T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

GÜVENLİ UYGULAMALAR İÇİN BİYOMETRİK TABANLI ŞİFRELEME ANAHTARI ÜRETİMİ

Samet ÖZTOPRAK

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

DANIŞMAN
Prof. Dr. Ahmet SERTBAŞ

II. DANIŞMAN
Yrd. Doç. Dr. Muhammed Ali AYDIN

Haziran, 2017

İSTANBUL

Bu çalışma 8.06.2017 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi:

Prof. Dr. Ahmet SERTBAŞ(Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi

Yrd. Doç. Dr. Muhammed Ali AYDIN
İstanbul Üniversitesi
Mühendislik Fakültesi

Doç. Dr. Rüya ŞAMLI
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Fırat KAÇAR
İstanbul Üniversitesi
Mühendislik Fakültesi

Yrd. Doç. Dr. Akhan AKBULUT
Kültür Üniversitesi
Mühendislik Fakültesi

20.04.2016 tarihli resmi gazetede yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Bu tez çalışmasında, giderek her alanda önem kazanan güvenlik konuları, özellikle sanal dünyada hayati bir öneme ulaşmış durumda, günümüzde artık devletler siber saldırıları yapan ülkeler ya da kurumlar ile sıcak savaş yapacak şekilde anayasal düzenlemelerin yapıldığı bir süreçte yaşamaktayız. Böyle bir süreçte biz de insanların üzerinde bulunan ve tekrar tekrar hatırlama ihtiyacı gerektirmeyecek biyometrik şifreleme alanında bu araştırmamı geliştirmem gerektirdiğini düşündüm ve buna göre tezimi hazırladım.

Öncelikle tez konusunu seçerken isteklerimi göz önünde bulundurup bana yardımcı olan tez danışmanım Prof. Dr. Ahmet SERTBAŞ ve Yrd. Doç. Dr. Muhammed Ali AYDIN'a teşekkürlerimi sunarım. Muhammed Ali hocam her mail gönderdiğim ve telefonla aradığımda bana karşılık vermeyi esirgemedi ve aynı zamanda yaptığım çalışmalarda daima yanımda olduğu için çok teşekkür ediyorum. Tüm eğitim hayatım boyunca benden maddi ve manevi desteklerini esirgemeyen her zaman yanımda olan sevgili aileme teşekkürlerimi bir borç bilirim.

Haziran 2017 Samet ÖZTOPRAK

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

Φ : Totient

∑ : Toplamsal

Kısaltmalar **Açıklama**

AES : Advanced Encryption Standard

c **:** Kod

d : Şifreli veri açma anahtarı

DES : Data Encryption Standard

DSS : Digital Signature Standard

D_n : Şifreli metni açma fonksiyonu

E_n : Şifreleme fonksiyonu

e : Şifreleme anahtarı

e_{pseudo} : Sahte açık anahtar

f_{value} **:** Parmak izi değeri

h : Yükseklik

n **:** Seçilen iki asal sayının çarpımı

NIST : National Institute of Standards and Technology

p **:** 1. Asal sayı

q **:** 2. Asal sayı

RSA : Rivest , Shamir, Adlemon

w : Genişlik

3DES : Triple Data Encryption Algorithm

ÖZET

GÜVENLİ UYGULAMALAR İÇİN BİYOMETRİK TABANLI ŞİFRELEME ANAHTARI ÜRETİMİ

YÜKSEK LİSANS TEZİ

Samet ÖZTOPRAK

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Ahmet SERTBAŞ

II. Danışman : Yrd. Doç. Dr. Muhammed Ali AYDIN

Giderek artan internet kullanımı yüzünden daha fazla sayıda kişi internet dolandırıcılığı olaylarına her gün maruz kalmaktadır. Kullanıcılar tarafından tanımlanan şifreler genellikle zayıf, kırılabilir ve tahmin edilebilir özelliktedir. Bu durumun üstesinden gelmenin en iyi yolu biyometrik şifreleme sistemleri kullanmak olarak görülmektedir. Biyometrik parametre tabanlı sistem temel olarak iki ana konu için çözüm önermektedir. Bu sorunlardan biri, şifreyi unutmak ya da kaybetmek olasılığıdır ki biyometrik sistemlerin doğası gereği böyle bir sorun söz konusu değildir. İkinci olarak, biyometrik parametre tarafından oluşturulan anahtarın benzersiz oluşudur. Bu amaçla, parmak izi bir matrise dönüştürülür ve bu matristen istenilen benzersiz bir anahtar üretilir. Bu benzersiz anahtarın yardımıyla açık anahtar üretilir. Bu tez çalışmasında, biyometrik parametreyi temel alan ve böylece açık anahtar üreterek RSA algoritmasının daha güçlü hale getirilmesini sağlayan bir algoritma önerilmiştir. Çalışmamızda kullanılan matematiksel yaklaşımın basitliği nedeniyle önerilen yöntem biyometrik parametreden tekil anahtarı üretmek için çok etkili ve hızlı bir yoldur. Oluşturulan bu yeni yöntemin güvenli olduğu RSA üzerinde yapılan kriptoanaliz yöntemleri üzerinden gösterilmiştir.

Haziran 2017, 55 sayfa.

Anahtar kelimeler: Asimetrik şifreleme, RSA, Biyometrik parametre, Parmak izi, Açık anahtar dağılımı

SUMMARY

BIOMETRIC BASED CRYTOGRAPHIC KEY GENERATION FOR SECURE APPLICATIONS

M.Sc. THESIS

Samet ÖZTOPRAK

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Computer Engineering

Supervisor : Prof. Dr. Ahmet SERTBAŞ

Co-Supervisor : Assist. Prof. Dr. Muhammed Ali AYDIN

More and more people are exposed to internet fraud every day because of the increasing use of the internet. The passwords defined by the users are generally weak, fragile and predictable. The best way to overcome this situation is to use biometric encryption systems. The biometric parameter based encryption systems basically suggest solutions for two main issues. One of these problems is the possibility of forgetting or losing the password, which is not the nature of biometric systems. Secondly, the uniqueness of the key generated by the biometric parameter. For this purpose, the fingerprint is transformed into a matrix, and a unique key is obtained from this matrix. The public key is generated by helping of this unique key. In this thesis study, an algorithm is proposed which is based on the biometric parameter, thus generating a public key, which makes the RSA algorithm stronger. Because of the simplicity of the mathematical approach used in our work, the proposed method is a very effective and rapid way to generate a unique key from a biometric parameter. Cryptoanalysis methods on RSA show that this new method is safer.

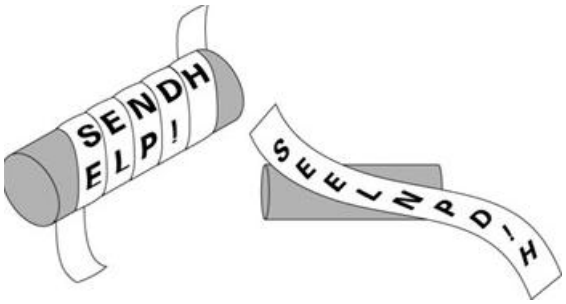
June 2017, 55 pages.

Keywords: Public cryptography, RSA, Biometric parameters, Fingerprint, Public key distribution

1. GİRİŞ

Bilinen en eski şifrelemenin ya da diğer bir deyişle bilgi korumanın tarihini uzmanlar M.Ö 2000 yılına kadar götürebilmektedir. Antik Mısırlıların, dini bir metni anlatan hiyerogliflerin bazı yerlerinde bazı işaretler kullandığı görülmüştür. İlginç olan ise bu sembollerin metnin farklı bölgelerinde işlenmiş olmasıdır. Bu yöntemi Mısırlılardan sonra Mezopotamyalılar ve İbraniler de kullanmışlardır. Bugün kullandığımız cryptography kelimesi Yunanca'da sırasıyla gizli ve yazı anlamlarına gelen kryptos ve graphein kelimelerinin birleşimi ile meydana gelmiştir.

Tarihte en ilgi çeken şifreleme sistemlerinden biri olarak, M.Ö. 7. yüzyılda Ispartalılar tarafından geliştirilen "scytale" isimli şifreleme sistemi gösterilebilmektedir. Bu sistemde, uzun bir şerit üzerine harfler yazılarak ve daha sonrasında bu şerit bir silindir üzerine sarılarak bir mesaj şifrelenmiştir. Sarılan bu metin açıldığında mesaj ortaya çıkmaktadır eğer sarılan çubuğun çapı sarıldığı silindirden daha büyük ya da daha küçükse anlamsız bir metin oraya çıkacaktır

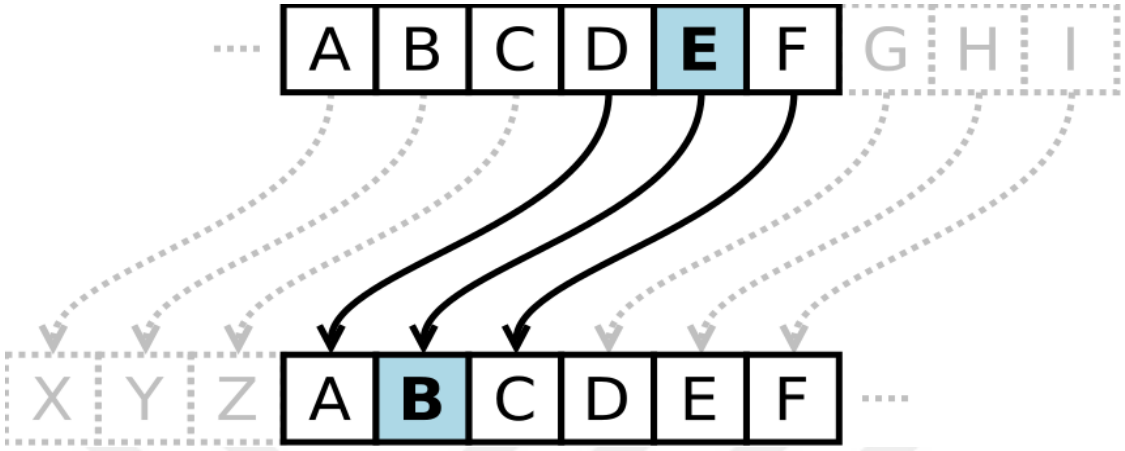


Şekil 1.1: Spartalılar tarafından geliştirilmiş olan scytale.

Tarihte yerini alan diğer bir önemli şifreleme algoritması da yaklaşık 2000 yıl önce Julius Ceasar (Sezar) tarafından Galya savaşlarında, kendi adıyla anılan şifreleme yönteminin kullanılmasıdır. Sözkonusu yöntem her harfin kendinden sonra gelen üçüncü harf ile değiştirilmesine dayandırılmıştır. Yöntemin geliştirilmesi ile monoalfabetik yerine koyma şifre algoritması ortaya konulmuştur.

Tarihte şifrelemeyi açıklamak için verilen en güzel örneklerden biride muhtemelen Kaydırma şifreleme yöntemidir. Bu yöntemde bütün harflerin belli bir sıraya göre dizildiğini düşünelim.

Bu sıralamada açık metindeki bir harf sıralamada belirtilen n adım ilerideki harf ile yer değiştirir bu şekilde şifreli metin elde edilmiş olur.



Şekil 1.2: Sezar kaydırma şeması.

Kaydırmalı şifreleme yönteminin genel şeması Şekil 1.2'de görülmektedir. Aşağıda gösterilen formüller ise kaydırmalı şifreleme yöntemi genel formülleridir.

$$E_n(x) = (x + n) mod\ 26 \quad (1)$$

$$D_n(x) = (x - n) mod\ 26 \quad (2)$$

Tablo 1.1: Sezar kaydırma şifrelemesi.

Açık	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Şifreli	QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD

Tablo 1.1'de Sezar kaymalı şifreleme algoritması için bir örnek uygulama verilmiştir. Burada İngilizce bir metin kullanılmıştır ve bu İngilizce metindeki her harfi sağa 3 adım ilerleterek şifreli metni oluşturulmuştur.

Kriptoanaliz, şifreleme algoritması güvenliğinin analizi ve zayıf yönlerini belirlenmesi işlemi olarak tanımlanmaktadır. Bu analiz, şifreleme algoritmalarının eksik taraflarının ortaya çıkarılması ve bunu önleyici tedbirler alınmasında kullanılmaktadır. Kriptoanaliz yöntemleri şifreleme yöntemlerine göre farklılık göstermekle birlikte, kaba kuvvet kriptoanaliz yöntemi tüm şifreleme algoritmalarında kullanılabilmektedir. Bu yöntem herhangi bir kullanıcının rastgele bir şekilde deneme yapması esasına dayanır. Kaba kuvvet saldırısı tüm şifreleme

yöntemlerinde olabilen, tüm şifreleme yöntemlerinin açık yanıdır. Sezar şifreleme yönteminden yine örnek verecek olursak, İngilizce'de 26 tane harf bulunduğundan, bu yöntemde şifrelenmiş metinde 26 olasılık denendiğinde şifreli metin ortaya çıkmaktadır. Diğer bir kriptoanaliz yöntemi olan sıklıkla kullanılan basit kelimeler saldırısı yöntemini kullanmak için öncelikle şifreli metnin yazıldığı dili bilmek gerekmektedir. Örneğin bu dil İngilizce ise, İngilizcede sıklıkla kullanılan "the, of, and, a, to, in, is, be" gibi kelimeler şifreli metnin kelime grupları içinden çıkarılarak şifreli metin bulunmuş olur.

