesi büyük pozitif tamsayıların asal çarpanlarına ayrılmasının zorluğu ile ilişkilidir. RSA, bu asimetrik şifreleme ailesinin bir üyesidir (Bozkurt 2005, Özcan 2006, Yavuz 2008, Paar ve Pelzl 2009, Kurt 2012).

Diffie-Hellman Anahtar Değişimi Protokolü (*Diffie-Hellman Key Exchange Protocol*), ElGamal Şifreleme Sistemi (*ElGamal Encryption System*) ve Sayısal İmza Algoritması (*Digital Signature Algorithm*) sonlu asal cisimlerde tanımlı ayrık logaritma problemi kullanılarak tasarlanmış yöntemlerdir (Bozkurt 2005, Özcan 2006, Yavuz 2008, Paar ve Pelzl 2009, Kurt 2012).

Eliptik Eğri Diffie-Hellman Anahtar Değişimi Protokolü (*Elliptic Curve Diffie-Hellman Key Exchange Protocol*) ve Eliptik Eğri Sayısal İmza Algoritmasında (*Elliptic Curve Digital Signature Algorithm*) ayrık logaritma problemi sonlu cisim uzayında yer alan eliptik eğriler üzerinde tanımlıdır (Bozkurt 2005, Özcan 2006, Yavuz 2008, Paar ve Pelzl 2009, Kurt 2012).

Eliptik eğri kripto sistemi diğer açık anahtarlı şifreleme sistemlerine nazaran daha kısa uzunlukta anahtarlar kullanmaktadır. Anahtar boyutunun küçülmesi ile güç tüketiminden, kullanılan bant genişliğinden ve işlem süresinden tasarruf edilmektedir (Chang vd. 2002).

Çizelge 2.1'de 80, 128, 192 ve 256 bit güvenlik seviyeleri için simetrik ve asimetrik şifreleme algoritmalarının tavsiye edilen parametre ve anahtar bit uzunlukları ifade edilmektedir (Paar ve Pelzl 2009).

Çizelge 2.1 Farklı güvenlik seviyeleri için önerilen bit uzunlukları (Paar ve Pelzl 2009)

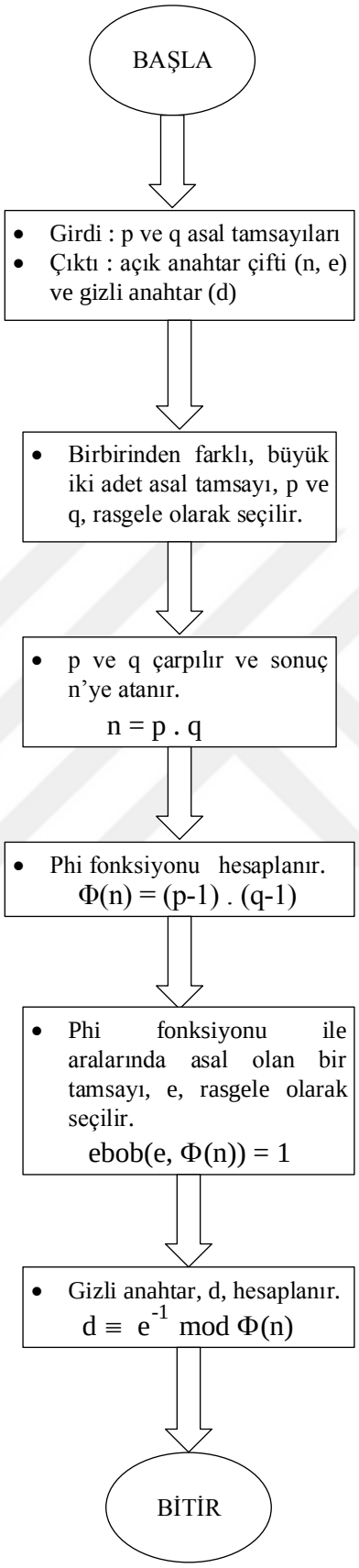
Şifreleme Ailesi	**Şifreleme Sistemi**	**Güvenlik Seviyesi (bit)**			
		80	128	192	256
Bir Tamsayının Çarpanlarına Ayrılması Problemi	RSA	1024	3072	7680	15360
Ayrık Logaritma Problemi	DH, DSA, ElGamal	1024	3072	7680	15360
Eliptik Eğriler	ECDH, ECDSA	160	256	384	512
Simetrik anahtarlı	AES, 3DES	80	128	192	256

2.1.2.1 RSA şifreleme sistemi

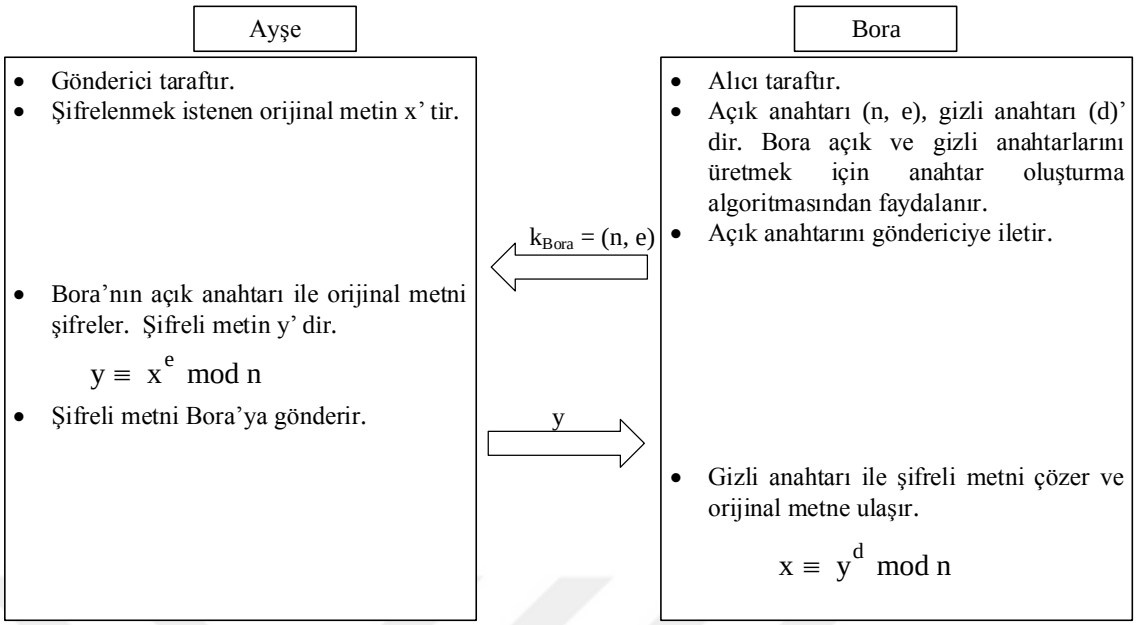
1977 yılında Ronald Rivest, Adi Shamir ve Leonard Adleman tarafından önerilmiş açık anahtarlı şifreleme şemasıdır. Rivest-Shamir-Adleman kripto sistemi olarak da bilinen RSA şifreleme sistemi yaygın olarak kullanılmaktadır. RSA ile sayısal imza üretilmesi ve anahtar paylaşılırken gizliliğin sağlanması hedeflenmektedir. RSA algoritmasının güvenli olup olmaması, büyük tamsayıların çarpanlarına ayrılması probleminin çözümünün kolay ya da zor olmasına bağlıdır. Tamsayıları çarpanlarına ayırma problemi tek yönlü bir fonksiyondur. RSA şifreleme yönteminde büyük iki tamsayı kolaylıkla çarpılmaktadır. Bunun aksine verilen çarpma sonucunu çarpanlarına ayırmak neredeyse olanaksızdır (Schneier 1996, Menezes vd. 1996, Çetin 2006, Paar ve Pelzl 2009).

RSA şeması anahtar oluşturulmasına, şifreleme ve şifre çözme işlemlerinin yapılmasına ve sayısal imza üretilmesine imkan sağlamaktadır. Şekil 2.11' de anahtar oluşturma adımları açıklanmaktadır (Çetin 2006, Paar ve Pelzl 2009).

Ayşe x metnini RSA şifreleme yöntemi ile şifrelemeyi amaçlamaktadır. Bora şifreli metnin, y, şifresini RSA şifre çözme yöntemini kullanarak çözmek ve orijinal metne ulaşmak istemektedir. Ayşe, orijinal metni Bora'nın açık anahtarı ile şifrelemekte ve Bora, kendi gizli anahtarı yardımıyla şifreli metnin şifresini çözerek orijinal metne erişmektedir. Şekil 2.12'de, RSA ile şifreleme ve şifre çözme işlemleri ayrıntılı bir şekilde izah edilmektedir (Paar ve Pelzl 2009).



Şekil 2.11 RSA anahtar oluşturma akış şeması (Paar ve Pelzl 2009)

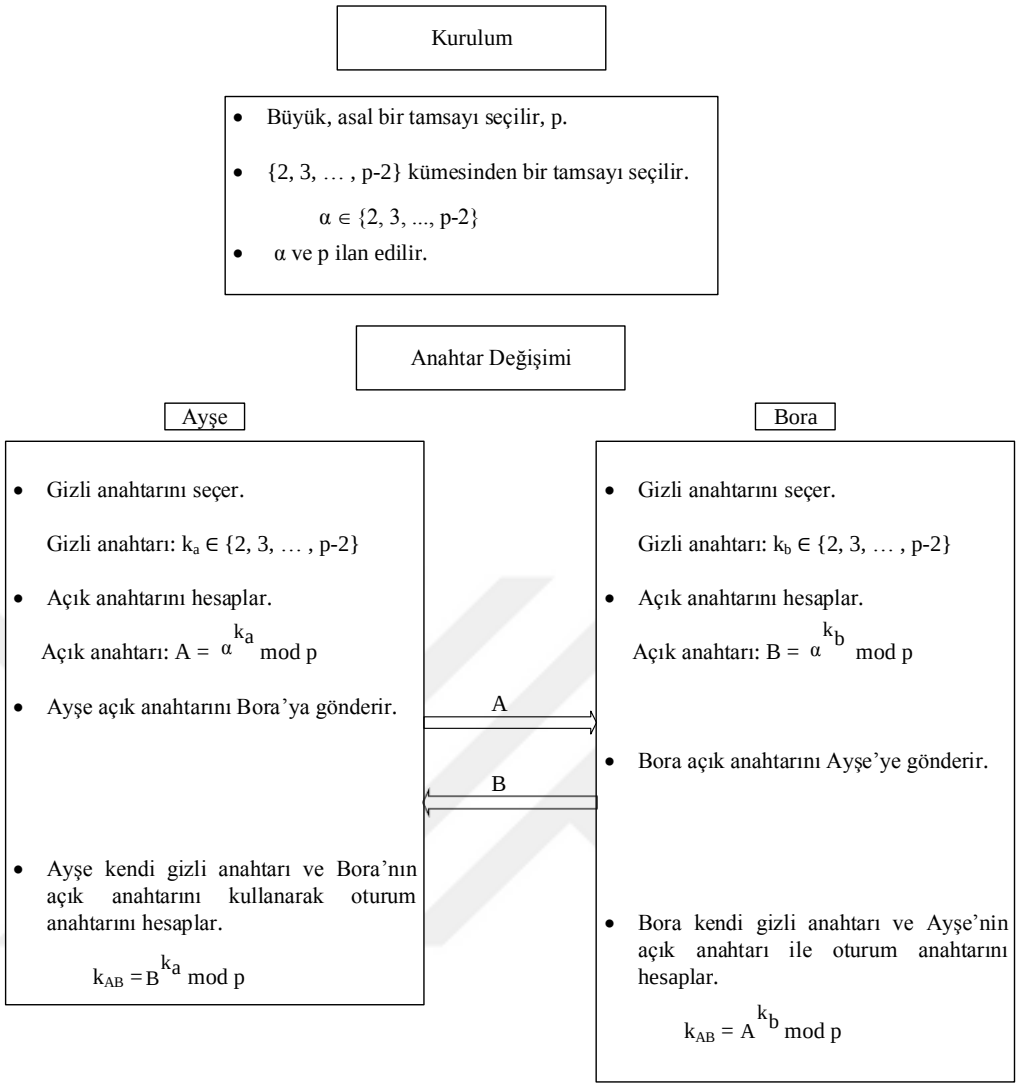


Şekil 2.12 RSA şifreleme ve şifre çözme algoritması (Paar ve Pelzl 2009)

2.1.2.2 Diffie-Hellman anahtar değişimi protokolü

Diffie-Hellman anahtar değişimi yöntemi 1976'da Whitfield Diffie ve Martin Hellman tarafından önerilmiştir. Literatürde yer alan ilk asimetrik yöntemdir. Bu yöntemin amacı anahtar dağıtımı problemine bir çözüm sunmaktır. Ayrık logaritma problemi temel alınarak tasarlanmıştır (Çetin 2006, Paar ve Pelzl 2009).

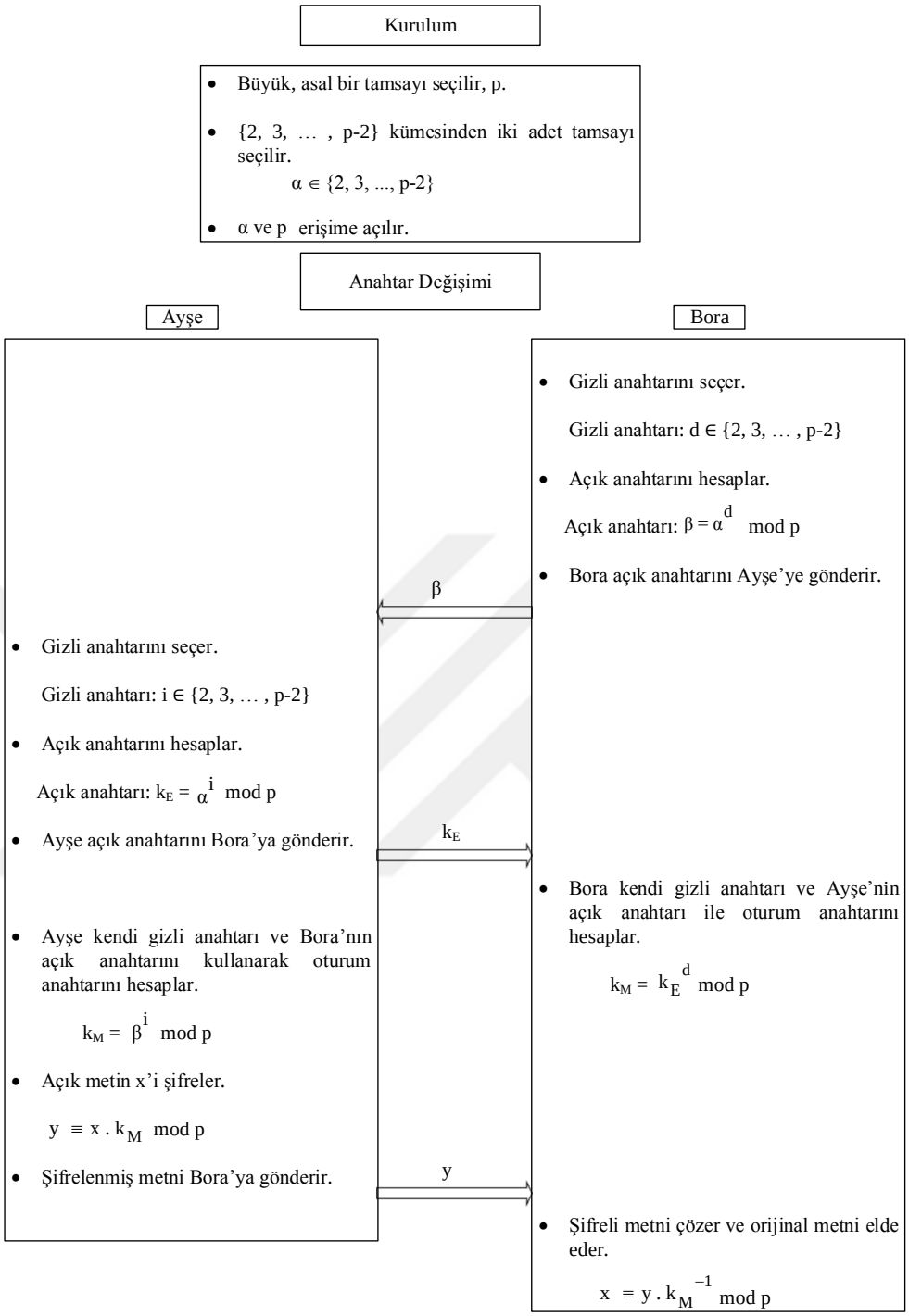
Diffie-Hellman anahtar değişimi yöntemi kurulum (*set-up*) ve anahtar değişimi (*key-exchange)* olmak üzere iki aşamalıdır. Kurulum adımında çalışma parametreleri α ve p oluşturulmaktadır. Bu parametrelere herkes tarafından erişilmektedir. Anahtar değişimi, haberleşme hattındaki tarafların ortak bir oturum anahtarı ürettikleri adımdır. Şekil 2.13' te kurulum ve anahtar değişimi ayrıntıları yer almaktadır (Paar ve Pelzl 2009).



Şekil 2.13 Diffie-Hellman anahtar değişimi protokolü (Paar ve Pelzl 2009)

2.1.2.3 ElGamal şifreleme sistemi

Taher Elgamal'in 1980'li yılların ilk yarısında ortaya attığı açık anahtarlı bir şifreleme tekniğidir. Güvenlik seviyesi ayrık logaritma probleminin zorluğu ile orantılıdır. Şekil 2.14'te görüldüğü üzere ElGamal şifreleme sisteminde kurulum ve anahtar değişimi işlemleri yapılmaktadır. Bu yönüyle Diffie-Hellman anahtar değişimi protokolü ile benzemektedir. Diffie-Hellman protokolünde kurulum aşamasını üçüncü bir taraf ElGamal yönteminde ise Bora gerçekleştirmektedir. ElGamal şifreleme tekniğinde kurulum adımı bir kez, şifreleme ve şifre çözme işlemleri mesaj gönderildiği sürece yürütülmektedir. (Paar ve Pelzl 2009, Stallings 2017).

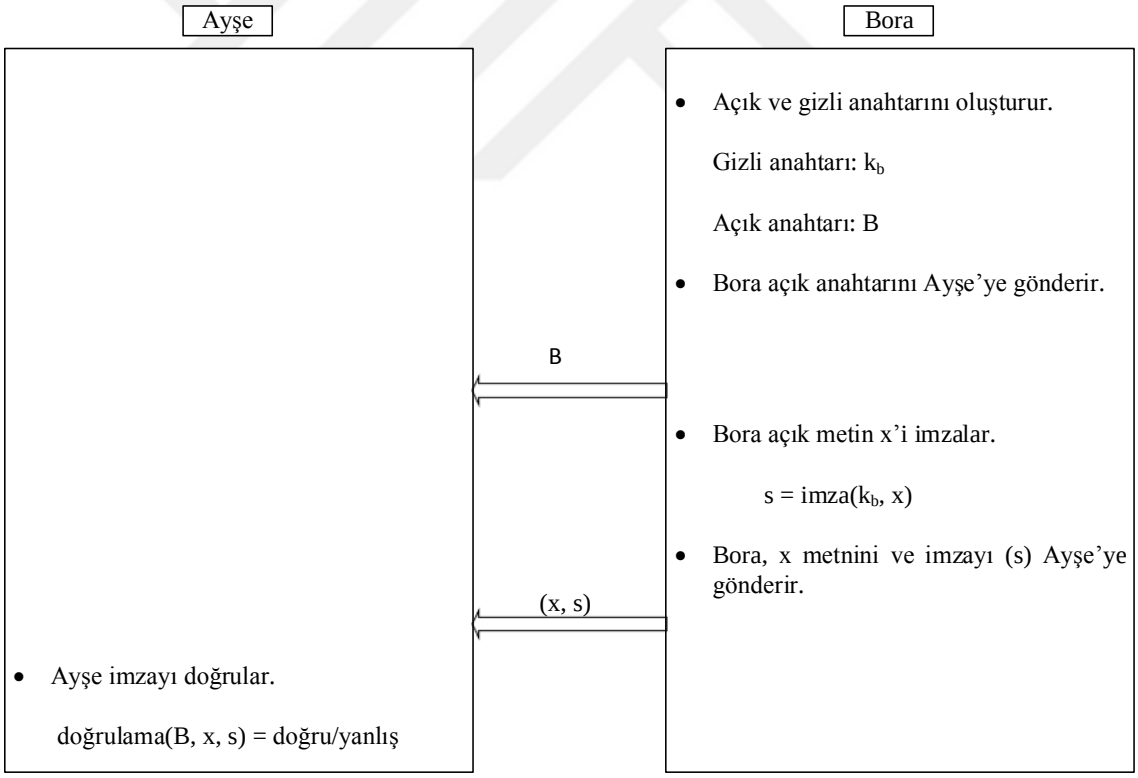


Şekil 2.14 ElGamal şifreleme tekniği (Paar ve Pelzl 2009)

2.1.2.4 Sayısal imza algoritması

1991'in ağustos ayında Ulusal Standartlar ve Teknoloji Enstitüsü tarafından önerilmiş bir algoritmadır. Açık anahtarlı sayısal imzalar bilgi kaynağını doğrulamada kullanılmaktadır (Schneier 1996, Paar ve Pelzl 2009).

Sayısal imza şemasının (Şekil 2.15) ilk aşaması Bora'nın açık anahtarını oluşturmasıdır. Bora açık anahtarını Ayşe ile paylaşmaktadır. Açık metin x'in özet (*hash*) fonksiyonunu gizli anahtarı ile şifreleyerek bir imza üreten Bora, x metnini ve bu imzayı Ayşe'ye göndermektedir. İmza, x metni olmadan bir anlam ifade etmemektedir. Ayşe x metni, imza ve Bora'nın açık anahtarı yardımıyla bilginin kaynağının Bora olup olmadığını doğrulamaktadır. Doğrulama işlemi 'Doğru (*True*)' ya da 'Yanlış (*False*)' ifadesi ile sonuçlanmaktadır (Özcan 2006, Paar ve Pelzl 2009).



Şekil 2.15 Sayısal imza algoritması (Paar ve Pelzl 2009)

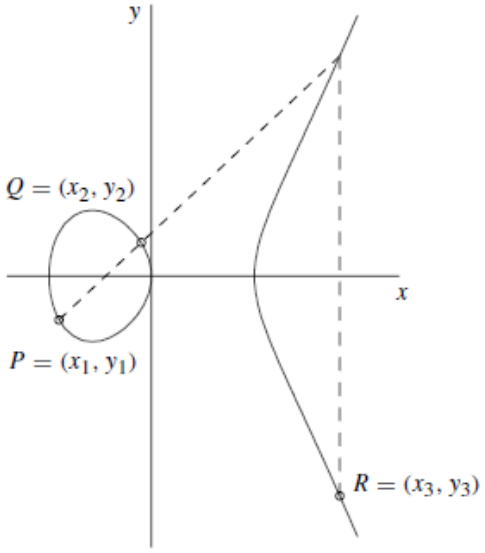
2.1.2.5 Eliptik eğri şifreleme sistemi

En yeni asimetrik şifreleme sistemidir. 19. yüzyılda eliptik eğriler üzerinde çalışmalar başlamıştır. Fakat eliptik eğrilerin kriptografi alanında kullanılması 1985 yılından sonradır. Eliptik eğri şifreleme algoritmalarının temelindeki matematiksel problem ayrık logaritma problemidir. Eliptik eğrilerin var olan açık anahtarlı şifreleme sistemlerinde kullanılabileceği düşünülmüştür. Bu düşünceyi savunan bilim insanları Neal Koblitz ve Victor Miller' dir. (Schneier 1996, Paar ve Pelzl 2009, Gülen 2014).

Açık anahtarlı şifreleme yöntemlerinde eliptik eğrilerin kullanımı bu yöntemlerin avantajlı olmasını sağlamaktadır. Uygulamaların daha kısa anahtar uzunlukları ile daha güvenli olması mümkün olmaktadır. Şifreleme, şifre çözme gibi işlemler daha hızlı yapılmakta, daha az miktarda güç tüketilmekte ve bellek kullanılmaktadır (Yavuz 2008).

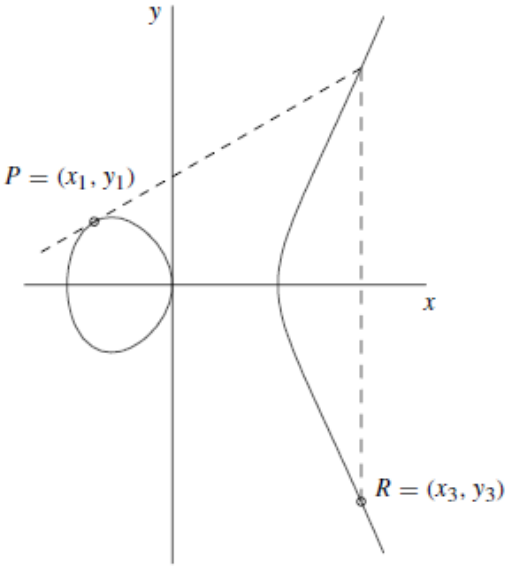
Eliptik eğrilerde iki grup işlemi söz konusudur. Nokta toplama (*point addition*) işleminde eliptik eğri üzerinde yer alan birbirinden farklı iki nokta toplanarak üçüncü bir nokta bulunmaktadır. Nokta aynılama (*point doubling*) işleminde eliptik eğri üzerindeki bir nokta aynılanarak yani kendisiyle toplanarak üçüncü bir nokta hesaplanmaktadır. Her iki işlem sonucunda hesap edilen nokta eğri üzerinde tanımlıdır. Eğri üzerinde üçüncü bir noktanın hesaplanmasında kiriş ve teğet kuralından yararlanılmaktadır (Paar ve Pelzl 2009).

Şekil 2.16'da eğri üzerinde tanımlı $P = (x_1, y_1)$ noktası ile $Q = (x_2, y_2)$ noktalarının toplanması ve $R = (x_3, y_3)$ noktasının bulunması işleminin geometrik gösterimi yer almaktadır. Başlangıçta P ve Q noktaları eğri üzerinde işaretlenmektedir. Her iki noktadan geçen bir doğru çizilmekte ve bu doğrunun eliptik eğri ile kesişim noktası bulunmaktadır. Bu kesişim noktasının x eksenine göre simetriği R noktasıdır (Hankerson vd. 2003, Paar ve Pelzl 2009).



Şekil 2.16 Nokta toplama işleminin geometrik gösterimi (Hankerson vd. 2003)

Şekil 2.17'de eğri üzerinde tanımlı $P = (x_1, y_1)$ noktasının aynılanması ile $R = (x_3, y_3)$ noktasının bulunması geometrik formda gösterilmektedir. İlk önce eğri üzerindeki P noktasına işaret koyulmaktadır. P noktasına bir teğet çizilmekte ve bu teğet eliptik eğri ile kesiştirilmektedir. Bu kesişim noktasının x eksenine göre simetriği R noktasını ifade etmektedir (Hankerson vd. 2003, Paar ve Pelzl 2009).



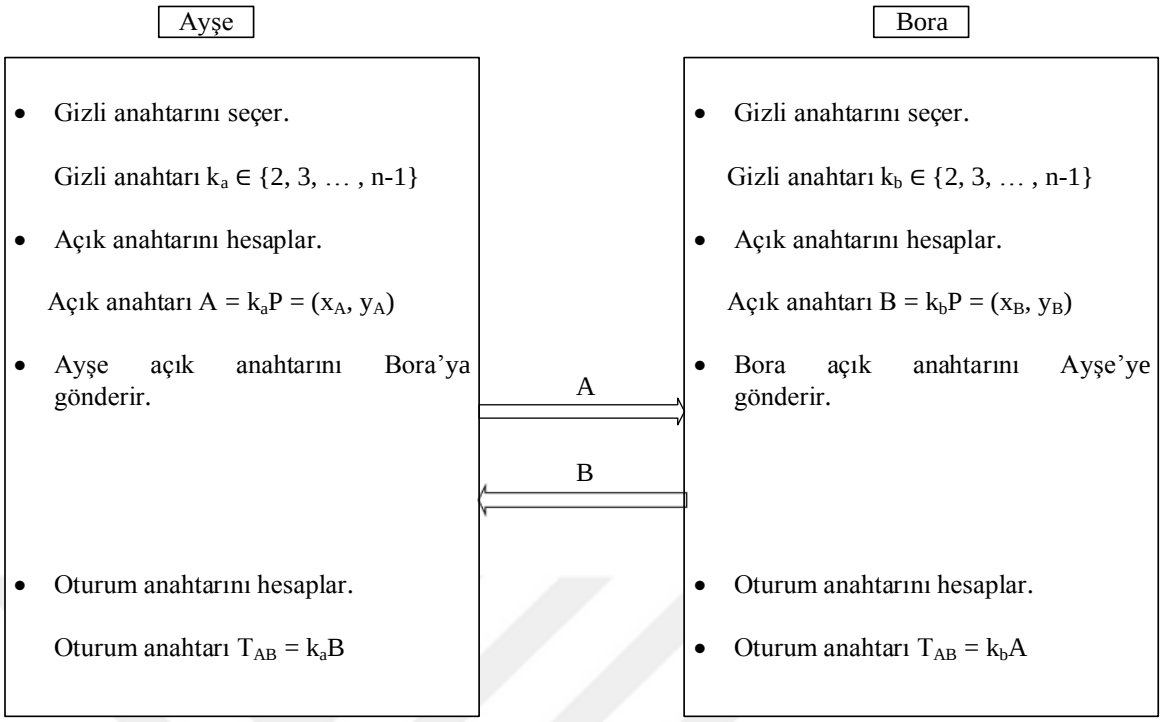
Şekil 2.17 Nokta aynılama işleminin geometrik gösterimi (Hankerson vd. 2003)

Diffie-Hellman Anahtar Değişimi Protokolü ve Sayısal İmza Algoritması için sonlu cisimlerde tanımlı eliptik eğrilerden yararlanılmaktadır. Bu eğrilerin mertebesi n olup genel denklemi denklem 2.5 ile verilmektedir. Denklem katsayıları a ve b sonlu cismin birer elemanıdır. p, büyük bir asal tamsayıyı temsil etmektedir. Kullanılan eliptik eğriler tekil olmayan (*non-singular*) eğrilerdir. Denklem 2.6'da bir eliptik eğrinin tekil olmaması için sağladığı koşul görülmektedir. $P = (x_P, y_P)$, eliptik eğri üzerinde bulunan bir noktadır (Paar ve Pelzl 2009).

$$E: y^2 \equiv x^3 + a \cdot x + b \bmod p \tag{2.5}$$

$$4 \cdot a^3 + 27 \cdot b^2 \neq 0 \bmod p \tag{2.6}$$

Diffie-Hellman anahtar değişimi protokolü (Şekil 2.18) sayesinde haberleşen taraflar arasında anahtar dağıtımı sorunu aşılmaktadır. Gönderici Ayşe, alıcı Bora açık anahtarlarını güvenli olmayan bir haberleşme yolu üzerinden birbirlerine iletmektedir. Açık anahtarların oluşturulmasında gizli anahtarlar kullanılmaktadır. Ayşe kendi gizli anahtarını ve Bora'nın açık anahtarını; Bora kendi kişisel anahtarını ve Ayşe'nin açık anahtarını kullanarak bir oturum anahtarı üretmektedir. Bu iki anahtarın birbirine eşit olması durumunda protokol başarıya ulaşmaktadır ve doğrulama gerçekleşmektedir. Ancak anahtarlar birbirinden farklıysa doğrulama sağlanmamaktadır (Paar ve Pelzl 2009).



