HASH FONKSİYONLARININ ADLİ BİLİŞİMDE UYGULAMALARI ve C++ İLE ŞİFRELEME ALGORİTMASI TASARIMI

HACI HASAN OKUYUCU

2020
YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

Tez Danışmanı
Doç. Dr. Muhammet Tahir GÜNEŞER

HASH FONKSİYONLARININ ADLİ BİLİŞİMDE UYGULAMALARI ve C++ İLE ŞİFRELEME ALGORİTMASI TASARIMI

Hacı Hasan OKUYUCU

T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır

Tez Danışmanı
Doç. Dr. Muhammet Tahir GÜNEŞER

KARABÜK
Ekim 2020

Hacı Hasan OKUYUCU tarafından hazırlanan "HASH FONKSİYONLARININ ADLİ BİLİŞİMDE UYGULAMALARI ve C++ İLE ŞİFRELEME ALGORİTMASI TASARIMI" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Muhammet Tahir GÜNEŞER .....
Tez Danışmanı, Elektrik-Elektronik Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Elektrik-Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 12/10/2020

Ünvanı, Adı SOYADI (Kurumu)	İmzası
Başkan : Prof. Dr. Mehmet KARALI (NEÜ)	.....
Üye : Doç. Dr. Muhammet Tahir GÜNEŞER (KBÜ)	.....
Üye : Doç. Dr. Ziyodulla YUSUPOV (KBÜ)	.....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Hasan SOLMAZ .....
Lisansüstü Eğitim Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Hacı Hasan OKUYUCU

ÖZET

Yüksek Lisans Tezi

HASH FONKSİYONLARININ ADLİ BİLİŞİMDE UYGULAMALARI ve C++ İLE ŞİFRELEME ALGORİTMASI TASARIMI

Hacı Hasan OKUYUCU

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı:
Doç. Dr. Muhammet Tahir GÜNEŞER

Ekim 2020, 70 sayfa

Bu çalışmada, bilişim sistemlerinde, elektronik delillerin incelenmesinden önce hesaplanan ve yalnızca ait olduğu delile özgü olan hash özetleme fonksiyonlarının matematiksel algoritması ve C++ programlama diliyle gerçekleştirilmesi yapılmıştır. Bu tezde öncelikle; bilişim, adli bilişim, elektronik delil, hash (özetleme) fonksiyonu gibi kavramların literatürdeki karşılıklarından bahsedilmiştir. Daha sonra adli bilişimde inceleme yapılması için delillerin uygun yazılım ve donanımlarla imajlarının alınmasından bahsedilmiş olup, hash fonksiyonları da bu aşamadan sonra önem kazanmıştır. Hash fonksiyonlarının kullanım amaçlarının başında veri güvenliği ve kimlik doğrulaması gelmektedir. İmajı alınan bir verinin hash değeri hesaplatıldıktan sonra imaj üzerinde bir değişiklik yapılması halinde hesaplatılan hash değeri de değişecektir ve bu sayede de veri üzerinde değişiklik yapılıp yapılmadığına bakılarak söz konusu verinin güvenliği sağlanmış olacaktır.

Bu tezde, C++ programlama dili yardımıyla, ekrana rastgele ve uzunluk fark etmeksizin girilen bir verinin hash değeri hesaplatılıp ekrana yazdırılmış ve kayıt altına alınmış daha sonra bu veride küçük bir değişiklik yapıp tekrardan hash değeri hesaplatılmış ve oluşan iki hash değerinin farklı olduğu görülmüştür. Böylece program sayesinde bir veri üzerinde yapılan değişiklik tespit edilmiştir.

Anahtar Sözcükler **:** Bilişim, Adli bilişim, Dijital belil, Hash fonksiyonu, Hash algoritması.

Bilim Kodu **:** 90505

ABSTRACT

M. Sc. Thesis

APPLICATIONS OF HASH FUNCTIONS IN FORENSIC COMPUTING AND ENCRYPTION ALGORITHM DESIGN WITH C++

Hacı Hasan OKUYUCU

Karabük University
Institute of Graduate Programs
Department of Electric and Electrıcal Engineering

Thesis Advisor:
Assoc. Prof. Dr. Muhammet Tahir GÜNEŞER
October 2020, 70 pages.

This information is for the realization of the hash summarization functions, which are calculated only for the examination of electronic evidences in the information systems and which are specific to the evidence that belongs to them, with the mathematical algorithm and C ++ programming language. In this thesis, firstly; information such as informatics, forensic informatics, electronic evidence, hash (summarizing) function in the literature are mentioned. Then, in forensic informatics, it is talked about obtaining the images of the evidence with appropriate software and hardware for the examination, and it has gained importance after this stage in the hash stage. It is for data security and authentication at the beginning of the use of hash functions. After calculating the hash value of a data received in the image, there may be a change in the image, while calculating the hash value will change, and it is checked whether it is used on the data. In this thesis, C ++ programming language options, the screen is random, regardless of the length, the hash value of a data

entered is calculated, usable, printed and recorded later in this data. This is done on a data thanks to the program.