Bu çalışmada, internet güvenliği için biyometrik parametrelerden oluşturulan tekil numarayı kullanarak şifreleme anahtarı oluşturulması hedeflenmiştir. Bilindiği gibi, giderek artan internet kullanımı ve artık banka işlemlerinden fatura ödeme ve diğer gündelik alış-veriş işlerimize kadar her şeyin internetten yapılabilir olması kötü niyetli bilgi güvenliği uzmanlarını teşvik etmektedir. Bu da çok ciddi güvenlik önlemlerinin alınmasını zorunlu kılmaktadır. Bu nedenle dünya üzerinde milyonlarca internet dolandırıcılığı vakasının önüne geçebilmek için yeni çözümler aranmaktadır. Gelecekte bunun en iyi çözümü olarak biyometrik parametreleri kullanmak olduğu görünmektedir. Dünyada biyometrik parametrelerin kullanımının yaygınlaşması hem unutulması ya da kaybedilme ihtimali olmayan nesneler olması hem de değişimi zor ya da bazılarının değişimi imkansız (DNA vb) kişiye özel değerler olmasından kaynaklanmaktadır ki bu özellikler onları güçlü bir güvenlik anahtarı haline getirmektedir. Bu kapsamda, çalışmamızda biyometrik çalışmalarda en güvenli parametrelerden biri olan parmak izi parametresi kullanılmıştır. Diğer yandan, bilinen en güvenilir biyometrik parametre olan DNA günümüzün bilgisayarları tarafından hızlı bir şekilde çözülemediğinden gelecekte yapılacak çalışma olarak görülmektedir.

Sonuç olarak, bu çalışmamızda güvenlik sorununa basit ve güvenilir bir çözüm sunulmuştur. Bu tezde, kullanılan matris gösterimi başka bir biyometrik parametre için de uygulanabilmektedir. Bu da geliştirdiğimiz matematiksel yöntemimizin kullanılabilirliğini arttırmaktadır. Özetle, bu çalışmada, RSA algoritması üzerine biyometrik parametre ekleyerek gönderilecek açık anahtarın biyometrik parametreye bağlı olarak gönderilmesi sağlanmıştır. Bu şekilde olan açık anahtarın daha güvenlikli olduğu kriptoanaliz yapılarak gösterilmiştir.

Bu çalışma toplamda beş bölümden oluşmaktadır. İlk bölüm giriş bölümü olup, bu bölümde şifrelemenin tarihi hakkında ve Sezar şifreleme algoritması hakkında bilgi verilmiştir.

İkinci bölümde literatür çalışmamızın yanı sıra, günümüzde kullanılan şifreleme yöntemleri ele alınarak bu yöntemlerinin kısa bir açıklaması yapılmıştır. Tarihte görülen ilk şifreleme sistemi olan simetrik şifreleme yöntemi ile yeni yaklaşım olan asimetrik şifreleme yöntemlerini karşılaştırması yapılmıştır. Çalışmada biyometrik parametrelerin karşılaştırılması yapılarak çalışma için uygun biyometrik parametre belirlenmiştir. RSA şifreleme sistemi hakkında genel bilgi verilmiştir.

Üçüncü bölümde temel alınan çalışmadaki temel alınan çalışmanın akış diyagramı tanıtılarak akabinde bunun üzerine yapılan çalışmamız tanıtılmıştır.

Dördüncü bölümde yapılan bu çalışma ve örnek alınan çalışmada açık anahtar üretmek için kullanılan tekil anahtar performansı ve uzunluğu karşılaştırılmıştır. Bu karşılaştırmanın sonuçları grafiksel olarak sunulmuştur.

2. GENEL KISIMLAR

2.1. LİTERATÜR TARAMASI

Biyometrik tanımlı bir şifreleme yapmak için parmak izini kullanmıştır, parmak tanımlama için kullanılan temel yöntemleri kullanarak 2 aşamalı parmak tanımlama sistemi üzerine çalışmıştır. Server üzerinde bu 2 aşamalı parmak izi verilerini kullanarak parmak izi verilerinin doğrulanma oranını yükseltmiştir. Parmak izinde bu çalışmada ve incelenen diğer makalelerde olduğu gibi, çok şekilde mesai harcanması bizimde çalışmamızda parmak izi kullanmaya yöneltmiştir[1]. Yeni bir biyometrik şifreleme anlayışı getirmişlerdir. Şifreleme parametresi olarak parmak izi, göz, yüz ve el kullanmışlardır. Bu doğrulamayı yaparken öklit mesafe parametresini kullanarak bu biyometrik parametreler ile yüksek bir doğrulama oranı yakalamıştır.

Bu yaklaşım doğruluk oranını belirtilen bir eşik değeri ile kontrol etmektedir. Belirtilen bu eşik değerinin üzerinde değerler doğru kabul edilecektir. Bu makale de çalışmamıza benzer bir çalışmadır çünkü bu çalışmada yine anahtar üretimi biyometrik parametre üzerinden yapılmaya çalışılmıştır[2]. Her geçen gün biyometrik parametre tabanlı şifreleme algoritmalarının artmasına dikkat çekmektedir, çalışmalarında iris, parmak izi, avuç içi, ses vb. klasik biyometrik parametrelerin performans ve kullanışlılık açısında şifreleme algoritmalarında kullanılması incelenmiştir. Detaylı veriler ve grafikler bu çalışmada verilmiştir. Çalışmamızın hangi biyometrik parametre üzerinde olması gerektiği hakkında bize bilgi vermektedir[3]. Biyometrik şifrelemede kullanılan kaba-kuvvet saldırıyı en aza indirmek için matematiksel hesaplamaları kullanarak bir çözüm üretmeye çalışmıştır. Kriptoanaliz çalışmamızda bu çalışma bize ilham kaynağı olmuştur[4]. Akıllı ev sistemlerinin güvenliği ses biyometrik parametresi tabanlı geliştirilmiştir. Nesnelerin interneti kavramı her geçen gün hayatımıza daha çok girmekte ve bu bizim nesneler ile iletişim üzerinde güvenlik önlemi almamızı daha çok gerektirmektedir. Bu çalışmada güvenli iletişimin ses parametresi ile nasıl sağlandığı görülmektedir[5]. Biyometrik tabanlı bir protokol yapısı geliştirmiştir. Bu protokolde parmak izi biyometrik parametresini asimetrik şifrelemede kullanılmıştır. Bu şekilde daha güvenli bir asimetrik şifreleme yapısının ortaya çıktığı tespit edilmiştir[6]. Biyometrik parametrelerin şifreli olarak tutulması ve başkasının eline geçmemesi için RSA ile biyometrik parametreyi şifreleme yoluna gidilmiştir. Bizim çalışmamızda, biz de biyometrik

parametreleri şifreli bir şekilde veritabanında tutması ve ağ üzerinde şifreli şekilde dolaştırılması esasına dayanmaktadır[7]. Parmak izindeki detay bilgilerini kullanarak matris oluşturulmuş ve daha sonrasında bu matris açık anahtar üretiminde kullanılmıştır. Aslında bizim çalışmamızın ilhan aldığı çalışma bu çalışma olarak gösterilebilir. Biz çalışmamızda daha geniş bir biyometrik parametre yelpazesine hitap etmeyi planlayıp bu bağlamda matematiksel bir formül ile matris oluşumuna gidilmiştir[8]. Simetrik ve asimetrik şifreleme yöntemleri karşılaştırmaktadır. Bunu yaparkende simetrik şifrleme için DES, asimetrik şifreleme için ise RSA algoritmasını karşılaştırarak güçlü ve zayıf yönlerini karşılaştırmıştır[9]. Çalışmamızda temel aldığımız parmak izi biyometrik parametresinin tanımlanması üzerine yapılan bir çalışmadır[10]. RSA, DES ve AES algoritmalarının performans olarak karşılaştırılması yapılmış ve sonuçlar gösterilmiştir[11]. Biyometrik parametreyi RSA şifreleme algoritmasına uygulayarak şifreleme sisteminin güvenilirliğini arttırmıştır. Performansındaki ilerleme grafiksel olarak gösterilmiştir[12]. İris biyometrik parametresini AES şifreleme algoritmasına entregre etmeye çalışmıştır. Bu işlem sırasında iris tanımlama sırasında olan bit hatalarını nasıl yönettiğiyle ilgili olarak bilgi sunmaktadır[13]. Günlük hayatta kullanılan anahtarların tekil anahtar olmadığını ve olmayacağını bunun önüne geçmek ve her kişiye özel tekil anahtar üretmek için biyometrik parametrelerin kullanılması gerektiğini öne sürmektedir[14]. Biyometrik parametrelerinin tanımlanması üzerine bir çalışma yapmıştır. Bunlarından iris, parmak izi gibi biyometrik parametreleri kıyaslayarak tanımlanma zorluklarını ve oranlarını göstermiştir[15]. Parmak izi detay bilgisi ile şifreleme sistemine anahtar bağlama gerçekleşir. Parmak izi detay bilgisi ile tekil bir anahtar üretmek asıl hedeftir[16]. RSA üzerinde 20 yıldır yapılan kriptoanaliz yöntemlerini göstermektedir. Bu kriptoanaliz yöntemlerini ile çalışmamızı kıyaslayarak yaptığımız algoritmanın güvenlik yönünden gücü ortaya konulmuştur[17]. Biyometrik parametleri kullanarak oluşturulan şifreleme sistemlerinin geleneksel şifreleme sistemlerine üstünlüğü ele alınmıştır. Makalede biyomterik tabanlı yeni bir şifreleme yönteminin üstünlükleri anlatılmıştır[18]. Simetrik ve Asimetrik şifreleme sistemlerini güçlüklerini ve güvenlik açıklarını ele alarak bize hangi şifreleme sistemi üzerinde devam etmemiz konusunda yardımcı olmuştur[19]. Biyometrik parametrelerin güvenlik ölçüleri ele alınarak incelenmesi, özellikle dilbilimsel olarak şifreleme yaklaşımın oluştrulması işlemlerini gerçekleştirmiştir[20]. Parmak izinden her seferinde üretilen sabit bir tekil anahtar üretmeye çalışmıştır. Bu oran kullanıcılar üzerinde 97.25% olarak görülmüştür[21]. Biyometrik parametre üzerinden şifreleme anahtarı

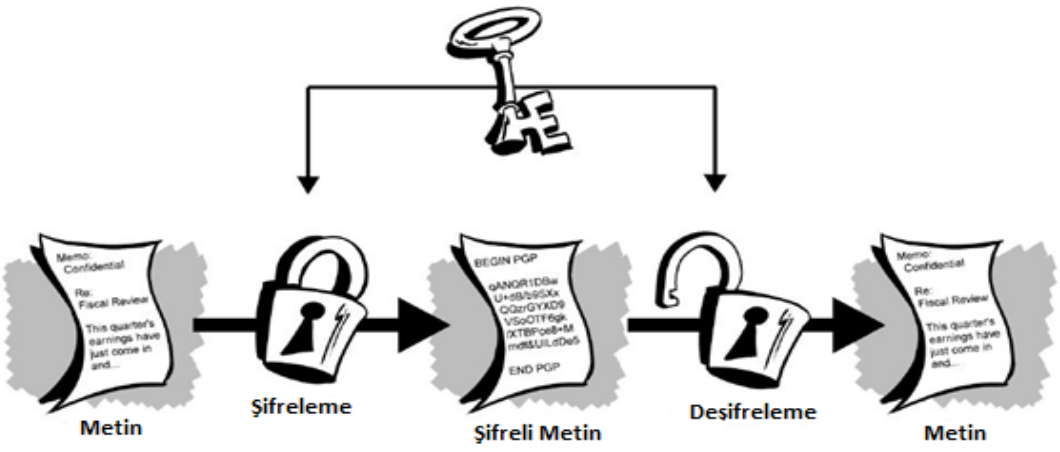
oluşturma üzerine yapılan bir çalışma fakat bu çalışmada anahtarın bir yerde tutulmasını önlemek için bir üretim şeması yaratılmıştır[22].

2.2. KULLANILAN ŞİFRELEME SİSTEMLERİ

Günümüzde kullanılan iki tip şifreleme sistemi vardır. Bunlardan biri en eskisi olan simetrik şifreleme sistemidir. Bu şifreleme sisteminde veriyi açan ve şifreleyen anahtar aynıdır. Bir diğeri de veriyi şifrelerken ve açarken farklı anahtarın kullanıldığı asimetrik şifreleme sistemidir. Bu sistemde veri şifrelenirken farklı bir anahtar kullanılır. Bu anahtar veriyi açamaz veriyi açarken de farklı bir anahtar kullanılır.

2.2.1. Simetrik Şifreleme Sistemleri

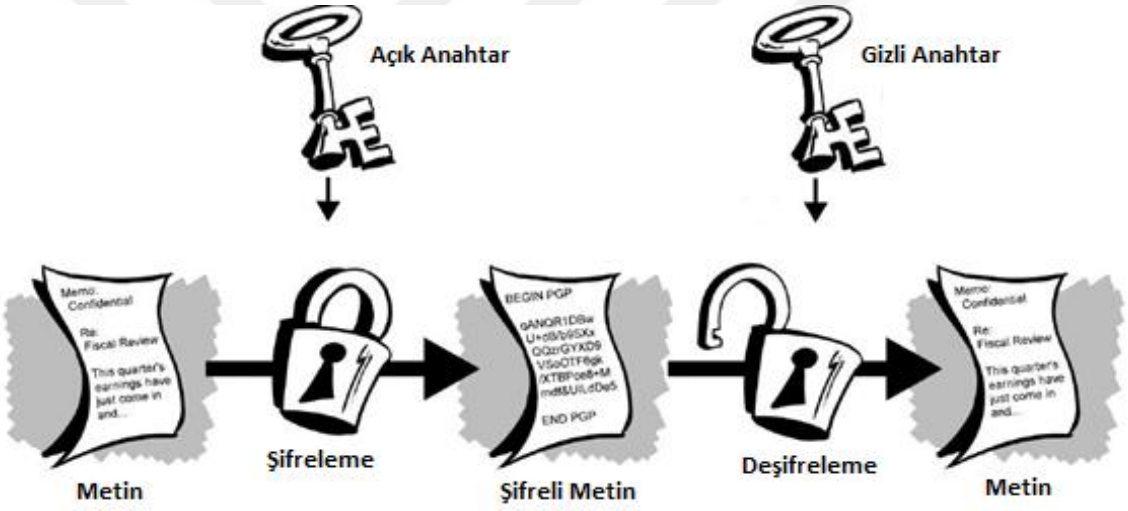
Simetrik şifreleme, adında da anlaşılacağı gibi verilen bilgiyi açmada ve şifrelemede aynı anahtarın kullanılması mantığına dayanır. Eski zamandan beri kullanılan şifreleme mantığıdır. Bu şifreleme yaklaşımında aynı anahtar kullanıldığı için oldukça karmaşık algoritma yapıları vardır. Bu şifreleme yöntemine ait sistemlere örnek olarak verilebilecek başlıca şifreleme sistemlerinden bazıları şunlar olarak gösterilebilir; Twofish, Serpent, AES (Rijndael), Blowfish, CAST5, RC4, DES şifreleme sistemleri verilebilir.