Şekil 2.18 Eliptik eğri Diffie-Hellman anahtar değişimi protokolü (Paar ve Pelzl 2009)

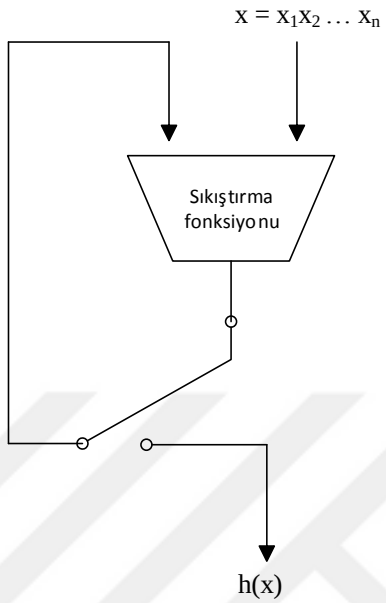
Sayısal İmza Algoritması üç adımda gerçekleştirilmektedir. İlk başta taraflar gizli anahtarlarını kullanarak açık anahtarlarını oluşturmaktadır. İkinci adımda gizli ve açık anahtarlar ile açık metin x'in imzası üretilmektedir. Son olarak imzanın geçerli olup olmadığı araştırılmaktadır (Paar ve Pelzl 2009).

2.2 Özet Fonksiyonları

Sayısal imza uygulamalarında yaygın olarak kullanılan özet fonksiyonları ile mesaj doğrulama ve veri bütünlüğü hedefleri karşılanmaktadır. Özet fonksiyonları, uzunluğu değişken olan bir girdiden sabit uzunlukta bir çıktı üretmeyi sağlamaktadır (Menezes vd. 1996, Paar ve Pelzl 2009, Stallings 2017).

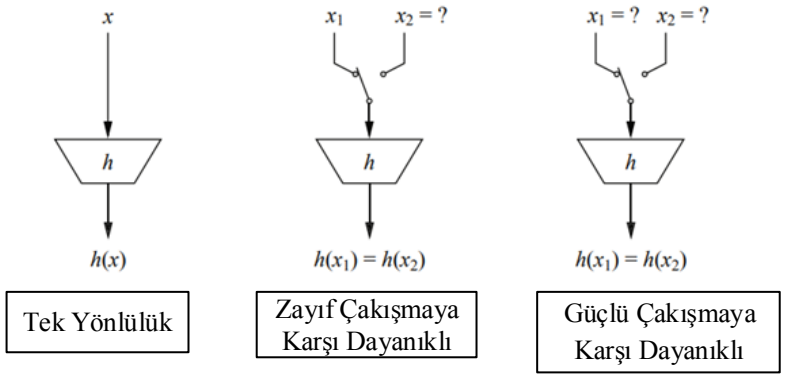
Sıkıştırma (*compression*) fonksiyonu bir özet fonksiyonunun en önemli bileşenidir. Girdi mesajının özeti, sıkıştırma fonksiyonunun son döngüsünde elde edilmektedir. Sıkıştırma fonksiyonu tekrarlı bir yapıdadır. Bu tasarımın adı Merkle-Damgard

yapısıdır. Merkle-Damgard özet fonksiyonu şekil 2.19' daki gibidir (Paar ve Pelzl 2009).



Şekil 2.19 Merkle-Damgard özet fonksiyonu (Paar ve Pelzl 2009)

Özet fonksiyonunun girdisi olan x mesajı rasgele, çıktısı sabit bir uzunluktadır. x mesajının özet fonksiyonu h(x) = z'nin hesaplanması kolaydır. Güvenli özet fonksiyonları (Şekil 2.20) tek yönlüdür (*one-wayness/preimage resistance*), zayıf çakışmaya (*second preimage resistance/weak collision resistance*) ve güçlü çakışmaya (*second preimage resistance/strong collision resistance*) karşı dayanıklıdır. Tek yönlü bir özet fonksiyonu için z = h(x) eşitliği verildiğinde x'in elde edilmesi imkansızdır. Zayıf çakışmaya karşı dayanıklı olan bir özet fonksiyonunda x_1 mesajı ve $h(x_1) = h(x_2)$ eşitliği verildiğinde x_2'nin bulunması mümkün olmamaktadır. Güçlü çakışmaya dirençli olan bir özet fonksiyonunda $h(x_1) = h(x_2)$ eşitliği verildiğinde $x_1 \neq x_2$ koşulunu sağlayan (x_1, x_2) çifti bulmak olanaksızdır (Paar ve Pelzl 2009).



Şekil 2.20 Güvenli özet fonksiyonlarının özellikleri (Paar ve Pelzl 2009)

Merkle-Damgard tasarımının yanı sıra sekiz farklı özet fonksiyonu geliştirilmiştir. Bu özet fonksiyonları MD2, MD3, MD4, MD5, RIPE-MD, Haval, N-Hash ve SHA algoritmalarıdır. Ronald Rivest tarafından geliştirilmiş olan özet fonksiyonları MD2, MD3, MD4 ve MD5'tir. MD2, güvenlik seviyesi bitlerin rasgele permütasyonu ile ilişkili olan bir özet fonksiyonudur. MD3 algoritması eksik yönlerinin çok olmasından dolayı kullanım aşamasına geçememiştir. Girdinin 32 bit, çıktının 128 bit olduğu MD4 mantıksal ve (and), veya (or), dışlayan veya (exclusive or), değil (not) operatörleri ile bit bit işlem yapmaktadır. MD5, RIPE-MD ve SHA, MD4'ün birer versiyonudur. MD5, MD4 ile aynı tasarıma sahip olup ve 128 bit çıktı üretmektedir fakat MD4'ten daha karmaşıktır. RIPE-MD, MD4'e kıyasla kripto analitik saldırılara daha dayanıklı olma özelliğine sahiptir. Haval özet fonksiyonu MD5'in değiştirilmesiyle elde edilmiştir. Haval algoritmasının girdileri 1024 bit iken çıktıları 128, 160, 192, 224 veya 256 bit uzunluğunda olabilmektedir. N-Hash özet fonksiyonunda 128 bitlik mesaj bloklarından 128 bitlik özet değeri üretilmektedir. MD4 ve MD5'in güvenlik zaafiyeti bulunmaktadır. Bu nedenle yeni bir mesaj özeti algoritması SHA önerilmiştir. SHA, NIST tarafından 1993 yılında yayınlanmış olan bir mesaj özetleme standardıdır (Paar ve Pelzl 2009, Budak 2010).

SHA-0, bilinen ilk SHA algoritmasıdır. SHA-1, SHA-0'ın değiştirilmiş ve gelişmiş halidir. SHA-0 ve SHA-1'in ürettiği çıktılar 160 bittir. MD5 algoritmasına kriptografik bir saldırı yapılması SHA-1'in kabul görmesi ve yaygın kullanılması ile sonuçlanmıştır. Asimetrik şifreleme yöntemlerinin yüksek güvenlik ihtiyacını karşılamak için üç farklı SHA-1 versiyonu geliştirilmiştir. 256 bitlik mesajların özeti için SHA-256, 384 bitlik

mesajların özeti için SHA-384 ve 512 bitlik mesajların özeti için SHA-512 algoritmalarından faydalanılmaktadır. SHA-224, simetrik şifreleme yöntemi 3DES'e yönelik bir özet algoritmasıdır. SHA-224, SHA-256, SHA-384 ve SHA-512; SHA-2 ailesinin birer üyeleridir. Çizelge 2.2'de MD5, SHA-1 ve SHA-2 ailelerinin başlıca parametreleri görülmektedir (Paar ve Pelzl 2009).

Çizelge 2.2 Özet fonksiyonlarının başlıca parametreleri (Paar ve Pelzl 2009)

Algoritma	Çıktı (bit uzunluğu)	Girdi (bit uzunluğu)	Döngü Sayısı	Çakışma Durumu
MD5	128	512	64	var
SHA-1	160	512	80	henüz yok
SHA-224	224	512	64	yok
SHA-256	256	512	64	yok
SHA-384	384	1024	80	yok
SHA-512	512	1024	80	yok

SHA-1 gibi özet fonksiyonlarında blok şifreleme yöntemleri kullanılabilmektedir. Orijinal x mesajı eşit uzunluklu i adet bloğa bölünmektedir. Mesaj blokları x_i, b boyutlu olan bir blok şifre ile şifrelenmektedir. $x_1x_2\ldots x_n$ mesaj bloğunun özeti hesap edilirken XOR işleminden yararlanılmaktadır. Özet fonksiyonlarının gerçekleştirilmesinde kullanılan blok şifreler çeşitlidir (Paar ve Pelzl 2009).

Özet fonksiyonları anahtarlı ya da anahtarsız olabilmektedir. Anahtarlı özet fonksiyonu mesaj doğrulama kodu (*message authentication code*) adıyla da anılmaktadır. Mesaj doğrulama kodu ile mesajın bütünlüğü korunmaktadır ve mesaj doğrulanmaktadır. Mesaj doğrulama kodu simetrik şifreleme yapılarını kullanmaktadır ve hız açısından avantajlıdır. Mesaj doğrulama kodu kullanılan kriptografik bir sistemde inkar edilememe hedefi sağlanamamaktadır. Mesaj doğrulama kodlarında bir simetrik anahtar kullanılarak hem doğrulama etiketi oluşturulmakta hem de bu etiketi doğrulama işlemi yapılmaktadır. Orijinal mesaj ile simetrik anahtarın bir fonksiyonu olan mesaj doğrulama kodları ile mesajın gönderim sırasında değiştirilip değiştirilmediğinden emin olunmaktadır. Mesaj doğrulama kodu SHA-1 gibi özet fonksiyonları veya AES ve benzeri blok şifreleme algoritmaları kullanılarak üretilebilmektedir (Çetin 2006, Paar ve Pelzl 2009).

2.3 Grup

Kapalılık ve birleşme özelliklerini sağlayan, her elemanın tersinin bulunduğu, bir birim (etkisiz) elemanın mevcut olduğu, iki değişkenli tek bir işlemin tanımlandığı kümelere grup denilmektedir.

Boş olmayan bir G kümesi ve bu küme üzerinde tanımlı iki değişkenli * işleminden oluşan bir grubun sahip olması gereken özellikler aşağıda sıralanmaktadır:

- * işlemi G'de kapalıdır.
  G'de tanımlı a ve b elemanları için a * b elemanı da G'de tanımlıdır.
- * işlemi G'de birleşme özelliğini sağlamaktadır.
  G'de tanımlı a, b ve c elemanları için a * (b * c) = (a * b) * c olmaktadır.
- G kümesinde * işleminin bir birim elemanı mevcuttur. G'de tanımlı birim elemanı e ile temsil edilmektedir.
  a * e = e * a = a eşitliği sağlanmaktadır.
- G kümesinde tanımlı her bir elemanın * işlemine göre tersi bulunmaktadır. G'de tanımlı a elemanının tersi a^{-1} ile gösterilmektedir.
  $a * a^{-1} = a^{-1} * a = e$ eşitliği sağlanmaktadır.

Kapalılık ve birleşme özelliklerini sağlayan fakat birim ve ters elemanları tanımlı olmayan bir grup yarı grup adıyla anılmaktadır.

Kapalılık ve birleşme özelliklerine sahip olan, birim ve ters elemanları bulunan bir grupta * işleminin değişme özelliği de mevcut ise bu grup değişmeli grup olarak adlandırılmaktadır. G'nin elemanları a ve b olmak üzere a * b = b * a eşitliği sağlanıyor ise * işleminin değişme özelliği vardır.

2.4 Halka (*Ring*)

Halkalarda iki işlem tanımlıdır. Boş olmayan bir R kümesi ve bu küme üzerinde tanımlı iki değişkenli + ve □ işlemlerinden oluşan bir halka aşağıdaki şartları yerine getirmektedir:

- Değişmeli bir grup olan R, + işlemine göre kapalılık, birleşme, değişme özelliklerini sağlamaktadır ve birim, ters elemanları bulunmaktadır.
- Bir yarı grup olan R, □ işlemine göre kapalıdır ve birleşmelidir. Birim ve ters elemanları yoktur.
- □ işlemi + işlemi üzerinde sağdan ve soldan dağılma özelliğine sahiptir. R'de tanımlı a, b ve c elemanları için sağdan dağılma a □ (b + c) = a □ b + a □ c, soldan dağılma (a + b) □ c = (a □ c) + (b □ c) eşitlikleri sağlanmaktadır.

Bir halkanın sıfır elemanı bu halkanın + işleminin etkisiz (sıfır) elemanı olmaktadır ve 0_R sembolü ile temsil edilmektedir. Birimli bir halkanın □ işlemine göre birim elemanı mevcuttur. Birim elemanın sembolü 1_R'dir. □ işlemine göre değişme özelliği olan bir halkanın adı değişmeli halkadır. Tek elemandan oluşan ve bu elemanı 0 olan halkaya sıfır halka ya da aşikar halka adı verilmektedir.

R halkasında bulunan etkisiz eleman olmayan a için a □ b = 0_R eşitliğini sağlayacak şekilde etkisiz eleman olmayan b var ise a, R halkasının sıfır bölenidir. R halkasının sıfır böleni yok ise R, bir tam halkadır denilmektedir. Tamlık bölgesi ile birimli, değişmeli ve sıfır böleni olmayan bir halka ifade edilmektedir.

2.5 Cisim (*Field*)

Bir F kümesi ve bu küme üzerinde tanımlı iki değişkenli + ve □ işlemlerinden oluşan bir cisim aşağıda belirtilen aksiyomları sağlamaktadır:

- Birimli ve değişmeli bir halka olan F'de + ve □ işlemlerine göre birim eleman tanımlıdır ve değişme özelliğine sahiptir.
- F kümesine + işleminin etkisiz elemanı dahil edilmediğinde F kümesi □ işlemine göre bir grup olmaktadır.

Bir cismin sıfır olmayan elemanlarının her biri için tek çarpımsal ters mevcuttur.

2.6 Sayı Teorisi

Sayı teorisi simetrik ve asimetrik kripto sistemlerin dayandığı matematiksel problemlerin anlaşılmasına yardımcı olmaktadır. Bu nedenle sayı teorisinin bilinmesi kritiktir. Sayı teorisi kapsamında bahsedilmekte olan tamsayı aritmetiği için (Stallings 2017, Afacan 2017); üs alma işlemi için (Çetin 2006); tek yönlü fonksiyon, Euler'in Phi fonksiyonu, üs alma işlemi, Kare Alma ve Çarpma, Öklid ve Genişletilmiş Öklid algoritmaları, Ayrık Logaritma Problemi, Fermat ve Euler teoremleri için (Paar ve Pelzl 2009, Stallings 2017) kaynaklarından yararlanılmaktadır. Kullanılan notasyon bu kaynaklarınki ile birebir aynı olup hiçbir katkı yapılmamıştır.

2.6.1 Tamsayılarda aritmetik işlemler

+ ve □ işlemlerine göre değişme özelliğine sahip, birim elemanı tanımlı ve sıfır bölensiz olan tamsayılar kümesi bir halkadır, ℤ ile simgelenmektedir. Tamsayılar kümesinde + işlemi için kapalılık, birleşme ile değişme özellikleri, sıfır ve ters elemanlar tanımlıdır. □ işlemi için kapalılık, birleşme ve değişme özellikleri sağlanmakta olup birim eleman bulunmaktadır. □ işleminin + işlemi üzerinde dağılma özelliği vardır.

Tamsayılar kümesinin birer elemanı a ve b olmak üzere b = a □ c eşitliğini sağlayan bir c bulunuyorsa ve c tamsayılar kümesinde tanımlı ise a, b'yi bölmektedir. c tamsayısını bulmaya yönelik işlem kalansız bölme işlemidir. Bölünen b, bölen a ve bölüm c'dir. b'nin çarpanları a ile c'dir. a ile b tamsayılarının ilgili olmaları hem b'nin a'yı bölmesi hem de a'nın b'yi bölmesi ile mümkündür. a ve b'nin ilgili olduğu a ≈ b ile ifade edilmektedir.

m ile n'nin tamsayılar kümesinde tanımlı ve m'nin sıfırdan farklı olduğu bilinmektedir. Kalanlı bir bölme işleminde n, m'ye bölündüğünde bölüm q ve kalan r'dir. q ve r bir tamsayıdır. r, 0'dan büyük veya eşit ve m'nin mutlak değerinden küçük olmaktadır.

Sadece 1'e ve kendisine bölünebilen tamsayılar asal tamsayılardır. 2, asal tamsayıların en küçüğüdür. 1'den büyük her tamsayı en az bir adet asal tamsayıya bölünebilmektedir. Asal tamsayı kümesinin elemanları sonsuz sayıdadır.

Değeri sıfırdan farklı olan m ve n tamsayılarının her ikisini de bölen sıfırdan büyük bir d tamsayı bulunabilirse d, m ve n'nin ortak bölenidir. Eğer d, hem m hem n'yi bölen en büyük tamsayı ise d'ye m ve n'nin en büyük ortak böleni denir. En büyük ortak böleni 1 olan tamsayılar aralarında asaldır.
a, b ve n birer tamsayı ve n sıfırdan büyük olmak üzere a mod n, a'nın n'ye bölümünden kalandır. a mod n ve b mod n işleminin sonucu birbirinin aynısı ise a ile b denktir. Bu denklik bağıntısı $a \equiv b \pmod{n}$ ile gösterilmektedir.

2.6.2 Tek yönlü fonksiyon

Genel bir ifadeyle tek yönlü bir f() fonksiyonu için y = f(x) hesabı hızlı ve kolaydır fakat $x = f^{-1}(x)$ hesabı imkansızdır.

2.6.3 Üs alma

Üs alma açık anahtarlı şifreleme yöntemlerinde sıklıkla faydalanılan bir işlemdir. Bir sayının üssünü almak o sayının üssü adedince kendisiyle çarpılması demektir. Örnek verilirse $b = a^e$ eşitliğinde b, a'nın e kez çarpılması ile hesaplanmaktadır.

2.6.4 Öklid (*Euclidean*) algoritması

Pozitif iki tamsayının en büyük ortak bölenini hesaplamak için Öklid algoritması kullanılmaktadır. Açık anahtarlı kriptografik yöntemlerde büyük tamsayılar ile çalışılmaktadır. Büyük tamsayıların çarpanlara ayrılması zor bir iştir. Büyük tamsayıların en büyük ortak bölenini hızlı bir şekilde bulmada Öklid algoritmasından faydalanılmaktadır.

Şekil 2.21'de Öklid algoritmasının nasıl çalıştığı izah edilmektedir.

2.6.5 Genişletilmiş Öklid algoritması (*Extended Euclidean Algorithm*)

Genişletilmiş Öklid algoritması pozitif iki tamsayının en büyük ortak böleninin yanı sıra çarpımsal terslerini de hesaplamaya imkan sağlamaktadır. Şekil 2.22'de genişletilmiş Öklid algoritmasının adımları sıralanmaktadır.

2.6.6 Euler'in Phi fonksiyonu ve Euler teoremi

Euler'in Phi fonksiyonu RSA başta olmak üzere açık anahtarlı şifreleme sistemlerinde kullanılan bir fonksiyondur. Eleman sayısı m olan ve pozitif tamsayıların oluşturduğu bir halkada m ile aralarında asal olan tamsayıların sayısı $\phi(m)$ ile temsil edilmektedir. Euler'in Phi fonksiyonunu küçük sayılar için hesaplamak kolay olmasına rağmen sayılar büyüdükçe bu hesabın yapılması oldukça yavaş ve zordur. Phi fonksiyonu m tamsayısı çarpanlarına ayrılarak kolaylıkla bulunmaktadır. $m = p_1^{e_1} \cdot p_2^{e_2} \cdots p_n^{e_n}$ şeklinde çarpanlarına ayrılırsa Phi fonksiyonunun değeri $\Phi(m) = (p_i^{e_i} - p_i^{e_i-1})$ eşitliğinin sonucuna eşittir. Örneğin eleman sayısı 6 olan $\mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}$ halkasında 6 çarpanlarına ayrılırsa $6 = 2^1 \square 3^1$'dir. $(2^1 - 2^0) \square (3^1 - 3^0)$ eşitliğinin sonucu 2 olmaktadır. Bu durumda $\phi(6) = 2$'dir.

Euler teoremine göre aralarında asal a ve m tamsayıları için $a^{\phi(m)} \equiv 1 \pmod m$ eşitliği sağlanmaktadır.

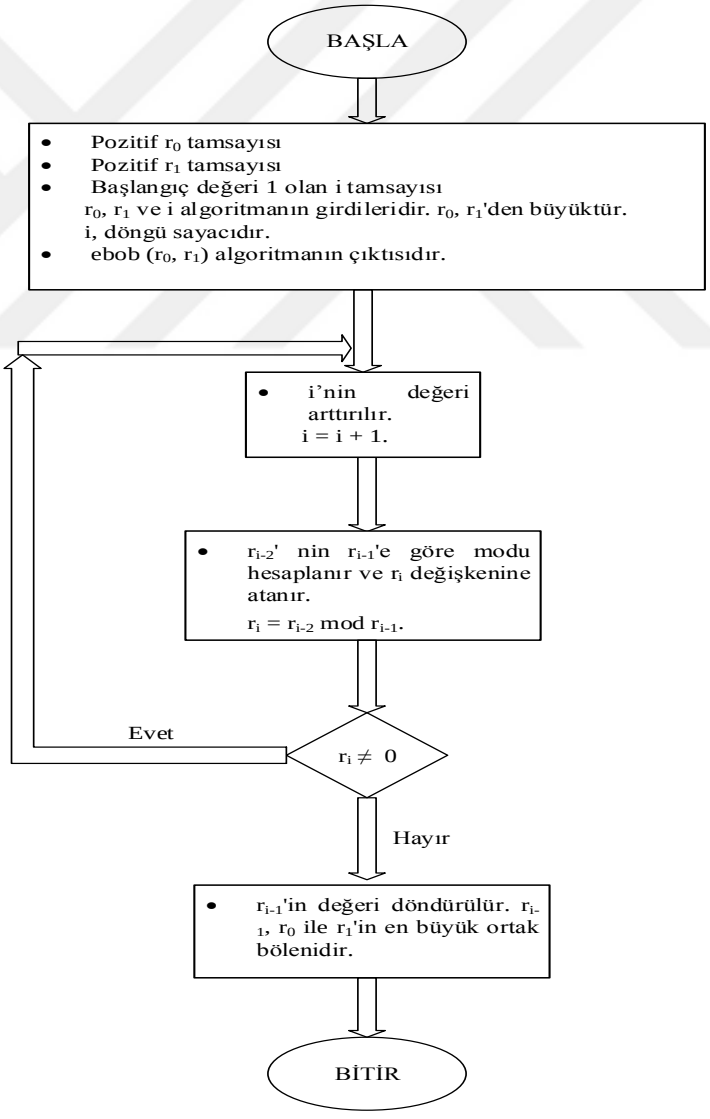
2.6.7 Fermat'ın küçük teoremi (*Fermat's Little Theorem*)

Akıllı kart uygulamalarında ya da üs alma işleminin hızlı olması gereken cihazlarda Fermat'ın küçük teoremini kullanmak avantajlıdır.

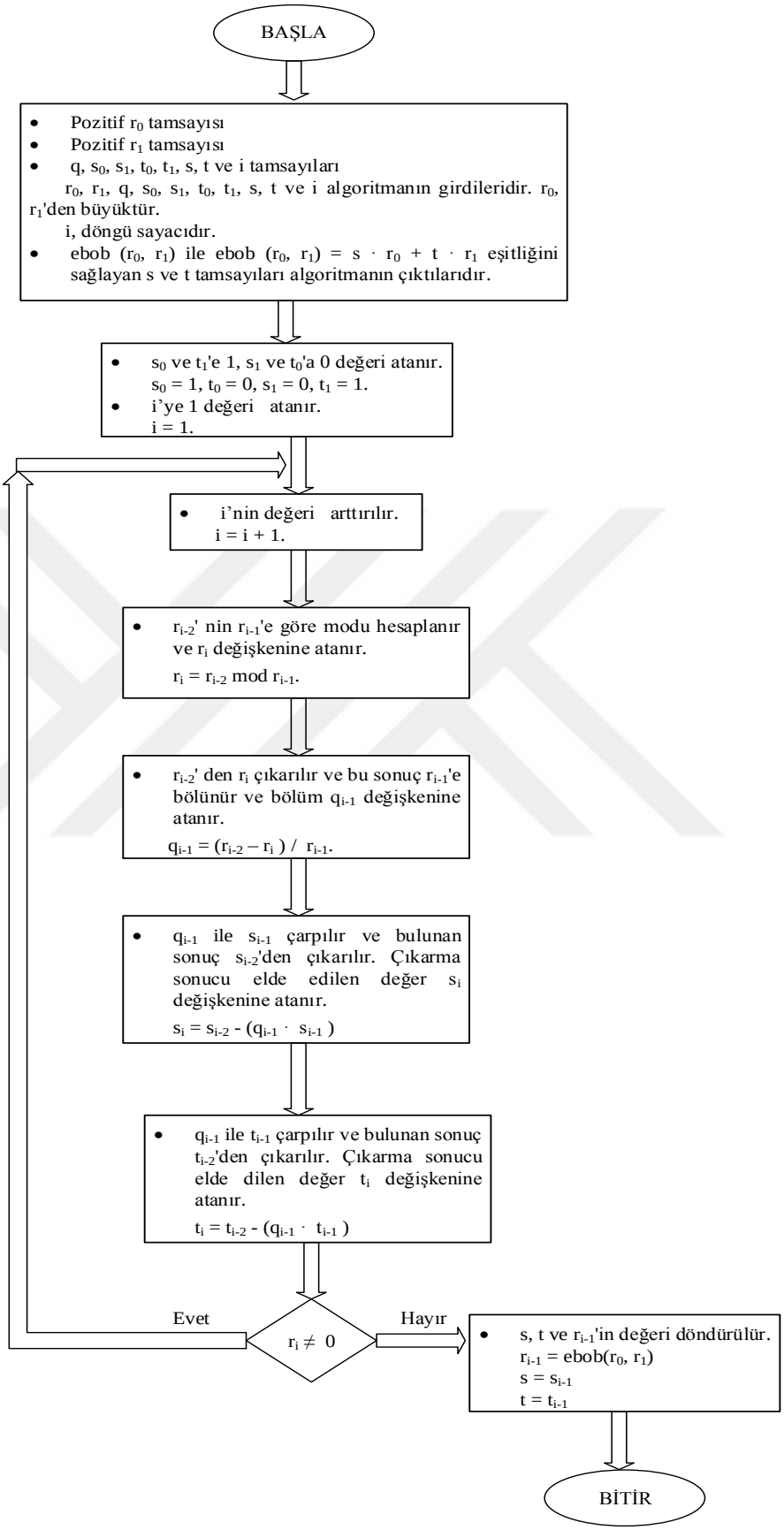
a bir tamsayı ve p bir asal sayı olmak üzere Fermat'ın küçük teoremi $a^p \equiv a \pmod p$ denkliğinin sağlandığını ifade etmektedir.

2.6.8 Ayrık logaritma problemi

Ayrık logaritma problemi sonlu ve periyodik Z_p^* grubunda tanımlıdır. Z_p^* grubunun elemanları sonlu sayıdadır ve belli aralıklarla yinelenmektedir. Derecesi (*order*) p-1 olmaktadır. Bu grubun en yüksek dereceli elemanlarına ilkel (*primitive*) kökler denilmektedir. α ve β, Z_p^* grubunun birer elemanıdır. p, asal bir tamsayıdır. α bir ilkel kök olmak üzere ayrık logaritma problemi $\alpha^x \equiv \beta \bmod p$ eşitliğinde $1 \leq x \leq p\text{-}1$ aralığında tanımlı x tamsayısını bulma problemidir.