Key Word **:** Computing, Forensic computing, Dıgıtal evidence, Hash function, Hash algorithm.

Science Code : 90505

TEŞEKKÜR

Bu tez çalışmasının planlanması ve yürütülmesinde, desteklerini hiçbir zaman esirgemeyen ve tezi bitirmem konusunda bolca motivasyon sağlayan değerli hocam ve danışmanım Sayın Doç. Dr. Muhammet Tahir GÜNEŞER'e sonsuz teşekkürlerimi sunarım.

Manevi desteğini her zaman hissettiğim, sevgili oğlum Kerem Fazıl'ın annesi sevgili eşim Sinemcan OKUYUCU'ya ve eğitim hayatımda kelimelerle ifade edemeyeceğim emekleri ve katkıları olan annem Serpil OKUYUCU'ya, babam Bahri OKUYUCU'ya ve kardeşlerime sonsuz teşekkürlerimi sunarım.

Tez yazımı konusunda tecrübelerini esirgemeyen değerli meslektaşım ve arkadaşım Yasin GENÇ'e katkılarından dolayı teşekkürlerimi sunarım.

Tanıştığım günden beri samimiyetini hissettiğim ve kardeşim bildiğim, üniversite hayatımın güzel geçmesine vesile olan yakın arkadaşım Adem KOŞAR'a teşekkürlerimi sunarım.

Sahip olduğu maddi ve manevi imkânlar yardımıyla, 4 çocuğunu da okutup ülke hizmetine adayan, gerek vizyonuyla gerek eğitim anlayışıyla geldiğim noktada çok büyük payı olan babam Bahri OKUYUCU'ya özel teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

KABUL ii
ÖZET iv
ABSTRACT vi
TEŞEKKÜR viii
İÇİNDEKİLER ix
ŞEKİLLER DİZİNİ xii
ÇİZELGELER DİZİNİ xiv
SİMGELER VE KISALTMALAR DİZİNİ xv

BÖLÜM 1 1
ADLİ BİLİŞİM TANIMI VE ADLİ BİLİŞİMİN SAFHALARI 1
1.1. BİLİŞİM KAVRAMI 1
1.2. BİLİŞİM SUÇU KAVRAMI 1
1.3. ADLİ BİLİŞİM KAVRAMI 2
1.4. ADLİ BİLİŞİMİN AŞAMALARI 3
1.4.1. Delil Toplama 5
1.4.2. Tanımlama/İnceleme 6
1.4.3. Çözümleme/Değerlendirme 7
1.4.4. Raporlama/Sunum 7
1.5. DİJİTAL DELİL 8
1.5.1. Dijital Delil Tanımı 8
1.5.2. Dijital Delillerin Bulundukları Yerler 9
1.5.3. Dijital Delil Kapsamı Ve Özellikleri 9
1.5.4. Dijital Delilllerin Manipule Edilmemesi 10

BÖLÜM 2 12
DİJİTAL DELİLLERİN ELDE EDİLMESİ VE İNCELENMESİ 12
2.1. İMAJ 12

Sayfa

2.2. YAZMAYA KARŞI KORUMA (Write Blocker) 13
2.3. İMAJ FORMATLARI 14
2.3.1. E01 Formatı 14
2.3.2. DD/RAW Formatı 15
2.3.3. AFF Formatı 15
2.4. İMAJ ALMA/OLUŞTURMA 15
2.4.1 İmaj Alma Yazılımları 15
2.4.1.1. FTK Imager 15
2.4.1.2. Encase Forensic Image 20
2.4.1.3. Forensic Image 21
2.4.1.4. Tableu Image 22
2.4.1.5. "dd" Komutu 22
2.4.1.6. Guymager 23
2.4.2. İmaj Alma Donanımları 23
2.4.2.1.Tableu TD2 23
2.4.2.2.Tableu TD3 24
2.4.2.3. DİTTO 25
2.5. İMAJI ANALİZ ETME YAZILIMLARI 25
2.5.1. Encase Forensic Yazılımı 25
2.5.2. Forensic ToolKit (FTK) Yazılımı 26
2.5.3. Cellebrate UFED 27
2.5.4. XRY Yazılımı 29
2.5.5. Forensic Explorer Yazılımı 30
2.5.6. The Sleuth Kit ve Autopsy Yazılımları 31