Şekil 2.1: Simetrik şifreleme sistemi.

2.2.2. Asimetrik Şifreleme Sistemleri

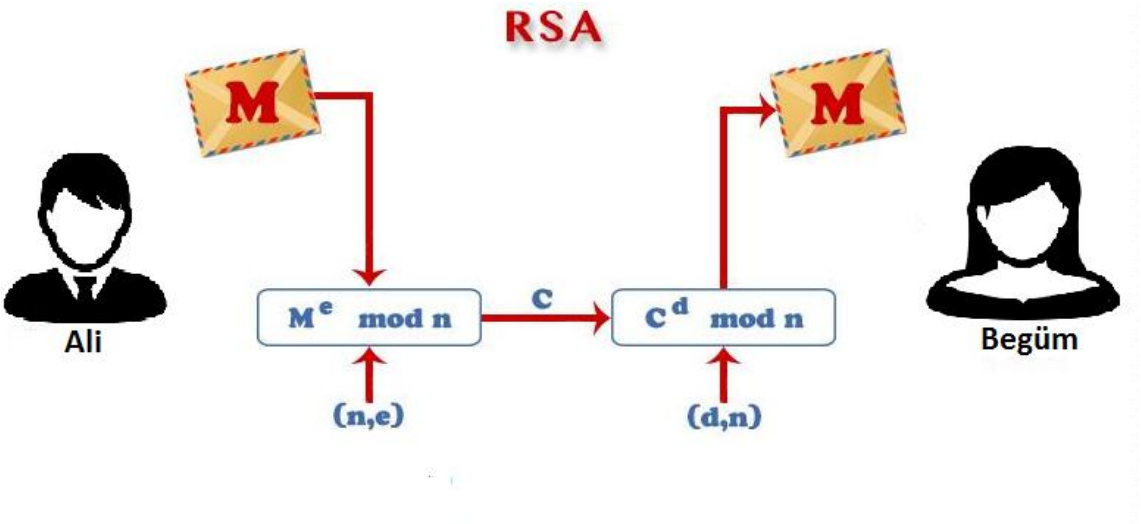
Asimetrik şifreleme, veriyi şifrelerken farklı, şifreli veriyi açarken farklı bir anahtar kullanılmasından dolayı bu şekilde isimlendirilmiştir. Diğer bir deyişle veriyi şifreleyen anahtar bu veriyi tekrar açamaz. Şifrelenen verinin tekrar açılabilmesi için şifreleme anahtarı ile birlikte üretilen anahtar kullanılmalıdır. Asimetrik algoritmalar simetrik şifreleme sistemlerine göre oldukça basittir fakat güvenlik yönünden daha üstündür. Bu şifreleme türünü ilk olarak 1977'de Ron Rivest, Adi Shamir ve Leonard Adleman adlı üç kişi tarafından bulunmuştur. Bulunan bu ilk asimetrik şifreleme sistemi adını bu üç bilim adamının soy isimlerinin baş harflerinden olacak şekilde (RSA) almıştır. Bu şifreleme yöntemine ait belli başlı şifreleme algoritmaları: Diffie–Hellman, DSS, Elliptic curve, Paillier, RSA.



Şekil 2.2: Asimetrik şifreleme sistemi.

2.2.2.1. Temel RSA Algoritması

RSA asimetrik şifrelemenin hem başlangıcı hem de en güzel örneğidir. Burada Begüm ile Ali arasında iletişim kurarken Ali'ye açık bir anahtar gönderilmektedir. Begüm ise açık anahtar ile şifrelenmiş metni kendine ait gizli bir anahtar ile açabilmektedir.



Şekil 2.3: Temel RSA algoritma yapısı.

1. Begüm şifreleme sisteminden bir anahtar çifti üretmesini ister. Veriyi şifrelemek için açık anahtar kullanılır, şifreli metni açmak için ise gizli anahtar kullanılır.
2. Ali Begüm'e bilgi göndermek için açık anahtarı kullanarak bilgilerini şifreler.
3. Ali şifreli veriyi alır ve gizli anahtarını bu şifreli veriyi açmak için kullanır.

2.2.3. RSA Şifreleme Algoritması

1. Iki adet asal sayı seçilir p ve q.
2. Hesaplama n = pq and
3. Totient hesapla $\Phi = (p-1)(q-1)$.
4. $1<e<\Phi$ ebob$(e, \Phi)=1$ hesapla.
5. d gizli anahtar hesaplanır de $=1 \bmod \Phi$.
6. (e,n) açık anahtar yaratılır.

Yukarıdaki RSA algoritması algoritmamızın temeli olan asimetrik şifreleme sistemine aittir. Buradan sonraki adımlar bu algoritma üzerinde yapılan değişiklikler hakkında olacaktır.

2.2.4. RSA Şifreleme Algoritmasının İspatı

RSA şifreleme algoritmasının matematiksel olarak yanılmayacağını ispatlanabilir. Bunu yaparken bazı teoremler kullanarak ispatı yapılabilir.

2.2.4.1. Fermat'ın Küçük Teorisi İspatı

İlk olarak Fermat'ın küçük teorisi ile RSA algoritmasının hep doğru sonuç üreteceğini ispatlayabiliriz. Bu teori Rivest, Shamir, and Adleman tarafından RSA algoritmasının ispatı için kullanılan ispat yöntemidir. Eğer p bir asal sayı ise onu hiçbir sayı bölemez, $a^{p-1} = 1\ mod(p)$, p ve q birer asal sayıdır. Aynı zamanda, e ve d sayıları da birer pozitif sayıdır, $ed = 1\ mod(\Phi(\mathrm{pq}))$, $\Phi = (\mathrm{p}-1)(\mathrm{q}-1)$ olduğu için söyle yazılabilir, $\mathrm{ed}-1 = \mathrm{h}(\mathrm{p}-1)(\mathrm{q}-1)$ h eksi olmayan bir sayıdır. Burada $m^{ed} = m\ mod(p)$, iki durum, $m = 0\ mod(p)$ ve $m \neq 0\ mod\ (p)$ nin herhangi bir katıdır, bu yüzden m^{ed} p' nin herhangi bir katıdır, bu yüzden $m^{ed} = 0 = m\ mod(p)$ dir.

$$m^{ed} = m^{(ed-1)}m = m^{h(p-1)(q-1)}m = (m^{p-1})^{h(q-1)}m = 1^{h(q-1)}m = m\ mod(p)$$

Burada Fermant'ın küçük teorisini kullanacak olursak $m^{(p-1)}\ mod(p)$ nin 1 ile yer değiştirdiğini düşünelim. Yine aynı şekilde m, q nun herhangi bir katıdır, m^{ed} q nin herhangi bir katıdır, bu yüzden $m^{ed} = 0 = m\ mod(q)$ dir.

$$m^{ed} = m^{(ed-1)}m = m^{h(p-1)(q-1)}m = (m^{q-1})^{h(p-1)}m = 1^{h(p-1)}m = m\ mod(p).$$

Bu şunu ispat eder ki verilen bir m sayısı e, d çifti için (ed = 1 mod Φ(pq)) olduğundan

$(m^e)^d = m\ mod(pq)$ dir.

2.2.4.2. Euler Teorisi İspatı

Burada bizim ispat etmek istediğimiz formül $\mathrm{m}^{ed} = m\ mod(n)$ dir. Buradaki n = pq yani iki farklı asal sayının çarpımıdır e ve d de pozitif ed = 1 mod Φ(n) dir, Bu noktadan hareketle e ve d pozitif olduğu için 1 + hΦ(n) şeklinde yazılabilir, m'in n' e göre asal olduğunu varsayalım.

$$m^{ed} = m^{1+\mathrm{h}\Phi(\mathrm{n})} = m(m^{\Phi(\mathrm{n})})^h = m(1)^h = m\ mod(n).$$

M n' e göre asal değilse verilen kanıt geçersizdir.

2.2.5. Simetrik ve Asimetrik Anahtarlı Şifrelemenin Karşılaştırılması

Bu bölümde simetrik ve asimetrik şifrelemenin avantaj ve dezavantajlarını ortaya koyarak hangisinin çalışmamız için kullanılmasının doğru olacağına karar verilecektir.

1. Şifrelemede kullanılacak veri arasında fark vardır. Simetrik şifrelemeli sistemde şifrelenebilecek veri miktarı daha yüksektir. Şifrelenen veri konusunda asimetrik şifrelemeli sistemler genel olarak geri kalmaktadır.
2. Asimetrik şifreleme sistemlerinde kullanılan anahtar uzunluğu asimetrik şifrelemeli sistemlere göre daha uzundur.
3. Asimetrik şifrelemeli sistemlerde, veri şifrelenen anahtar ile açılamaz bu iş için üretilen diğer anahtar kullanılır. Asimetrik şifreleme sistemlerinde ise aynı anahtar şifreleme ve çözme işlemini gerçekleştirir.
4. Simetrik şifrelemeli sistemlerde anahtarın sık sık değiştirilmesine gerek duyulur çünkü bu anahtar şifreleme ve şifre çözmede kullanıldığı için çok önemlidir. Asimetrik şifreleme sistemlerinde ise bu kadar sık anahtar değişimine ihtiyaç yoktur çünkü ne de olsa şifreleme yapan anahtar şifreli veriyi açamamaktadır.
5. Asimetrik şifrelemeli sistemler elektronik imzanın ortaya çıkmasının da temelini atmıştır.

2.2.5.1. AES, DES ve RSA Algoritmalarının Karşılaştırılması

Tablo 2.1'de üç şifreleme sistemi arasında yapılmıştır; bunlardan ikisi simetrik diğeri asimetrik anahtarlama sistemine sahip üç şifreleme sistemi arasındaki ilişkiyi daha iyi görmemizi sağlamaktadır. Aşağıdaki üç şifreleme algoritması simetrik ve asimetrik şifreleme için en iyi örnek olan algoritmalar olduğu için seçilmiştir, bu parametrelerin incelenmesi, çalışmamızı asimetrik şifreleme üzerine yapılmasını teşvik etmiştir. Bunu nedeni DES ve AES sistemin daha hızlı olmasına rağmen güvenlik ön plana çıktığı için RSA algoritması bizim daha iyi bir seçim olarak görülmektedir[9][12]. Bu çalışmadaki temel amaç güvenliği olabildiğince ileri götürerek güçlü bir şifreleme algoritması oluşturabilmektir. Çalışmamızda algoritmanın hızlı bir yapıya sahip olması ikinci plana atılmıştır.

Tablo 2.1: AES, DES ve RSA karşılaştırılması.

Etken	**AES**	**DES**	**RSA**
Geliştirilme	*2000*	*1977*	*1978*
Anahtar boyutu	*128,192, 256 bit*	*56 bit*	*>1024 bit*
Engel boyutu	*128 bit*	*64 bit*	*En az 512 bit*
Şifreleme ve Çözme anahtar uzunluğu	*Aynı*	*Aynı*	*Farklı*
Ölçeklenebilirlik	*Ölçeklenemez*	*Değişken anahtar boyutundan dolay ölçeklenebilir*	*Ölçeklenemez*
Algoritma	*Simetrik*	*Simetrik*	*Simetrik değil*
Şifreleme	*Hızlı*	*Makul*	*Yavaş*
Çözme	*Hızlı*	*Makul*	*Yavaş*
Enerji tüketimi	*Yavaş*	*Yavaş*	*Hızlı*
Güvenlik	*Yüksek güvenlik*	*Yeterince güvenli değil*	*En güvenli*
Anahtar mevduat	*Gerekli*	*Gerekli*	*Gerekli*
Kalıtılan Açıklar	*Kaba kuvvet saldırısı*	*Kaba kuvvet, lineer ve diferansiyel kriptanaliz saldırısı*	*Kaba kuvvet ve oracle saldırısı*
Kullanılan anahtar	*Şifreleme ve çözme için aynı anahtar*	*Şifreleme ve çözme için aynı anahtar*	*Şifreleme ve çözme için farklı anahtar*
Tekrarlanma	*10/12/14*	*16*	*1*
Uyarım hızı	*Hızlı*	*Hızlı*	*Hızlı*
Sahte Anahtar	*İspatlanmadı*	*Yok*	*Yok*
Yazılım ve donanım uygulaması	*hızlı*	*Donanımda yazılıma göre daha iyi*	*Fark etmez*
Şifreleme ve Çözme algoritması	*Farklı*	*Farklı*	*Aynı*

2.3. BİYOMETRİK PARAMETRELERİN KARŞILAŞTIRILMASI

Tablo 2.2'de biyometrik parametre tablosu en çok kullanılan biyometrik parametrelerin olumlu ve olumsuz yönlerini göstermektedir. Bu tablo çalışmamızda hangi biyometrik parametrenin seçilmesinin doğru olacağı hakkında bize bilgi vermektedir. İlk olarak yüz parametresi ele alacak olursak kalıcılık ve verim açısından düşük olduğu için bu seçenek elenmektedir. El şekli parametresine gelince bu parametre tüm kıstaslar da ortalama bir

izlenim verdiği için kullanımı başarımı yüksek bir sistem meydana getirmeyecektir. Iris parametresi için birçok parametre yüksek değerde olsa da en olumsuz özelliği yanılma olasılığı oldukça yüksek bu seçenek de çalışmamız için uygun değildir. Tuşlama parametresi davranışsal bir parametre ve matris haline getirilmeye müsait değil bu yüzden çalışmamız için kullanılması uygun değildir. İmza parametresi birçok yönden düşük bir başarıma sahip yine aynı şekilde ses parametresinde yüksek bir başarım göstermemektedir. Geriye en iyi seçenek olarak parmak izi kalmaktadır.