Şekil 2.21 Öklid algoritması (Paar ve Pelzl 2009)



Şekil 2.22 Genişletilmiş Öklid algoritması (Paar ve Pelzl 2009)

2.7 Sonlu Cisim Aritmetiği

Şifreleme mekanizmalarında kullanılan eliptik eğrilerin aritmetik özelliklerinin bilinmesi gereklidir. Eliptik eğri aritmetiğinin temelinde sonlu cisimler vardır. Sonlu cisim aritmetiğinin anlatımında (Mano 1992, Hankerson vd. 2003, Stallings 2017) kitapları takip edilmektedir ve kullanılan notasyon hiçbir katkı yapılmadan sunulmaktadır.

Pek çok şifreleme algoritmasının uygulanmasında kullanılan sonlu cisimler matematikçi Evariste Galois tarafından keşfedilmiştir. Bu nedenle Galois cisimleri olarak da anılmaktadır. Özellikle Gelişmiş Şifreleme Standardı (AES) ve eliptik eğrilerle şifreleme yöntemleri sonlu cisim aritmetiği üzerine kuruludur. İki çeşit sonlu cisim vardır: 1. p elemanlı sonlu cisimler, 2. p^n elemanlı sonlu cisimler. Sonlu asal cisimler p elemanlı; sonlu ikili cisimler ve sonlu genişleme cisimleri p^n elemanlı sonlu cisimlerin örnekleridir.

2.7.1 p mertebeli sonlu cisimler

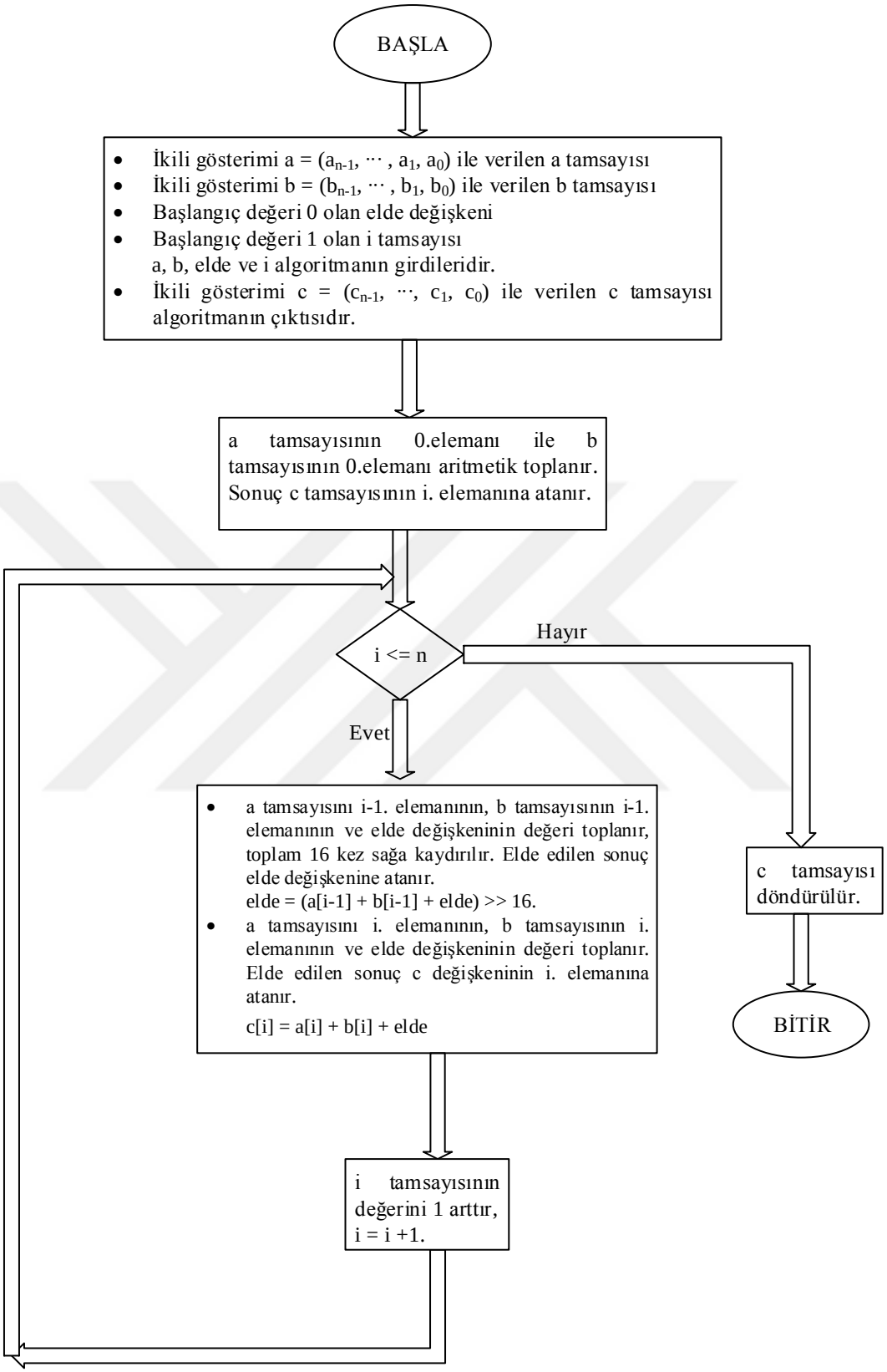
Mertebesi p olan bir sonlu cisim GF(p) ile temsil edilmektedir, bu cismin eleman sayısı p'dir ve p bir asal sayıdır. GF(p) asal cisminin elemanları {0, 1, …, p-1} olup aritmetik işlemler mod p'ye göre yapılmaktadır. {0, 1, …, p-1} kümesinin sıfırdan farklı olan tüm elemanları p asal sayısı ile aralarında asaldır ve sıfırdan farklı her elemanın bir çarpımsal tersi mevcuttur. İki sayı birbiri ile çarpıldığında sonucun bit sayısı bu iki sayının bit sayısının iki katı kadardır. Bu nedenle çarpma sonucunun mod p'ye göre indirgenmesine gerek duyulmaktadır. İndirgeme işlemi maliyetlidir. Eliptik eğri şemalarının hızı çarpma hızı ile doğrudan ilişkilidir. NIST tarafından önerilen asal sayılar kullanılarak hızlı indirgeme yapılabilmekte ve böylelikle çarpma işlemi hızlandırılabilmektedir.

Sonlu asal cisimlerde tanımlı aritmetik işlemler toplama, çıkarma, çarpma, çarpımsal ters

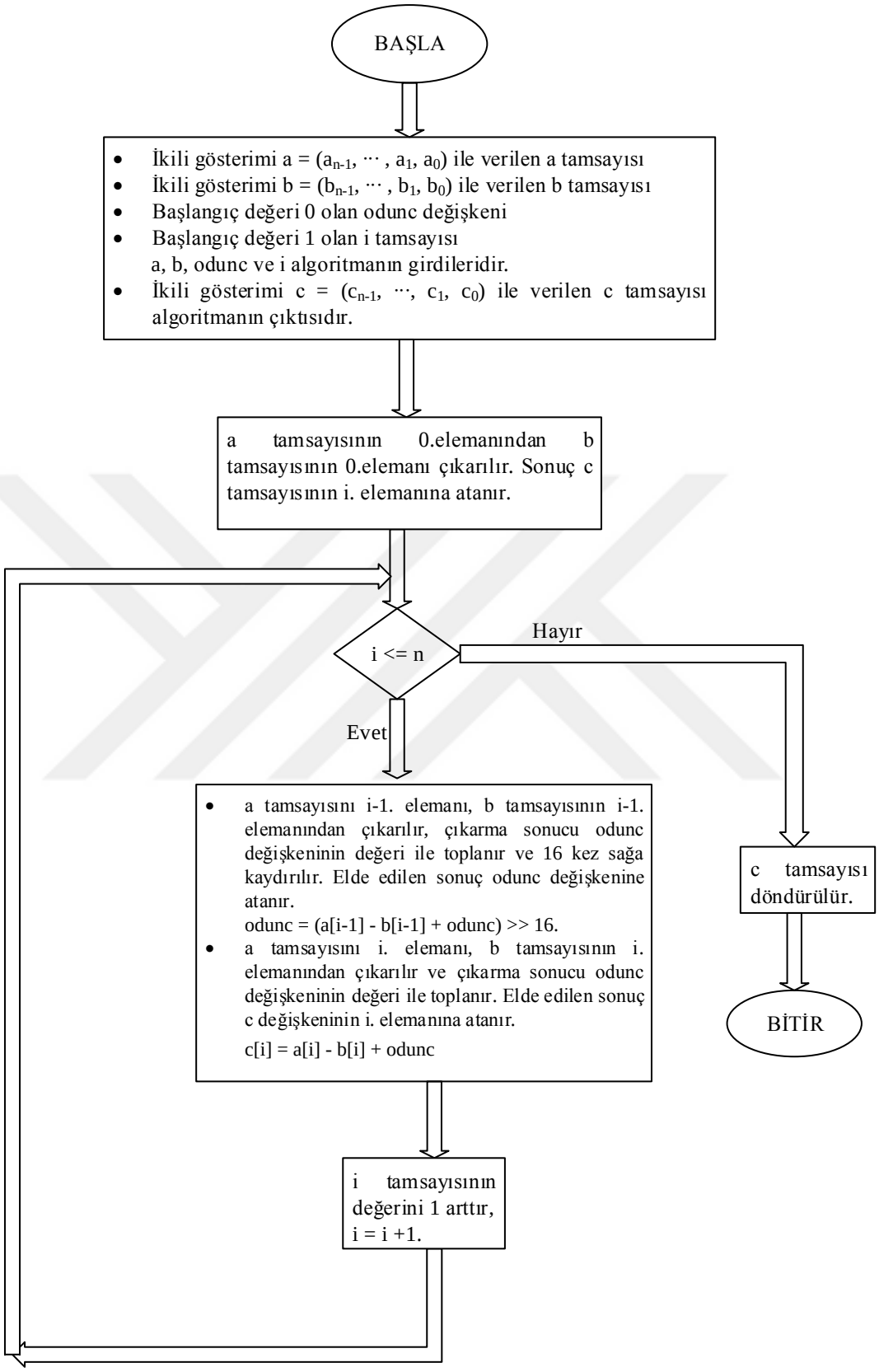
alma ve bölmedir. Şekil 2.23, 2.24, 2.25, 2.26 ve 2.27 sırasıyla toplama, çıkarma, çarpma, çarpımsal tersi alma ve bölme işlemlerine ait akış diyagramlarıdır.

Booth algoritması yardımıyla işaretli tamsayılar çarpılmaktadır. Şekil 2.25'te akış diyagramı verilen algoritmada negatif sayıların ikiye tümleyen formu kullanılmaktadır. Q tamsayısının n+1. biti ile ifade edilmek istenen A ve Q tamsayıları aritmetik olarak bir bit sağa kaydırılmadan önce Q tamsayısının 0. bitinin değeridir. Q tamsayısının n. biti, A ve Q aritmetik olarak bir bit sağa kaydırıldıktan sonra Q tamsayısının n-1. bitinin değerine eşit olmaktadır.

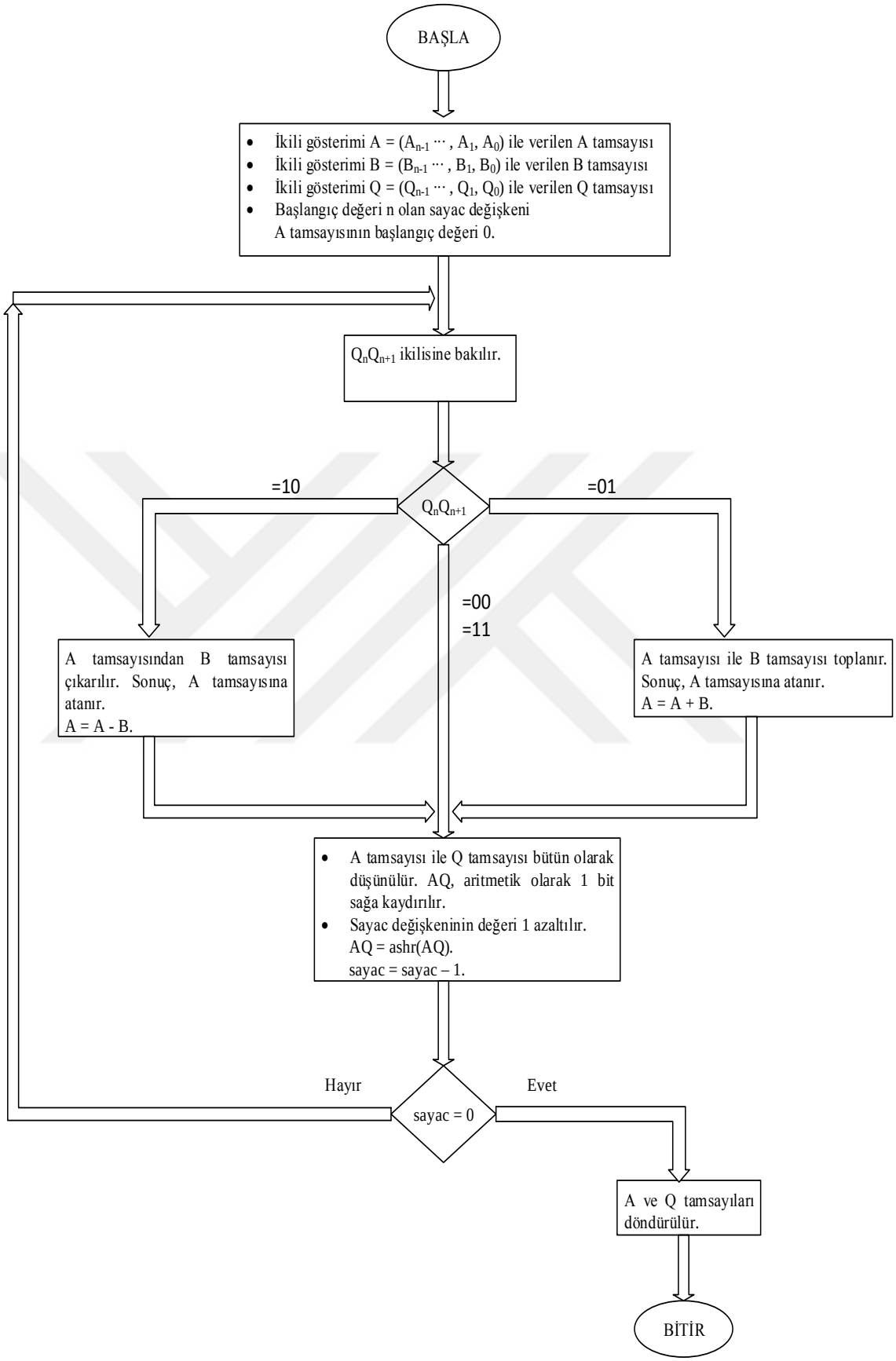
Bir tamsayı aritmetik olarak bir bit sağa kaydırılmak isteniyor olsun. Bu tamsayı aritmetik olarak bir bit sağa kaydırıldığında tamsayının n-1. biti ve n-2. biti eşittir. Bir başka deyişle bu tamsayının en soldaki bitinin değeri hiç değişmemektedir.



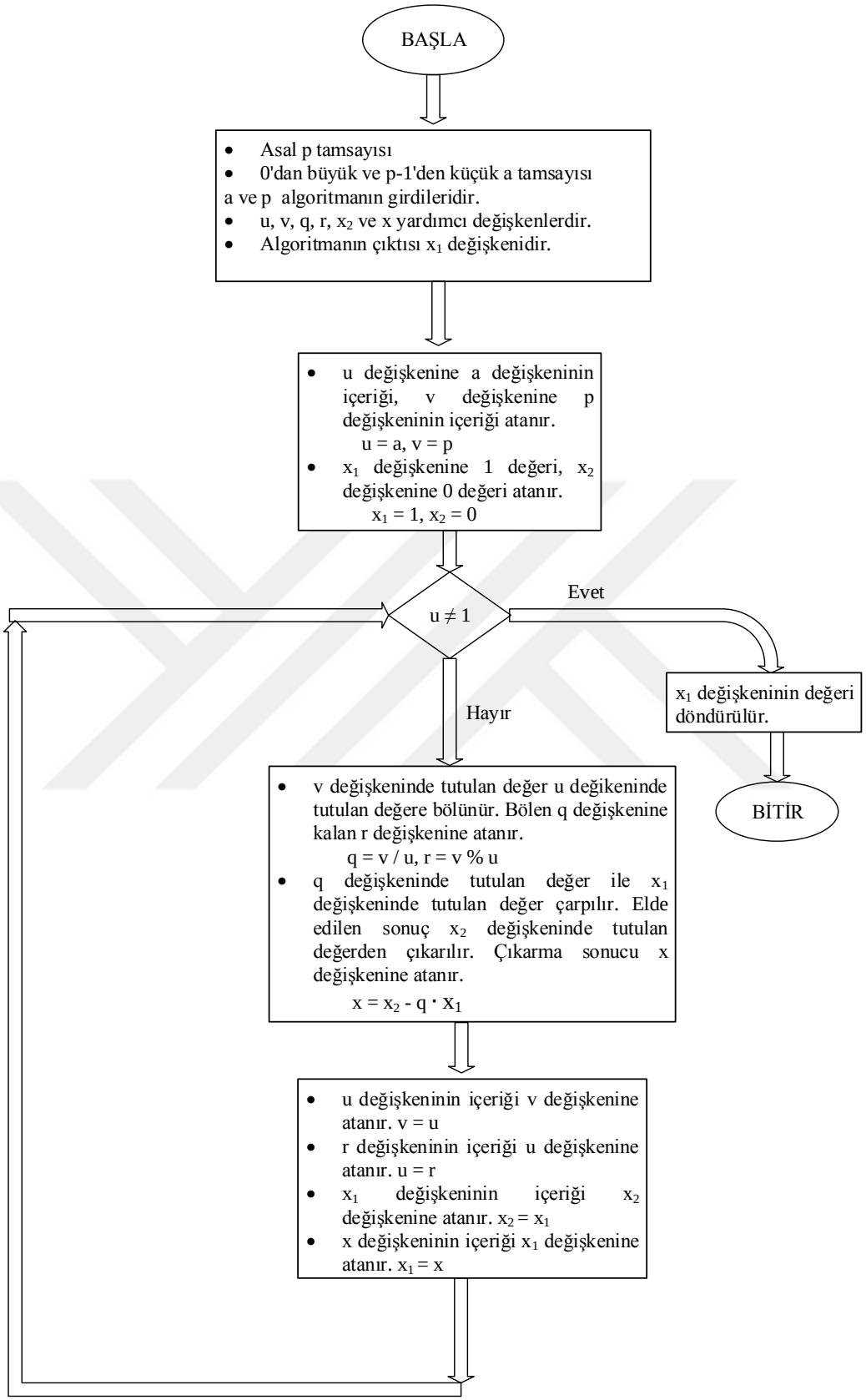
Şekil 2.23 Çok duyarlı toplama algoritması (Hankerson vd. 2003)



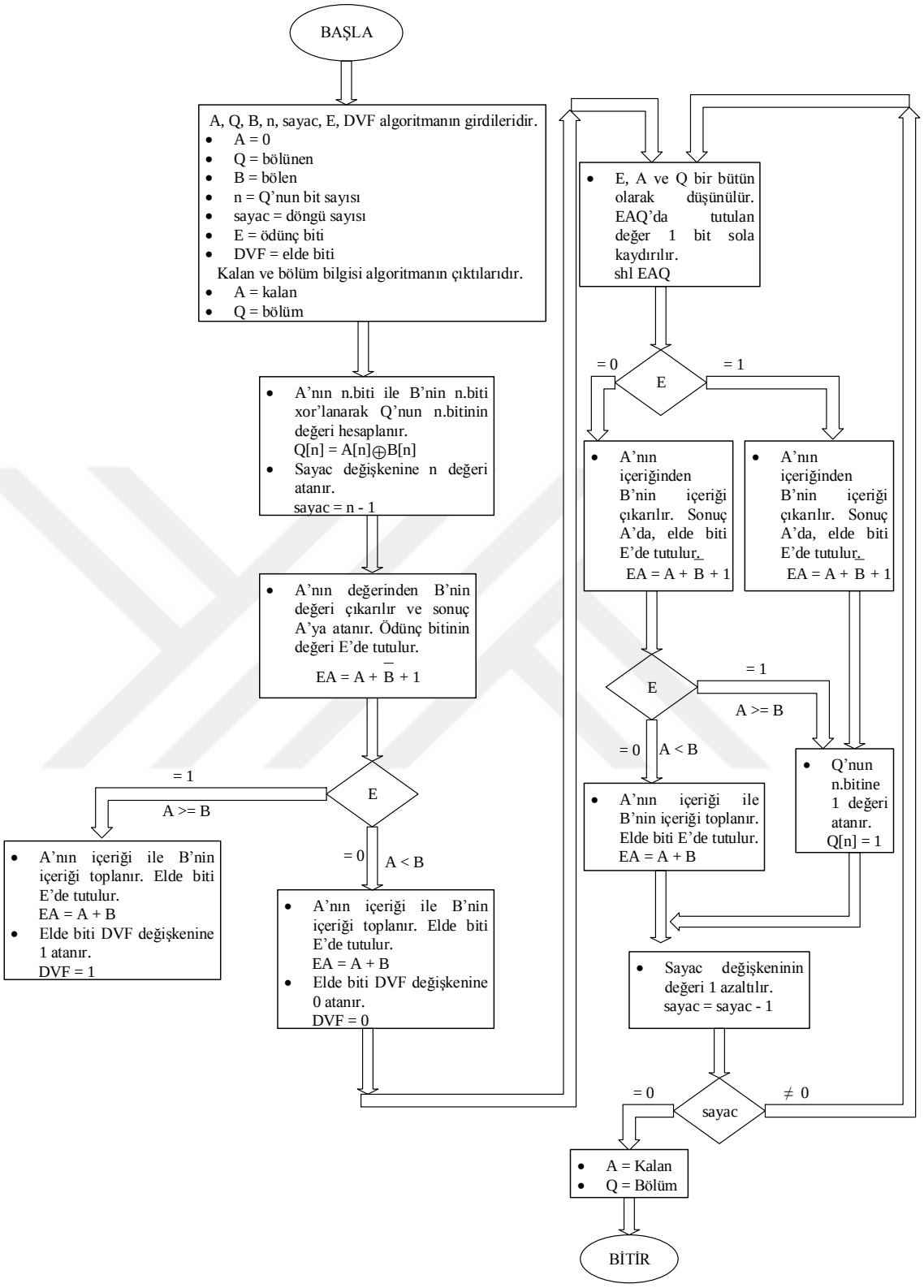
Şekil 2.24 Çok duyarlı çıkarma algoritması (Hankerson vd. 2003)



Şekil 2.25 İşaretli tamsayılar için çarpma algoritması (Mano 1992)



Şekil 2.26 Asal cisimlerde çarpımsal ters alma algoritması (Hankerson vd. 2003)



Şekil 2.27 İşaretli tamsayılarda bölme algoritması (Mano 1992)

2.7.2 p^n mertebeli sonlu cisimler

Mertebesi p^n olan bir sonlu cisim $GF(p^n)$ ile simgelenmektedir. p asal ve n pozitif bir tamsayıdır. Kriptografik algoritmaların güvenlik seviyesinin daha yüksek olmasını sağlaması nedeniyle $GF(2^n)$ formundaki sonlu cisimler kriptografide önemli bir yere sahiptir.

İkili cisim veya karakteristiği iki olan sonlu cisim olarak da anılan $GF(2^n)$ cismini oluşturma temel polinom aritmetiğinin kullanımı ile mümkündür. $GF(2^n)$ cisminin mertebesinin 2^n olması cismin eleman sayısının 2^n olmasına karşılık gelmektedir. Cismin elemanları derecesi maksimum n-1 olan ikili polinomlardır. Polinomların katsayıları {0, 1} kümesinin bir elemanıdır. $GF(2^n)$ cisminin genel polinom denklemi eşitlik (2.7)' de verilmektedir.

$$GF(2^n) = \{a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0 : a_i \in \{0, 1\}\} \quad (2.7)$$

Sonlu ikili cisimlerde toplama, çarpma, kare alma ve çarpımsal tersi bulma olmak üzere dört adet aritmetik işlem tanımlıdır. Aritmetik toplama, çarpma, kare alma ve çarpımsal ters işlemleri için kullanılan algoritmalara ait sözde kodlar sırasıyla şekil 2.28, şekil 2.29, şekil 2.30 ve 2.31 ile verilmektedir.

$GF(2^n)$ cisminde iki polinomun toplanması bit bit dışlayan veya (xor) işlemine karşılık gelmektedir. $GF(2^n)$ sonlu cisminde çarpma işleminin gerçekleştirilebilmesi için sağdan sola kaydır ve topla algoritması kullanılmaktadır. Sonlu ikili cisimlerde kare alma işlemi birbirine eşit iki sayının çarpılmasıdır. Bu nedenle çarpma için kullanılan algoritma kare alma işlemi için de geçerlidir. Kare alma işlemine ait sözde (*pseudo*) kod herhangi bir kaynaktan edinilmemiş, çarpma işleminin sözde (*pseudo*) kodundan faydalanılarak yazılmıştır. Çarpma işleminin sözde (*pseudo*) kodundan tek farkı başlangıçta bir atama komutunun bulunmasıdır.

Derecesi maksimum n-1 olan $GF(2^n)$ sonlu cisminde tanımlı bir a(z) polinomu a, f(z) indirgeme polinomu f ile gösterilmektedir. Eğer ebob (a, f) ≡ 1 ise a polinomunun

çarpımsal tersi bulunabilmektedir. Mertebesi n olan başka bir deyişle derecesi n olan indirgenemez bir f(z) polinomu seçilmektedir. f(z) polinomunun indirgenemez olması ile bu polinomun kendisinden daha düşük dereceli ikili polinomlar cinsinden ifade edilemediği yani çarpanlara ayrılamadığı kastedilmektedir.

Girdi:

İkili gösterimi a = (a_{n-1}, …., a_0) ile verilen a tamsayısı

İkili gösterimi b = (b_{n-1}, …., b_0) ile verilen b tamsayısı

Çıktı:

İkili gösterimi c = (c_{n-1}, …., c_0) şeklinde olan c tamsayısı

c = a + b

Algoritma:

1. for i = 0 : n-1

   c[i] = a[i] $\oplus$ b[i]

   end
2. c değeri döndürülür.

Şekil 2.28 Toplama işlemi için sözde kod (Hankerson vd. 2003)

Girdi:

İkili gösterimi a = (a_{n-1}, …., a_0) ile verilen a tamsayısı

İkili gösterimi b = (b_{n-1}, …., b_0) ile verilen b tamsayısı

İkili gösterimi f = (a_{n-1}, …., a_0) ile verilen f tamsayısı

Çıktı:

c = a · b mod f

Algoritma:

```
1. if a[0] = 1
       c = b
2. else
       c = 0
   end
3.  for i = 1 : n-1
3.1.   if b[n-1] = 1
           b = b << 1
           b = b + f
3.2.   else
           b = b << 1
       end
3.3.   if a[i] = 1
               c = c + b
       end
   end
4. c değeri döndürülür.
```

Şekil 2.29 Çarpma işlemi sözde kodu (Hankerson vd. 2003)

Girdi:

İkili gösterimi a = (a_{n-1}, …., a_0) ile verilen a tamsayısı

İkili gösterimi b = (b_{n-1}, …., b_0) ile verilen b tamsayısı

İkili gösterimi f = (a_{n-1}, …., a_0) ile verilen f tamsayısı

Çıktı:

c = a · b mod f

Algoritma:

```
1. b = a
2. if a[0] = 1
       c = b
3. else
       c = 0
   end
4.  for i = 1 : n-1
4.1.   if b[n-1] = 1
           b = b << 1
           b = b + f
4.2.   else
           b = b << 1
       end
4.3.   if a[i] = 1
               c = c + b
       end
   end
5. c değeri döndürülür.
```

Şekil 2.30 Kare alma işlemine ait sözde kod (Hankerson vd. 2003)

Girdi:
İkili gösterimi a = (a_{m-1}, …., a_0) ile verilen a tamsayısı
İkili gösterimi f = (f_{n-1}, …., f_0) ile verilen f tamsayısı
$m \leq n$
Çıktı:
a^{-1} mod f
Algoritma:
1. u = a, v = f
2. $g_1 = 1$, $g_2 = 0$
3. while ($u \neq 1$)
3.1. j = m – n
3.2. if j < 0
u ↔ v, $g_1 \leftrightarrow g_2$, j = -j
end
3.3. u = u + v << j, $g_1 = g_1 + g_2$ << j
end
4. g_1 değeri döndürülür.

Şekil 2.31 Çarpımsal ters alma işleminin sözde kodu (Hankerson vd. 2003)

Şekil 2.31'daki algoritmada yer alan ↔ operatörü ile değişkenlerin içeriği değiş-tokuş edilmektedir. Bir polinomu z^j ile çarpma işlemi bu polinomun j bit sola kaydırılması işlemi ile eşdeğerdir.

2.8 Eliptik Eğri Aritmetiği

Eliptik eğri kripto sistemi için cisim aritmetiğinin yanında seçilen eliptik eğri aritmetiği de önemlidir. Eliptik eğriler ile ilgili tanımlamalar ve eliptik eğri aritmetiği (Hankerson vd. 2003) kitabı birebir takip edilerek anlatılmaktadır ve notasyonda hiçbir değişiklik yapılmamaktadır.