BÖLÜM 3 32
KRİPTOGRAFİK HASH FONKSİYONLARI 32
3.1. KRİPTOLOJİ VE KRİPTOGRAFİ NEDİR? 32
3.1.1. Kriptoloji 32
3.1.2. Kriptoloji Tarihi 32
3.2. HASH FONKSİYONU NEDİR? 35
3.3. HASH FONKSİYONLARININ MATEMATİKSEL ALGORİTMASI 37

Sayfa

3.3.1. MD5 Algoritması ve Güvenilirliği 37
3.3.1.1. MD5 Algoritması 37
3.3.1.2. Algoritmanın Güvenilirliği 39
3.3.2. SHA1 Algoritması ve Güvenilirliği 40
3.3.2.1. SHA1 Algoritması 40
3.3.2.2. Algoritmanın Güvenilirliği 42
3.3.3. MD5 ve SHA1 Algoritmalarının Karşılaştırılması 43
3.4. HASH FONKSİYONLARININ KULLANIM ALANLARI VE UYGULAMADA KARŞILAŞILAN PROBLEMLER 43
3.4.1. Hash Fonksiyonlarının Kullanım Alanları 43
3.4.2. Hash Fonksiyonlarının Uygulamadaki Problemleri 44
3.4.2.1. Sabit Disklede (HDD)'lerde karşılaşılan Hash Problemler 44
3.5. DELİL BÜTÜNLÜĞÜ VE HASH FONKSİYONU 46

BÖLÜM 4 51
C++ PROGRAMLAMA DİLİ İLE BİR VERİNİN HASH DEĞERİNİ HESAPLAYAN PROGRAM ALGORİTMASI 51
4.1. PROGRAMIN ALGORİTMASI ve ÇALIŞMASI 51
4.1.1.Programın Algoritması 51
4.1.2. Programın Çalışması 60
4.2. PROGRAMIN DOĞRULUĞU ve SONUÇLAR 62

BÖLÜM 5 66
SONUÇLAR 66
KAYNAKLAR 68

ÖZGEÇMİŞ 70

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Yazma korumanın çalışma şekli. 14
Şekil 2.2. FTK Imagerden imajı başlatma adımı. 16
Şekil 2.3. FTK Imagerden imajı başlatma adımı 2. 17
Şekil 2.4. FTK Imagerden imajı başlatma adımı 3. 17
Şekil 2.5. FTK Imagerden imajı başlatma adımı 4. 18
Şekil 2.6. FTK Imagerden imajı başlatma adımı 5. 18
Şekil 2.7. FTK Imagerden imajı başlatma adımı 6. 19
Şekil 2.8. FTK Imagerden imajı başlatma adımı 7. 19
Şekil 2.9. FTK Imagerden imajı başlatma adımı 8. 20
Şekil 2.10. Encase Forensic Imager yazılımı arayüzü. 21
Şekil 2.11.Tableu TD2 cihazı görünümü. 24
Şekil 2.12. Tableu TD3 cihazı görünümü. 24
Şekil 2.13.Ditto cihazı görünümü. 25
Şekil 2.14. UFED yazılımı arayüz görünümü. 29
Şekil 2.15. Forensic Explorer programının arayüz görünümü. 31
Şekil 3.1.Örnek olarak alınmış 400 bitin 512 bit olarak ifade edilmesi. 38
Şekil 3.2.MD5 Algoritması çalışma mantığı. 38
Şekil 3.3.SHA1 Algoritması çalışma mantığı. 41
Şekil 3.4.Algoritmasının ilk adımı. 41
Şekil 3.5.Bir Sabit diskin iç yapısı 44
Şekil 3.6. İçine herhangi bir veri yazılmamış boş word belgesi. 47
Şekil 3.7. Boş dökümanın hash değeri. 47
Şekil 3.8. Karabük1 yazısı eklenmiş word dökümanı. 48
Şekil 3.9. Karabük1 yazılmış word dökümanının hash değeri. 48
Şekil 3.10. Karabük2 yazılmış word dökümanı. 49
Şekil 3.11. Karabük2 yazılmış word dökümanının hash değeri. 49
Şekil 4.1. Algoritmanın akış diyagramı. 59
Şekil 4.2. C++ yardımıyla yazılan algoritmanın basite indirgenmiş hali. 61
Şekil 4.3. Orijinal metin ve Hash değeri. 62
Şekil 4.4. Örnek 1 ve Hash değeri. 62

Sayfa

Şekil 4.5. Örnek 2 ve Hash değeri. 63
Şekil 4.6. Örnek 3 ve Hash değeri. 63
Şekil 4.7. Örnek 4 ve Hash değeri. 63
Şekil 4.8. Örnek 5 ve Hash değeri. 64
Şekil 4.9. Örnek 6 ve Hash değeri. 64