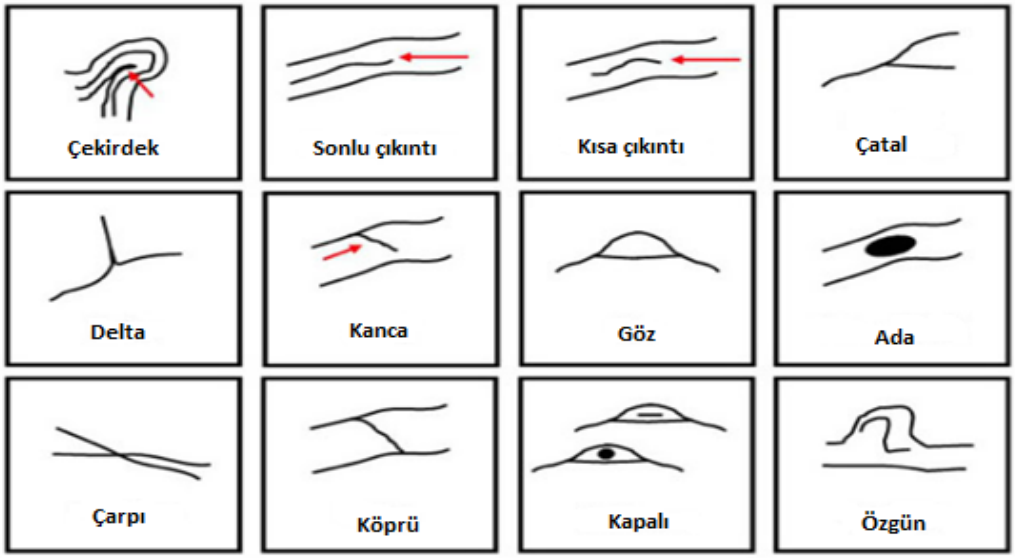
Tablo 2.2: Biyomterik parametre karşılaştırma tablosu.

Parametre	**Evrensellik**	**Açıklık**	**Kalıcılık**	**Edinilme**	**Verim**	**Uygunluk**	**Yanılma**
Yüz	*Y*	*D*	*O*	*Y*	*D*	*Y*	*Y*
Parmak izi	*O*	*Y*	*Y*	*O*	*Y*	*O*	*O*
El şekli	*O*	*O*	*O*	*Y*	*O*	*O*	*O*
İris	*Y*	*Y*	*Y*	*O*	*Y*	*D*	*D*
Tuşlama	*D*	*D*	*D*	*O*	*D*	*O*	*O*
İmza	*D*	*D*	*D*	*Y*	*D*	*Y*	*Y*
Ses	*O*	*D*	*D*	*O*	*D*	*Y*	*Y*

Y: Yüksek, *O*: Orta, *D*: Düşük [10]

2.4. PARMAK İZİ DETAY BİLGİSİ DESENLERİ

Şekil 2.4'te parmak izinde bulunan detay bilgilerinin en sıklıkla karşımıza çıkan desenleri gösterilmiştir[11]. Temel alınan çalışmada bu desenler kullanılarak tekil anahtar kullanılmaktadır. Üretilen bu tekil anahtar asimetrik şifrelemede kullanılmaktadır.



Şekil 2.4: Parmak izi detay desenleri.

Parmak izi biyometrik parametresi kullanıldığı zaman bu desenler tanımlamada çok önemli rol oynamaktadır. Parmak izinde bu desenler aranır ve koordinatları kayıt edilerek tanımlama bu bilgi üzerinden yapılır. Parmak izindeki tüm detaylar karşılaştırılmaz çünkü parmak izi günlük elle yapılan işler sonucunda bir önceki parmak izi ile tutarlı sonuçlar vermeyebilir. Bazen bu desenler bile yetmeyebilir bir de belli bir eşik değeri konularak parmakta meydana gelebilecek değişikleri tahammül edilebilir bir seviyede tutmak gerekmektedir.

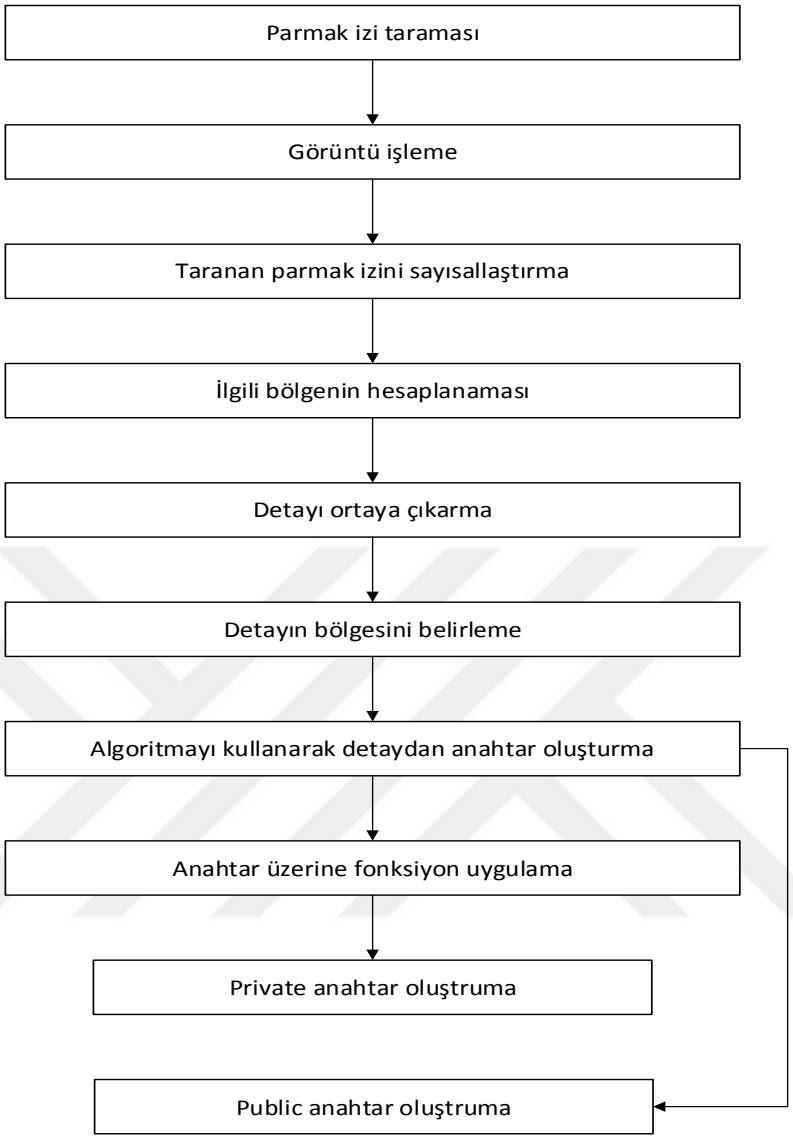
3. MALZEME VE YÖNTEM

3.1. PARMAK İZİ TABANLI RSA ŞİFRELEME

Biyometrik parametrenin günümüzde kullanımın artması ile güvenliği sağlayan anahtarlarda bu biyometrik parametrelerden üretilmeye başlanmıştır. Bunun üzerine birçok makale ve proje geliştirilmiştir. Bizim çalışmamızda bu bağlamda olup amacımız biyometrik parametreden anahtar üretmeye dayanmaktadır. Çalışmamıza esin kaynağı olabilecek çalışmalar incelenmiş ve çalışmamızın mantığı ve ruhuna en uygun olan makale bizim için başlangıç noktası olarak seçilmiştir.

Seçtiğimiz makale olan ve yine RSA tabanlı, biyometrik parametreden anahtar üretme fikrine dayanan şifreleme sistemidir. Bu sistemde parmak izinden anahtar üretilirken parmak izinde var olan çeşitli desenlerin kordinatları baz alınarak oluşturulan veri dizisi daha sonrasında RSA tabanlı bir açık anahtar oluşturmak için kullanılır[8].

Biyometrik parametreden anahtar üretimi için çeşitli yollar kullanılmıştır. Bu yollardan bir tanesi olan çalışma Şekil 3.1'de örnek alınan şemaya ait akış şeması görülmektedir.

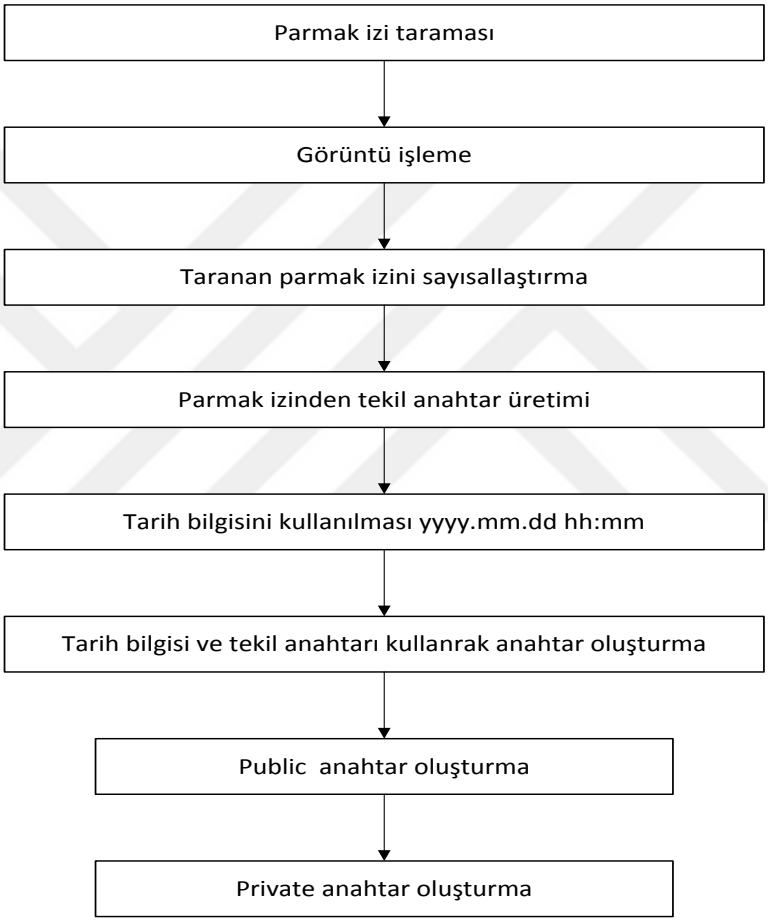


Şekil 3.1: Temel alınan şifreleme algoritması akış şeması.

Şekil 3.1'de akış şeması örnek alınan çalışmadaki tekil anahtar üretme aşamalarını göstermektedir[8]. Buradaki çalışmada parmak izi biyometrik parametresi kullanılmıştır. Parmak izi ilk önce sayısallaştırarak matris haline getirilir. Daha sonrasında, parmak izinden detay bilgileri çıkartılarak bir dizide tutulur. Algoritmayı kullanarak bu detaylardan anahtar oluşturulur. Anahtar üzerine fonksiyon uygulandığında buradan açık ve gizli anahtar elde edilir.

3.2. ÖNERİLEN PARMAK İZİ TABANLI RSA ŞİFRELEME

Biyometrik parametrelerin şifreleme sistemlerinde daha fazla kullanılması ile biyometrik parametrelerden anahtar üretimi fikride artık daha öne çıkmaya başladı. Anahtarın değiştirilmesi güvenlik açıklarını önlemede etkin olduğu için farklı anahtar üretimini için biyometrik parametre kullanımı gayet kullanışlı görülmektedir. Şekil 3.2'de çalışmamıza ait anahtar üretimi şeması görülmektedir.



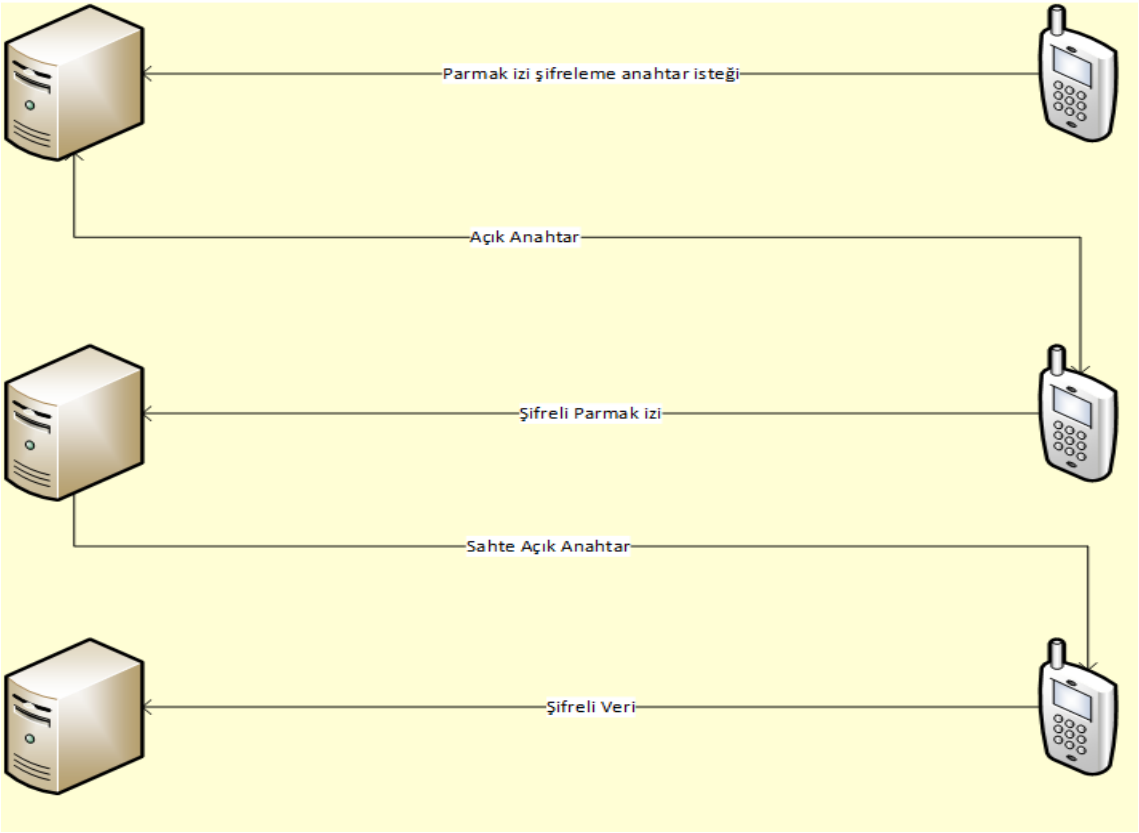
Şekil 3.2: Biyometrik tabanlı RSA algoritması akış şeması.