2.8.1 Eliptik eğrilere giriş

Bir K cismi üzerinde tanımlı E eliptik eğrisinin denklemi (2.8) eşitliği ile verilmektedir. Bu eşitlik Weierstrass denklemi adıyla anılmaktadır.

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (2.8)$$

a_1, a_2, a_3, a_4 ve a_6 katsayıları K cisminin bir elemanıdır ve E eliptik eğrisinin diskriminantı 0'dan farklıdır. E eliptik eğrisinin diskriminantı Δ sembolü ile gösterilmektedir. Eşitlik (2.9), E eliptik eğrisinin diskriminantını ifade etmektedir.

$$\begin{aligned} \Delta &= -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2 d_4 d_6 \\ d_2 &= a_1^2 + 4a_2 \\ d_4 &= 2a_4 + a_1 a_3 \\ d_6 &= a_3^2 + 4a_6 \\ d_8 &= a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2 \end{aligned} \quad (2.9)$$

K cismi temel bir cisimdir. K cismine ait bir sonlu genişleme cisminin L olduğu varsayılmaktadır. Bu durumda K üzerinde tanımlı E eliptik eğrisi K'nin sonlu genişleme cismi L'de de tanımlı olmaktadır. Eşitlik (2.10)'da E üzerinde bulunan L rasyonel nokta kümesi belirtilmektedir. E eliptik eğrisi üzerinde bulunan L rasyonel noktaları x ve y, eşitlik (2.8)'deki eliptik eğri denklemini sağlamaktadır. ∞, izdüşümsel koordinatlarda tanımlı Weierstrass denkleminin sonsuzda bulunan tek noktasıdır. Bu sonsuzluk noktası K'nin tüm sonlu genişleme cisimleri için rasyonel bir nokta olarak değerlendirilmektedir.

$$E(L) = \{(x,y) \in L \times L: y^2 + a_1 xy + a_3 y - x^3 - a_2 x^2 - a_4 x - a_6 = 0\} \cup \{\infty\} \quad (2.10)$$

2.8.2 Basitleştirilmiş Weierstrass eşitlikleri

K cismi üzerinde tanımlı E eliptik eğrisinin (2.8)'de yer alan eşitliğinin uygun değişken değiştirme ile basitleştirilmesi mümkün olmaktadır. Basitleştirilmiş Weierstrass eşitlikleri K cisminin karakteristiğine göre farklılık göstermektedir. K cisminin

karakteristiği 2 veya 3 olabilmektedir ya da hem 2'den hem 3'ten farklı bir değer alabilmektedir.

K'nin karakteristiği 2 ve 3'ten farklı olduğunda eşitlik (2.11) ile değişken değiştirme yapılmaktadır. Eşitlik (2.8), (2.12)'deki eşitliğe dönüştürülmektedir. E eliptik eğrisinin denklemi ve bu eliptik eğrinin diskriminantı sırasıyla eşitlik (2.12) ve (2.13)'teki gibidir.

$$(x, y) \rightarrow \left(\frac{x - 3a_1^2 - 12a_2}{36}, \frac{y - 3a_1 x}{216} - \frac{a_1^3 + 4a_1 a_2 - 12a_3}{24}\right) \quad (2.11)$$

$$y^2 = x^3 + ax + b \quad (2.12)$$

$$\Delta = -16\,(4a^3 + 27b^2) \quad (2.13)$$

K'nin karakteristiği 2'ye eşit ise iki durum göz önünde bulundurulmalıdır:

a_1, 0'dan farklı ise o zaman değişken değiştirme eşitliği (2.14) olup (2.8) eşitliği (2.15) eşitliğine dönüştürülmektedir. E eliptik eğrisinin denklemi eşitlik (2.15) ve bu eliptik eğrinin diskriminantı eşitlik (2.16) ile verilmektedir. Böyle bir eğriye süper tekil olmayan (*non-supersingular*) eliptik eğri denilmektedir.

$$(x, y) \rightarrow \left(a_1^2 x + \frac{a_3}{a_1}, a_1^3 y + \frac{a_1^2 a_4 + a_3^2}{a_1^3}\right) \quad (2.14)$$

$$y^2 + xy = x^3 + a x^2 + b \quad (2.15)$$

$$\Delta = b \quad (2.16)$$

a_1'in 0'a eşit olduğu durumda değişken değiştirme eşitliği (2.17) olmaktadır. Denklemi (2.8) olan eliptik eğriden (2.18) olan eliptik eğri elde edilmektedir. Elde edilen eliptik eğrinin diskriminant formülü eşitlik (2.19)'da görülmektedir. Bu eliptik eğri süper tekil (*supersingular*) eliptik eğri olarak adlandırılmaktadır.

$$(x, y) \rightarrow (x + a_2, y) \tag{2.17}$$

$$y^2 + cy = x^3 + a\,x + b \tag{2.18}$$

$$\Delta = c^4 \tag{2.19}$$

K'nin karakteristiği 3 olarak verildiğinde a_1^2'nin $-a_2$ ile eşit olduğu ve olmadığı durumlar için farklı değişken değiştirme eşitlikleri kullanılmaktadır:

$a_1^2 \neq -a_2$ ise eşitlik (2.20) ile değişken değiştirme yapılmaktadır ve (2.8) eşitliği (2.21) eşitliğine dönüşmektedir. Süper tekil olmayan E eliptik eğrisinin yeni eşitliği (2.21), diskriminantı (2.22) ile formülize edilmektedir.

$$(x, y) \rightarrow (x + \frac{d_4}{d_2}, y + a_1 x + a_1 \frac{d_4}{d_2} + a_3)$$

$$d_2 = a_1^2 + a_2 \tag{2.20}$$

$$d_4 = a_4 - a_1\,a_3$$

$$y^2 = x^3 + a\,x^2 + b \tag{2.21}$$

$$\Delta = a^3\,b \tag{2.22}$$

$a_1^2 = -a_2$ ise eşitlik (2.23) yardımıyla değişen değiştirme yapılarak (2.8) eşitliğinden (2.12) eşitliği türetilmektedir. E eliptik eğrisi süper tekil bir eliptik eğridir. Eşitlik (2.24) bu eliptik eğrinin diskriminantını belirtmektedir.

$$(x, y) \rightarrow (x, y + a_1 x + a_2) \tag{2.23}$$

$$\Delta = -a^3 \tag{2.24}$$

2.8.3 Grup kuralları

E, K cisminde tanımlı bir eliptik eğri olmak üzere kiriş-ve-teğet kuralı kullanılarak eliptik eğri üzerinde bulunan iki nokta toplanmakta ve üçüncü bir nokta hesaplanmaktadır. Bu üçüncü nokta da bu eliptik eğrinin bir elemanıdır. Değişmeli bir grup olan ve eliptik eğrilerle şifreleme yöntemlerinde kullanılan E(K) noktalar kümesinin etkisiz elemanı sonsuzdur.

Grup işlemleri iki çeşittir: 1. Nokta Toplama (*Point Addition*), 2. Nokta Aynılama (*Point Doubling*). Birbirinden farklı iki noktanın toplanması nokta toplama işlemidir. Nokta aynılama ise bir noktanın kendisi ile toplanması demektir.

Afin koordinatlarda tanımlı grup kuralları temel sonlu cisim K'nin türüne ve karakteristiğine göre farklılık göstermektedir.

Temel sonlu bir asal cisim K'nin karakteristiği 2 ve 3'ten farklı olduğunda (K = GF(p), p asal ve p> 3) (2.12) eşitliği ile verilen basitleştirilmiş Weierstrass formundaki eliptik eğriler için grup kuralları denklem 2.25, 2.26, 2.27, 2.28, 2.29, 2.30, 2.31, 2.32, 2.33, 2.34 ve 2.35'te yer almaktadır.

(i) Etkisiz eleman:

$$\text{Tüm } P \in E(K) \text{ için } P + \infty = \infty + P = P \tag{2.25}$$

(ii) Toplama işlemine göre ters eleman:

$$P = (x, y) \in E(K) \text{ ise } (x, y) + (x, -y) = \infty \tag{2.26}$$

$$-P = (x, -y), -P \in E(K) \tag{2.27}$$

$$-\infty = \infty \tag{2.28}$$

(iii) Nokta toplama:

$$P = (x_1, y_1) \in E(K), Q = (x_2, y_2) \in E(K), P \neq \pm Q \tag{2.29}$$

$$P + Q = (x_3, y_3) \tag{2.30}$$

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2 \tag{2.31}$$

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right) (x_1 - x_3) - y_1 \quad (2.32)$$

(iv) Nokta aynılama:

$$P = (x_1, y_1) \in E(K),\ P \neq -P,\ 2P = (x_3, y_3) \quad (2.33)$$

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1} \right)^2 - 2x_1 \quad (2.34)$$

$$y_3 = \left(\frac{3x_1^2 + a}{2y_1} \right)(x_1 - x_3) - y_1 \quad (2.35)$$

Temel sonlu bir ikili cisim K = GF(2^n) üzerinde tanımlı süper tekil olmayan eliptik eğri formu için (2.9) basitleştirilmiş Weierstrass eşitliği ile tanımlanan grup kuralları denklem 2.36, 2.37, 2.38, 2.39, 2.40, 2.41, 2.42, 2.43, 2.44, 2.45, 2.46, 2.47 ve 2.48'de verilmektedir.

(i) Etkisiz eleman:

$$\text{Tüm } P \in E(K) \text{ için } P + \infty = \infty + P = P \quad (2.36)$$

(ii) Toplama işlemine göre ters eleman:

$$P = (x, y) \in E(K) \text{ ise } (x, y) + (x, x + y) = \infty \quad (2.37)$$

$$-P = (x, x + y),\ -P \in E(K) \quad (2.38)$$

$$-\infty = \infty \quad (2.39)$$

(iii) Nokta toplama:

$$P = (x_1, y_1) \in E(K),\ Q = (x_2, y_2) \in E(K),\ P \neq \pm Q \quad (2.40)$$

$$P + Q = (x_3, y_3) \quad (2.41)$$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \quad (2.42)$$

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1 \quad (2.43)$$

$$\lambda = \frac{y_1 + y_2}{x_1 + x_2} \quad (2.44)$$

(iv) Nokta aynılama:

$$P = (x_1, y_1) \in E(K),\ P \neq -P,\ 2P = (x_3, y_3) \tag{2.45}$$

$$x_3 = \lambda^2 + \lambda + a = x_1^2 + \frac{b}{x_1^2} \tag{2.46}$$

$$y_3 = x_1^2 + \lambda x_3 + x_3 \tag{2.47}$$

$$\lambda = x_1 + \frac{y_1}{x_1} \tag{2.48}$$

Temel sonlu bir ikili cisim K = GF(2^n) üzerinde tanımlı süper tekil eliptik eğri formu için (2.18) basitleştirilmiş Weierstrass eşitliği ile tanımlanan grup kuralları denklem 2.49, 2.50, 2.51, 2.52, 2.53, 2.54, 2.55, 2.56, 2.57, 2.58 ve 2.59'da sıralanmaktadır.

(i) Etkisiz eleman:

$$\text{Tüm } P \in GF(2^n) \text{ için } P + \infty = \infty + P = P \tag{2.49}$$

(ii) Toplama işlemine göre ters eleman:

$$P = (x, y) \in E(K) \text{ ise } (x, y) + (x, y + c) = \infty \tag{2.50}$$

$$-P = (x, y + c),\ -P \in E(K) \tag{2.51}$$

$$-\infty = \infty \tag{2.52}$$

(iii) Nokta toplama:

$$P = (x_1, y_1) \in E(K),\ Q = (x_2, y_2) \in E(K),\ P \neq \pm Q \tag{2.53}$$

$$P + Q = (x_3, y_3) \tag{2.54}$$

$$x_3 = \left(\frac{y_2 + y_1}{x_2 + x_1}\right)^2 + x_1 + x_2 \tag{2.55}$$

$$y_3 = \left(\frac{y_2 + y_1}{x_2 + x_1}\right)(x_1 + x_3) + y_1 + c \tag{2.56}$$

(iv) Nokta aynılama:

$$P = (x_1, y_1) \in E(K),\ P \neq -P,\ 2P = (x_3, y_3) \tag{2.57}$$

$$x_3 = \left(\frac{x_1^2 + a}{c}\right)^2 - 2x_1 \tag{2.58}$$

$$y_3=\left(\frac{x_1^2+a}{c}\right)(x_1+x_3)+y_1+c \qquad (2.59)$$

2.8.4 Grup mertebesi (*Group Order*)

E'nin, GF($q = p^m$) cisminde tanımlı bir eliptik eğri olduğu bilinmektedir. Grup mertebesi ile eliptik eğri üzerinde bulunan nokta sayısı ifade edilmektedir. #E(GF(q)), E eliptik eğrisinin mertebesini simgelemektedir.

Weierstrass eşitliği, denklem (2.8)'deki gibi olan E eliptik eğrisinin mertebesi 1'den büyük veya 1'e eşit ve 2 □ q + 1'den küçük veya 2 □ q + 1'e eşit olmalıdır. Bu eşitliğin her $x \in GF(q)$ için en fazla iki çözümü mevcuttur. Hasse teoremi (Denklem 2.60) E eliptik eğrisinin mertebesinin aralığının daralmasını sağlamaktadır. $[q+1-2\sqrt{q}, q+1+2\sqrt{q}]$ aralığı Hasse aralığı adıyla anılmaktadır.

$$q + 1 - 2\sqrt{q} \leq \#E(GF(q)) \leq q + 1 + 2\sqrt{q} \qquad (2.60)$$

Hasse teoreminin alternatif bir gösterimi bulunmaktadır (Denklem 2.61). t'ye, eliptik eğrinin izi (*trace*) denilmektedir. $2\sqrt{q}$, q'ya kıyasla oldukça küçük olduğu için E eliptik eğrisinin mertebesi yaklaşık olarak q'ya eşit olmaktadır.

$$\#E(GF(q)) = q + 1 - t \text{ , } |t| \leq 2\sqrt{q} \qquad (2.61)$$

E eliptik eğrisinin mertebesinin denklem (2.61)'e eşit olması için aşağıdaki üç koşuldan birisinin sağlanması gerekmektedir:

i. $t \not\equiv 0 \pmod p$ ve $t^2 \leq 4$□q

ii. n, tek bir sayı ise (a) t = 0 veya

(b) p = 2 ve $t^2 = 2$□q veya

(c) p = 3 ve $t^2 = 3$□q

iii. n, çift bir sayı ise (a) $t^2 = 4 \square q$ ya da

(b) $p \not\equiv 1 \pmod 3$ ve $t^2 = q$ ya da

(c) $t = 0$ ve $p \not\equiv 1 \pmod 4$

2.8.5 Süper tekil (*Supersingular*) ve süper tekil olmayan (*Non-Supersingular*) eliptik eğriler

Karakteristiği p olan GF(q) cismi göz önüne alınmaktadır. p, t'yi bölüyorsa GF(q) üzerinde tanımlı eliptik eğriler süper tekil eliptik eğriler olarak adlandırılmaktadır. Ancak p'nin, t'yi bölmediği durumda GF(q) üzerinde tanımlı eliptik eğriler süper tekil olmayan eliptik eğrilerdir.

2.8.6 Grup yapısı

E, GF(q) üzerinde tanımlı bir eliptik eğridir. n_1 ve n_2 pozitif tamsayılar ve E eliptik eğrisinin mertebesinin $n_1 \cdot n_2$ olduğu bilinmektedir. n_2, 1'e eşit ise E eliptik eğrisi periyodik bir grup oluşturmaktadır. n_2, 1'den büyük olan küçük bir tamsayı ($n_2 = 2$, 3 veya 4) ise E, yaklaşık olarak periyodiktir. GF(q) üzerindeki eliptik eğrilerin çoğunun periyodik ya da yaklaşık olarak periyodik olması gereklidir.

2.8.7 İzomorfizma (*Isomorphism*) sınıfları

İzomorfizma, sonlu bir cisimde tanımlanan eliptik eğriler için bir denklik bağıntısıdır. Karakteristiği 2 ve 3'ten farklı sonlu bir cisim olan K = GF(q)'de E_1 ve E_2 eliptik eğrileri tanımlıdır. Denklem (2.62) ve (2.63), sıralı olarak E_1 ve E_2 eliptik eğrilerinin Weierstrass eşitlikleridir. E_1 ve E_2'nin K cismi üzerinde izomorfik olmaları $u^4\bar{a} = a$ ve $u^6\bar{b} = b$ eşitliklerini sağlayan bir u'nun varlığı ile mümkündür. Denklem (2.63), (2.62)'nin (2.64) ile değişken değiştirme yapılarak dönüştürülmüş halidir.

K = GF(q)'de tanımlı eliptik eğrilerin izomorfizma sınıflarının sayısı $q \equiv 1 \pmod{12}$ için $2 \cdot q + 6$, $q \equiv 5 \pmod{12}$ için $2 \cdot q + 2$, $q \equiv 7 \pmod{12}$ için $2 \cdot q + 4$ ve $q \equiv 11 \pmod{12}$ için $2 \cdot q$' dur.

$$E_1 : y^2 = x^3 + ax + b \quad (2.62)$$

$$E_2 : y^2 = x^3 + \bar{a}x + \bar{b} \quad (2.63)$$

$$(x, y) \rightarrow (u^2x, u^3y) \quad (2.64)$$

Sonlu bir ikili cisim olan K = GF(2^n)'de tanımlı süper tekil olmayan eliptik eğriler E_1 ve E_2 için Weierstrass eşitlikleri E_1 için denklem (2.65) ve E_2 için (2.66)'da görüldüğü gibidir. $\bar{a} = s^2 + s + a$ eşitliğini sağlayan s, GF(2^n) sonlu cisminin bir elemanıdır. Ancak ve ancak (2.67) ve (2.68) eşitlikleri sağlandığı takdirde E_1 ve E_2 eliptik eğrileri K üzerinde izomorfiktir.

GF(2^n) cisminde iz fonksiyonu Tr: GF(2^n) $\rightarrow$ GF(2^n), $Tr(a) = a + a^2 + a^{2^2} + ... + a^{2^{n-1}}$ eşitliği ile tanımlanmaktadır. Denklem (2.69)'daki değişken değiştirme eşitliği yardımıyla (2.65) eşitliğinden (2.66) eşitliği elde edilmektedir.

$$E_1 : y^2 + xy = x^3 + ax^2 + b \quad (2.65)$$

$$E_2 : y^2 + xy = x^3 + \bar{a}x^2 + \bar{b} \quad (2.66)$$

$$b = \bar{b} \quad (2.67)$$

$$Tr(a) = Tr(\bar{a}) \quad (2.68)$$

$$(x, y) \rightarrow (x, y + sx) \quad (2.69)$$

K = GF(2^n)'de tanımlı süper tekil olmayan izomorfik eliptik eğri sınıfları $2^{n+1} - 2$ adettir.

$\gamma \in$ GF(2^n) için $Tr(\gamma) = 1$ olmaktadır. İzomorfizma sınıfı için temsilciler kümesi denklem (2.70)'te belirtilmektedir.

$$\{ y^2 + xy = x^3 + ax^2 + b \mid a \in \{0, \gamma\}, b \in GF^*(2^n) \} \quad (2.70)$$

Weierstrass eşitliği denklem (2.71) ile verilen süper tekil olmayan eliptik bir eğrinin mertebesi 2'nin katıdır. $\gamma \in GF(2^n)$ için $Tr(\gamma) = 0$ olduğu zaman eliptik eğrinin mertebesi 4 ile bölünebilmektedir.

$$E : y^2 + xy = x^3 + \gamma x^2 + b \tag{2.71}$$

2.8.8 İzdüşümsel (*Projective*) koordinatlar

Asal ve ikili sonlu cisimler üzerinde tanımlı eliptik eğrilerde nokta toplama ve nokta aynılama işlemlerinde bir kez ters alma işlemi ve birçok kez aritmetik çarpma işlemi yapılmaktadır. Ters alma işleminin maliyeti çarpma işlemine nazaran oldukça pahalıdır. Bu nedenle izdüşümsel koordinatları kullanarak noktaları göstermek avantajlı olmaktadır.

Bir noktanın izdüşümsel koordinatları X, Y ve Z bileşenlerinden oluşmaktadır. İzdüşümsel bir nokta (X : Y : Z) şeklinde gösterilmektedir. P(K), temel sonlu bir K cisminin tüm izdüşümsel noktalar kümesidir. Temel K cisminde tanımlanan, Weierstrass denklemi (2.8)'de görülen eliptik bir eğrinin izdüşümsel formunu elde etmek için x yerine $(\frac{X}{Z^c})$, y yerine $(\frac{Y}{Z^d})$ konulmaktadır ve payda eşitlemesi yapılmaktadır. c ve d pozitif tamsayılardır.

Karakteristiği 2 ve 3 olmayan temel sonlu asal K = GF(q) cisminin bir elemanı olan, denklemi (2.12)'de verilmiş bir eliptik eğri için standart ve Jacobian izdüşümsel koordinatlar önerilmektedir:

I. Standart İzdüşümsel Koordinatlar

   c ve d pozitif tamsayılarının değeri 1'dir. Z sıfırdan farklı olmak üzere denklem (2.12)'de x yerine $(\frac{X}{Z})$, y yerine $(\frac{Y}{Z})$ yazılarak payda eşitlendiğinde yeni

denklem (2.72) bulunmaktadır. Sonsuzluk noktası, ∞, (0 : 1 : 0) noktasına eşittir. (X : Y : Z) izdüşümsel noktasının toplamaya göre tersi (X : -Y : Z) noktasıdır.

$$Y^2Z = X^3 + a\,XZ^2 + b\,Z^3 \tag{2.72}$$

II. Jacobian İzdüşümsel Koordinatlar

c, 2'ye ve d, 3'e eşittir. Sıfıra eşit olmayan bir Z için denklem (2.12)'de x yerine $(\frac{X}{Z^2})$, y yerine $(\frac{Y}{Z^3})$ yazılarak payda eşitlendiğinde yeni denklem (2.73) elde edilmektedir. Sonsuzluk noktası (1 : 1 : 0) noktasına eşittir. (X : Y : Z) izdüşümsel noktasının toplamaya göre tersi (X : -Y : Z) noktasına eşit olmaktadır.

$$Y^2 = X^3 + a\,XZ^4 + b\,Z^6 \tag{2.73}$$

İzdüşümsel koordinatlarda nokta toplama ve nokta aynılama işlemlerine ait sözde (*pseudo*) kod sıralı olarak şekil 2.32 ve 2.33'te görülmektedir.

Girdiler :

$P = (X_1 : Y_1 : Z_1)$

$Q = (X_2 : Y_2 : Z_2)$

$E : Y^2Z = X^3 + aXZ^2 + bZ^3$

$b_3 = 3 \cdot b$

t_0, t_1, t_2, t_3, t_4 ve t_5 geçici değişkenler

Çıktı :

$P + Q = (X_3 : Y_3 : Z_3)$

Algoritma Adımları

1. $t_0 = X_1 \cdot X_2$	11. $t_4 = t_4 \cdot t_5$	21. $Z_3 = X_3 + Z_3$	31. $t_2 = a \cdot t_2$
2. $t_1 = Y_1 \cdot Y_2$	12. $t_5 = t_0 + t_2$	22. $X_3 = t_1 - Z_3$	32. $t_4 = t_4 + t_2$
3. $t_2 = Z_1 \cdot Z_2$	13. $t_4 = t_4 - t_5$	23. $Z_3 = t_1 + Z_3$	33. $t_0 = t_1 \cdot t_4$
4. $t_3 = X_1 + Y_1$	14. $t_5 = Y_1 + Z_1$	24. $Y_3 = X_3 \cdot Z_3$	34. $Y_3 = Y_3 + t_0$
5. $t_4 = X_2 + Y_2$	15. $X_3 = Y_2 + Z_2$	25. $t_1 = t_0 + t_0$	35. $t_0 = t_5 \cdot t_4$
6. $t_3 = t_3 \cdot t_4$	16. $t_5 = t_5 \cdot X_3$	26. $t_1 = t_0 + t_1$	36. $X_3 = t_3 \cdot X_3$
7. $t_4 = t_0 + t_1$	17. $X_3 = t_1 + t_2$	27. $t_2 = a \cdot t_2$	37. $X_3 = X_3 - t_0$
8. $t_3 = t_3 - t_4$	18. $t_5 = t_5 - X_3$	28. $t_4 = b_3 \cdot t_4$	38. $t_0 = t_3 \cdot t_1$
9. $t_4 = X_1 + Z_1$	19. $Z_3 = a \cdot t_4$	29. $t_1 = t_1 + t_2$	39. $Z_3 = t_5 \cdot Z_3$
10. $t_5 = X_2 + Z_2$	20. $X_3 = b_3 \cdot t_2$	30. $t_2 = t_0 - t_2$	40. $Z_3 = Z_3 + t_0$

Şekil 2.32 $E : y^2 = x^3 + ax + b$ nokta toplama sözde kodu (Renes vd. 2016)

Girdiler :

$P = (X : Y : Z)$

$E : Y^2Z = X^3 + aXZ^2 + bZ^3$

$b_3 = 3 \cdot b$

t_0, t_1, t_2 ve t_3 geçici değişkenler

Çıktı :

$2P = (X_3 : Y_3 : Z_3)$

Algoritma Adımları

1. $t_0 = X \cdot X$	11. $X_3 = t_1 - Y_3$	21. $t_0 = Z_3 + t_0$	31. $Z_3 = Z_3 + Z_3$
2. $t_1 = Y \cdot Y$	12. $Y_3 = t_1 + Y_3$	22. $t_0 = t_0 + t_2$	
3. $t_2 = Z \cdot Z$	13. $Y_3 = X_3 \cdot Y_3$	23. $t_0 = t_0 \cdot t_3$	
4. $t_3 = X \cdot Y$	14. $X_3 = t_3 \cdot X_3$	24. $Y_3 = Y_3 + t_0$	
5. $t_3 = t_3 + t_3$	15. $Z_3 = b_3 \cdot Z_3$	25. $t_2 = Y \cdot Z$	
6. $Z_3 = X \cdot Z$	16. $t_2 = a \cdot t_2$	26. $t_2 = t_2 + t_2$	
7. $Z_3 = Z_3 + Z_3$	17. $t_3 = t_0 - t_2$	27. $t_0 = t_2 \cdot t_3$	
8. $X_3 = a \cdot Z_3$	18. $t_3 = a \cdot t_3$	28. $X_3 = X_3 - t_0$	
9. $Y_3 = b_3 \cdot t_2$	19. $t_3 = t_3 + Z_3$	29. $Z_3 = t_2 \cdot t_1$	
10. $Y_3 = X_3 + Y_3$	20. $Z_3 = t_0 + t_0$	30. $Z_3 = Z_3 + Z_3$	

Şekil 2.33 $E : y^2 = x^3 + ax + b$ nokta aynılama sözde kodu (Renes vd. 2016)

Denklem (2.12)'de a katsayısının -3'e eşit olması özel bir durumdur. Denklemi $E: y^2 \equiv x^3 - 3x + b \pmod p$ şeklinde olan eliptik eğrilere kısa Weierstrass eliptik eğrileri denilmektedir. a = -3 olduğu durumlar için nokta toplama ve nokta aynılama algoritmalarının izdüşümsel koordinatlardaki sözde (*pseudo*) kodları şekil 2.34 ve şekil 2.35'teki gibidir.