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 3.1. Sezar şifrelemesi giriş-çıkış tablosu..... 33
Çizelge 3.2. Harflerin kullanılma sıklıkları..... 34
Çizelge 3.3. Hash değerlerinin hesaplanması. 36
Çizelge 3.4. Oluşan hash değerlerinin karşılaştırılması. 50
Çizelge 4.1. Hash değerlerinin karşılaştırılması. 64

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

CD : Compact Disk
CMK : Ceza Muhakemeleri Kanunu
DCO : Device Configuration Overlay
DVD : Digital Versatile Disc
FAT : File Allocation Table
FTK : Forensic Tool Kit
HPA : Host Protected Area
IDE : Integrated Development Environment
MB : Mega Byte
MD : Message Digest
NTFS : New Technology File System
PATA : Parallel Advanced Technology Attachment
RAM : Random Access Memory
SATA : Serial Advanced Technology Attachment
SCSI : Small Computer System Interface
SD : Solid Disk
SHA : Secure Hash Algorithm
SSD : State Solid Disk
USB : Universal Serial Bus

BÖLÜM 1

ADLİ BİLİŞİM TANIMI VE ADLİ BİLİŞİMİN SAFHALARI

1.1. BİLİŞİM KAVRAMI

Bilişim; günümüzde; teknik, ekonomik ve toplumsal ilişkilerde kullanılan bilginin akla uygun ve düzenli bir biçimde başta bilgisayarlar olmak üzere benzeri elektronik aygıtlar aracılığıyla işlenmesi olarak tanımlanmaktadır [1].

1.2. BİLİŞİM SUÇU KAVRAMI

İlk çağdan günümüze kadar insan hayatını doğrudan ve dolaylı olarak etkileyen birçok faktör olmuştur. Bu faktörlerin başında teknoloji gelmektedir. Teknoloji gerek ortaya çıkması gerekse gelişmesi bakımından da insan hayatına birçok kavramın girmesine sebep olmuştur. Ayrıca teknoloji geliştiği için birçok alanda örneğin; sağlık, ekonomi, finans gibi alanlarda köklü değişikleri beraberinde getirmiştir. Bu köklü değişiklikler temelde hayatımızı kolaylaştırırken öte yandan da bazı tedbirleri almamızı zorunlu kılmıştır. Örneğin bankacılık alanında, bir bankanın sahip olduğu müşteri portfolyosunun yanlış kişilerin eline geçmemesi için bu bilginin olağanüstü şekilde korunması zorunluluğunu doğmuştur. Dolayısı ile de insan yaşamında "bilgi"nin de önemli bir güç olabileceği ve korunması zorunluluğu çıkmıştır. Gündeme önemli bir güç olarak giren "bilgi" nin işe yaraması için doğruluğu, bütünlüğü ve gizliliği gibi temel sorunları da beraberinde getirmiştir.

Bilgi kavramı, önemli bir güç ve silah olduğundan, onu alıp kötü emelleri için kullanmak isteyen şahısların da hedefi haline gelmiş ve bilginin muhafazası ve paylaşımında güvenlik çok daha ciddi bir husus haline gelmiştir. Bu bağlamda, suç ve suçlu kavramı, bütün dünyada

bilinen ve farklı şekillerde tanımlanan bir sonuç olarak ortaya çıkmıştır. Bilişim suçları Türkiye'de ve dünya da farklı şekillerde tanımlanmıştır:

- Bilişim suçları genel olarak, "verilere veya veri işleme bağlantısı olan sistemlere veya sistemin düzgün işlevsel işleyişine karşı, bilişim sistemleriyle işlenen suçlar" şeklinde tanımlanmaktadır.
- Türk Ceza Hukukunda bilişim suçları "Bilişim Alanında Suçlar" ve "Özel Hayata ve Hayatın Gizli Alanına Karşı İşlenen Suçlar" bölümünde işlenmiştir.
- Amerikada ise bu terim "computer-assisted crime" (bilgisayarla işlenen suç), "crimes agains computer" (bilgisayara karşı işlenen suç), "computer-related crime" (bilgisayarla bağlantılı suç) şeklinde tanımlanmıştır [2].

1.3. ADLİ BİLİŞİM KAVRAMI

Adli bilişim; hukuki bir olayın adli vakaya dönüştüğü durumlarda; bu vakanın sonuçlandırılması için kullanılan delillerin elektronik ortamda tanımlanması, tespit edilmesi, incelenmesi ve raporlanması aşamaları ile doğrudan ilişkilidir. Bu bilgiye bakarak adli bilişimi tanımlamak gerekirse, veri olarak elde edilen delillerin tespit edilmesi, analiz edilmesi ve ilgililer tarafından kolayca anlaşılır seviyede sonuca bağlanması aşamalarının tamamı şeklinde tanımlanabilir.