Şekil 3.2'de akış şeması biyometrik tabanlı geliştirdiğim RSA algoritmasına aittir. Temel alınan çalışmada, matris parmak izinden çıkartılan detaylar sayesinde yapılmıştır[8]. Önerdiğimiz yöntemde biyometrik matristen oluşturulan tekil anahtar veritabanında tutulur ve bu anahtar üzerinden açık anahtar oluşturulur. Açık anahtar oluştururken her seferinde farklı bir anahtarın oluşturulması için zaman temel alınır. Daha sonrasında oluşturulan açık anahtardan temel RSA daki olduğu gibi gizli anahtar oluşturulur ve her seferinde farklı

anahtar çifti oluşmuş olur. Bunun güvenlik açısından önemli etkisi vardır. Bunu sonuçlarını ileriki bölümlerde gösterilecektir.

3.2.1. Şifreleme Protokolü

Biyometrik parametrenin şifreleme sistemlerinde kullanımının genel gösterim şeması Şekil 3.3'de gösterilmiştir. Buradaki gösterim şeması biyometrik parametrenin şifreleme sistemi üzerine uygulandığı takdirde kullanıcı ile sunucu arasında nasıl bir akış olacağını gösterilmektedir.



Şekil 3.3: Biyometrik tabanlı RSA algoritması protokolü.

Şekil 3.3'te biyometrik tabanlı RSA şifreleme algoritması şeması görülmektedir. Burada, kullanıcı sunucudan parmak izinin tekil anahtarını oluşturulur. Bunun nedeni, oluşturulan bu tekil anahtardan açık anahtar üretmek için faydalanılacak olması hem de veritabanında parmak izi değil bu veri tutularak gizlilik sağlanmasıdır. Parmak izinin ağda serbest bir şekilde dolaşmaması için bir açık anahtar gönderilir ve bu anahtar ile şifreleme yapılır. Şifrelenen parmak izi şifreli hali açılarak bu matristen tekil anahtarı üretilir. Bu aşamadan sonra kullanıcıya oluşturulan tekil anahtarı temelinde üretilen açık anahtar gönderilir. Bu açık

anahtar ağ üzerinde görülmemesi için belli bir çerçeve yapısındadır. Bu çerçeve yapısında zaman ve şifreli açık anahtar birlikte verilerek karmaşa arttırılır. Kullanıcıya, (ddMMyyyyhhmm + şifreli açık anahtar) formatında bir çerçeve ulaştırılır. Bunun nedeni, elimizdeki açık anahtardan gizli anahtara ulaşılmasında kullanılan çeşitli yöntemlerin önüne geçebilmektir. Açık anahtar zaman bazında yapılan bir formül ile çarpılarak sayı çok büyük bir sayı haline getirilir. Bu sayede, aradaki sahte alıcıların bu sayıyı anlamlandırması daha zorlaşır. Gönderilecek açık anahtar parmak izi resim objesinden üretildiği için kişisel olarak farklılık göstermektedir. Bu da her defasında farklı bir açık anahtar oluşturmamızı sağlamaktadır. Bu da önemli bir güvenlik önlemidir.

3.2.2. Tekil Anahtar Elde Etme İşlemleri

Tablo 3.1 ‘de parmak izi biyometrik parametresinin alındıktan sonraki işlem adımları tek tek gösterilmiştir. Burada alınan parmak izi resminin hangi işlemlerden geçtiği görülmektedir.

Tablo 3.1: Biyometrik parametrenin resimden tekil anahtar oluşturma.

Parmak izi resmi	
Resmi boyutlama.	$\begin{bmatrix} a_{1,1} & \cdots & a_{1,m} \\ \vdots & \ddots & \vdots \\ z_{n,1} & \cdots & z_{n,m} \end{bmatrix}_{nxm}$
Ortalama değeri bulma.	$avg = \sum_{k=1,l=1}^{m,n} a_{k,l} \,/2$
Ortalama değere göre 1 ya da 0 matrisinin belirlenmesi.	$a_{m,n} = \begin{cases} 1, & avg < a_{m,n} \\ 0, & avg \geq a_{m,n} \end{cases}$
Tekil anahtarı oluşturacak 2 matrisin elde edilmesi.	$\begin{bmatrix} \sum_{k=0}^{m} a_{0,k} \\ \vdots \\ \sum_{k=0}^{m} a_{n,k} \end{bmatrix} \begin{bmatrix} a_{0,0} & \cdots & a_{0,m} \\ \vdots & \ddots & \vdots \\ a_{n,0} & \cdots & a_{n,m} \end{bmatrix}_{nxm} \begin{bmatrix} \sum_{k=0}^{n} a_{k,0} & \ldots & \sum_{k=0}^{n} a_{k,m} \end{bmatrix}$
Tekil anahtar elde edilmesi.	$a = [\sum_{k=1}^{m} a_{1,k} \quad \cdots \quad \sum_{k=1}^{m} a_{n,k} \quad \sum_{k=1}^{m} a_{k,1} \quad \cdots \quad \sum_{k=1}^{m} a_{k,n}]$
Sıfır filtresinden geçirme	$v = \begin{cases} a_m, & 0 < a_m \\ , & 0 = a_m \end{cases}$
Açık değer kümesini oluşturulması	$S_1 = V_1.$ $S_n = S_{n-1} * V_n \,, S_n = \{S_n \% 10000,\ S_n \geq 10000\}$

Tablo 3.1'de görüldüğü üzere veri tabanında parmak izi maskeli olarak saklanmaktadır. Kişisel bilgiler veritabanında ya da ağda asla gönderilmemelidir. Gelen parmak izi önce işlem için istenilen şekilde yeniden boyutlanır. Bundan sonra matris olarak açılır ve matrisin veritabanında ham halde durmaması ve de şifrelemede işlemindeki açık anahtar üretiminde kullanılacak tekil anahtarın üretilmesi için bu matris kullanılır. Bu tekil anahtar veritabanımızda tutulacaktır. Aynı zamanda, kişisel parametre olan parmak izi, iris, avuç içi gibi biyometrik parametreler de bu geliştirilen algoritmada kullanılabilir. Yapılan bu işlem biyometrik parametrenin maskelenmesinin yanı sıra şifreleme işleminde de temel rol oynayacaktır.

Tekil anahtar söyle oluşturulur: nxm bir matrisimiz vardır. Bu matrisin her bir satırındaki değerler toplanır ve o satırın başına yazılır böylece n elemanlı bir matris oluşturulur. Matrisin her bir kolonundaki değerler toplanır ve m elemanlı bir matris oluşur. Bu iki matris bir biri ile birleştirilir. Bu matrisimiz bizim tekil matrisimiz olacaktır. Bu matris sayesinde parmak izi bilgimizi maskelememizin yanında bu matris üzerinden onu sorgulama yapmak mümkün olacaktır. Daha önceki çalışmada parmak izinden detay bilgileri çıkarılarak bu matris oluşturulmuştur[8].

3.2.2.1. Tekil Matris - Karakter Seti Dönüşümü

Tekil anahtarı üretirken oluşacak değerleri saklamak için bir karakter tablosu üretilir. Bu tablo aracılığı ile karakterleri eşleyerek bu değerlerin hem sadece bir karakter dizisi olarak saklanması, hem de veritabanında gizli olarak saklanması sağlanmış olur, burada kullanılan karakterler sistemi geliştirecek kişi tarafında seçilebilir. Karakter sayısı w+h olmalıdır. Bunun nedeni tekil anahtarın değerlerini saklayabilecek sayıda olması gerekmektedir. Bu karakterler için örnek bir şablon Tablo 3.2'de gösterilmiştir.

Tablo 3.2: Karakter değer dönüşüm tablosu.

0	*0*	**32**	*W*	**64**	*A*	**96**	*O*	**128**	*ε*	**160**	*τ*	**192**	ח	**224**	פּ
1	*1*	**33**	*X*	**65**	*B*	**97**	*П*	**129**	*ζ*	**161**	*υ*	**193**	ט	**225**	פ
2	*2*	**34**	*V*	**66**	*Г*	**98**	*P*	**130**	*η*	**162**	*φ*	**194**	י	**226**	ף
3	*3*	**35**	*Z*	**67**	*Д*	**99**	*Σ*	**131**	*θ*	**163**	*χ*	**195**	ך	**227**	צ
4	*4*	**36**	*a*	**68**	*E*	**100**	*T*	**132**	*ι*	**164**	*ψ*	**196**	כ	**228**	ץ
5	*5*	**37**	*b*	**69**	*Z*	**101**	*Y*	**133**	*κ*	**165**	*ω*	**197**	ך	**229**	ק
6	*6*	**38**	*c*	**70**	*H*	**102**	*Φ*	**134**	*λ*	**166**	א	**198**	ל	**230**	ר
7	*7*	**39**	*d*	**71**	*Θ*	**103**	*X*	**135**	*μ*	**167**	בּ	**199**	ל	**231**	שׁ
8	*8*	**40**	*e*	**72**	*I*	**104**	*Ψ*	**136**	*ν*	**168**	ב	**200**	מ	**232**	שׂ
9	*9*	**41**	*f*	**73**	*K*	**105**	*Ω*	**137**	*ξ*	**169**	ג	**201**	ם	**233**	תּ
10	*A*	**42**	*g*	**74**	*Λ*	**106**	*α*	**138**	*o*	**170**	ד	**202**	נ	**234**	ת
11	*B*	**43**	*h*	**75**	*M*	**107**	*β*	**139**	*π*	**171**	ה	**203**	ן	**235**	س
12	*C*	**44**	*i*	**76**	*N*	**108**	*γ*	**140**	*ρ*	**172**	ו	**204**	ס	**236**	ق
13	*D*	**45**	*j*	**77**	*Ξ*	**109**	*δ*	**141**	*σ*	**173**	ז	**205**	ע	**237**	ف
14	*E*	**46**	*k*	**78**	*B*	**110**	*У*	**142**	*e*	**174**	*ч*	**206**	*{*	**238**	-
15	*F*	**47**	*l*	**79**	*Г*	**111**	*Ф*	**143**	*ё*	**175**	*ш*	**207**	*(*	**239**	_
16	*G*	**48**	*m*	**80**	*Д*	**112**	*Х*	**144**	*ж*	**176**	*щ*	**208**	*[*	**240**	|
17	*H*	**49**	*n*	**81**	*E*	**113**	*Ц*	**145**	*з*	**177**	*ы*	**209**	*)*	**241**	*:*
18	*I*	**50**	*o*	**82**	*Ё*	**114**	*Ч*	**146**	*и*	**178**	*э*	**210**	*]*	**242**	@
19	*J*	**51**	*p*	**83**	*Ж*	**115**	*Ш*	**147**	*й*	**179**	*ю*	**211**	=	**243**	Ç
20	*K*	**52**	*q*	**84**	*З*	**116**	*Щ*	**148**	*к*	**180**	*я*	**212**	*}*	**244**	Ö
21	*L*	**53**	*r*	**85**	*И*	**117**	*Ъ*	**149**	*л*	**181**	*!*	**213**	*?*	**245**	Ş
22	*M*	**54**	*s*	**86**	*Й*	**118**	*Ы*	**150**	*м*	**182**	‘	**214**	*	**246**	İ
23	*N*	**55**	*t*	**87**	*К*	**119**	*Ь*	**151**	*н*	**183**	“	**215**	\	**247**	Ğ
24	*O*	**56**	*u*	**88**	*Л*	**120**	*Э*	**152**	*о*	**184**	^	**216**	.	**248**	Ü
25	*P*	**57**	*v*	**89**	*М*	**121**	*Ю*	**153**	*п*	**185**	#	**217**	,	**249**	Ç
26	*Q*	**58**	*w*	**90**	*Н*	**122**	*Я*	**154**	*р*	**186**	+	**218**	;	**250**	ö
27	*R*	**59**	*x*	**91**	*О*	**123**	*а*	**155**	*с*	**187**	$	**219**	<	**251**	ş
28	*S*	**60**	*v*	**92**	*П*	**124**	*б*	**156**	*т*	**188**	%	**220**	>	**252**	ı
29	*T*	**61**	*z*	**93**	*Р*	**125**	*в*	**157**	*ф*	**189**	½	**221**	*æ*	**253**	ğ
30	*U*	**62**	*A*	**94**	*С*	**126**	*г*	**158**	*х*	**190**	&	**222**	~	**254**	ü
31	*V*	**63**	*Б*	**95**	*Т*	**127**	*д*	**159**	*ц*	**191**	/	**223**	“	**255**	ç

3.2.3. Biyometrik Tabanlı Şifreleme Algoritması

Aşağıdaki algoritma yeni biyometrik tabanlı şifreleme RSA şifreleme algoritmasına aittir. Açık anahtar ve gizli Anahtar RSA'da olduğu gibi aynı mantıktadır. Açık anahtar bir metni şifrelemeye gizli anahtar ise o açık anahtar ile şifrelenen metni açmaya yaramaktadır.

3.2.4. Biyometrik Tabanlı RSA Algoritması

1. İki adet asal sayı seçilir p ve q.
2. Hesaplama n = p.q.
3. Totient hesaplama Φ = (p-1)(q-1).
4. $\text{publicKey}_{\text{value}}(\text{date})$ < e < Φ ve ebob(e,Φ)=1.
5. d gizli anahtar hesaplanır d.e = 1 mod(Φ)
6. $e_{pseudo} = \text{e} * \text{getNumber}(\text{date})$.