Girdiler :

$P = (X_1 : Y_1 : 1)$

$Q = (0 : 1 : 0)$

$E : Y^2Z = X^3 + aXZ^2 + bZ^3$

$a = -3$

t_0, t_1, t_2, t_3 ve t_4 geçici değişkenler

Çıktı :

$P + Q = (X_3 : Y_3 : Z_3)$

Algoritma Adımları

1. $t_0 = X_1 \cdot X_2$	12. $X_3 = t_1 + t_2$	23. $Z_3 = t_1 - X_3$	34. $t_0 = t_0 - t_2$
2. $t_1 = Y_1 \cdot Y_2$	13. $t_4 = t_4 - X_3$	24. $X_3 = t_1 + X_3$	35. $t_1 = t_4 \cdot Y_3$
3. $t_2 = Z_1 \cdot Z_2$	14. $X_3 = X_1 + Z_1$	25. $Y_3 = b \cdot Y_3$	36. $t_2 = t_0 \cdot Y_3$
4. $t_3 = X_1 + Y_1$	15. $Y_3 = X_2 + Z_2$	26. $t_1 = t_2 + t_2$	37. $Y_3 = X_3 \cdot Z_3$
5. $t_4 = X_2 + Y_2$	16. $X_3 = X_3 \cdot Y_3$	27. $t_2 = t_1 + t_2$	38. $Y_3 = Y_3 + t_2$
6. $t_3 = t_3 \cdot t_4$	17. $Y_3 = t_0 + t_2$	28. $Y_3 = Y_3 - t_2$	39. $X_3 = t_3 \cdot X_3$
7. $t_4 = t_0 + t_1$	18. $Y_3 = X_3 - Y_3$	29. $Y_3 = Y_3 - t_0$	40. $X_3 = X_3 - t_1$
8. $t_3 = t_3 - t_4$	19. $Z_3 = b \cdot t_2$	30. $t_1 = Y_3 + Y_3$	41. $Z_3 = t_4 \cdot Z_3$
9. $t_4 = Y_1 + Z_1$	20. $X_3 = Y_3 - Z_3$	31. $Y_3 = t_1 + Y_3$	42. $t_1 = t_3 \cdot t_0$
10. $X_3 = Y_2 + Z_2$	21. $Z_3 = X_3 + X_3$	32. $t_1 = t_0 + t_0$	43. $Z_3 = Z_3 + t_1$
11. $t_4 = t_4 \cdot X_3$	22. $X_3 = X_3 + Z_3$	33. $t_0 = t_1 + t_0$	

Şekil 2.34 $E : y^2 = x^3 - 3x + b$ nokta toplama sözde kodu (Renes vd. 2016)

Girdiler :

P = (X : Y : Z)

E : $Y^2Z = X^3 + aXZ^2 + bZ^3$

a = -3

t_0, t_1, t_2 ve t_3 geçici değişkenler

Çıktı :

2P = (X_3 : Y_3 : Z_3)

Algoritma Adımları

1. $t_0 = X \cdot X$	11. $Y_3 = X_3 + Y_3$	21. $t_3 = Z_3 + Z_3$	31. $X_3 = X_3 - Z_3$
2. $t_1 = Y \cdot Y$	12. $X_3 = t_1 - Y_3$	22. $Z_3 = Z_3 + t_3$	32. $Z_3 = t_0 \cdot t_1$
3. $t_2 = Z \cdot Z$	13. $Y_3 = t_1 + Y_3$	23. $t_3 = t_0 + t_0$	33. $Z_3 = Z_3 + Z_3$
4. $t_3 = X \cdot Y$	14. $Y_3 = X_3 \cdot Y_3$	24. $t_0 = t_3 + t_0$	34. $Z_3 = Z_3 + Z_3$
5. $t_3 = t_3 + t_3$	15. $X_3 = X_3 \cdot t_3$	25. $t_0 = t_0 - t_2$	
6. $Z_3 = X \cdot Z$	16. $t_3 = t_2 + t_2$	26. $t_0 = t_0 \cdot Z_3$	
7. $Z_3 = Z_3 + Z_3$	17. $t_2 = t_2 + t_3$	27. $Y_3 = Y_3 + t_0$	
8. $Y_3 = b \cdot t_2$	18. $Z_3 = b \cdot Z_3$	28. $t_0 = Y \cdot Z$	
9. $Y_3 = Y_3 - Z_3$	19. $Z_3 = Z_3 - t_2$	29. $t_0 = t_0 + t_0$	
10. $X_3 = Y_3 + Y_3$	20. $Z_3 = Z_3 - t_0$	30. $Z_3 = t_0 \cdot Z_3$	

Şekil 2.35 E : $y^2 = x^3 - 3x + b$ nokta aynılama sözde kodu (Renes vd. 2016)

Sonlu ikili bir K = GF(2^n) cisminde tanımlı, denklemi (2.15)'te belirtilmiş eliptik bir eğri için üç adet izdüşümsel koordinat sistemi önerilmektedir:

I. Standart İzdüşümsel Koordinatlar

c ve d tamsayıları 1'e eşittir. $Z \neq 0$ için denklem (2.15)'te x yerine $(\frac{X}{Z})$, y yerine $(\frac{Y}{Z})$ yazılmakta, paydalar eşitlenmekte ve denklem (2.74) elde edilmektedir. Sonsuzluk noktası (0 : 1 : 0) noktası olmaktadır. (X : Y : Z) izdüşümsel noktasının toplamaya göre tersi (X : X + Y : Z) noktasına karşılık gelmektedir.

$$Y^2Z + XYZ = X^3 + a\,X^2Z + b\,Z^3 \qquad (2.74)$$

II. Jacobian İzdüşümsel Koordinatlar

c = 2 ve d = 3'tür. $Z \neq 0$ olmak üzere denklem (2.15)'te x yerine $(\frac{X}{Z^2})$, y yerine $(\frac{Y}{Z^3})$ yazılmakta, paydalar eşitlenmekte ve yeni denklem (2.75) olmaktadır. Sonsuzluk noktası (1 : 1 : 0), (X : Y : Z) izdüşümsel noktasının toplamaya göre tersi (X : X + Y : Z) noktasıdır.

$$Y^2 + XYZ = X^3 + a X^2 Z^2 + b Z^6 \quad (2.75)$$

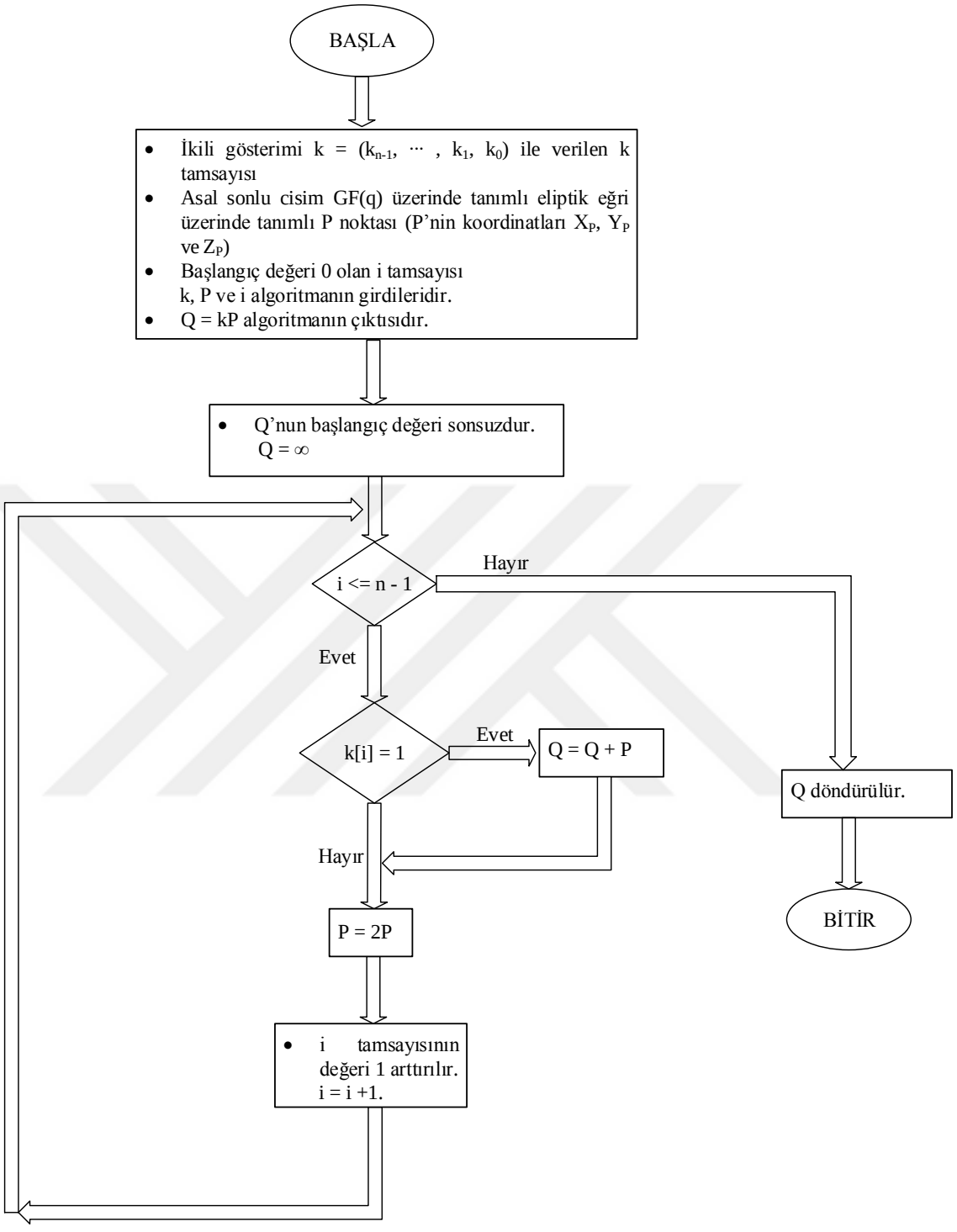
III. Lopez-Dahab İzdüşümsel Koordinatlar

c, 1'e ve d, 2'ye eşittir. $Z \neq 0$ olmak üzere denklem (2.15)'te x yerine $(\frac{X}{Z})$ ve y yerine $(\frac{Y}{Z^2})$ yazılmaktadır. Paydalar eşitlendikten sonra nihai denklem (2.76)'daki gibidir. (1 : 0 : 0), sonsuzluk noktasıdır. (X : Y : Z) noktasının toplama işlemine göre tersi (X : X + Y : Z) noktası olmaktadır.

$$Y^2 + XYZ = X^3 + a X^2 Z^2 + b Z^4 \quad (2.76)$$

2.8.9 Nokta çarpım algoritması

E eliptik eğrisi GF(q) sonlu asal cisminde tanımlanmış ve bu eliptik eğri üzerinde bir P noktası bulunmaktadır. k tamsayısı ile P noktasının çarpılmasına nokta çarpım veya skaler çarpım adı verilmektedir. kP ile k nokta çarpım P işlemi sembolleştirilmektedir. Nokta çarpım için şekil 2.36'da yer alan sağdan sola doğru ikiyle çarp ve topla algoritmasından faydalanılmaktadır. Nokta çarpma algoritması nokta toplama ve nokta aynılama işlemlerini içermektedir.



Şekil 2.36 Sağdan sola doğru ikiyle çarp ve topla algoritması (Hankerson vd. 2003)

Eliptik eğrilerin şifrelemede kullanılmaya başlanmasından bu yana eliptik eğrilerle şifreleme üzerinde pek çok çalışma yapılmıştır. Bu bölümde eliptik eğrilerle şifrelemenin kullanıldığı bir takım uygulamalara yer verilmektedir.

Weimerskirch vd. (2001) avuç içi işletim sistemi ile çalışan cihazlara sonlu ikili cisimlerde tanımlı ve anahtar uzunluğu 163 bit olan eliptik eğrilerin kullanıldığı şifreleme yöntemini uygulayarak güvenlik seviyesinin yeterli düzeyde olup olmadığını incelemiştir.

Chang vd. (2002) OpenSSL0.9.6.b programında Ultra-80 ve Yopy sunucularında RSA şifreleme yöntemini, eliptik eğrilerin kullanıldığı sayısal imza ve Diffie-Hellman anahtar değişimi algoritmalarını çalıştırmış ve her bir algoritmanın işlem yapma hızlarını hesaplamıştır.

Qiu ve Xiang (2004) tarafından eliptik eğrilerle şifreleme yöntemleri kullanılan anahtar boyutu, bilgisayar sayısı, şifrelerin kırılma süresi ve hafıza gereksinimi bakımından RSA kripto sistemleri ile karşılaştırılmış ve eliptik eğri şifreleme sistemlerinin avantajları tespit edilmiştir.

Bozkurt (2005) Güvenli Soket Katmanı Tokalaşma Protokolü'nü eliptik eğrilerle şifreleme tekniklerini kullanarak gerçekleştirmiş ve bu uygulamayı performans açısından değerlendirmiştir.

Çetin (2006) temel eliptik eğri kriptografisinin bir simülasyonunu geliştirmek üzerine çalışmalar yapmıştır.

Değirmenci (2006) Eliptik Eğri Sayısal İmza algoritmasını yazılım ortamında uygulamıştır. Geliştirme ortamı Visual C++ 6.0 versiyonudur. Uygulama Microsoft Windows XP işletim sistemi üzerinde, Visual C++ 6.0 geliştirme ortamında, C++ programlama dili kullanılarak yapılmıştır.

Özcan (2006) tarafından eliptik eğri nokta çarpma metodları yazılım ortamında uygulanmış ve uygulamanın performans değerlendirmesi yapılmıştır.

Kumar ve Paar (2006) ikili cisimlerde tanımlı standartlara uygun eliptik eğrilerin kullanıldığı yüksek alan optimizasyonu sağlayan eliptik eğri şifreleme işlemcisi

tasarlamıştır. Bu tasarımda ters almanın verimli bir şekilde hesaplanabilmesi için hızlı kare alma uygulaması kullanılmaktadır. Bit sayısı 113-193 arasında değişen eliptik eğriler için Montgomery nokta çarpma yöntemi kullanılarak afin koordinatlarda eliptik eğri işlemcisinin ASIC uygulaması önerilmektedir. 0,35 μm CMOS teknolojisi ve 10k ile 18k arasında kapı alanı kullanımı kısıtlı kapasitedeki cihazlarda eliptik eğri şifreleme sistemi tasarımını çekici hale getirmektedir.

RFID etiketleri, ek fonksiyonlara sahip yeni nesil barkod sistemleridir. Batina vd. (2007) tarafından ortaya konulan uygulama gömülü bir ortamda geliştirilmiş ve sahteciliği önlemek amacıyla RFID etiketlerden faydalanmıştır. Bu kısımda çalışmada sahteciliği önlemede hangi açık anahtar tabanlı kimlik doğrulama protokolünün faydalı olduğu araştırılmaktadır. Ayrıca eliptik eğri kripto sistemleri kullanılarak tanımlama yapılıp yapılamayacağı tartışılmakta ve eliptik eğri algoritmalarının RFID etiketlerinde uygulanabilirliği ortaya konmaktadır. Farklı uygulama türleri karşılaştırılmakta ve yan kanal saldırılarına karşı önlemlerin maliyeti araştırılmaktadır.

Bock vd. (2008) ikili cisimlerde tanımlı eliptik eğrilerin kullanıldığı, açık anahtar kripto sistemini kullanan entegre bir kimlik doğrulama modülüne sahip bir RFID etiketin donanım uygulamasını önermiştir.

Braun vd. (2008) hafif kriptografik cihazlarda asimetrik tekniklerin gerçekleştirilmesine yönelik bir konsept üzerinde çalışmışlardır ve RFID etiketlerinin toplu olarak kullanıldığı uygulamalarda kimlik doğrulaması yapmayı sağlayan eliptik eğri kriptografisini temel alan bir uygulama tasarlamıştır.

Puttman vd. (2008) eliptik eğri şifreleme sistemleri için donanım hızlandırıcısı tasarlamıştır.

Yavuz (2008) sonlu asal cisimlerde tanımlı eliptik eğri kripto sistemlerini FPGA ile gerçeklemiştir.

Sarıtaş (2010) eliptik eğri şifreleme yöntemi yardımıyla iki konsol ekran arasında mesajlaşmaya olanak sağlayan bir uygulama geliştirmiştir.

Kurt (2012) Menezes Vanstone eliptik eğri şifreleme algoritmasını bazı özellikler ekleyerek geliştirmiş ve bu algoritmayı yazılım ortamında modellemiştir.

Agarwal vd. (2014) tek ve çoklu işlemci mimarisine sahip FPGA tabanlı gömülü sistemlerde kullanılan eliptik eğri şifreleme tekniklerinin performansını analiz etmiştir.

Kodali vd. (2015) tarafından eliptik eğri şifreleme sistemlerinde çarpma işleminin hızlı yapılabilmesi ve düşük güç tüketmesi amacıyla karma çarpıcı yapısı ileri sürülmüştür.

Renes vd. (2016) karakteristiği 2 ve 3 olmayan temel K cismi üzerindeki kısa Weierstrass eğrilerinde nokta toplama ve nokta aynılama işlemlerini daha fazla cisim toplaması, daha az cisim çarpması yaparak iyileştirmiştir.

Singh vd. (2016) hız ve hafıza bakımından farklı özelliklere sahip üç farklı test makinesinde eliptik eğrilerin kullanıldığı şifreleme sistemlerinin anahtar üretme, şifreleme, şifre çözme, imza üretme ve doğrulama aşamalarını yaptırmış ve bu işlemlerin gerçekleşme sürelerini hesap etmiştir.

3. MATERYAL ve YÖNTEM

3.1 Materyal

Bu çalışmada CryptoGPS, ELLI, ALIKE ve Kimlik Tabanlı İmza (*Identity Based Signature*) yöntemlerinin yazılım yoluyla hayata geçirilmesi, başarım analizinin yapılması ve bu mekanizmaların hafif sıklet olma durumunun kaynak kısıtları ile ilişkilendirilerek değerlendirilmesi amaç edinilmiştir. Bu bölümde konu edilen ISO/IEC 29192 standardında tanımlı olan bu dört algoritmanın üçünde eliptik eğriler kullanılırken birisi RSA asimetrik şifreleme şemasını temel almaktadır (Anonymous 2013).

3.2 Yöntem

ISO/IEC 29192 standardında asimetrik hafif sıklet doğrulama mekanizmaları ELLI pasif RFID etiketlerinde, cryptoGPS işlem yükü kısıtlı cihazlarda, kimlik tabanlı imza yöntemi kablosuz sensör ağlarında ve ALIKE temassız işlemlerde kullanılmak üzere geliştirilmiş algoritmalardır. Bu kısımda asimetrik cryptoGPS, kimlik tabanlı imza, ALIKE ve ELLI protokollerine yer verilmektedir (Anonymous 2013).

3.2.1 CryptoGPS

Eliptik eğrilerde ayrık logaritma problemine dayanan tek taraflı bir doğrulama yöntemi olan cryptoGPS Girault, Poupard ve Stern tarafından tasarlanmıştır. Sınırlı depolama yeteneği ve oldukça düşük işlem maliyeti nedeniyle hafif sıklet olma özelliğine sahiptir. Bu yöntemde gönderici ve alıcı taraftan sadece birisi doğrulanmaktadır. Gönderici ve alıcı taraflar sırayla iddia sahibi (*claimant*) ve doğrulayıcı (*verifier*) şeklinde ifade edilmektedir. Doğrulayıcı tarafından iddia sahibinin iddia edilen, herkesin bildiği temel bir noktanın ayrık logaritmasını içeren eliptik eğriyi bilip bilmediği doğrulanmaktadır (Anonymous 2013).

3.2.1.1 Kullanılan parametreler

Açık Anahtar (*Public Key*): İddia sahibinin kimliğinin doğrulanmasında kullanılan ve herkes tarafından bilinen veridir.

Kişisel Anahtar (*Private Key*): Yalnızca iddia sahibi tarafından bilinen, gizli tutulan veridir.

Tanık Sayı (*Witness*) : İddia sahibinin kimliğini doğrulamasını sağlayan bir parametredir. W ile gösterilmektedir.

Talep (*Challenge*) : Bir yanıt üretmek için gizli parametrelerle birlikte kullanılmaktadır ve d ile temsil edilmektedir.

Jeton (*Taken*) : Bir şifreleme tekniği kullanılarak üretilmiş bilgiler içeren mesajlardır.

Yanıt (*Response*) : İddia sahibi tarafından üretilen ve doğrulayıcı tarafından iddia sahibini doğrulamak amacıyla kullanılan bir parametredir. D sembolü kullanılarak ifade edilmektedir.

Kupon (*Coupon*) : Yalnızca bir kez kullanılacak olan önceden hesap edilmiş sayı çiftidir.

δ : Talebi temsil eden rasgele bit dizisinin uzunluğudur.
β : Talebin alabileceği en büyük değerin bit sayısıdır.
ρ : İddia sahibi tarafından üretilen rasgele bit dizisinin uzunluğudur.
r : İddia sahibi tarafından üretilen rasgele bit dizisidir.
σ : İddia sahibine ait kişisel anahtarın bit sayısıdır.
h(x) : x mesajının özet fonksiyonudur (Anonymous 2013).

3.2.1.2 Güvenlik gereksinimleri

Çalışma alanı parametreleri mekanizmanın işleyişini düzenleyecek şekilde seçilmektedir. Bu parametrelere güvenilir bir şekilde erişilmektedir.

Her iddia sahibi aynı eliptik eğriye (E) ve bu eliptik eğri üzerinde tanımlı bir P noktasına sahiptir. Bu eliptik eğrinin tanımlandığı cismin boyutu q ve P noktasının mertebesi n olmaktadır. Kullanılan eliptik eğri ve bu parametreler kümesi çalışma alanı parametresi ya da iddia sahibi parametresi olarak anılmaktadır.

Her P noktası, eliptik eğri ayrık logaritma problemini tanımlamak için temel bir noktadır. Eğri üzerinde tanımlı P ile rasgele bir J noktasının eşitliği J = [k]P olmak üzere [0, n-1] aralığında tanımlı bir k noktasının sayısal değerini bulmak imkânsızdır.

İddia sahibinin kişisel bir anahtarı bulunmaktadır.

Doğrulayıcı, iddia sahibinin kişisel anahtarına karşılık gelen açık anahtarının doğrulanmış bir kopyasına sahiptir.

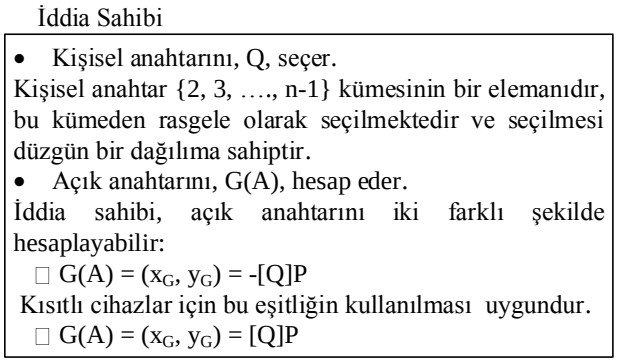
Doğrulayıcı, yeni rasgele bit dizileri üretebilen bir takım ekipmanlara sahiptir. Tek kullanımlık sayı çiftlerinin kullanılamadığı durumda iddia sahibi yeni rasgele bit dizileri üretebilmektedir.

Mekanizmada bir özet fonksiyonu kullanılması gerekiyor ise o zaman ortak bir özet fonksiyonu üzerinde anlaşılmaktadır (Anonymous 2013).

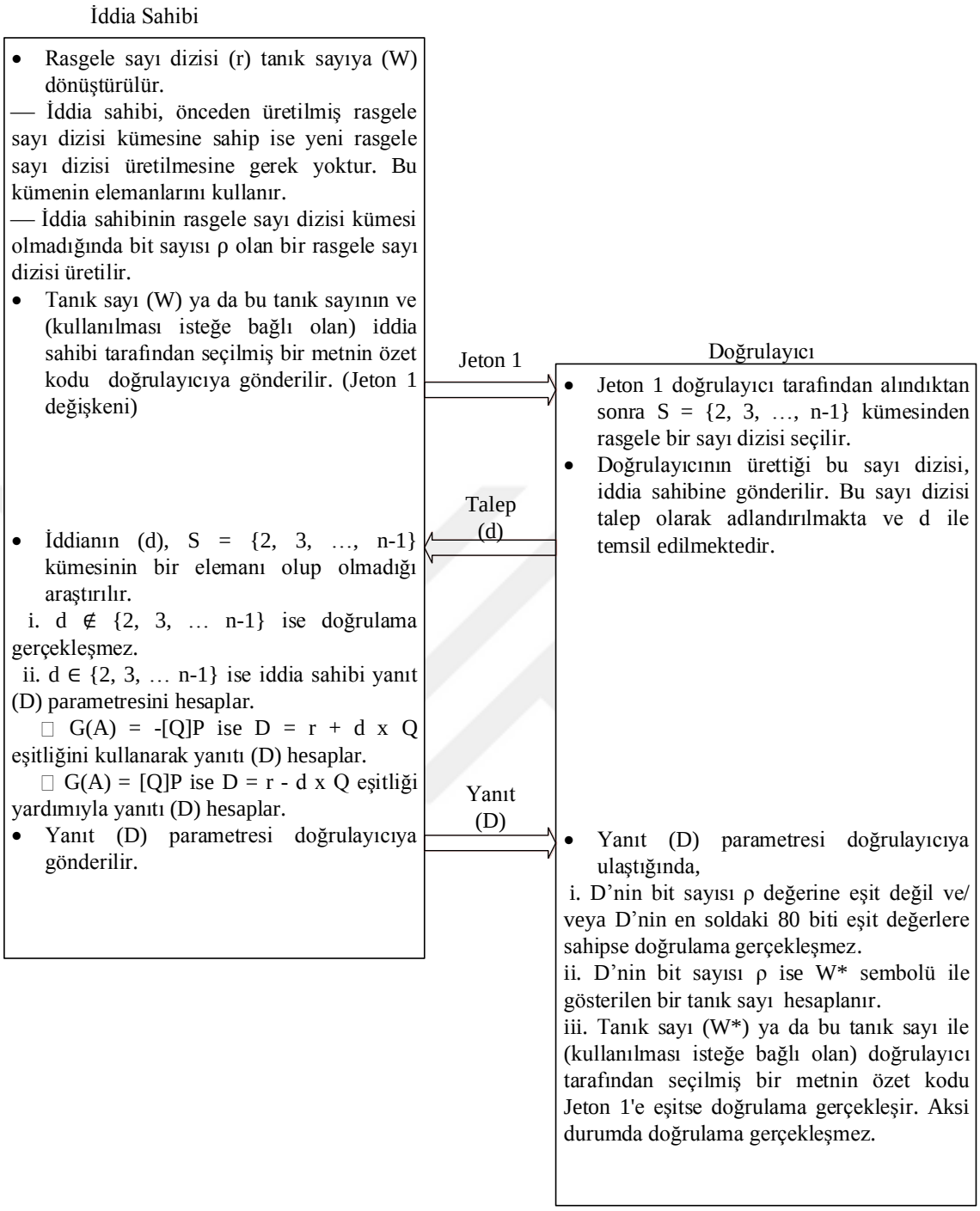
3.2.1.3 CryptoGPS yönteminin adımları

CryptoGPS, anahtar üretme ve tek taraflı doğrulama aşamalarından oluşan bir doğrulama yöntemidir. Anahtar üretme (Şekil 3.1) iddia sahibinin açık anahtarını

hesapladığı adımdır. Tek taraflı doğrulamada (Şekil 3.2) doğrulayıcı tarafından iddia sahibinin doğrulanıp doğrulanmadığına karar verilmektedir (Anonymous 2013).



Şekil 3.1 CryptoGPS için anahtar üretme (Anonymous 2013)

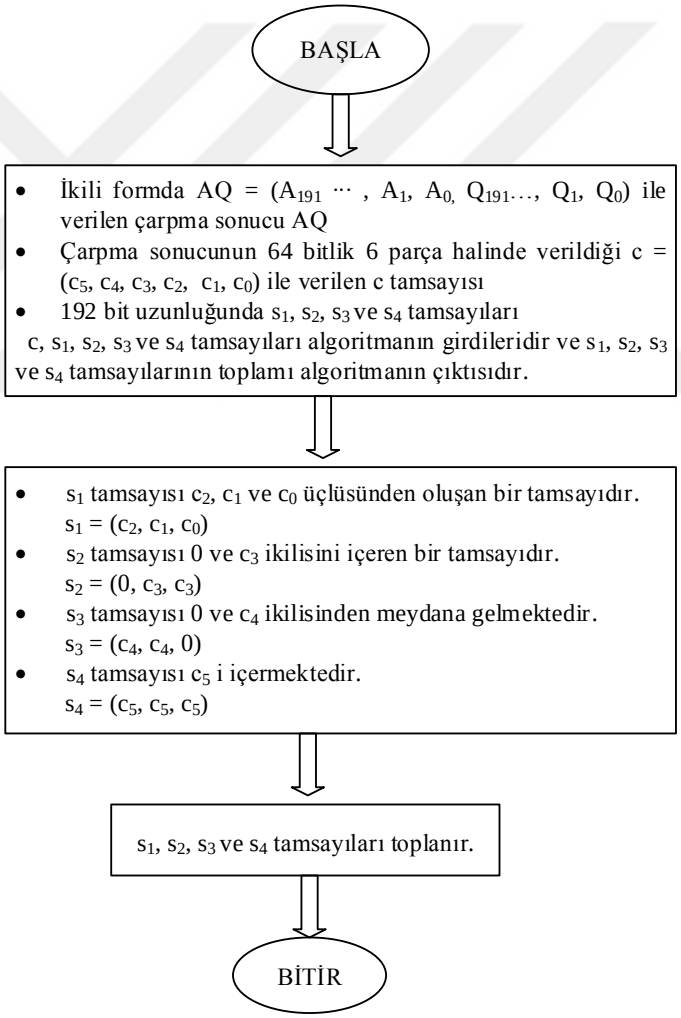


Şekil 3.2 CryptoGPS için doğrulama (Anonymous 2013)

Bu yöntemde mertebesi 192 bit olan sonlu bir asal cisimde tanımlanmış bir eliptik eğri kullanılmaktadır ve bu eliptik eğrinin kısa Weierstrass eşitliği denklem 2.12'deki gibi olmaktadır. İddia sahibinin kişisel anahtarından nokta çarpım algoritması (Şekil 2.36) yardımıyla açık anahtarı elde edilmektedir. Nokta çarpım nokta toplama (Şekil 2.34) ve

nokta aynılama (Şekil 2.35) işlemlerini kapsamaktadır. Hem nokta toplama hem nokta aynılamada toplama (Şekil 2.23), çıkarma (Şekil 2.24), çarpma (Şekil 2.25), çarpımsal ters alma (Şekil 2.26) ile bölme (Şekil 2.27) işlemleri yapılmaktadır (Anonymous 2013).

192 bitlik iki sayı çarpımının maksimum 384 bit uzunluğunda olması beklenmektedir. Bellek tüketiminden tasarruf etmek ve çarpma sonucunun en fazla 192 bit uzunluklu olmasını sağlamak için indirgeme yapılması gerekmektedir. $P_{192} = 2^{192} - 2^{64} - 1$ asal sayısına göre hızlı indirgeme algoritması adımları şekil 3.3'te açıklanmaktadır.



Şekil 3.3 P-192 için hızlı modüler indirgeme algoritması (Hankerson vd. 2003)

3.2.2 Kimlik tabanlı imza yöntemi

Kimlik tabanlı imza yönteminin geliştiricileri Bellare, Namprempe ve Neven olup Schnorr'un kimlik doğrulama şeması temel alınarak imza doğrulaması yapılmaktadır. İmza sahibi ile doğrulayıcının önceden iletişime geçmesi gerekmemektedir. Kimlik tabanlı imza yönteminin hafif sıklet olarak değerlendirilmesinin sebebi kısıtlı kapasite özellikli cihazlarda etkili sonuçlar ortaya koymasıdır (Liu vd. 2010, Anonymous 2013).