Adli bilişim; bilgisayar ağlarından, veri depolama cihazlarından elde edilen datayı, mahkemenin talebini göz önünde bulundurarak ve mahkemeler tarafından kabul edilebilir ölçüde delilleri toplama, sınıflandırma ve analiz etme ile ilgilenir [3]. Dolayısı ile de, adli bilişim de diğer adli bilimlerde olduğu gibi adaletin sağlanmasına yardımcı olur. Adli Bilişim dünya genelinde "Computer Forensic" olarak kullanılsa da, tam olarak Adli Bilişim kavramını karşılayamamaktadır [4]. Computer Forensic, bilgisayar incelemesi anlamına gelirken; Adli Bilişim ülkemizde, Cep Telefonu, Tablet, Simkart, Server, Fax Cihazı, Telsiz, Switch vs. gibi tüm elektronik cihazları ve ayrıca şifreleme, şifre çözme gibi alanları da

kapsamaktadır. Bu sebepten dolayı, Adli Bilişim Türkiye'de daha geniş bir alana yayılmış denilebilir.

Adli Bilişim, dünya literatüründe de farklı şekillerde tanımlanmıştır:

- Adli Bilişim, bilgisayar biliminin yasal sürece katkıda bulunmak üzere uygulanması bilimi ve sanatıdır.
- Adli Bilişim en basit tanımıyla, bir yargılama esnasında kullanılabilecek potansiyel delillerin belirlenmesi için bilgisayar araştırma ve analiz tekniklerinin kullanılmasıdır. Bilgisayardaki verilerin korunması, tanılanması, çıkarılması, dökümü ve yorumunu içerir ancak bunun yanında hukuki kurallar, süreçler, delillerin bütünlüğü gibi konulara da riayet ederek bulunan veriler hakkında rapor yazılmasını da kapsar.
- Delilin değiştirilmeden ve orijinalinin bozulmadan elde edilmesi, elde edilen verilerin değişime uğratılmadan analiz edilmesi adli bilişimin temelini oluşturmaktadır [5].

1.4. ADLİ BİLİŞİMİN AŞAMALARI

Bir adli vakanın aydınlatılabilmesi için, vakanın meydana geldiği yerde tespit edilen delillerden faydalanılır. Adli Bilişim vakalarında da normal vakalarda olduğu gibi deliller bulunmakta ancak buradaki deliller dijital(sayısal) olarak adlandırılmaktadır [6].

Dijital delil, "Bilişim sistemlerinin ya da bilgileri otomatik olarak işleme tabi tutma yetisine sahip olan elektronik cihazların depolama birimlerinde bulunan ve suç ile ilgili delil sayılabilecek ve bu suçu aydınlatabilecek verilerdir.

Dijital olmayan deliller olay yerinde rahatlıkla tespit edilebilirken, dijital (sayısal) deliller için aynı durum söz konusu olmayacaktır. Tahmin edildiği üzere, olay yerinde fark edilemeyen yada gözden kaçan herhangi bir dijital delil, ciddi manada kayıplara sebep olabilir. Bu yüzden, olay yerinde bulunabilecek her türlü elektronik

cihaz ya da veri, delil olarak değerlendirilmeli ve bu şekilde işlem yapılmalıdır. Ayrıca klasik deliller, inceleme aşamasında iken rahatlıkla kontrol edilebilir gözle görülebilir ve suç oranı tespit edilebilirken, dijital delillerde bu durum söz konusu değildir. Bir dijital delilden suç oranını tespit etmek için o delile bazı işlemler yapılması gerekmektedir [7].

Dijital deliller ile elektronik ortamda doğru ve tarafsız bir analiz yapmak kolay değildir ve klasik delil incelemelerine göre daha karmaşık ve pahalı bir teknolojik altyapı gerektirmektedir. Dijital deliller verinin işlenmiş her türlü çeşidi olabilir.

Türk ve dünya literatüründe dijital (sayısal) delil olarak nitelendirilebilecek verileri şu şekilde sıralayabiliriz:

- Video Görüntüleri
- Fotoğraflar
- Ses Kayıtları
- Yazı Programları
- Bilgisayar Programları
- İletişim Kayıtları
- Gizli/Şifreli Dosyalar
- Cep Telefonlarına İndirilen Uygulamalar
- İnternet Ortamından İndirilen Dosyalar
- Arama Kayıtlar/Cevapsız çağrılar
- Gelen/Gönderilen SMSler
- Kaydedilmiş Notlar ve Mesajlar

Adli Bilişim sistemlerinde ya da adli bilişim vakalarında, mevcut duruma hukuki geçerlilik kazandırmak ve mahkemeye sunabilmek için, olay yerinden tespit edilen dijital delillerin birtakım işlemlerden geçmesi gerekmektedir. Burada bahsedilen işlemler ise adli bilişim aşamaları olarak adlandırılmaktadır. Bu aşamaları temel olarak 4 başlıkta inceleyebiliriz [8].