Yukarıda görüldüğü gibi temel RSA şifreleme algoritmasında olduğu gibi iki adet asal sayı seçilir. Bunlar bir biri ile çarpılır. Totient(Φ) de yine temel RSA da olduğu gibi seçilen asal sayılardan bir çıkartılarak bu iki sayının çarpımı olarak hesaplanır. Burada, temel RSA'dan farklı olduğu ilk yer olarak, seçilen açık anahtar biyometrik parametre yardımı ile oluşturulan tekil anahtar üzerinden üretilir. Daha sonrasında bu açık anahtara uygun bir gizli anahtar oluşturulur. Seçilen açık anahtar zaman temelli sayı üreten getNumber fonksiyonudur, bu fonksiyonun ürettiği değer ile açık anahtar çarpılır ve büyük bir sayı üretilerek açık anahtarın açık bir şekilde gönderilmesi önlenir.

3.2.4.1. publicKey Fonksiyonu

Bu fonksiyonumuzda ise parametre olarak yine date türünde ve yine ddMMyyyyhhmm formatında olacaktır.

publicKey (hhmm) = (hh + mm)*5 % (lenght) (3)

Örnek olarak;

Tarih: 081220161220

publicKey (1220) = (hh+mm)*5 % (m+n) = 16. sıradaki sayıdan yararlanılacaktır.

Aşağıda örnek bir tekil anahtar kümesi yer almaktadır. 16. sıradaki değer açık anahtarın belirlenmesinde rol oynayacaktır. Şöyle ki seçilen sayısı ile Φ sayısının aralarında asal olması gerekmektedir. Eğer seçilen bu sayı ile açık anahtarın çarpımı Φ sayısı ile aralarında asal değilse bu sefer seçilen sayı bir artırılır bu seferde değilse böyle arttırılmaya sonuca ulaşılıncaya kadar devam edilir.

Tablo 3.3: Parmak izinden üretilmiş örnek bir değer kümesi.

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	**16.**	17.
3	15	120	960	7680	61440	614400	14400	44000	40000	80000	60000	60000	20000	40000	**48000**	…

3.2.4.2. getNumber Fonksiyonu

Bu fonksiyon parametre olarak tarih türünde bir değer alır. Bu tarih formatı ddMMyyyyhhmm şeklinde olmalıdır. Bu gönderilen formatın ilk 8 değeri yani gün ay yıl bu fonksiyonda kullanılır.

$$\text{getNumber}(\text{ddMMyyyy}) = yyyy^3 + mm^2 + \text{dd} \qquad (4)$$

Örnek olarak;

Frame : **08122016**1220*2630126419608*

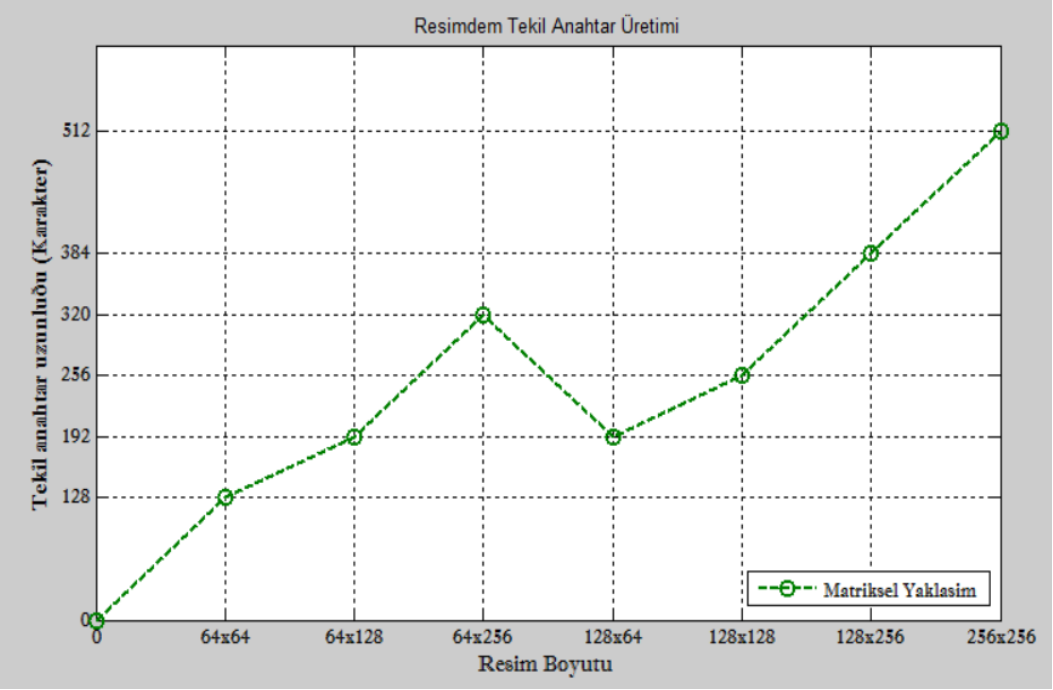
$\text{getNumber}(\mathbf{08122016}) = yyyy^3 + mm^2 + \text{dd} =$ 8193540248

Açık anahtar = *2630126419608* / 8193540248 = ***321*** açık anahtarı elde etmiş olduk.

Yukarıda bizim elimize gelen bir çerçeve vardır. Bu çerçevede ddMMyyyy değerleri kullanılarak fonksiyon değeri hesaplanarak gelen değere bölünür ve açık anahtarımız elde edilir.

4. BULGULAR

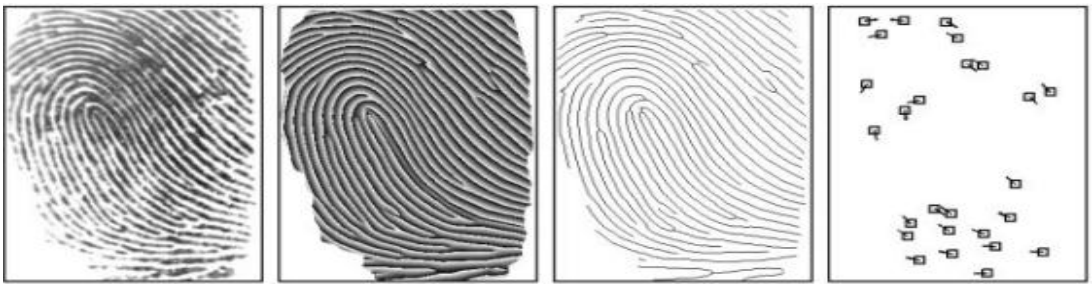
Şekil 4.1 tekil anahtar elde edilirken ürettiğimiz tekil anahtarın boyutunun resmin piksel sayısına göre nasıl değiştiği gösterilmektedir. Aşağıdaki grafikte görüldüğü üzere anahtarın boyutu eğer yükseklik ve genişliği sırasıyla h, w ile ifade edecek olursak anahtar uzunluğu h+w şeklinde olacaktır. Bununda performans etkisi olacaktır. Bu grafiğin elde edilişi Ek-3'deki Matlab kodunda gösterilmiştir.



Şekil 4.1: Resim boyutuna göre tekil anahtar üretimi.

Geliştirilen biyometrik RSA algoritması sayesinde temel RSA algoritmasının kriptoanaliz yapılan bazı açıkları bu geliştirilen algoritma sayesinde giderilmiş olmaktadır.

4.1. PARMAK İZİ DETAY BİLGİSİ VE ÖNERİLEN YAKLAŞIMIN KARŞILAŞTIRILMASI



Şekil 4.2: Parmak izinden detay bilgisi çıkarma.

Şekil 4.2'de örnek alanın projedeki tekil anahtarın çıkarılması için kullanılan yöntem gösterilmiştir. Buradaki parmak izi detay bilgilerinin koordinatlarını dizide tutularak tekil anahtar oluşturulmaktadır[8]. Önceki çalışmadaki anahtar uzunluğunu belirlemek için ortalama koordinat değerleri bulunarak bunların hesaplanması ile başlanmaktadır.

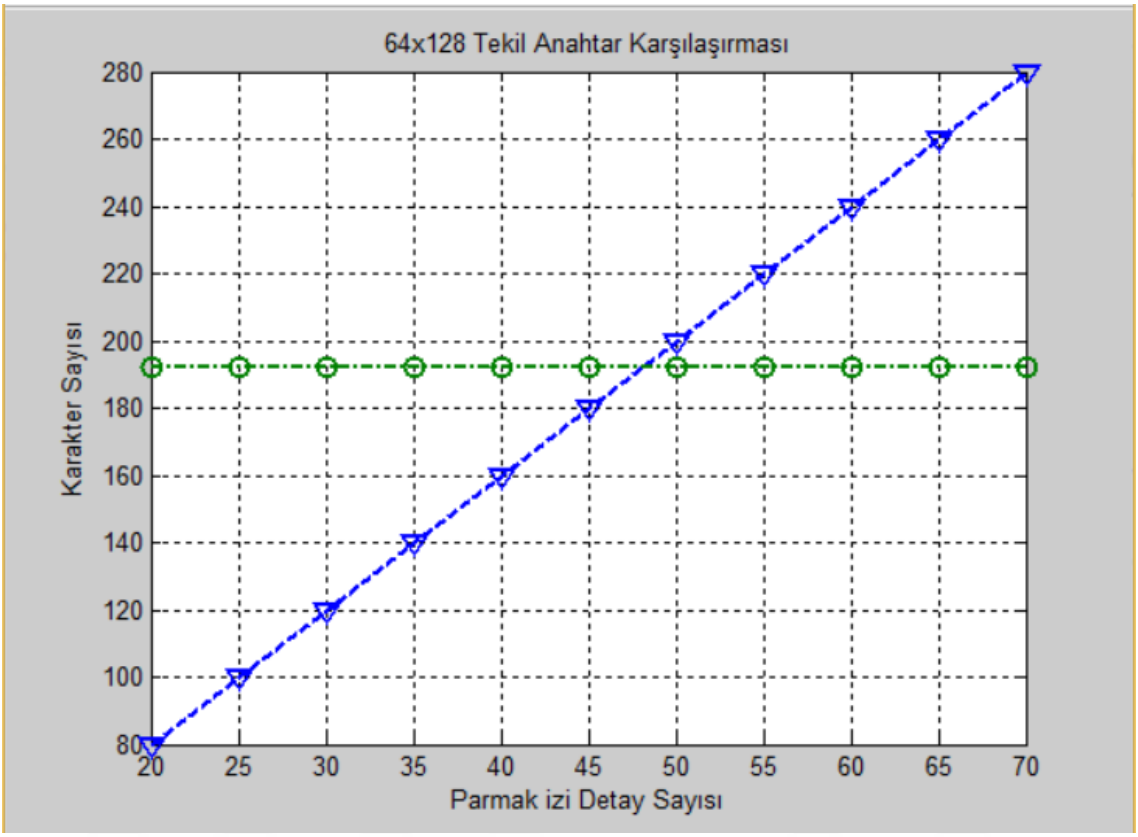
Ortalama kod uzunluğu formülü aşağıdaki şekildedir:

$$\sum_{i=1}^{n} w_{i\ x\ lenght(c_i)} \tag{5}$$

Y ekseni (128) ortalama uzunluğu $\frac{9}{128} * 1 + \frac{90}{128} * 2 + \frac{157}{128} * 3 \cong 2{,}58$

X ekseni (64) ortalama uzunluğu $\frac{9}{64} * 1 + \frac{55}{64} * 2 \cong 1{,}86$

Koordinatların ortalama uzunluğu $2.58 + 1.86 = 4.04$



Şekil 4.3: Anahtarın karakter sayısı karşılaştırması.

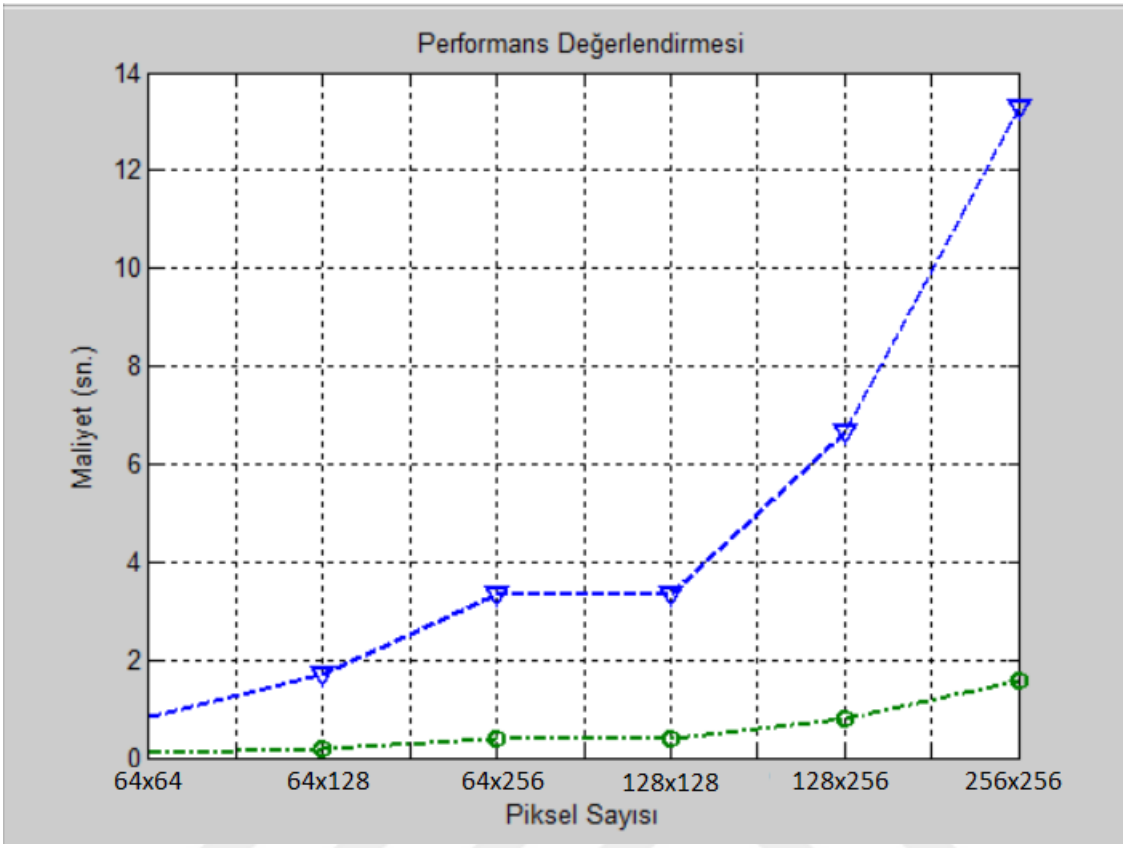
Şekil 4.3'te örnek alınan makaledeki parmak izi detay bilgisinden çıkarılan detay bilgilerinin koordinatları ile oluşturulan tekil anahtarın karakter sayısı ve bu çalışmada ele alınan yaklaşımda 192 karakterden oluşan sabit bir tekil anahtar üretilmektedir. Bu sabit anahtar çalışmasında matristen elde edilen bir anahtar fakat her parmak izindeki detay bilgisi 20 ile 70 arasında değişmektedir [10]. Bu durum göz önüne alındığında anahtar uzunluklarının parmak izi detay bilgisine göre nasıl değiştiği yukarıda gösterilmiştir. Matris üzerinden elde edilen parmak izinde bu değer daima sabit ve 192 olacaktır (64x128 piksel bir parmak izi resmi). Koordinat değerleri için ortalama karakter sayısı hesaplanarak bu değerler üzerinden tekil anahtar değeri belirlenmiştir.