3.2.2.1 Kullanılan parametreler

Kimlik tabanlı imza yönteminde bir açık anahtar ve kişisel anahtarın yanı sıra aşağıda sıralanan parametreler kullanılmaktadır:

İmzalama Anahtarı (*Signing Key*): Sadece imza sahibinin kullanmasının mümkün olduğu gizli bilgilerdir.

İmza (*Sign*): İmzalama anahtarı ile bir mesajın imzasının oluşturulması işlemidir.

ID: İkili formatta bir dizi olmakla birlikte kimlik veya doğrulama bilgisini temsil etmektedir.

h: Özet fonksiyonudur (Anonymous 2013).

3.2.2.2 Güvenlik gereksinimleri

Çalışma alanı parametreleri mekanizmanın işleyişini düzenlemektedir. Çalışma alanındaki tüm bileşenlerin seçilen parametrelere güvenilir bir şekilde erişmesi mümkün olmaktadır.

Çalışma alanında bir E eliptik eğrisi, bu eliptik eğri üzerinde tanımlı temel bir P noktası yer almaktadır. Bu P noktasının mertebesi n ile simgelenmektedir.

Eliptik eğri ayrık logaritma problemini tanımlayan temel bir nokta P, eğri üzerinde tanımlı rasgele bir J için J = [k]P eşitliğinde [0, n-1] aralığında tanımlı bir k noktasının sayısal değerini bulmak olanaksızdır.

İmza sahibi ve sunucunun rasgele bit dizileri üreten bir takım ekipmanları vardır.

Çalışma alanında tanımlı tüm bileşenlerin ortak kullandığı özet fonksiyonu saldırılara karşı dayanıklıdır.

Sunucu tarafından rasgele seçilmiş sıfırdan farklı ve n'den küçük yeni bir t sayısı ve P noktası T açık anahtarını hesap etmede rol almaktadır. T = [t]P eşitliği açık anahtarın bulunmasına yardımcı olmaktadır.

Herkes tarafından bilinen açık anahtarlar E eliptik eğrisi, T noktası, P noktası ve n sayısıdır, t sayısı kişisel anahtardır ve gizli tutulmaktadır (Anonymous 2013).

3.2.2.3 Kimlik tabanlı imza yönteminin adımları

Bu yöntem anahtar oluşturma, imzalama ve doğrulama olmak üzere üç aşamada uygulanmaktadır. Anahtar üretme adımında imza sahibinin talebiyle sunucu imzalama anahtarı üretmektedir. İmzalama adımında rasgele uzunluktaki bir m mesajının imzalama anahtarı kullanılarak imzalanması gerçekleştirilmektedir. Doğrulama adımında ise imza sahibinin doğrulanıp doğrulanmadığı belirlenmektedir (Anonymous 2013).

Şekil 3.4, 3.5 ve 3.6'da sırası ile anahtar üretme, imzalama ve doğrulama adımlarında yapılan işlemler görülmektedir.

- Güvenilir sunucu, sıfırdan farklı ve n'den küçük bir r sayısını rasgele olarak seçer. Bu r sayısını ve kullanılan eliptik eğri üzerinde yer alan temel P noktasını kullanarak R = [r]P eşitliği yardımıyla (x_R, y_R) koordinatlarını hesap eder.
- Güvenilir sunucu ana gizli anahtarı t'yi ve r sayısını kullanarak s'yi bulur.
- R ve s ikilisi imza sahibinin imzalama anahtarı olmaktadır.

Şekil 3.4 Kimlik tabanlı imza yöntemi için anahtar üretme (Anonymous 2013)

- İmza sahibi, sıfırdan farklı ve n'den küçük rasgele bir y tamsayısı seçer.
- Kullanılan eliptik eğri üzerinde yer alan temel P noktasını ve y sayısını kullanarak Y = [y]P eşitliği aracılığıyla Y = (x_Y, y_Y) koordinatlarını hesap eder.
- İmzalama anahtarını, {R, s}, kullanarak z tamsayısını bulur.
- İmza sahibi A ve m mesajının imzası {Y, R, z} olacaktır.

Şekil 3.5 Kimlik tabanlı imza yöntemi için imzalama (Anonymous 2013)

- Doğrulayıcı, m mesajı için imza sahibinin kimlik bilgisi ID ile imza sahibinin imzasını, {Y, R, z}, doğrulamak için c = h(x_Y || x_R || m) formülünü kullanarak bir c tamsayısı hesaplar.
- Doğrulayıcı tarafından bulunan bu c sayısı,

  [z]P = Y + [c]R + [c x h(x_R || ID)]T

eşitliğinde yerine konur. Eşitlik sağlanırsa imza sahibi doğrulanır. Aksi durumda imza sahibi doğrulanmaz ve mekanizma başarısızlıkla sonuçlanmıştır.

Şekil 3.6 Kimlik tabanlı imza yöntemi için doğrulama (Anonymous 2013)

Anahtar üretmedeki s sayısı, (3.1) formülü kullanılarak hesaplanmaktadır. (3.2) eşitliğini sağlayabilen bir kişisel anahtar doğru olarak üretilmiştir. İmzalamada z tamsayısının hesaplanmasında kullanılan formül (3.3)'te ifade edilmektedir (Anonymous 2013).

$$s = r + h(x_R \| ID) \times t \bmod n \quad (3.1)$$

$$[s]P = R + [h(x_R \| ID)]T \quad (3.2)$$

$$z = y + h(x_Y \| x_R \| m) \times s \bmod n \quad (3.3)$$

Bu yöntem mertebesi 160 bit olan sonlu bir asal cisimde tanımlanmış bir eliptik eğriyi kullanmaktadır ve bu eliptik eğrinin kısa Weierstrass eşitliği denklem 2.12'de verilmektedir. Nokta çarpım (Şekil 2.36) kimlik tabanlı imza yönteminin tüm aşamalarında bulunan bir işlemdir. Nokta çarpımda nokta toplama (Şekil 2.32) ve nokta aynılama (Şekil 2.33) işlemleri yapılmaktadır. Hem nokta toplama hem nokta aynılama toplama (Şekil 2.23), çıkarma (Şekil 2.24), çarpma (Şekil 2.25), çarpımsal ters alma (Şekil 2.26) ile bölme (Şekil 2.27) işlemlerini içermektedir (Anonymous 2013).

3.2.3 ALIKE

Zaman sınırlamalarının çok fazla olduğu temassız işlemlerde kullanılmak amacıyla tasarlanmış olan ALIKE mekanizmasının temeli RSA açık anahtarlı şifreleme sistemi üzerine kuruludur. Bu algoritma düşük maliyetli okuyucuların kullanımına olanak vermektedir. Bu mekanizmada bir doğrulayıcı ve bir iddia sahibi bulunmaktadır. Doğrulayıcı tarafından iddia sahibi doğrulanmaktadır. Ek olarak doğrulayıcının ve iddia sahibinin güvenli bir şekilde haberleşmeleri için bir oturum anahtarı oluşturulmaktadır. Hızlı şifre çözme yeteneğine sahiptir. Şifre çözme işlemini iddia sahibi yapmaktadır (Coron vd. 2010, Anonymous 2013).

3.2.3.1 Kullanılan parametreler

ALIKE mekanizmasında açık anahtar, kişisel anahtar, talep ve yanıt bilgisine ek olarak kullanılan parametreler aşağıda listelenmektedir:

E_K: Blok şifreleme fonksiyonudur. Şifreleme anahtarı K'dir.,
u: Şifreleme fonksiyonunda kullanılan K anahtarının bit uzunluğudur.
v: Şifreleme fonksiyonunda kullanılan blok mesajın bit uzunluğudur.
α: Modun bit sayısıdır. $2^{\alpha-1} \leq \text{mod} < 2^{\alpha}$, $\alpha = |\text{mod}|$.

σ: İddia sahibine ait açık anahtarın sertifikasıdır. İddia sahibini doğrulamada kullanılmaktadır.
w: Güvenlik parametresidir.

ρ_1, ρ_2, …: Modun asal çarpanlarıdır. Artan sırada ifade edilmektedir. $\rho_1 < \rho_2 < \ldots$

$f_0(u, x)$: $f_0(u, x) = 0 \parallel x \parallel \ldots\ldots \parallel 0 \parallel x \parallel 0 \parallel x^*$. x^*, x' in en önemli bitlerini simgelemektedir.
$0 \parallel x \parallel \ldots\ldots \parallel 0 \parallel x \parallel 0 \parallel x^*$ ifadesinin bit sayısı u' dur.

$f_1(u, x)$: $f_1(u, x) = 1 \parallel x \parallel \ldots\ldots \parallel 1 \parallel x \parallel 1 \parallel x^*$. x^*, x' in en önemli bitlerini simgelemektedir.
$1 \parallel x \parallel \ldots\ldots \parallel 1 \parallel x \parallel 1 \parallel x^*$ ifadesinin bit sayısı u' dur (Anonymous 2013).

3.2.3.2 Güvenlik gereksinimleri

Seçilen çalışma alanı parametreleri mekanizmanın işleyişini düzenleyen parametrelerdir. Çalışma alanındaki tüm bileşenler tarafından seçilen parametrelere erişme güvenilir bir şekildedir.

İddia sahiplerinin kullandığı asal çarpanlar farklılık göstermektedir. Bu sayede çıktılara herhangi biri kolaylıkla erişemeyecektir.

Çalışma alanındaki tüm bileşenler aynı blok şifreyi (E_K) kullanmaktadır. Anahtar boyutu u, blok boyutu v, anahtar K ile ifade edilmektedir. u ve v'nin, 128 bit uzunlukta ya da 128 bitten daha fazla bit uzunluğunda ve u, v'ye eşit veya v'den daha büyük bir değerde olması gerekmektedir. Blok şifrenin gizli anahtarının maksimum bit uzunluğu v-1'dir. Bit sayısı u olan K gizli anahtarı, $f_0(u, x)$ ve $f_1(u, x)$ fonksiyonları ile bit uzunluğu v-1 olan x gizli anahtarı kullanılarak hesaplanmaktadır. Blok şifreleme anahtarının ilk bitinin ya 0 ya da 1 olduğu bilinmektedir. Böylelikle blok şifrenin birbirinden bağımsız iki farklı kullanımı mümkün olmaktadır.

Her doğrulayıcı ve iddia sahibi, yeni rasgele bit dizileri üretmek için ekipmanlara sahiptir (Anonymous 2013).

3.2.3.3 ALIKE mekanizmasının adımları

ALIKE yönteminde üç adım mevcuttur. İlk adımda iddia sahibinin açık anahtarı N ve kişisel anahtarı t hesap edilmektedir. İddia sahibine ait açık anahtar $P_A = (N, e)$; kişisel anahtar $S_A = (\rho_1, t)$ ile temsil edilmektedir. ρ_1 asal sayısının bit uzunluğu olan w tamsayısı pozitif olup değeri 2.v'den büyük olmalıdır. İkinci sıradaki tek yönlü doğrulama başarıyla tamamlandığında doğrulayıcı iddia sahibini kabul etmektedir. Anahtar üretmede şekil 3.7'de ve tek yönlü doğrulamada 3.8'de belirtilen işlemler yürütülmektedir. ALIKE mekanizması, oturum anahtarının hesaplanması ile son bulmaktadır. Oturum anahtarının hesaplanması isteğe bağlıdır ve oturum anahtarı S_K, (r, k) sayı çifti kullanılarak eşitlik (3.4) yardımıyla oluşturulabilmektedir (Anonymous 2013).

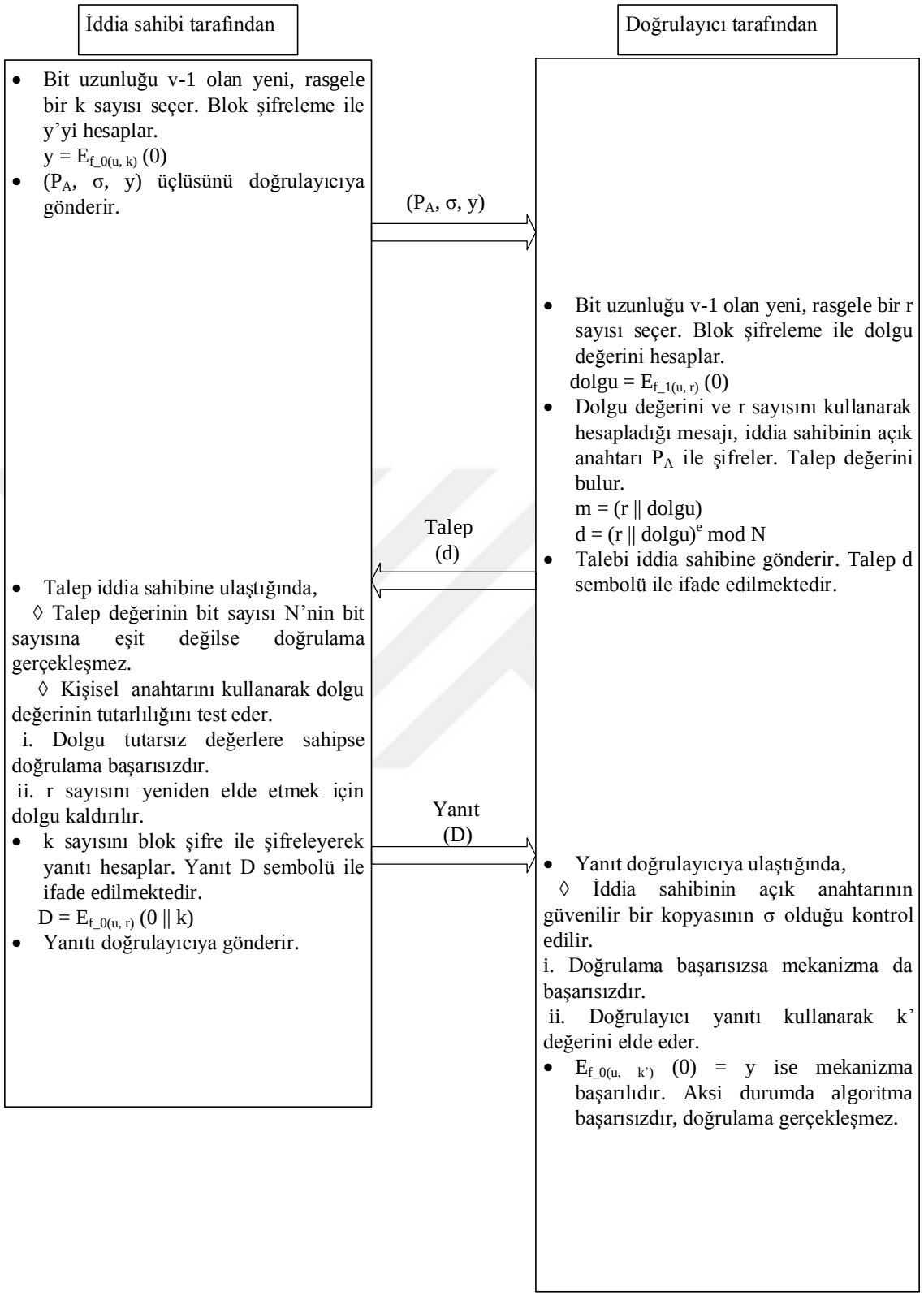
İddia sahibi tarafından

- $|\rho_1| = w$ ve ebob $(e, \rho_1 - 1) = 1$ olacak şekilde uygun bir ρ_1 asal sayısı ve e sayısı seçilmelidir. e üssü herkese açık bir çalışma alanı parametresidir. e'nin 11 olarak seçilmesi pratik bazı avantajlara sahiptir.
- $|\rho_2| = \alpha - w$ olacak şekilde bir ρ_2 asal sayısı seçilmelidir.
- ρ_1 ve ρ_2 asal sayıları çarpılır. Bu çarpımın bit sayısı α sayısına eşit olmaktadır. $|\rho_1 \times \rho_2| = \alpha$.

İddia sahibi, mod N'nin iki asal çarpanı olan ρ_1 ve ρ_2 sayılarını gizli tutmalıdır ve $|\rho_1| << |\rho_2|$ olmalıdır.

- $N = \rho_1 \times \rho_2$ ve $t = e^{-1} \bmod (\rho_1 - 1)$ hesaplanmalıdır. (N, e) ikilisi açık anahtarı, (ρ_1, t) ikilisi kişisel anahtarı temsil etmektedir.

Şekil 3.7 ALIKE için anahtar üretme (Anonymous 2013)



Şekil 3.8 ALIKE için tek yönlü doğrulama (Anonymous 2013)

$$S_K = r \oplus k \tag{3.4}$$

ALIKE mekanizmasında kullanılan ρ_1 asal sayısı 352 bit, N sayısı 1280 bit uzunluğundadır. N, işaretli tamsayılar için çarpma algoritması (Şekil 2.25); t, asal cisimlerde çarpımsal ters alma algoritması (Şekil 2.26) kullanılarak hesaplanmaktadır. Her iki algoritmada da çok duyarlı toplama (Şekil 2.23), çıkarma (Şekil 2.24) ve işaretli tamsayılarda bölme (Şekil 2.27) işlemleri yapılmaktadır. Doğrulamada CLEFIA blok şifresinden faydalanılmaktadır. Anahtar uzunluğu (u) ve mesaj bloğunun boyutu (v) 128 bit olmaktadır (Anonymous 2013).

CLEFIA; 128 bit uzunluklu girdilerin 128, 192 ya da 256 bit anahtarlar ile şifrelenerek 128 bitlik şifreli metinlerin üretildiği blok bir şifreleme tekniğidir. Bu teknikte iki tür Feistel ağı mimarisi kullanılabilmektedir. Girdi, kullanılan anahtar 128 bit ise 4 bölüme; 192 veya 256 ise 8 bölüme ayrılmaktadır. ALIKE mekanizmasında uzunluğu 128 bit olan girdi, 32 bitlik 4 parça halinde şifrelenmektedir ve kullanılan şifreleme anahtarı 128 bittir (Anonymous 2012, Anonymous 2013).

3.2.4 ELLI

ELLI, asimetrik şifreleme tabanlı bir meydan okuma-karşılık verme (*challenge-response*) protokolüdür. Kriptografik meydan okuma-karşılık verme protokolünde başlangıçta doğrulayıcı rasgele bir tamsayı seçmektedir. Rasgele bir sayının seçilmesi düzgün dağılıma sahiptir. Bu rasgele tamsayı kullanılarak talep değişkeninin değeri hesap edilmektedir ve talep değişkeni ile iddia sahibine ait bir gizli bilgi göz önüne alınarak yanıt değişkeni hesaplanmaktadır. Yanıt değişkeni yardımıyla iddia sahibi doğrulanmaktadır. Bu gizli bilgi yalnızca iddia sahibi tarafından bilinmekte olup doğrulama süresi boyunca saklı tutulmaktadır. Meydan okuma-karşılık verme mekanizmalarında rasgele sayıların kullanılması ile tahmin edilemezlik sağlanmaktadır ve böylece saldırılar engellenmektedir (Menezes vd. 1996, Anonymous 2013).

ELLI protokolü pasif RFID etiketlerde kullanılmak amacıyla tasarlanmıştır. Eliptik eğrilerde ayrık logaritma problemine dayanan bir doğrulama yöntemidir. Kullanılan eliptik eğri sonlu ikili cisimler, $GF(2^n)$, üzerinde tanımlıdır ve eliptik eğri üzerindeki nokta sayısı 2^n'dir. Bu doğrulama protokolünde bir iddia sahibi ve bir doğrulayıcı

bulunmaktadır. Bu doğrulama protokolünde yalnızca iddia sahibi doğrulanmaktadır, yani tek yönlü doğrulama (*unilateral*) yapılmaktadır. İzdüşümsel(*projective*) koordinat sistemi kullanılmaktadır. Eliptik eğri üzerindeki noktaların y koordinatı kullanılmamaktadır (Anonymous 2013).

3.2.4.1 Kullanılan parametreler

(Anonymous 2013)'te verildiği şekilde ELLI protokolünün çalışma alanı parametreleri:

1. Mertebesi 2^n olan sonlu bir GF(2^n) cismi,
2. Mertebesi 2^n olan bir indirgeme fonksiyonu f(x),
3. GF(2^n) üzerinde tanımlı sıradan bir eliptik eğri,
4. Eliptik eğri üzerindeki bir P = (x_P, y_P) noktası,
5. P noktasının mertebesi q_1 asal sayısı,
6. E: $y^2 = x^3 + ax^2 + b$ ile verilen eliptik eğri denklemindeki b değişkeni,
7. İddia sahibine ait bir açık anahtar,
8. İddia sahibine ait bir kişisel anahtar,
9. Doğrulayıcının açık anahtarı,
10. Doğrulayıcının kişisel anahtarıdır.

3.2.4.2 Güvenlik gereksinimleri

İddia sahibinin kişisel bir anahtarı olmalıdır.

İddia sahibi, GF (2^n) sonlu cisminde toplama ve çarpma işlemlerini yapabilmelidir.

İddia sahibi, tanımlı eliptik eğri denklemindeki b katsayısını kullanarak Çarpma_1 ve Çarpma_2 fonksiyonunu çalıştırabilmelidir.

Doğrulayıcı, iddia sahibinin kişisel anahtarı kullanılarak hesaplanmış olan iddia sahibine ait açık anahtarı bilmelidir.

Doğrulayıcı, tanımlı eliptik eğri üzerindeki temel P noktasını ve bu P noktasının mertebesi q_1'i bilmelidir.

Doğrulayıcı GF(2^n) sonlu cisminde toplama, çarpma ve bölme işlemlerini yapabilmelidir.

Doğrulayıcı q_1 asal sayısından küçük rasgele pozitif tamsayılar üretebilmelidir (Anonymous 2013).

3.2.4.3 ELLI protokolünün adımları

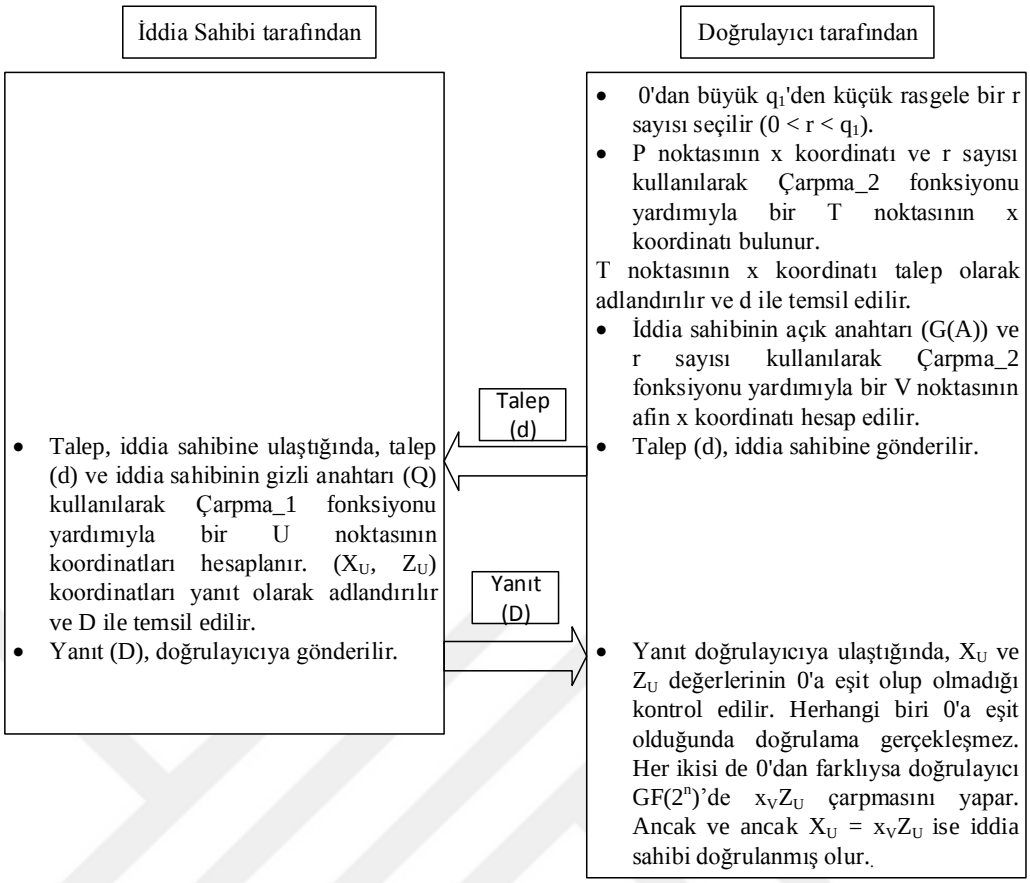
ELLI protokolü iki adımda gerçekleştirilmektedir: 1. Anahtar Üretme, 2. Tek yönlü doğrulama. Anahtar üretme aşamasında iddia sahibine ait bir gizli anahtar ve bir açık anahtar üretilmektedir (Şekil 3.9). Tek yönlü doğrulama doğrulayıcının iddia sahibini doğruladığı aşamadır (Şekil 3.10).

- İddia sahibi {2, 3, …, q_1-1} kümesinden rasgele bir tamsayı seçer.

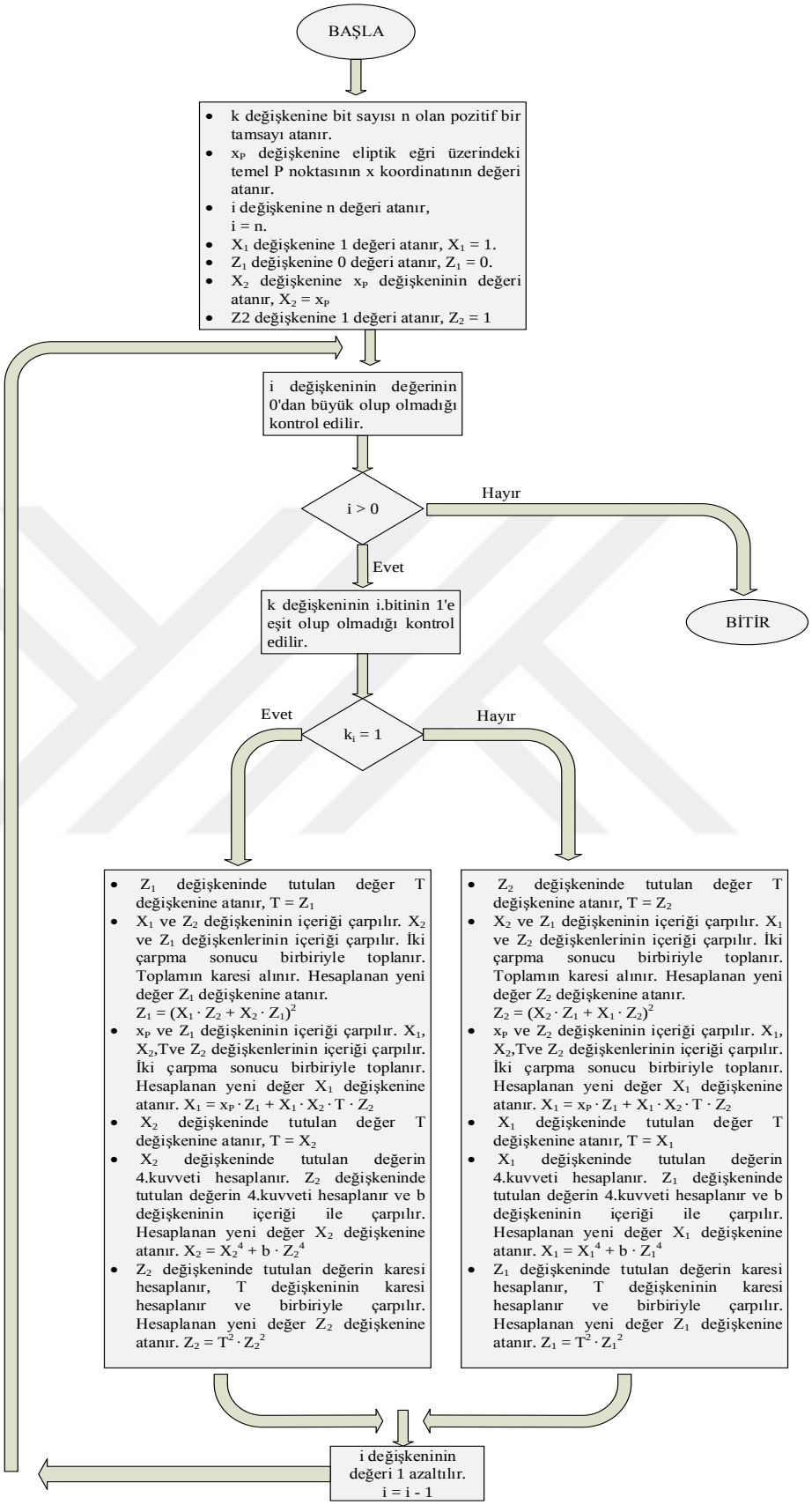
Bu tamsayı iddia sahibinin kişisel anahtarıdır ve Q ile ifade edilmektedir.

- İddia sahibi, kişisel anahtarını (Q) ve temel P noktasının x koordinatını kullanarak Çarpma_2 fonksiyonu yardımıyla açık anahtarını hesaplar. İddia sahibine ait açık anahtar G(A) ile gösterilir.