- Delil Elde Etme ve Toplama
- Tanımlama/İnceleme
- Değerlendirme/Çözümleme
- Raporlama/Sunum

1.4.1. Delil Toplama

Delil toplama aşaması adli bilişimin ilk adımı ve bundan sonraki aşamaların da yalnızca toplanan deliller üzerinden yapılacağı göz önünde bulundurulursa en önemli adımı denilebilir. Burada, delillerin toplama şekli CMK'nın 116. maddesine uygun olarak yapılması gerekmektedir. Ayrıca, yukarıda da bahsedildiği gibi deliller toplanırken her türlü medyanın delil olabileceği düşünülerek hareket edilmelidir.

Dijital delillerin toplanması esnasında, bu delillerden veri kaybı olma ihtimali de göz önünde bulundurularak, öncelikle olay yeri belirlenmeli ve buraya yetkisi olmayan kişilerin girmemesi ve sadece uzman kişilerin delil toplama işlemini yapması gerekmektedir. Böylece sayısal delillerin zarar görmeden toplanması sağlanmış olacaktır. Ayrıca burada, delil toplama esnasında fiziki delillerin de bulunabileceği, örneğin parmak izi, unutulmamalı ve bu delillerin de bütünlüğünün korunması gerekmektedir.

Sayısal deliller, olay yerinde toplanmaya başlanmadan önce fotoğrafları çekilmeli, medyaların konumları belirlenerek not edilmeli, sonrasında ise çalışan cihazlar sırasıyla tespit edilmelidir. Delil toplama sırasında, açık bir cihazın olması, mutlaka daha dikkat edilmesi gereken bir konudur. Çünkü açık cihaz kapatıldığında yeniden açılması için şifre istenecek ve şifre tespit edilemezse delilin imajı alınamayacak ve dolayısıyla incelene yapılamayacaktır. Aynı durum kapalı cihazlar için de geçerlidir. Şöyleki, olay yerinde kapalı olarak bulunan bir cihaz asla açılmamalıdır. Örneğin bir bilgisayar kapalı iken açılırsa, işletim sistemi devreye girecek ve cihazın yapılandırma dosyalarına erişim sağlayacak ve ileride delil sayılabilecek verilere zarar verebilecektir. Dosyalara erişim tarihi de bazen delil olarak sayılabileceği için bu durum oldukça sakıncalıdır. Ayrıca, işletim sistemi açılırken, oluşturabileceği

geçici dosyalar, disk alanında silinmiş verilerin üzerine yazılacak ve buradaki silinmiş verilerin geri getirilememesine sebep olacaktır ve delil bütünlüğü bozulacaktır.

Bütün bu sebeplerden dolayı, olay yerinde bulunan cihazlar özenle tespit edilmeli ve usule uygun olarak toplanmalıdır.

Adli bilişimde inceleme aşamasında, incelenecek veri orijinal üzerinden değil, orijinalinden alınan bir kopya üzerinden yapılmalıdır. Nitekim orijinalin üzerinden yapılırsa delil zarar görebilir ve geri dönülmez biçimde veriler kaybolabilir.

1.4.2. Tanımlama/İnceleme

Delilin birebir kopyası alındıktan sonra Tanımlama/İnceleme safhasına geçilir. Bu safhada, elimizde depolama cihazında mevcut veriler haricinde bir de kullanıcının daha önce cihazında mevcut olan ancak üzerine veri yazıldıkça adresleri silinen veriler bulunmaktadır. Silinen veriler ya da üzerine başka veri yazıldığı için kaybolan veriler de adli bilişimin konusu olup öteki dijital delillerden hiçbir farkı olmayan verilerdir. İnceleme esnasında silinen verilerin de incelemesi yapılacak ve varsa suç unsuru tespit edilmeye çalışılacaktır. Örneğin ziyaret edilen bir web sitesi ya da cep telefonuna indirilmiş ve silinmiş bir uygulama suç unsuru olabilir.