Tablo 4.1: Algoritma performans karşılaştırılması**.**

Matrisel yaklaşım	**Parmak izi detay yaklaşımı**

for(h=0; h>H; h++){
x = 0;
for(w=0; w>W; w++){
x = x + matrix(h,w);
}
registerKey(x);
}
for(w=0; w>W; w++){
x = 0;
for(h=0; h>H; h++){
x = x + matrix(w,h);
}
registerKey(x);
}
	for(h=0; h>H; h++)
	for(w=0; w>W; w++)
	for(p=0; p<12;p++)
	if(minutiae(w,h,p))
	registerKey(w,h)

Tablo 4.1'de asimptotik karmaşıklık hesaplaması yapılarak her iki algoritmanın da performans değerlendirmesi yapılmıştır. Bu ölçümleri yaparken Intel® Core(TM) i7-4710MQ CPU @ 2.50GHz işlemcili, 8,00 GB Ram kapasiteli ve 64-bit işlerim sistemi olan bilgisayar kullanılmıştır.



Şekil 4.4: Performans karşılaştırması.

Şekil 4.4'de görüldüğü üzere parmak izi detay bilgisinin alınması için tüm piksellerin gezilerek bu pikseller üzerinde parmak izi detay bilgisinin olup olmadığı belirlenmesi gerekmektedir. Bunun maliyeti yukarıda gösterilmiştir. Matriksel yöntemde ise sadece satır ve sütunların toplam değerleri bulunduğu için maliyet oldukça düşüktür. Değerler Tablo 3.1'de gösterilmiştir. Ek-5'de performans değerlendirmesinin matlab kodu verilmiştir.

4.2. KRİPTOANALİZ

Şifreleme algoritmalarının zayıf yönlerini bulmaya kriptoanaliz denir. Her şifreleme algoritması için kriptoanaliz yapılmaktadır. Elbette RSA algoritması içince bu analizler yapılmıştır. Bizim geliştirdiğimiz biyometrik tabanlı RSA algoritmasının bazı kriptoanaliz yöntemlerine karşı nasıl güçlü bir hale geldiği görülecektir.

4.2.1. Düşük Açık Anahtar Saldırısı

4.2.1.1. RSA Üzerindeki Düşük Açık Anahtar Saldırısı.

Aynı mesaj farklı seçilmiş asal sayılardan oluşan yeni bir RSA sisteminde aynı açık anahtar tarafından şifrelenir, bu şifreli metinler aşağıdaki formül ile elde edilebilir.

$$c = c_b(n_c n_d)[(n_c n_d)^{-1}]_{n_b} + c_c(n_b n_d)[(n_b n_d)^{-1}]_{n_c} + c_d(n_b n_c)[(n_b n_c)^{-1}]_{n_d} \quad (6)$$

$$c = m^{\mathrm{e}} \ (mod\ n) \quad (7)$$

Buradaki n_b, n_c, n_d değerleri iki asal sayının çarpılması sonucu elde edilen değerler.

$c = c_b(n_c n_d)[(n_c n_d)^{-1}]_{n_b} + c_c(n_b n_d)[(n_b n_d)^{-1}]_{n_c} + c_d(n_b n_c)[(n_b n_c)^{-1}]_{n_d}$

Mesaj değeri m = 102 modül değerler sırası ile $n_b = 377, n_c = 391, n_d = 589$.

$c_b = 102^3 \ (mod\ 377)$ = 330.

$c_c = 102^3 \ (mod\ 391)$ = 34.

$c_d = 102^3 \ (mod\ 589)$ = 419.

Yukarıdaki değerler elde edilir. Bu değerleri kullanarak;

$t_b = c_b(n_c n_d)[(n_c n_d)^{-1}]_{n_b}$ = $330(391.589)[(391.589)^{-1}]_{377}$ = 24471571740.

$t_c = c_c(n_b n_d)[(n_b n_d)^{-1}]_{n_c}$ = $34(377.589)[(377.589)^{-1}]_{391}$ = 505836734.

$t_d = c_d(n_c n_b)[(n_c n_b)^{-1}]_{n_d}$ = $419(377.391)[(377.391)^{-1}]_{589}$ = 35452267942.

$$c = t_b + t_c + t_d \ mod(n_b n_c n_d) = 1061208.$$

$m = \sqrt[3]{1061208}$ = 102. Yukarıda görüldüğü üzere mesaj elde edildi.

4.2.1.2. Biyometrik Tabanlı RSA Üzerindeki Düşük Açık Anahtar Saldırısı.

Açık anahtar havuzu = 3, 5, 6, 8, 11, 13, 15, 17, 18 olarak farz edelim.

$t_1 = 06{:}03,\ t_2 = 09{:}02,\ t_3 = 14{:}17$ Zaman saat: dakika

Açık anahtar havuzu = 3, 5, 6, 8, 11, 13, 15, 17, 18

n_b = 377, Açık = **3**

n_c = 391, Açık = **5**

n_d = 589, Açık = **6**

$c_b = 102^3 \ (mod\ 377)$ = 330.

$c_c = 102^5 \ (mod\ 391) = 272.$

$c_d = 102^6 \ (mod\ 589) = 39.$

$t_b = c_b(n_c n_d)[(n_c n_d)^{-1}]_{n_b}$ = $330(391.589)[(391.589)^{-1}]_{337}$ = 24471571740.

$t_c = c_c(n_b n_d)[(n_b n_d)^{-1}]_{n_c}$ =$272(377.589)[(377.589)^{-1}]_{391}$ = 4046693872.

$t_d = c_d(n_c n_b)[(n_c n_b)^{-1}]_{n_d}$ = $39(377.391)[(377.391)^{-1}]_{589}$ = 3299853102.

$$c = t_b + t_c + t_d \ mod(n_b n_c n_d) = 41002096.$$

$m = \sqrt[3]{41002096}$ = 334.82.

$m = \sqrt[5]{41002096}$ = 33,30.

$m = \sqrt[6]{41002096}$ = 18,56.

Görüldüğü gibi doğru sonuç hiçbir açık anahtar için doğru bulunamadığı için bu saldırı yöntemi saf dışı bırakılmış olur.

4.2.2. Ortak Mod Saldırısı

4.2.2.1. RSA Üzerindeki Ortak Mod Saldırısı.

Eğer ebob$(e_1, e_2) = 1$ denklem sağlanırsa, çözüm için o zaman (s_1, s_2) çifti vardır. Şöyle ki ; $e_1 s_1 + e_2 s_2 = 1$ ve elimizdeki şifreli veriler için n aynı olmalıdır. Buradaki n sayısı seçilen iki asal sayının çarpımıdır.

Elimizde 2 adet şifreli metin olduğunu varsayalım aynı zamanda açık anahtarlar.

$c_1 = M^{e_1} \, mod \, (n) \, ve \, c_2 = M^{e_2} \, mod \, (n)$

$c_1^{s_1} c_2^{s_2} = (M^{e_1})^{s_1} (M^{e_2})^{s_2}$

$= (M^{e_1 s_1})(M^{e_2 s_2})$

$= (M^{e_1 s_1 + e_2 s_2})$

$= (M^1)$.

4.2.2.2. Biyometrik Tabanlı RSA Üzerindeki Ortak Mod Saldırısı.

Eğer ebob$(e_1, e_2) = 1$ denklem sağlanırsa, o zaman bir (s_1, s_2) çifti aranabilir. Uyguladığımız yöntemde her zaman (n,e) çifti system tarafından üretilir. Ortak bir n değeri üzerinden açık anahtar üretilmediği için bu saldırı yöntemi sonuçsuz kalacaktır.

Üretilen anahtar çiftleri (n_1, e_1) ve (n_2, e_2) dir.

Bunun sonucu olarak ebob(e_1, e_2) =1 sağlanması durumuna bakılmaz çünkü açık anahtarlar ortak bir n den üretilmemektedir.

N sabit olmadığı ebob(e_1, e_2) =1 denklemi sağlanmaz.

Bu sayede bu yöntem içinde önlem alınmış oldu.

4.2.3. Açık Anahtar Saldırısı.

4.2.3.1. RSA Üzerindeki Açık Anahtar Saldırısı.

Kullanıcıya gönderilecek anahtar çifti (n, açık anahtar) şeklinde olmaktadır. Buradaki n sunucu tarafta seçilen iki adet asal sayının çarpımıdır. Böyle olunca temel RSA algoritması yine savunmasız kalmaktadır.

Örnek olarak;

Tablo 4.2: Ele geçirilen n ve açık anahtar çifti.

n	Açık anahtar
10142789312725007	5

$$Floor\left[\sqrt{10142789312725007}\right] = 100711415$$

Tablo 4.2'de n sayısı iki asal sayının çarpımıdır. Geometrik ortasından başlayarak bu asal sayılar bulunmaya çalışılacaktır. Çünkü iki yakın boyuttaki asal sayının çarpımı ile n sayısı bulunmaktadır.

10142789312725007 mod(100711415) = 100711367

10142789312725007 mod(100711413) = 100711373

10142789312725007 mod(100711411) = 100711387

10142789312725007 mod(100711409) = 0

öyleyse birinci asal sayı p = 100711409, ikinci asal sayı q = n / p = 10142789312725007 / 100711409 = 100711423 dir. Böylece Φ = (p - 1)(q - 1) = 100711408.100711422 = 10142789111302176 değeri bulunur. Burdaki değerinden de = 1 mod(Φ) = d.5 = 1 mod(10142789111302176) Buradan gizli anahtarımızı buluruz, gizli anahtarı bulurken kullanılan Matlab kodu Ek kısmında, Ek-1 adlı başlığın altında verilmiştir. e = 8114231289041741.

İspat;

Veri = 2121

$c = 2121^5 = mod(10142789312725007) = 2353154715334573$ Şifreli veri.

$m = 2353154715334573^{\,8114231289041741} = mod(10142789312725007) = 2121$

Yukarıdaki görüldüğü üzere temel RSA anlayışındaki şifreli veriyi çözmeye yarayan gizli anahtara ulaşılmıştır. Artık şifreli metin çözülebilir.

4.2.3.2. Biyometrik Tabanlı RSA Üzerindeki Açık Anahtar Saldırısı.

Bizim yaptığımız biyometrik tabanlı şifrelemede açık anahtar açık bir şekilde gönderilmemektedir. Buda geliştirdiğimiz sistemin önemli taraflarından biridir. Bu sayede açık anahtar saldırısı temel RSA da gösterildiği gibi kolay olmayacaktır.

Örnek olarak;

Tablo 4.3: Örnek verilen RSA bilgileri.

p	q	n	Φ
100711409	100711423	10142789312725007	10142789111302176

Yukarıdaki örnekte olduğu gibi n = 10142789312725007 olarak seçilmiştir. Asal sayılarda bir önceki bölümde olduğu gibi seçilmiştir. Biyometrik tabanlı RSA algoritmasında açık anahtarın seçimini hatırlayacak olursak açık anahtar üretilen tekil anahtardan elde edilir. Örnek bir tekil anahtar havuzu Tablo 4.3 de verilmiştir. Oradan 16. Sırada olan 48000 sayısını seçtiğimizi düşünelim. Bu sayı baz alındığında açık anahtar 48001 olarak bulunur. Çünkü açık anahtar ile Φ sayısının en büyük ortak böleni 1 olmalıdır (ebob(e,Φ)=1). Matlab kodu Ek kısmında, Ek-2 adlı başlığın altında verilmiştir.

Tarihi **08122016**1220 olarak farz edelim.
getNumber(**08122016**) $= yyyy^3 + mm^2$+ dd = 8193540248.
e_{pseudo} = 8193540248 * 48001 = 393298125444248 olarak sahte açık anahtar.

Tablo 4.4: Elegeçirilen n ve açık anahtar çifti.

n	Açık anahtar
10142789312725007	08122016393298125444248

Açık anahtarımız bir çerçeve şeklinde gönderilmektedir. Bu çerçevenin formatı şu şekildedir, Tarih + publicKey.

Örnek olarak ; 08122016393298125444248.

Buradaki tarih kısmı fark edilebilir fakat elde edilecek olan açık anahtar 393298125444248 gerçek anahtar olmadığı için n sayısını oluşturan asal sayılar bulunsa bile şifreli veriyi açmak için kullanılan gizli anahtara ulaşım gerçekleşmez. Kişi fonksiyonu ele geçirmekte zorunda kalır, aksi takdirde bu saldırıda boşa çıkmış olacaktır

5. TARTIŞMA VE SONUÇ

Günümüzde, internet bir vazgeçilmez haline gelmiştir. İnternetin maliyetinin düşmesi, interneti iletişim için kullanımını daha cazip hale getirmektedir. Ödemeler ve başvurular artık internet üzerinden daha basit ve esnek bir şekilde yapılabilir tabi bu da beraberinde güvenlik problemlerini getirmektedir.