Şekil 3.9 ELLI için anahtar üretme (Anonymous 2013)



Şekil 3.10 ELLI için tek yönlü doğrulama (Anonymous 2013)



Şekil 3.11 Montgomery çarpma algoritması akış diyagramı (Anonymous 2013)

Çarpma_1 fonksiyonu ile izdüşümsel koordinatlarda nokta çarpım, Çarpma_2 fonksiyonu ile afin koordinatlarda nokta çarpım işlemi yapılmaktadır. Çarpma_1 ve Çarpma_2 fonksiyonları Montgomery çarpma algoritmasını kullanmaktadır. Montgomery çarpma algoritmasının akış şeması şekil 3.11'de görülmektedir. Algoritmanın girdileri pozitif bir tamsayı olan $k = (k_n, \ldots, k_1)$, tanımlı eliptik eğrinin b parametresi ve temel P noktasının x koordinatı; çıktısı (X_1, Z_1) noktalarıdır. Bu algoritmada gerçekleştirilen aritmetik işlemler $GF(2^n)$'de tanımlı toplama (Şekil 2.28), çarpma (Şekil 2.29) ve kare alma (Şekil 2.30) işlemleridir (Anonymous 2013).

Afin koordinatlarda nokta çarpma işleminde (Çarpma_2 fonksiyonu) Montgomery çarpmasına ek olarak çarpımsal ters işlemi (Şekil 2.31) yapılmaktadır. İzdüşümsel koordinatlarda nokta çarpma işleminde (Çarpma_1 fonksiyonu) sadece Montgomery çarpması yapılmaktadır (Anonymous 2013).

Çarpma_1 fonksiyonunda yapılan aritmetik işlemler ve sayıları çizelge 3.1' de; Çarpma_2 fonksiyonunda yapılan aritmetik işlemler ve sayıları çizelge 3.2' de belirtilmektedir.

Çizelge 3.1 Çarpma_1 fonksiyonu

Aritmetik İşlem	İşlem Sayısı
Toplama	3 (her bir döngüde)
Çarpma	6 (her bir döngüde)
Kare alma	5 (her bir döngüde)

Çizelge 3.2 Çarpma_2 fonksiyonu

Aritmetik İşlem	İşlem Sayısı
Toplama	3 (her bir döngüde)
Çarpma	6 (her bir döngüde) + 1 (algoritma sonunda)
Kare alma	5 (her bir döngüde)
Çarpımsal ters	1 (algoritma sonunda)

4. ARAŞTIRMA BULGULARI

Bir yazılım uygulamasının başarımı kod satır sayısı, gerekli RAM/ROM bayt miktarı, her bir komutun yürütülmesi sırasında harcanan zaman (*execution time*) ve veri işleme hacmi olarak ifade edilen saniyede üretilen bit sayısı (*throughput*) ile ölçülebilmektedir. Bu bölümde tez çalışmasında yapılmış yazılım uygulamalarının başarımı analiz edilmektedir. Veri işleme hacmi kullanılan değişkenlerin bit sayısının işlem süresine bölünmesiyle hesap edilmektedir.

Tez kapsamındaki yazılım uygulamaları CodeBlocks derleyicisinde çalıştırılmıştır. CodeBlocks derleyicisindeki yazılım uygulaması 16 bit mikroişlemcilerde kullanıma yöneliktir. Yazılım uygulamalarında her bir değişken dizi tipinde ve dizi elemanları 16 bit uzunlukta tanımlanmıştır. Programlama dili olarak C kullanılmıştır.

Çizelge 4.1'de sonlu asal cisimlerde tanımlı 160 bit aritmetik toplama, çıkarma, çarpma işlemlerinin yazılım uygulamalarına ait başarım metriklerinden kod satır sayısı, bayt cinsinden RAM/ROM kullanımı, işlem süresi ve veri işleme hacmi sonuçları yer almaktadır. Veri işleme hacminin hesaplanmasında toplama için eşitlik (4.1), çıkarma için eşitlik (4.2) ve çarpma için eşitlik (4.3) kullanılmaktadır.

Çizelge 4.1 Sonlu asal cisimlerde 160 bit işlemlerin yazılım başarım metrikleri

İşlem	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
Toplama	Code Blocks	36	1.321	0,001	160.000
Çıkarma	Code Blocks	36	1.321	0,001	160.000
Çarpma	Code Blocks	107	3.190	0,002	80.000

$$\text{Veri işleme hacmi} = 160 \text{ bit} / 0,001 \text{ s} = 160.000 \text{ bit/sn} \quad (4.1)$$

$$\text{Veri işleme hacmi} = 160 \text{ bit} / 0,001 \text{ s} = 160.000 \text{ bit/sn} \quad (4.2)$$

$$\text{Veri işleme hacmi} = 160 \text{ bit} / 0,002 \text{ s} = 80.000 \text{ bit/sn} \quad (4.3)$$

Sonlu asal bir cismin elemanı olan 160 bit uzunluğundaki iki sayının toplanması 0,001 saniye, çıkarılması 0,001 saniye ve çarpılması 0,002 saniye sürmektedir. Toplama ve çıkarma işlemi bellekte 1.321 bayt, çarpma işlemi bellekte 3.190 bayt kullanmaktadır. Toplama ve çıkarma algoritmaları ile kıyaslandığında çarpma algoritmasının işlem süresinin daha uzun olduğu ve daha fazla bellek kullandığı görülmektedir.

Çizelge 4.2'de sonlu asal cisimlerde tanımlı 192 bit aritmetik toplama, çıkarma, çarpma, indirgenmiş çarpma, bölme, çarpımsal ters alma algoritmalarının yazılım uygulamalarına ait performans metriklerinden kod satır sayısı, bayt cinsinden RAM/ROM kullanımı, işlem süresi ve veri işleme hacmi sonuçları verilmektedir. Toplama, çıkarma, çarpma, indirgenmiş çarpma, bölme ve çarpımsal ters alma algoritmalarının veri işleme hacmi sırayla (4.4), (4.5), (4.6), (4.7), (4.8), (4.9) eşitlikleri ile hesaplanmaktadır.

Çizelge 4.2 Sonlu asal cisimlerde 192 bit işlemlerin yazılım başarım metrikleri

İşlem	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
Toplama	Code Blocks	36	1.402	0,001	192.000
Çıkarma	Code Blocks	36	1.400	0,001	192.000
Çarpma	Code Blocks	107	3.287	0,002	96.000
İndirgenmiş Çarpma	Code Blocks	110	3.875	0,001	192.000
Bölme	Code Blocks	160	4.712	0,002	96.000
Çarpımsal Ters Alma	Code Blocks	236	6.765	0,006	32.000

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,001 \text{ s} = 192.000 \text{ bit/sn} \quad (4.4)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,001 \text{ s} = 192.000 \text{ bit/sn} \quad (4.5)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,002 \text{ s} = 96.000 \text{ bit/sn} \quad (4.6)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,001 \text{ s} = 192.000 \text{ bit/sn} \quad (4.7)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,002 \text{ s} = 96.000 \text{ bit/sn} \quad (4.8)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,006 \text{ s} = 32.000 \text{ bit/sn} \quad (4.9)$$

Sonlu asal bir cisimde tanımlı 192 bit uzunluğundaki iki sayının toplanması için işlem süresi 0,001 saniye ve bellek kullanımı 1.402 bayttır; çıkarılması için işlem süresi 0,001 saniye ve bellek kullanımı 1.400 bayttır; çarpılması için işlem süresi 0, 002 saniye ve bellek kullanımı 3.287 bayttır. 192 bit iki sayının çarpım sonucu hızlı modüler indirgeme algoritması (Şekil 3.3) ile indirgenmektedir. Hızlı modüler çarpma algoritmasında çarpma sonucunun Federal Bilgi İşleme standardının (*Federal Information Processing Standard*) 186-3 sayılı yayınında tanımlanan $P_{192} = 2^{192} – 2^{64} – 1$ asal sayısına göre modu alınmaktadır. İndirgenmiş çarpma işlemi 0,001 saniyede yapılmaktadır ve 3.875 bayt bellek kullanmaktadır. 192 bit bir sayının 192 bit bir sayıya bölünmesi işleminin yapılma süresi 0,002 saniyedir ve bellek kullanımı 4.172 bayttır. 192 bit uzunluğundaki bir sayının çarpımsal tersi 0,006 saniyede bulunmaktadır ve çarpımsal ters bulma algoritması bellekte 6.765 bayt tutmaktadır. Çarpma algoritmasının indirgenmesi ile çarpma işleminin yapılma süresi kısalmıştır, veri işleme hacmi artmıştır ancak bellek kullanımında artış meydana gelmiştir. Başarımı en düşük olan işlem çarpımsal ters almadır. Çarpımsal ters alma algoritmasında bellek tüketimi fazla, işlem süresi uzun ve veri işleme hacmi düşüktür.

Çizelge 4.3'te sonlu asal bir cisimde tanımlı 1280 bit aritmetik toplama, çıkarma, çarpma ve çarpımsal ters alma işlemlerinin yazılım uygulamaları için başarım sonuçları sunulmaktadır. Yazılım başarımı kod satır sayısı, bayt cinsinden RAM/ROM kullanımı, işlem süresi ve veri işleme hacmi metrikleri kullanılarak analiz edilmektedir. Toplama, çıkarma, çarpma ve çarpımsal ters alma işlemlerinin veri işleme hacmi sıralı olarak eşitlik (4.10), (4.11), (4.12) ve (4.13)'te verilmektedir.

Çizelge 4.3 Sonlu asal cisimlerde 1280 bit işlemlerin yazılım başarım metrikleri

İşlem	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
Toplama	Code Blocks	47	2.092	0,006	213.333
Çıkarma	Code Blocks	47	2.094	0,006	213.333
Çarpma	Code Blocks	82	3.093	0,023	55.652
Çarpımsal Ters Alma	Code Blocks	224	6.462	0,026	49.231

$$\text{Veri işleme hacmi} = 1280 \text{ bit} / 0,006 \text{ s} = 213.333 \text{ bit/sn} \quad (4.10)$$

$$\text{Veri işleme hacmi} = 1280 \text{ bit} / 0,006 \text{ s} = 213.333 \text{ bit/sn} \quad (4.11)$$

$$\text{Veri işleme hacmi} = 1280 \text{ bit} / 0,023 \text{ s} = 55.652 \text{ bit/sn} \quad (4.12)$$

$$\text{Veri işleme hacmi} = 1280 \text{ bit} / 0,026 \text{ s} = 49.231 \text{ bit/sn} \quad (4.13)$$

1280 bit uzunluğundaki iki sayının toplanması işleminin süresi 0,006 saniye, çıkarılması işleminin süresi 0,006 saniye ve çarpılması işleminin süresi 0,023 saniye olarak bulunmuştur. 1280 bit bir sayının çarpımsal tersi 0,026 saniyede hesaplanmıştır. Toplama işlemi bellekte 2.092 bayt, çıkarma işlemi 2.094 bayt, çarpma işlemi 3.093 bayt ve çarpımsal ters alma işlemi 6.462 bayt yer tutmaktadır. Çarpımsal ters alma diğer işlemler ile karşılaştırıldığında bellek kullanımı en fazla, işlem süresi en uzun ve veri işleme hacmi en düşük olan işlemdir.

Sonlu asal cisimlerde tanımlı 160 bit ve 192 bit sayılar ile yapılan aritmetik işlemlerde 1280 bit sayılar ile yapılan aritmetik işlemlere göre işlem süresi daha kısadır. Bit sayısındaki önemli ölçüde artış işlem süresinin uzamasına sebep olmaktadır.

Sonlu asal bir cisimde tanımlı bir eliptik eğride nokta toplama, nokta aynılama ve nokta çarpım işlemleri yazılımda uygulanmıştır. Bu eliptik eğrinin parametrelerinin bit sayısı maksimum 192'dir. Örneğin bu eliptik eğri üzerindeki bir P noktasının x ile y koordinatları, bu eliptik eğrinin grup mertebesi yani eleman sayısı maksimum 192 bit uzunlukta bir değer olmaktadır. Eliptik eğri denklemindeki a katsayısının -3'e eşit veya

-3'ten farklı olduğu durumlar söz konusudur. a katsayısının -3 olduğu eliptik eğri FIPS 186-3 sayılı yayınında tanımlanmış bir eliptik eğridir. a katsayısının -3 seçilme sebebi eliptik eğri işlemlerinin performansını iyileştirmektir.

Sonlu asal cisimlerde tanımlı, grup mertebesi maksimum 192 bit olan eliptik eğrilerde nokta toplama ve nokta aynılama işlemlerinin yazılım uygulamalarının başarım metriklerinden kod satır sayısı, RAM/ROM boyutu, işlem süresi ve veri işleme hacmi sonuçları çizelge 4.4'te verilmektedir.

a = -3 iken nokta toplama ve nokta aynılama algoritmalarının veri işleme hacminin hesabı için sırasıyla eşitlik (4.14) ile (4.15) kullanılmaktadır. a ≠ -3 iken nokta toplama ve nokta aynılama algoritmalarının veri işleme hacmi sıralı olarak eşitlik (4.16) ve (4.17)'de hesap edilmektedir.

Çizelge 4.4 Sonlu asal cisimlerde tanımlı, grup mertebesi maksimum 192 bit olan eliptik eğrilerde nokta toplama ve nokta aynılama işlemlerinin yazılım başarım metrikleri

a = -3	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
Nokta Toplama	Code Blocks	499	14.654	0,007	27.429
Nokta Aynılama	Code Blocks	498	14.153	0,009	21.333
a ≠ -3	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
Nokta Toplama	Code Blocks	498	14.305	0,012	16.000
Nokta Aynılama	Code Blocks	478	13.548	0,013	14.769

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,007 \text{ s} = 27.429 \text{ bit/sn} \quad (4.14)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,009 \text{ s} = 21.333 \text{ bit/sn} \quad (4.15)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,012 \text{ s} = 16.000 \text{ bit/sn} \quad (4.16)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,013 \text{ s} = 14.769 \text{ bit/sn} \quad (4.17)$$

Sonlu asal cisimlerde tanımlı, grup mertebesi maksimum 192 bit olan eliptik eğrilerde nokta çarpım işleminin yazılım uygulamalarının başarım metriklerinden kod satır sayısı, RAM/ROM boyutu, işlem süresi ve veri işleme hacmi sonuçları çizelge 4.5'te görüldüğü gibidir. Nokta çarpım işlemi sağdan sola ikiyle çarp ve topla algoritması (*Left-to-right double and add algorithm*) ile yapılmaktadır. Nokta çarpım algoritmasının veri işleme hacmi a = -3 iken eşitlik (4.18), a ≠ -3 iken eşitlik (4.19)'daki gibidir.

Çizelge 4.5 Sonlu asal cisimlerde tanımlı, grup mertebesi maksimum 192 bit olan eliptik eğrilerde nokta çarpım işleminin yazılım başarım metrikleri

	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
a = -3	Code Blocks	660	18.779	0,013	14.769
a ≠ -3	Code Blocks	640	17.779	0,028	6.857

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,013 \text{ s} = 14.769 \text{ bit/sn} \quad (4.18)$$

$$\text{Veri işleme hacmi} = 192 \text{ bit} / 0,028 \text{ s} = 6.857 \text{ bit/sn} \quad (4.19)$$

a katsayısı -3 olarak seçildiğinde nokta toplama işleminin işlem süresinin 0,007 saniye ve RAM/ROM kullanımının 14.654 bayt; nokta aynılama işleminin işlem süresinin 0,009 saniye ve RAM/ROM kullanımının 14.153 bayt; nokta çarpım işleminin işlem süresinin 0,013 ve RAM/ROM kullanımının 18.779 bayt olduğu görülmektedir. a katsayısı -3'ten farklı bir değer aldığında nokta toplama işlemi için işlem süresi 0,012 saniye ve RAM/ROM kullanımı 14.305 bayt; nokta aynılama işlemi için işlem süresi 0,013 saniye ve RAM/ROM kullanımı 13.548 bayt; nokta çarpım işlemi için işlem süresi 0,028 saniye ve RAM/ROM kullanımı 17.779 bayt olarak elde edilmiştir. a'nın -3 olması durumunda eliptik eğri işlemleri daha hızlı yapılmaktadır fakat daha fazla bellek kullanmaktadır.

Çizelge 4.6'da CLEFIA blok şifresi için 128 bit ara (*intermediate*) anahtar hesabının yazılım uygulamasının başarım metriklerinden kod satır sayısı, RAM/ROM boyutu, işlem süresi ve veri işleme hacmi sonuçları görülmektedir. Eşitlik (4.20)'de CLEFIA blok şifresinin ara anahtar hesabının veri işleme hacmi belirtilmektedir.

Çizelge 4.6 CLEFIA blok şifresi için ara anahtar hesabının yazılım başarım metrikleri

Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
Code Blocks	220	8.752	0,004	32.000

$$\text{Veri işleme hacmi} = 128 \text{ bit} / 0,004 \text{ s} = 32.000 \text{ bit/sn} \quad (4.20)$$

ALIKE mekanizmasının anahtar üretme aşamasının yazılım uygulamasının başarım metriklerinden kod satır sayısı, RAM/ROM boyutu, işlem süresi ve veri işleme hacmi sonuçları çizelge 4.7'de analiz edilmektedir. Eşitlik (4.21)'de ALIKE mekanizmasının anahtar üretme adımının veri işleme hacmi görülmektedir.

Çizelge 4.7 ALIKE algoritması anahtar üretme adımının yazılım başarım metrikleri

Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
Code Blocks	250	7.743	0,042	30.476

$$\text{Veri işleme hacmi} = 1280 \text{ bit} / 0,042 \text{ s} = 30.476 \text{ bit/sn} \quad (4.21)$$

Sonlu ikili cisimlerde tanımlı 163 ve 193 bit sayılar kullanılarak yapılan çarpma, çarpımsal ters bulma işlemlerinin yazılım uygulamalarının başarım metriklerinden kod satır sayısı, RAM/ROM boyutu, işlem süresi ve veri işleme hacmi sonuçları çizelge 4.8'de yer almaktadır. 163 bit sayılarda çarpma işleminin veri işleme hacmi sonucu eşitlik (4.22)'de ve çarpımsal ters alma işleminin eşitlik (4.24)'te sunulmaktadır. 193 bit sayılarda çarpma işleminin veri işleme hacmi sonucu eşitlik (4.23)'te ve çarpımsal ters alma işleminin veri işleme hacmi sonucu eşitlik (4.25)'teki gibi olmaktadır.

Çizelge 4.8 Sonlu ikili cisimlerde tanımlı çarpma ve çarpımsal ters alma işlemlerinin yazılım başarım metrikleri

Çarpma					
Bit Sayısı	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
163	Code Blocks	86	2.431	0,004	40.750
193	Code Blocks	85	2.582	0,007	27.571
Çarpımsal Ters					
Bit Sayısı	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
163	Code Blocks	130	3.252	0,01	16.300
193	Code Blocks	130	3.298	0,01	19.300

$$\text{Veri işleme hacmi} = 163 \text{ bit} / 0,004 \text{ s} = 40.750 \text{ bit/sn} \quad (4.22)$$

$$\text{Veri işleme hacmi} = 193 \text{ bit} / 0,007 \text{ s} = 27.571 \text{ bit/sn} \quad (4.23)$$

$$\text{Veri işleme hacmi} = 163 \text{ bit} / 0,01 \text{ s} = 16.300 \text{ bit/sn} \quad (4.24)$$

$$\text{Veri işleme hacmi} = 193 \text{ bit} / 0,01 \text{ s} = 19.300 \text{ bit/sn} \quad (4.25)$$

Tıpkı sonlu asal cisimlerde olduğu gibi sonlu ikili cisimlerde de çarpımsal ters alma işleminin çarpma işlemine göre işlem süresi daha uzun ve bellek tüketimi daha fazladır.

Sonlu ikili bir cisimde tanımlı iki eliptik eğri versiyonu için nokta çarpım işlemi yazılımda uygulanmıştır. Nokta çarpım işlemi Montgomery algoritması ile yapılmaktadır. 163 versiyonunda eliptik eğri parametrelerinin bit sayısı maksimum 163 bit, 193 versiyonunda eliptik eğri parametrelerinin bit sayısı maksimum 193 bit olabilmektedir. Örneğin 163 versiyonu için eliptik eğri üzerindeki bir P noktasının x ile y koordinatları, bu eliptik eğrinin grup mertebesi yani eleman sayısı maksimum 163 bit iken; 193 versiyonu için eliptik eğri üzerindeki bir P noktasının x ile y koordinatları, bu eliptik eğrinin grup mertebesi yani eleman sayısı maksimum 193 bittir.

Sonlu ikili cisimlerde tanımlı 163 ve 193 eliptik eğri versiyonları için nokta çarpım işleminin yazılım başarım metriklerinden kod satır sayısı, RAM/ROM boyutu, işlem süresi ve veri işleme hacmi sonuçları çizelge 4.9'daki gibidir. Nokta çarpım işleminin

veri işleme hacmi 163 versiyonu için eşitlik (4.26)'da; 193 versiyonu için (4.27)'de belirtilmektedir.

Çizelge 4.9 Sonlu ikili cisimlerde nokta çarpım işleminin yazılım başarım metrikleri

Versiyon	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
163	Code Blocks	432	12.096	0,103	1.583
193	Code Blocks	431	12.165	0,116	1.664

$$\text{Veri işleme hacmi} = 163 \text{ bit} / 0{,}103 \text{ s} = 1.583 \text{ bit/sn} \quad (4.26)$$

$$\text{Veri işleme hacmi} = 193 \text{ bit} / 0{,}116 \text{ s} = 1.664 \text{ bit/sn} \quad (4.27)$$

ELLI tek yönlü doğrulama tekniğinin yazılım uygulaması iki aşamada gerçekleştirilmiştir. İlk olarak anahtar üretme adımı sonrasında doğrulama adımı yazılımda uygulanmıştır.

ELLI algoritmasının 163 ve 193 olmak üzere iki çeşidi mevcuttur. ELLI-163 versiyonunda 163 bit uzunluğunda anahtarlar kullanılarak doğrulama yapılmaktadır. ELLI-193 versiyonunda doğrulama işlemi için 193 bit uzunluğunda anahtarlar kullanılmaktadır.

ELLI algoritmasının anahtar üretme ve doğrulama adımlarının CodeBlocks derleyicisindeki yazılım uygulamalarının başarım metriklerine ait sonuçları sıralı olarak çizelge 4.10 ve 4.11'de verilmektedir. Anahtar üretme adımında ELLI-163 versiyonu için veri işleme hacmi eşitlik (4.28), ELLI-193 versiyonu için eşitlik (4.29) ile bulunmaktadır. Doğrulama adımında ELLI-163 versiyonunun veri işleme hacmi sonucu eşitlik (4.30)'da, ELLI-193 versiyonunun veri işleme hacmi sonucu eşitlik (4.31)'de yer almaktadır.

Çizelge 4.10 ELLI algoritmasının anahtar üretme adımının yazılım başarım metrikleri

Versiyon	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
ELLI-163	Code Blocks	446	12.383	0,115	1.417
ELLI-193	Code Blocks	445	12.438	0,14	1.379

Çizelge 4.11 ELLI algoritmasının doğrulama adımının yazılım başarım metrikleri

Versiyon	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
ELLI-163	Code Blocks	452	12.489	0,074	2.203
ELLI-193	Code Blocks	452	12.565	0,123	1.569

$$\text{Veri işleme hacmi} = 163 \text{ bit} / 0{,}115 \text{ s} = 1.417 \text{ bit/sn} \quad (4.28)$$

$$\text{Veri işleme hacmi} = 193 \text{ bit} / 0{,}14 \text{ s} = 1.379 \text{ bit/sn} \quad (4.29)$$

$$\text{Veri işleme hacmi} = 163 \text{ bit} / 0{,}074 \text{ s} = 2.203 \text{ bit/sn} \quad (4.30)$$

$$\text{Veri işleme hacmi} = 193 \text{ bit} / 0{,}123 \text{ s} = 1.569 \text{ bit/sn} \quad (4.31)$$

ELLI algoritmasının CodeBlocks derleyicisindeki yazılım uygulamasının başarım metriklerine ait sonuçlar çizelge 4.12'deki gibidir. ELLI-163 algoritmasının veri işleme hacmi eşitlik (4.32)'de, ELLI-193 algoritmasının veri işleme hacmi eşitlik (4.33)'te hesap edilmektedir.

Çizelge 4.12 ELLI algoritmasının CodeBlocks'taki yazılım uygulamasına ait başarım metrikleri

Versiyon	Derleyici	Kod Satır Sayısı	RAM/ROM Boyutu (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (*throughput*) bit/s
ELLI-163	Code Blocks	480	13.207	0,154	1.058
ELLI-193	Code Blocks	478	13.249	0,173	1.116

$$\text{Veri işleme hacmi} = 163 \text{ bit} / 0{,}154 \text{ s} = 1.058 \text{ bit/sn} \quad (4.32)$$

$$\text{Veri işleme hacmi} = 193 \text{ bit} / 0{,}173 \text{ s} = 1.116 \text{ bit/sn} \quad (4.33)$$

ELLI asimetrik doğrulama yönteminin kaynak kısıtları olan, internete bağlı cihazların bulunduğu bir uygulama ortamının güvenliğini sağlaması istenmektedir. Bu durumda çapraz derleyici (*cross compiler*) kullanımına ihtiyaç duyulmaktadır. Arduino MAC, Linux ve Windows işletim sistemlerinde çalışabilen çapraz bir platformdur (Anonymous 2019a).

ELLI algoritmasının yazılım uygulaması Arduino 1.8.12 emülatöründe de yapılmıştır ve yazılım uygulamasında C programlama dili tercih edilmiştir. ELLI mekanizmasının 8 bitlik mikro denetleyiciye yönelik yazılım uygulaması için Arduino Uno Rev 3 mikro denetleyici kartı kullanılmıştır. Bu mikro denetleyici kartının saat frekansı (*clock frequency*) 16 MHz, anlık hafızası (*flash memory*) 32 KB, statik RAM kapasitesi (SRAM) 2 KB ve elektrikle yazılıp silinebilen ROM (EEPROM) kapasitesi 1 KB'tır. Program kodu anlık hafızada tutulmaktadır (Anonymous 2019b, Anonymous 2019c, Anonymous 2019d, Anonymous 2019e).

ELLI asimetrik doğrulama protokolünün Arduino emülatöründe yapılmış yazılım uygulamasının kod satır sayısı, statik RAM, anlık hafıza kullanımı ve işlem süresi sonuçları çizelge 4.13'te yer almaktadır. ELLI-163 algoritması için veri işleme hacmi eşitlik (4.34)'te; ELLI-193 algoritması için veri işleme hacmi eşitlik (4.35)'te belirtilmektedir.

Çizelge 4.13 ELLI algoritmasının Arduino'daki yazılım uygulamasına ait başarım metrikleri

Versiyon	Derleyici	Kod Satır Sayısı	SRAM (bayt)	Anlık hafıza (*flash memory*) (bayt)	İşlem Süresi (saniye)	Veri İşleme Hacmi (throughput) bit/s
ELLI-163	Arduino 1.8.12	466	316	6.718	76,820	2,122
ELLI-193	Arduino 1.8.12	467	332	6.856	130,679	1,477

$$\text{Veri işleme hacmi} = 163 \text{ bit} / 76,820 \text{ s} = 2,122 \text{ bit/sn} \quad (4.34)$$

$$\text{Veri işleme hacmi} = 193 \text{ bit} / 130,679 \text{ s} = 1,477 \text{ bit/sn} \quad (4.35)$$

CodeBlocks derleyicisinde çalıştırılan ELLI algoritmasında 16 bit değişkenlerin kullanılması algoritmadaki döngü sayılarının azalmasını dolayısıyla işlem sürelerinin

kısalmasını sağlamıştır. Ama 16 bit değişkenler ile işlem yapılması belleğin daha fazla kullanılmasına neden olmuştur. Arduino emülatöründe maksimum 2 Kilobayt RAM varlığında çalıştırılan ELLI algoritmasında 8 bit değişkenler ile işlem yapılması sonucunda döngü sayısı artmış ve bu durumda ELLI algoritmasının işlem süresi uzamıştır.

5. TARTIŞMA ve SONUÇ

Düşük seviyeli assembly dilinin komut kümesi oldukça dardır. Assembly dilinde program akışının yönlendirilmesi için atlama komutları kullanılmaktadır ve bazı komutlar sadece belirli yazmaçlar kullanılarak yürütülebilmektedir. Yazmaç sayısının az olması nedeniyle belleklerde işlem yapmak kaçınılmazdır. Atlama komutları işlem süresinin artmasına sebep olmaktadır. Yukarıda bahsedilen dezavantajların yanı sıra eliptik eğri ve sonlu cisim aritmetiğinin karmaşıklığı nedeniyle bu tezdeki yazılım uygulamalarında assembly dili tercih edilmemiştir.

C dilinde değişken tiplerinin fazla olması, dizi, gösterge gibi değişken tiplerinin kullanılabilmesi, fonksiyonların varlığı eliptik eğri ve sonlu cisim aritmetiğinin uygulanmasını kolaylaştırmaktadır. Bu nedenle bu tezdeki yazılım uygulamalarında C dili kullanılmıştır.

Tez çalışmaları kapsamında sonlu asal cisimlerde tanımlı toplama, çıkarma, çarpma, indirgenmiş çarpma, bölme, çarpımsal ters alma; sonlu ikili cisimlerde tanımlı çarpma ve çarpımsal ters alma algoritmaları yazılım ile gerçekleştirilmiştir. Sonlu asal cisim uzayında bulunan eliptik eğrilerdeki nokta toplama ve nokta aynılama işlemlerinin yazılım uygulamaları izdüşümsel koordinatlar kullanılarak yapılmıştır. Nokta çarpım işlemi için sonlu asal cisimlerde sağdan sola ikiyle çarp ve topla algoritması; sonlu ikili cisimlerde Montgomery algoritması yazılımda uygulanmıştır. CLEFIA blok şifresi için ara anahtar hesabı yazılımda gerçeklenmiştir. ALIKE mekanizmasının yazılım uygulamasında iddia sahibine ait açık ve kişisel anahtarlar üretilmiştir. ELLI protokolünün hem 163 hem 193 bit versiyonunun anahtar üretme ve tek yönlü doğrulama adımları yazılım yoluyla yapılmıştır. Yazılım uygulamalarında ilk olarak bellek tüketiminin en aza indirilmesi, ikinci olarak kısa işlem süresi ve yüksek veri işleme hacmi hedeflenmiştir. 8 ve 16 bit işlemcilere yönelik gerçeklenmiş yazılım uygulamalarının başarımı kod satır sayısı, RAM/ROM boyutu, işlem süresi ve veri işleme hacmi metrikleri göz önünde bulundurularak değerlendirilmiştir.