Dijital bir delilin elektronik ortamda adli bilişim yazılım ve donanımları ile bir kopyasının alındıktan sonra elde edilen bu veriye o delilin imajı denir. Buradaki kopya alma işlemi, sadece o anda mevcut olan veriler değil, silinmiş verilerin de bilgisini kapsamaktadır. Bir elektronik delilden alınan kopyaya adli bilişimde o delillin imaj denilmektedir.

İmaj konusu ilerleyen bölümde daha detaylı olarak anlatılacaktır. İmaj tam olarak anlamlı bir veri değildir ve anlamlı hale getirilmesi bu safhada yapılır. Bu yüzden, tanımlama/inceleme safhasına veriye anlam kazandırma safhası da denir. Veriye anlam kazandırma, adli bilişim yazılımları ve donanımları kullanılarak yapılır. Bu

safhada kullanılan adli bilişim yazılım ve donanımları ilerleyen bölümlerde detaylı bir şekilde anlatılacaktır.

1.4.3. Çözümleme/Değerlendirme

Bu aşamada, yapılan incelemeler ve analizler sonucunda bulunan veri türleri ayıklanarak hangi verilerin hangi sebeplerle ve hangi ölçüde yetkili makama sunulacağının belirlendiği aşamadır. Bir imaj dosyasından çıkan her türlü data delil olarak değerlendirilir ancak hepsi suç unsuru olmayabilir. Bu aşamada daha önceden suç unsuru olarak belirlenen ve şüpheliyi "suçlu" hale getirebilecek suç unsuru, elde edilen imaj ile kıyaslanır. Şayet imajdan bir suç unsuru çıkarsa bu durum usule uygun olarak kayıt altına alınır ve adli bilişimin son aşaması olan raporlama aşamasına geçilir. Bu aşamaya ayrıştırma ve temizleme aşaması da denebilir. Ayrıca bu aşama, adli bilişimin söz konusu vaka ile ilgili olarak kurallara uygunluğunun da denetiminin yapıldığı aşamadır.

Sonuç olarak, adli bilişim safhalarının en can alıcı ve suçun net şekilde tespit edildiği aşamadır denebilir.

1.4.4. Raporlama/Sunum

Delil toplama, inceleme ve çözümleme aşamalarından sonra adli bilişimin son adımı olan raporlama aşamasına geçilir. Raporlama aşamasında, delil üzerinde yapılan tüm işlemler raporda mutlaka yer almalıdır. Bunun sebebi, adli bilişimde çok önemli bir konu olan delil bütünlüğüdür. Delil bütünlüğü, delil üzerinde incelenme yapılırken dahi korunması gereken ve herhangi bir değişiklik olması durumunda mutlaka kayıt altına alınması gereken bir konudur. Bununla birlikte, raporlama aşamasında, raporun açık seçik, anlaşılır bir ifadeyle, talep edilen incelemenin ve sorulan sorunun cevabını vermeye yönelik olacak şekilde uzman kişiler tarafından yazılması gerekmektedir. Aslında bu aşama, bir delile başından sonuna kadar hangi işlemlerin yapıldığının ve ilgili makamlarca belirlenmiş suç unsurlarının delil üzerinde ne derece mevcut olduğunun belirtildiği bir sonuç bildirme yazısı olduğundan rapor

yazımına çok dikkat edilmeli ve sözcük seçiminde çağrıştırdığı anlamlar göz önünde bulundurulmalıdır. Olası bir yanlışlığa ve yanlış anlaşılmaya imkân verilmemelidir.

Bütün bu işlemler eksiksiz bir şekilde yapıldıktan ve rapor ilgili makama gönderildikten sonra Raporlama/Sunum aşaması tamamlanmış olur.

1.5. DİJİTAL DELİL

1.5.1. Dijital Delil Tanımı

Türk hukuk sistemi göz önünde bulundurulduğunda, ceza hukuku ve ceza yargılamasında en doğru ve objektif sonuca ulaşmak hukukun en önemli amaçlarındandır. Bu amacı gerçekleştirmek için ise çoğu zaman deliller kullanılır. Bazı yargılama davalarında ise, davayı sonuca bağlayacak tek etmen deliller olacaktır ki bu da zaten çok önemli olan delilleri çok daha önemli bir konuma getirecektir. Türk hukuk sistemine göre delil, "kanıt", ya da " ispat vasıtası" olarak da tanımlanmıştır ve burada bahsi geçen delillerin, gerçeği ortaya çıkarması amacıyla kullanıldığı için şeklen de bir sınırlaması yoktur [9].