Hayatımızda internet kullanımına güzel bir örnek olarak internet bankacılığını ele alabiliriz. Bugünkü internet bankacılığı ilk etapta dört ya da altı haneli şifre ve daha sonrasında telefona gelen kod numara ile giriş üzerine kurulmuştur. Dört haneli bir şifreyi tahmin etmek ya da ele geçirmek çok zor olmasa gerek, ama biyometrik bir parametreyi ele geçirmek ve kullanmak çok zor, tahmini ise neredeyse imkansız hale gelecektir. Öbür taraftan, bu kısa şifrelerin hatırlanması daha basit bir şey değildir, herhangi bir şifremizi unuttuğumuz durumlar birçok kişinin başına gelmiştir. Biyometrik parametreli şifreleme yöntemi sayesinde şifre ezberlemeye gerek kalmadan güvenilir hatırlamaya gerek kalmayan bir şifre edinilmesi mümkün hale gelecektir.

Bu çalışmayı hazırlarken, temelde K. Ankit ve J. Rekha ikilisinin çalışması bizim için ilham kaynağı olmuştur [8]. Bu kaynakta RSA açık ve kapalı anahtar üretimi parmak izi biyometrik parametresine bağlı olarak yapılmıştır. Bu bağlamda parmak izinde bulunan detaylar baz alınarak bir matris yaratılmış ve bu matris üzerinden anahtar üretimi yapılmıştır. Bizim çalışmamızda genel bir matematiksel formül yardımı ile resim şeklinde olan her biyometrik parametre için kullanışlı bir algortima meydana getirilmiştir.

Bu tez çalışmasında RSA algoritması üzerine biyometrik parametre ekleyerek gönderilecek açık anahtarın artık biyometrik parametreye bağlı olarak gönderilmesi sağlanmıştır. Bu şekilde olan açık anahtarın daha güvenli olduğu kriptoanaliz bölümünde gösterilmiştir. Bu işlemi yaparken aynı zamanda zaman baz alınmaktadır. Tarihteki saat ve dakika bizim hangi anahtarı seçeceğimizi belirleyerek her defasında aynı anahtarın gönderilmesinin önüne geçer, böylelikle daha güçlü bir şifreleme sistemi oluşturulmuş olur.

Gelecekteki çalışmamız yine biyometrik parametreler üzerine olup, gelecekte her bireyin tekil olan DNA parçası üzerinden yapılacaktır. Bunun nedeni DNA hariç diğer biyometrik parametreler olan parmak izi, iris, avuç içi ve diğer biyometrik parametrelerin zamanla değişime uğrayan biyometrik parametreler olmasıdır. Bizim, bu tip çalışmalar için hiçbir zaman değişmeyen ve insan hücrelerinin tümünde aynı olan, beşikten mezara kadar tekillik arz eden bir anahtara ihtiyacımız vardır. Bu yüzden, çalışmamız DNA biyometrik parametresi olarak yapılması gerekmektedir. DNA şifreleme sisteminde en büyük engel olarak karşımıza büyük sarmal yapısının bir anda çözülememesi şeklinde çıkmaktadır. Buna rağmen gelecekte DNA kimlik tanımlamada ve güvenli iletişimi sağlamada başta rol oynayacağı şimdiden kesin gözü ile bakılmaktadır. DNA'nın insanlar arasında sadece 0.1% gibi bir oranı farklı olarak görülmektedir, yaklaşık olarak bir insanın 3 milyar genomu olduğu düşünüldüğünde bizi 3 milyon genom grubu ilgilendirmektedir. Geri kalan baz çiftleri insanı insan yapan özellik genleridir [24]. Bu DNA parametresinin tüm alanlarda kullanılmasının önündeki en büyük engel olarak bugünkü işlemcilerin DNA sarmalını etkin bir sürede çözmeyeceği gerçeğidir. Bu durum kuantum bilgisayarların devreye girilmesi ile ortadan kalkabileceği düşünülmektedir.

KAYNAKLAR

[1]. Mohammed, A. F., 2016, Biometric based authentication using two-stage fingerprint privacy protection for file storage on server, *International journal of computer science and mobile computing*, 5 (3), 377 – 387.

[2]. Guo, F., Susilo, W. ve Mu, Y., 2016, Distance-Based Encryption: How to embed fuzziness in biometric-based encryption, *Transactions on information forensics and security*, 11 (2), 247 – 257.

[3]. Chakraborty, S. ve Bandyopadhyay, S. K., 2016, Emerging biometric technology-A review, *International journal of advances in computer science and technology*, 5 (1), 8 – 22.

[4]. Murakami, T., Ohki, T. ve Takahashi, K., 2016, Optimal sequential fusion for multibiometric cryptosystems, *National institute of advanced industrial science and technology*, 11 (3), 1 – 16.

[5]. Joymala, O. ve Khare, N. Securing, 2016, A smart home network using voice biometric, *International journal of application or Innovation in Engineering & Management*, 5 (2), 113 – 118.

[6]. Zhao, T., Ran, Q., Yuan, L., Chi, Y. ve Ma, J., 2015, Image encryption using finger print as key based on phase retrieval algorithm and public key cryptography, *Optics and Lasers in Engineering*, 72 (3), . 12 – 17.

[7]. Ramesh, A. ve Setty, S. P., 2013, Analysis on biometric encryption using RSA Algoritm, *International journal of multidisciplinary educational research*, 2 (11), 303 – 307.

[8]. Ankit, K. ve Rekha, J., 2016, Biometrics as a Cryptographic Method for Network Security, *Indian journal of science and technology*, 9 (22), 2 – 6.

[9]. Kumar, Y., Munjal, R. ve Rekha, J., 201, Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities ve Countermeasures, *International journal of computer science and management studies*, 11 (3), 2 – 6.

[10]. Jain, A. K., Feng, J. ve Nandakumar, K., *Fingerprint Matching*, http://biometrics.cse.msu.edu/Publications/Fingerprint/JainFpMatching_IEEEComp10.pdf, [Ziyaret tarihi: 5 Şubat 2017].

[11]. Mahajan, P. ve Sacheva, A., 2013, A study of encryption algorithms AES, DES and RSA for security, *Global journal of computer science and technology network*, 13 (15), 14 – 22.

[12]. Manoria, M., Shrivastava, A. K., Thakur, S. S. ve Sinha, D., 2011, *Secure biometric cryptosystem for distributed system*, http://www.idc-

online.com/technical_references/pdfs/information_technology/Secure%20Biometric.pdf , [Ziyaret tarihi: 28 Şubat 2017].

[13]. Hao, F., Anderson, R. ve Daugman, J., 2005, *Combining cryptography with biometrics effectively*, http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-640.pdf, [Ziyaret tarihi: 10 Mart 2017].

[14]. Chandra, S., Paul, S., Saha, B. ve Mitra, S., 2013, Generate an encryption key by using biometric cryptosystems to secure transferring of Data over a Network, *Journal of Computer Engineering*, 12 (1), 16 – 22.

[15]. Rathgeb, C. ve Uhl, A., 2011, A survey on biometric cryptosystems and cancelable biometrics, *Journal on Information Security*, 1 (3), 1 – 25.

[16]. Jin, Z., Teoh, A. B. J., Goi, B. ve Tay, Y., 2016, *A new biometric key binding and its implementation for finger print minutiae-based representation*, http://www.sciencedirect.com/science/article/pii/S0031320316000959, [Ziyaret tarihi: 14 Mart 2017].

[17]. Boneh, D., 2017, *Twenty years of attacks on the RSA Cryptosystem*, https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf, [Ziyaret tarihi: 3 Mart 2017].

[18]. Uludag, U., Pankanti, S., Prabhakar, S. ve Jain, A. K., 2004, *Biometric cryptosystems: issues and challenges, Proceedings of The IEEE*, 92 (6), 14 – 22.

[19]. Kumar, Y., Munjal, R. ve Sharma, H., 2011, Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures, *International journal of computer science and management studies*, 11 (3), 60 – 63.

[20]. Ogiela, L., Ogiela, M. R. ve Ogiela, U., 2011, Comparison of biometric and linguistic secret sharing protocols, *Lecture notes on data engineering and communications technologies*, 2 (3), 501 – 505.

[21]. Panchal, G. ve Samanta, D., 2016, Comparable features and same cryptography key generation using biometric fingerprint image, *Advances in electrical, electronics, information, communication and bio-informatics*, 16 (2), 50 – 55.

[22]. Balakumar, P. ve Venkatesan, R., 2012, A Survey on biometrics based cryptographic key generation schemes, *International journal of computer science and information technology & security*, 2 (1), 80 – 85.

[23]. Srivastava, P., 2003, Drug metabolism and individualized medicine, *Division of pharmacokinetics and drug metabolism central drug research institute*, 4 (1), 33 – 44.

EKLER

Ek -1 : Gizli anahtarı bulmak için kullanılan matlab kodu.

```
clc
d = uint64(8114231289000000);
e = uint64(5);
f = uint64(10142789111302176);
x = 1;
while x
   if mod(d*e,f)==1
      d
      x = 0;
   end;
   d = d + 1;
end;
```

Ek -2 : Biyometrik tabanlı RSA için açık anahtarı bulmakta kullanılan matlab kodu.

```
clc
d = double(48000);
f = double(10142789111302176);
x = 1;
while x
   if ebob(d,f)==1
      d
      x = 0;
   end;
   d = d + 1;
end;
```

Ek -3 : Parmakizi resminden tekil anahtar üretme grafiği matlab kodu.

```
function createfigure1(X1, Y1)
%CREATEFIGURE1(X1,Y1)
%  X1:  vector of x data
%  Y1:  vector of y data

%  Auto-generated by MATLAB on 20-Feb-2017 21:08:02

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1,'YTick',[0 128 192 256 320 384 512],...
   'YGrid','on',...
   'XTickLabel',{'0','64x64','64x128','64x256','128x64','128x128','128x256','256x256'},...
   'XTick',[0 10 20 30 40 50 60 70],...
   'XGrid','on',...
   'FontName','Imprint MT Shadow');
box(axes1,'on');
hold(axes1,'all');

% Create plot
plot(X1,Y1,'Parent',axes1,'MarkerSize',8,'Marker','o','LineWidth',2,...
   'LineStyle','--',...
   'Color',[0 0.498039215686275 0],...
   'DisplayName','Matriksel Yaklasim');

% Create xlabel
xlabel('Resim Boyutu','FontSize',11,'FontName','Imprint MT Shadow');

% Create ylabel
ylabel('Tekil anahtar uzunluğu (Karakter)','FontSize',11,...
   'FontName','Imprint MT Shadow');

% Create title
title({'Resimdem Tekil Anahtar Üretimi'});

% Create legend
legend1 = legend(axes1,'show');
set(legend1,...
   'Position',[0.688511326860838 0.131868131868131 0.20752427184466
0.0490842490842491]);
```

Ek -4 : Parmak izi ve matrisel yaklaşımın karşılaştırılması.

```
X1 = [20 25 30 35 40 45 50 55 60 65 70];
Y1 = [20*4 25*4 30*4 35*4 40*4 45*4 50*4 55*4 60*4 65*4 70*4];
Y2 = [192 192 192 192 192 192 192 192 192 192 192];

% Create multiple lines using matrix input to plot
plot1 = plot(X1,Y1,X1,Y2,'LineWidth',2);
set(plot1(1),'MarkerSize',8,'Marker','v','LineStyle','--');
set(plot1(2),'MarkerSize',8,'Marker','o','LineStyle','-.');

xlabel({'Parmak izi Detay Sayısı'});

% Create ylabel
ylabel({'Karakter Sayısı'});

% Create title
title({'64x128 Tekil Anahtar Karşılaşırması'});

grid on
```

Ek -5 : Parmak izi ve matrisel yaklaşımın performans değerlendirmesi.

```
clc;

hh = [64 64 64 128 128 256];
ww = [64 128 256 128 256 256];

xx = [1 2 3 4 5 6];

for j = 1:6

Height = hh(j);
Weight = ww(j);
avarage = 0;
im = imread('fingerPrint1.jpg'); %% numbers.png  numbers3.jpg
rImage = imresize(im,[Height,Weight]);

for x = 1 :Height
   for y = 1 : Weight
      rImage(x, y);
   end
end

avarage = ceil(mean(mean(rImage)))

for x = 1 :Height
   for y = 1 : Weight
      if rImage(x, y) < avarage
         rImage(x, y) = '1';
      else
         rImage(x, y) = '0';
      end;
   end
end

rImage = char(rImage);

d = clock
for p = 1 :1

   for x = 1 :Height
      coloumnValue = 0;
      for y = 1 : Weight
         coloumnValue = coloumnValue + str2num(rImage(x, y));
      end
      coloumnValues(x) = coloumnValue;
```

```
   end
```

Ek -5 (Devamı)

```
 for x = 1 :Weight
       rowValue = 0;
       for y = 1 : Height
          rowValue = rowValue + str2num(rImage(y, x));
       end
       rowValues(x) = rowValue;
   end
end

c = clock

timeInterval = c-d;
timeInterval(6);

intervalMat(j) = timeInterval(6);

d = clock
for p = 1 :1
   for x = 1 :Height
      for y = 1 :Weight
         for z = 1:12
            if (minutiae(x,y,z,rImage))
               z;
            end
         end
      end
   end
end

c = clock
timeIntervalMinutiae = c-d;
timeIntervalMinutiae(6);

intervalMinituae(j) = timeIntervalMinutiae(6);

xx(j) = j;
end

intervalMat
intervalMinituae
xx

plot(xx,intervalMat,'--o',xx,intervalMinituae,'-.v','LineWidth',2)
```

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Samet Öztoprak
Doğum Yeri	Kadiköy
Doğum Tarihi	15.02.1989
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	5543435267
E-Posta Adresi	oztoprak.samet@gmail.com
Web Adresi	https://www.linkedin.com/in/samet-%C3%B6ztoprak-30824777/

Eğitim Bilgileri	
Lisans	
Üniversite	Kocaeli Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	19.02.2014

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Programı
Mezuniyet Tarihi	08.05.2017

Makale ve Bildiriler
[1]. Öztoprak, S., Aydın, M. A. ve Atmaca, T., 2016, Energy-Efficiency for heterogeneous wireless networks by using hand-off approach, 10th International conference on application of information and communication technologies, 22 – 26 Nisan 2016 Valensiya, İspanya, ISBN: 978-1-61208-473-2, 40 – 44.