Tez kapsamında 16 bit mikroişlemcilere yönelik yazılım uygulamaları için CodeBlocks derleyicisi kullanılmıştır. 8 bitlik mikroişlemcilerde kullanılabilecek yazılım uygulamaları Arduino emülatöründe çalıştırılmıştır. Her iki platformda da yazılım uygulamaları C dili ile yapılmıştır.

Malik (2010), sonlu asal cisimlerde tanımlı 160 bit aritmetik toplama, çıkarma ve çarpma işlemlerinin yazılım uygulamalarını çalışma frekansı 160 MHz olan, gelişmiş Harvard mikroişlemci mimarisine sahip TMS320VC5416 sayısal sinyal işlemcisinde (*digital signal processor*) yapmıştır. Bu sayısal sinyal işlemcisi 16 bit bir işlemcidir, RAM boyutu 128K x 16 bit, ROM boyutu 16K x 16 bittir. Yazılım uygulamalarında assembly dili kullanılmaktadır. Sonlu asal cisimlerde tanımlı 160 bit sayılar ile yapılan aritmetik toplama, çıkarma ve çarpma işlemlerinin literatür çalışmasında ve tez çalışmasında elde edilen işlem süreleri milisaniye cinsinden çizelge 5.1'deki gibidir.

Çizelge 5.1 Farklı platformlarda sonlu asal cisimlerde tanımlı 160 bit aritmetik işlemlerin işlem süreleri

Çalışma	Platform	İşlem	İşlem süresi
Malik (2010)	16 bit Sayısal Sinyal İşlemcisi assembly dili	Toplama	0,00197
		Çıkarma	0,00223
		Çarpma	0,01788
Tez	16 bit CodeBlocks C dili	Toplama	1,0
		Çıkarma	1,0
		Çarpma	2,0

Sonlu asal cisimlerde tanımlı 160 bit aritmetik toplama, çıkarma ve çarpma işlemleri tezdeki yazılım uygulamalarında Malik (2010) tarafından gerçekleştirilen yazılım uygulamalarından daha uzun sürmektedir. Malik (2010) tarafından gerçeklenen uygulamanın assembly dilinde olmasının kaynak kullanımını daha etkin hale getirdiği ve işlem süresi bakımından daha iyi sonuçlar alınmasına katkı sağladığı düşünülmektedir. Literatür çalışmasında işlemlerin yapılma sürelerinin daha kısa olması bu durumun bir sonucu olarak değerlendirilmektedir. Tez çalışmasındaki uygulamalarda işlemlerin daha uzun sürme nedeninin C dilinin kullanılması olduğu düşünülmektedir.

Brown vd. (2001), sonlu asal cisimlerde tanımlı 192 bit aritmetik toplama, çıkarma, indirgenmiş çarpma ve çarpımsal ters alma algoritmalarının yazılım uygulamalarını çalışma frekansı 400 MHz olan Pentium II işlemcisinde çalıştırmıştır. Yazılım uygulamaları C dili ile yapılmıştır. Derleyici olarak Microsoft C derleyicisi kullanılmıştır. İşlemler 32 bit değişkenler ile yapılmıştır. Sonlu asal cisimlerde tanımlı 192 bit aritmetik toplama, çıkarma, indirgenmiş çarpma ve çarpımsal ters alma işlemlerinin literatür çalışmasında ve tez kapsamındaki yazılım uygulamasında ulaşılan işlem süresi sonuçları milisaniye cinsinden çizelge 5.2'de yer almaktadır.

Çizelge 5.2 Farklı platformlarda sonlu asal cisimlerde tanımlı 192 bit aritmetik işlemlerin işlem süreleri

Çalışma	Platform	İşlem	İşlem süresi
Brown vd. (2001)	32 bit Microsoft C derleyicisi C dili	Toplama	0,000235
		Çıkarma	0,000243
		İndirgenmiş Çarpma	0,001268
		Çarpımsal Ters	0,014621
Tez	16 bit CodeBlocks C dili	Toplama	1,0
		Çıkarma	1,0
		İndirgenmiş Çarpma	1,0
		Çarpımsal Ters	6,0

Çizelge 5.2'deki sonuçlara göre aritmetik toplama, çıkarma, indirgenmiş çarpma ve çarpımsal ters alma işlemleri 32 bit değişkenlerin kullanıldığı Microsoft C derleyicisinde 16 bit değişkenlerin kullanıldığı CodeBlocks derleyicisine kıyasla daha hızlı yapılmaktadır. Tez kapsamında gerçekleştirilen yazılım uygulamalarının işlem sürelerinin daha uzun olmasının mimari farklılıklardan kaynaklandığı söylenebilir. 32 bit değişkenlerin kullanıldığı algoritmalarda döngü sayıları ile işlemlerin tekrarlanma sayıları 16 bit değişkenlerin kullanıldığı algoritmalardakine göre daha az olmaktadır. Bu durumun işlem sürelerinin kısalmasını sağladığı düşünülmektedir.

Sonlu asal cisimlerde tanımlı, grup mertebesi maksimum 192 bit olan eliptik eğrilerin kullanıldığı ve a'nın -3'e eşit olduğu nokta çarpım işleminin literatürde yer alan farklı platformlardaki yazılım uygulamasında ve tez çalışmasında elde edilen işlem süreleri milisaniye cinsinden çizelge 5.3'te verilmektedir.

Çizelge 5.3 Sonlu asal cisimlerde tanımlı, grup mertebesi maksimum 192 bit olan eliptik eğrilerde nokta çarpım algoritmasının farklı platformlardaki işlem süreleri

Çalışma	Platform	İşlem süresi
Wenger ve Werner (2011)	16 bit MSP430 C dili	2,926
Wenger ve Verner (2011)	16 bit PIC24 C dili	5,616
Tez	16 bit CodeBlocks C dili	13,0

Wenger ve Werner (2011) yazılım uygulamalarını çalışma frekansı 8 MHz olan 16 bit mikroişlemci mimarisine sahip MSP430F1611 mikrodenetleyicisinde ve çalışma frekansı 16 MHz olan 16 bit mikroişlemci mimarisine sahip PIC24FJ96GA006 mikrodenetleyicisinde gerçekleştirmiştir. Yazılım uygulamalarında C dili tercih edilmiştir. Tez kapsamında gerçekleştirilen uygulamanın işlem süresinin Wenger ve Werner (2011) tarafından yapılan uygulamaya göre daha uzun olduğu görülmektedir. Tez kapsamında gerçekleştirilen uygulamada nokta çarpım algoritmasında kullanılan aritmetik işlemlerin her biri değer döndüren, parametre alan bir fonksiyon olarak tanımlanmıştır. Nokta çarpım algoritmasının yürütülmesi sırasında bu fonksiyonlar tekrar tekrar çağrılmaktadır. Bu durumun işlem süresinin uzamasına yol açtığı düşünülmektedir.

Lopez ve Dahab (2002) sonlu ikili cisimlerde tanımlı olan, 163 bit uzunluğunda değişkenlerin çarpılması ve 163 bit bir değişkenin çarpımsal tersinin alınması işlemlerini yazılımda uygulamıştır. Uygulama çalışma frekansı 300 MHz olan, 64 bitlik işlemlerin yapıldığı Sun UltraSPARC makinesinde çalıştırılmıştır. Yazılım

uygulamasında C++ dili ile LiDIA adlı hazır kütüphane kullanılmıştır. Literatür çalışmasının ve tezdeki uygulamanın işlem süreleri milisaniye cinsinden çizelge 5.4'te sunulmaktadır.

Çizelge 5.4 Sonlu ikili cisimlerde 163 bit aritmetik işlemlerin farklı platformlardaki işlem süreleri

Çalışma	Platform	İşlem	İşlem süresi
Lopez ve Dahab (2002)	64 bit Sun UltraSPARC C++ dili	Çarpma	0,0105
		Çarpımsal Ters	0,0961
Tez	16 bit CodeBlocks C dili	Çarpma	4,0

Çizelge 5.4'teki sonuçlar incelendiğinde tezdeki çalışmada çarpma ve çarpımsal ters alma işlemlerinin daha uzun sürede yapıldığı görülmektedir. Tez kapsamında yapılan uygulamaların daha uzun işlem sürelerine sahip olması mimari farklılıklar ile ilişkilendirilebilir. Lopez ve Dahab (2002) tarafından gerçeklenen uygulamada işlemlerde 64 bit değişkenler kullanılmaktadır. Bit sayısı daha fazla olan değişkenlerin kullanımı sonucunda döngü sayıları ve işlemlerin tekrarlanma sayıları azalmıştır. Dolayısıyla bu literatür çalışmasındaki işlem süreleri daha kısadır. Kullanılan değişkenlerin bit sayısının azalması döngü sayılarının ve işlemlerin tekrarlanma sayılarının artmasına neden olmaktadır. Tez çalışmasındaki uygulamalarda işlem sürelerinin daha uzun olmasının bu durumdan kaynaklandığı düşünülmektedir.

Yan ve Shi (2006) sonlu ikili cisimlerde tanımlı, grup mertebesi maksimum 163 bit olan eliptik eğrilerde nokta çarpım algoritmasının yazılım uygulamasını program hafızasının 12 kilobayt, veri depolama hafızasının 4 kilobayt olduğu 8 bit ATmega128 mikroişlemcisinde çalıştırmıştır. Yazılım uygulamasında C dili kullanılmıştır. Kumar ve Paar (2006) çalışma frekansı 13.56 MHz olan, işlem sürelerinin Modelsim simülatöründe ölçüldüğü, Java tabanlı iki tip eliptik eğri kriptografisi işlemcisi tasarlamıştır. Eliptik eğri kriptografisi işlemcileri ASIC tasarımlardır. Eliptik eğri kriptografisi işlemcisinin birinde yazmaç sayısı daha fazladır, bir başka deyişle ekstra yazmaç kullanımı söz konusudur. Bu eliptik eğri kriptografisi işlemcilerinde işlemler bit

bit yapılmaktadır. Kumar ve Paar (2006) tasarladıkları eliptik eğri kriptografisi işlemcilerinde sonlu ikili cisimlerde tanımlı eliptik eğrilerde nokta çarpım işlemini uygulamıştır. Nokta çarpım işlemi afin koordinatlarda ve Montgomery algoritması ile yapılmaktadır. Literatürde yer alan çalışmalarda ve tez çalışmasında sonlu ikili cisimlerde tanımlı eliptik eğrilerde nokta çarpım işleminin yazılım uygulamasının sonuçları çizelge 5.5'te verilmektedir. Birim işlem süresi için saniye, bellek kullanımı için bayttır.

Çizelge 5.5 Sonlu ikili cisimlerde tanımlı eliptik eğrilerde nokta çarpım işleminin farklı platformlardaki yazılım başarım sonuçları

Çalışma	Platform		Bit sayısı	İşlem süresi	Bellek kullanımı
Yan ve Shi (2006)	8 bit ATmega128 C dili		163	13,9	12.412
			193	----	----
Kumar ve Paar (2006)	Eliptik eğri kriptografisi işlemcisi Java dili	Ekstra yazmaç var	163	0,0279	10.718
			193	0,0388	12.686
		Ekstra yazmaç yok	163	0,0318	9.633
			193	0,0417	11.401
Tez	16 bit CodeBlocks C dili		163	0,103	12.096

Sonlu ikili cisimlerde tanımlı, grup mertebesi maksimum 163 bit olan eliptik eğrilerde nokta çarpım işleminin tezde yapılan yazılım uygulamasının başarımı Yan ve Shi (2006) tarafından yapılan yazılım uygulamasına göre daha yüksektir. Tez kapsamındaki yazılım uygulamasında 16 bit değişkenler ile çalışılması sonucunda algoritmada yürütülen döngülerin ve tekrarlı işlemlerin sayısı azalmıştır. Bu durumun işlem süresi bakımından avantaj sağladığı düşünülmektedir.

Kumar ve Paar (2006) tarafından tasarlanan eliptik eğri kriptografisi işlemcilerinde çalıştırılan sonlu ikili cisimlerde tanımlı eliptik eğrilerde nokta çarpım algoritmasında tez çalışmasındaki yazılım uygulamasına kıyasla işlem süresi daha kısa ve bellek kullanımı daha azdır. Eliptik eğri kriptografisi mikroişlemcilerinde gerçekleştirilen

uygulama bir donanım uygulamasıdır. Nokta çarpım işlemi yazmaçlar kullanılarak yapılmaktadır. Sonlu ikili cisimlerde tanımlı eliptik eğrilerde nokta çarpım işleminin donanım uygulamasının başarımı tez kapsamındaki yazılım uygulamasına göre daha yüksektir. Yazmaç kullanımı ile daha az sayıda aritmetik çarpma işlemi yapılmaktadır. Ekstra yazmaçların varlığı ile bellek kullanımı artmakta, işlem süresi kısalmaktadır. Tez kapsamındaki yazılım uygulamasında yazmaç kullanılmaması, aritmetik çarpma işleminin daha fazla yapılması nedeniyle işlem sürelerinin daha uzun olduğu düşünülmektedir.

Literatürde 128 bit CLEFIA blok şifresinin 16 bit mikrodenetleyicide yapılan herhangi bir yazılım uygulamasına rastlanmamıştır. Tez çalışmalarında CLEFIA blok şifresinde ara anahtar üretilmesine yönelik yapılan yazılım uygulaması sonuçlarında kodun 8.752 bayt bellek tükettiği, 4 milisaniyede yürütüldüğü ve veri işleme hacminin 32.000 bit/s olduğu bulunmuştur.

Literatürde hafif sıklet bir doğrulama mekanizması olan ALIKE için 16 bit mikrodenetleyicide gerçekleştirilen yazılım uygulamaları ile karşılaşılmamıştır. Tez kapsamında 16 bit mikroişlemcilerde kullanım amacıyla yazılım yoluyla gerçeklenen ALIKE anahtar üretme adımının uygulama sonuçlarına göre RAM/ROM bayt sayısı 7.743 bayt, işlem süresi 0,042 saniye ve veri işleme hacmi 30.476 bit/s olarak elde edilmiştir.

Literatürde 163 bit ve 193 bit versiyonları bulunan ELLI tek taraflı doğrulama algoritmasının 8 ve 16 bit mikrodenetleyicilerde gerçeklenen yazılım uygulamalarına rastlanmamıştır. ELLI algoritmasının CodeBlocks derleyicisindeki yazılım uygulamasında Arduino emülatöründekine kıyasla işlem süresi daha kısa, bellek kullanımı daha fazla olmaktadır. ELLI algoritmasının Arduino emülatöründeki yazılım uygulamasında CodeBlocks derleyicisindekine göre daha az bellek kullanılmaktadır ama işlemler daha uzun sürede yapılmaktadır. İşlem süresi ile bellek tüketimi arasında bir ödünleşim olduğu söylenebilir. Bir başka deyişle işlem süresinin kısalması bellek kullanımının artmasına; bellek kullanımının minimuma indirilmesi işlem süresinin uzamasına sebep olmaktadır.

Bu tez çalışması sonlu asal cisimlerde tanımlı aritmetik işlemlerin, nokta toplama, nokta aynılama ve nokta çarpım algoritmalarının; sonlu ikili cisimlerde tanımlı ikili cisimlerde aritmetik işlemlerin, nokta toplama, nokta aynılama ve nokta çarpım algoritmalarının; ISO/IEC standardında tanımlı hafif sıklet asimetrik doğrulama algoritmalarının yazılımda uygulanabildiğini göstermiştir. Tez kapsamındaki yazılım uygulamalarının bir çoğunda işlem süreleri literatürdeki yazılım uygulamalarına göre daha uzundur. CodeBlocks derleyicisinin işlemciyi fazla meşgul eden bir derleyici olduğu düşünülmektedir. Tez çalışmasında sonlu cisim işlemleri ve eliptik eğri işlemleri C dili ile kodlanarak yazılımda gerçeklenmiştir. Tamsayı, dizi, gösterge tipinde değişkenler kullanılmış; sonlu cisim işlemleri ve eliptik eğri işlemlerinin uygulanması için değer döndüren parametre alan fonksiyonlar tanımlanmıştır. Yazılım başarımını birden çok faktör etkilemektedir. Bu faktörlerden bazıları mikroişlemci mimarisi, mikroişlemcinin çalışma frekansı, değişkenlerin bit sayısı, yazılım uygulamasının çalıştırıldığı derleyici/emülatör, yazılım dilidir. Kısıtlı cihazlarda tek yönlü asimetrik doğrulama algoritmalarını hazır bir kütüphane üzerinden uygulamak ve bu algoritmaların yazılım uygulamalarının başarımını iyileştirmek ileride bir çalışma konusu olarak değerlendirilebilir.

KAYNAKLAR

Afacan, E. 2017. Kriptografiye Giriş Şifreleme Teorisi. Epos, 111, Ankara.

Agarwal, S., Saha, S., Paul, R. ve Chakrabarti, A. 2014. Performance Evaluation of ECC in Single and Multi Processor Architectures on FPGA Based Embedded System, In: Elsevier Science and Technology. Venugopal, K.R., Patnaik, L.M. (eds), Elseiver Publications, 140-147, Bangalore.

Altan, K., Kaşkaloğlu, K., Kındap, N., Özakın, Ç., Saygı, Z., Yıldırım, E., Yıldırım, M., ve Yıldız, S. 2004. Kriptografiye Giriş Ders Notları, Orta Doğu Teknik Üniversitesi, Uygulamalı Matematik Enstitüsü, Kriptografi Anabilim Dalı, 141, Ankara.

Anonymous. 2008. Web Sitesi: https://www.slideshare.net/ErdenerUyanPhd/kriptolojinin-temelleri-ve-elektronik-mza-altyaps-26122008?from_action=save. Erişim Tarihi: 22.01.2021.

Anonymous. 2009. National Institute of Standards and Technology, Digital Signature Standard (DSS). FIBS Publication 186-3, 131 s., Gaithersburg.

Anonymous. 2012. ISO/IEC 29192-2: 2012, Information technology - Security techniques – Lightweight Cryptography – Part 2: Block Ciphers. BSI Standards Publication, 41 s., Switzerland.

Anonymous. 2013. ISO/IEC 29192-4: 2013, Information technology-Security techniques–Lightweight Cryptography – Part 4: Mechanism using asymmetric techniques. BSI Standards Publication, 42 s., Switzerland.

Anonymous. 2019a. Web Sitesi: https://www.arduino.cc/en/Guide/Introduction, Erişim Tarihi: 26.06.2019.

Anonymous. 2019b. Arduino Uno R3 Datasheet. Web Sitesi: https://www.fecegypt.com/uploads/dataSheet/1522237550_arduino%20uno%20r3.pdfErişim Tarihi: 26.06.2019.

Anonymous. 2019c. Web Sitesi: https://www.farnell.com/datasheets/1682209.pdf, Erişim Tarihi: 26.06.2019.

Anonymous. 2019d. Web Sitesi: https://www.mekatronikegitim.com/arduino_uno_r3_ozellikleri/, Erişim Tarihi: 26.06.2019.

Anonymous. 2019e. ATmega8U2 Datasheet. Web Sitesi: https://pdf1.alldatasheet.com/datasheet_pdf/view/313553/ATMEL/ATMEGA8U2.html, Erişim Tarihi: 26.06.2019.

Anonymous. 2020. Web Sitesi:
https://tr.wikipedia.org/wiki/Feistel_şifresi. Erişim T arihi: 25.01.2021.

Batina, L., Guajardo, J., Kerins, N., Tuyls, P. ve Verbauwhede, I. 2007. Public-Key Cryptography for RFID-Tags. Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 19-23 March 2007, White Plains, NY, U.S.A.

Benissa, M. ve Good, T. 2008. Hardware Performance of e-Stream Phase-Ⅲ Stream Cipher Candidates. The State of the Art of Stream Ciphers, 13-14 February, Lausanne/Switzerland.

Bock, H., Braun, M., Dichtl, M., Hess, E., Heyszl, J., Kargl, W., Koroschetz, H., Meyer, B. ve Seuschek, H. 2008. A Milestone Towards RFID Products Offering Asymmetric Authentication Based On Elliptik Curve Cryptography, January 2008.

Bozkurt, Ö. Ö. 2005. Eliptik Eğri Şifreleme Kullanarak Güvenli Soket Katmanı Protokolü' nün Gerçeklenmesi ve Performansının Değerlendirilmesi. Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 106, İstanbul.

Braun, M., Hess, E., ve Meyer, B. 2008. Using Elliptic Curves on RFID Tags. International Journal of Computer Science and Network Security, Vol 8, February 2008, Munich, Germany.

Brown, M, Hankerson D., Lopez J., Menezes A. 2001. Software Implementation of the NIST Elliptic Curves Over Prime Fields. In: Naccache D. (eds) Topics in Cryptology — CT-RSA 2001. CT-RSA 2001. Lecture Notes in Computer Science, vol 2020. Springer, Berlin, Heidelberg.

Budak, B. 2010. Güvenli Özet Algoritması. Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, 103, Ankara.

Chang, S., Gupta, V., ve Gupta, S. 2002. Performance Analysis of Elliptic Curve Cryptography for SSL, Conference WISE '02 Proceedings of the 1st ACM Workshop on Wireless Security, 87-94, Atlanta / USA.

Coron, J.S., Gouget, A., Paillier, P. and Villegas, K. 2010. SPAKE: a Single-party Public-key Authenticated Key Exchange Protocol for Contact-less Applications, In: Financial Cryptography and Data Security. Sion, R., Curtmola, R., Dietrich, S., Kiayias, A., Miret, J.M., Sako, K. ve Sebé, F. (eds), 107-122, Berlin.

Çetin, Ö. 2006. Eliptik Eğri Kriptografisi. Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, 96, Ankara.

Değirmenci, F. 2006. Eliptik Eğriler ile Sayısal İmza. Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, 76, Ankara.

Ertan, O., Kılıç , A. ve Yıldız, H. 2018. Lightweight Cryprography in 5G Machine-Type Communication, In: Networks of The Future Architectures, Technologies, and Implementations. Elhokdr, M., Hassan, Q.F. ve Shahrestani, S. (eds), CRC Press, 127-131, Florida.

Engin, E. 2017. İkinci Dereceden Çok Değişkenli Polinom Sistemlerine Dayanan İmzalama Algoritmalarının Bileşenlerinin Analizi ve Açık Kaynak Kodlu Uygulamaları. Yüksek Lisans Tezi, On Dokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 31, Samsun.

Gülen, U. 2014. Implementing Elliptic Curve Cryptography for Wireless Sensor Networks. Yüksek Lisans Tezi, Bahçeşehir Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 39, İstanbul.

Hankerson, D., Menezes, A. ve Vanstone, S. 2003. Guide to Elliptic Curve Cryptography. Springer, 382, USA.

Karri, R., Kuznetsov, G., ve Goessel, M. 2003. Parity-Based Concurrent Error Detection of Substitution-Permutation Network Block Ciphers, In: Lecture Notes in Computer Science 2779, Walter, C. D. (eds), 113-124, Springer-Verlag, Berlin.

Kodali, R. K., Kumar, S., Jain, V. ve Lakshmi, B. 2015. Hybrid Multiplier for ECC. 2015 IEEE Recent Advances in Intelligent Computational Systems, 10-12 December 2015, Trivandrum/India.

Koyutürk, R. 2020. Çok Değişkenli Polinom Sistemlerine Dayanan Kuantum Sonrası Güvenilir Şifreleme Sistemleri ve Açık Kaynak Kodlu Uygulamaları. Yüksek Lisans Tezi, Ege Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Anabilim Dalı, 92, İzmir.

Kumar, S. S. ve Paar, C. 2006. Are Standards Compliant Elliptic Curve Cryptosystems Feasible on RFID? Workshop on RFID Security, 2006, Germany.

Kurt , M. 2012. Eliptik Eğri Şifreleme Algoritmasının Uygulaması ve Analizi. Yüksek Lisans Tezi, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 96, Edirne.

Liu, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E. ve Culler, D. 2005. TinyOS: An Operating System for Sensor Networks, In: Ambient Intelligence. Weber, W., Rabaey, J. M. ve Aarts, E. (eds), Springer, 115-148, Berlin.

Lopez, J., Dahab, R. 1999. Fast Multiplication on Elliptic Curves Over $GF(2^m)$ without precomputation. In: Koç Ç.K., Paar C. (eds) Cryptographic Hardware and Embedded Systems. CHES 1999. Lecture Notes in Computer Science, vol 1717, pages 316-327. Springer, Berlin, Heidelberg.

Madden, S., Polastre, J., Szewczyk, R., ve Gay, D. 2005. TinyOS: An Operating System for Sensor Networks, In: Ambient Intelligence, 115-148, Springer-Verlag, Berlin.

Malik, M.Y. 2010. Efficient Implementation ofElliptic Curve Cryptography Using Low-power Digital Signal Processor. IEEE The 12th International Conference on Advanced Communication Technology (ICACT), 7-10 February 2010, Gangwon, Korea.

Mano, M. 1992. Computer System Architecture.

McKay, K. A., Bassham, L., Sönmez Turan, M. ve Moua, N. 2017. Report on Lightweight Cryptography. National Institute of Standards and Technology, US Department of Commerce, March 2017, Gaithersburg/USA.

Menezes, A., Van Oorscott, P., and Vanstone, S. A. 1996. Handbook of Applied Cryptography. CRC Press, 816, Florida.

Orlando, G., and Paar, C. 2001. A Scalable GF(p) Elliptic Curve Processor Architecture for Programmable Hardware. In C. K. Koc, D. Naccache , and C. Paar, editors, Workshop on Cryptographic Hardware and Embedded Systems - CHES 2001, volume LNCS 2162, pages 348-363. Springer-Verlag, May 14-16, 2001.

Özcan, A. B. 2006. Performance Analysis of Elliptic Curve Multiplication Algorithms for Elliptic Curve Cryptography. Master Thesis, Middle East Technical University, Graduate School of Natural and Applied Sciences, Electrical and Electronics Engineering Department, 98, Ankara.

Paar, C., Pelzl, J. 2009. Understanding Cryptography A Textbook for Students and Practitioners. Springer, 382, Berlin.

Panda, A.K., Rajput, P ve Shukla, B. 2012. FPGA Implementation of 8, 16 and 32 Bit LFSR with Maximum Length Feedback Polynomial using VHDL, 2012 International Conference on Communication Systems and Network Technologies, 11-13 May 2012, Rajkot, India.

Puttman, C., Shokrollahi, J., Porrmann, M. ve Ruckert, U. 2008. Hardware Accelerators for Elliptic Curve Cryptography. Advances in Radio Science, 6, 259-264.

Qiu, Q. ve Xiong, Q. 2003. Research on Elliptic Curve Cryptography. The 8th International Conference on Computer Supported Cooperative Work in Design Proceedings, 26-28 May 2004, Xiamen/China.

Renes, J., Costello, C. , ve Batina, L. 2016. Complete addition formulas for prime order elliptic curves, Annual International Conference on the Theory and Applications of Cryptographic Techniques, 403-428, 28 April 2016, Berlin.

Sarıtaş, H. 2010. Dijital İmza Uygulamasının Eliptik Eğri Şifreleme Yöntemi Kullanılarak Gerçekleştirilmesi. Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik-Bilgisayar Eğitimi Anabilim Dalı, 96, İstanbul.

Schneier, B. 1996. Applied Cryptography, Second Edition : Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc., 1027, New Jersey.

Singh, S. R., Khan, A. K. ve Singh, S. R. 2016. Performance Evaluation of RSA and Elliptic Curve Cryptography. 2016 2nd International Conference on Contemporary Computing and Informatics. 14-17 December 2016, Noida/India.

Soysaldı, M. 2018. Çok Değişkenli Polinom Sistemlerine Dayalı Kuantum Bilgisayarlar Sonrası Güvenilir Yeni Kimlik Doğrulama ve İmzalama Şemaları. Yüksek Lisans Tezi, On Dokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 66, Samsun.

Stallings, W. 2017. Cryptography and Network Security Principles and Practice. Pearson, 767, Malezya.

Suo, H., Wan, J. Zou, C. and Liu, J. 2012. Security in the Internet of Things. 2012 International Conference on Computer Science and Electronics Engineering, 23-25 March 2012, IEEE, Hangzhou/China.

Weimerskirch, A., Paar, C. and Shantz, S.C. 2001. Elliptic Curve Cryptography on a Palm OS Device. 6th Australasian Conference on Information Security and Privacy, 11-13 July 2001, Macquarie University, Sydney/Australia.

Wenger, E., and Werner, M. 2006. Evaluating 16-bit Processors for Elliptic Curve Cryptography. In: Smart Card Research and Advanced Applications, volume LNCS 7079, pages 166-181. Springer, Berlin, Heidelberg.

Yan, H., and Shi, Z.J. 2006. Studying Software Implemantations of Elliptic Curve Cryptography. Third International Conference on Information Technology: New Generations, 10-12 April 2006, IEEE, Las Vegas/USA.

Yavuz, İ. 2008. Eliptik Eğri Kriptosisteminin FPGA Üzerinde Gerçeklenmesi. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, 92, İstanbul.

Yavuzer Aslan, F. 2012. Blok Şifrelerde Kullanılan Doğrusal Dönüşüm Yapılarının İncelenmesi. Yüksek Lisans Tezi, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 86, Edirne.