Adli bilişim sistemlerinde kullanılan delillerin; yerine getirdiği amaç bakımından normal delillerden hiçbir farkı bulunmamaktadır. Ancak isimlendirme olarak farkları bulunmaktadır. Adli bilişim sistemlerinde kullanılan delillere dijital(sayısal, elektronik) delil denmektedir. Bilişim sistemlerinde, bakıldığı zaman, elde edilen delillerin isimlendirilmesi konusunda standart bir kelime kullanılmamaktadır. Bu bilgi doğrultusunda günümüzde sıkça kullanılan; e-delil, dijital delil ve elektronik delil kelimeleri aynı anlama gelmektedir.

Elektronik deliller, suçun içeriği ne olursa olsun bu suçu açıklığa kavuşturmada kullanılacak elektronik ortamda kayıtlı olan her türlü bilgi olarak da ifade edilebilir. Elektronik deliller fiziksel ve mantıksal olmak üzere iki farklı şekilde kullanılabilirler. Fiziksel anlamda elektronik(dijital) deliller, verinin elle

dokunulabilir bir cihazdaki halini temsil ederken, mantıksal manada ise verinin sanal olarak varlığını ifade eder.

1.5.2. Dijital Delillerin Bulundukları Yerler

Adli bilişim sistemlerinde, dijital deliller; elektronik aygıtların kendisi ya da bu aygıtların depolama birimlerinde tutulan datalar olabilir. Bu durumda da dijital delillerin nerede olabileceği kolayca tahmin edilebilir. Örneğin, bir bilgisayarın hard diskinde, bir flash bellekte, hafıza kartında, SSDlerde, cep telefonlarında, faks cihazında, internet tarayıcı geçmişinde, modemlerde, e-posta kayıtlarında, log kayıtlarında, sohbet ve mesajlaşma programlarında, kablosuz internet noktalarında, elektronik imzalarda vs. gibi içerisinde veri olabilecek her yerde bulunabilirler [10].

1.5.3. Dijital Delil Kapsamı Ve Özellikleri

Dijital delillerin yetkili makam tarafından delil olarak sayılması için ve bütünlüğünün bozulmaması için, teknik olarak taşıması gereken bazı özellikleri vardır. Bu özellikler, akla uygunluk, kabul edilebilirlik, eksiksizlik, gerçeklik, güvenilirlik, manipülasyona uğramamış olması ve tekrar edilebilirlik olarak sıralanabilir. Esasında bu özelliklerin çoğu normal delillerde de bulunmaktadır. Ancak dijital deliller yapısı ve muhteviyatı sebebiyle çok kolay kaybedilebilir, değiştirilebilir ya da delil olma özeliğinden çıkarılabilirler. Bu hassas yapılarından dolayı öteki delillerden farklı şekilde değerlendirilirler. Örneğin dijital deliller, kolaylıkla kopyalanabilir, çoğaltılabilir, manyetik alan ve sıcaklıktan dolayı bozulabilir ve içerikleri kolaylıkla değiştirilebilir oldukları için de bu delillerin muhafazasında çok dikkat etmek gerekmektedir. Bütün bu değişken durumlar göz önünde bulundurularak dijital delillerin kapsamı ve dikkat edilmesi gereken özellikleri adli bilişim sistemlerinde şu şeklide belirlenmiştir [11]:

Dijital Delillerin Bütünlüğü: Bu kavram, bir dijital delilin, her ne şartta olursa olsun daima korunması gerektiği ve orijinal haliyle kalması gerektiği anlamına gelmektedir. Aksi takdirde, delil üzerinde sonradan yapılan incelemeler farklı

sonuçlar verebilir ve bu da itiraz makamlarının elini güçlendirebilir. Bu yüzden dijital deliller üzerinde çok kolay bir şekilde değiştirme, silme ve kopyalama yapılabileceği için bütünlüğünü sağlamak zor olacaktır.

Dijital Delillerin Doğrulanması: Dijital delilin ait olduğu kişinin gerçekten delil sahibi olup olmadığı çok iyi tespit edilmelidir. Çünkü söz konusu dijital delilin kopyasının oluşturulması çok kolay olduğu için bir başkası da farklı bir iddia ortaya atabilir. Bu durum şu şekilde sağlanabilir: Delillerin ilk tespit edildiği ve toplanmaya başlandığı anda, başka şahitlerin de olay yerinde olup toplanan delillerin tutanakla kayıt altına alınması ve delil sahibine de bu tutanağın imzalatılmasıyla sağlanmış olur.

Dijital Delillerin İnkâr Edilememesi: Dijital delillerin doğrulanması ilkesi ile doğrudan ilişkili olup, delil sahibi tarafından delilin içeriği ve özelliklerinin sonradan inkâr edilememesi gerekmektedir. Buradaki farklılık, delillerin içeriğinin de inkâr edilemez oluşudur.

Dijital Delillerin Doğruluğu: Dijital delillerin olay yerinden toplanma şekli, doğru bir teknikle ve doğru kişiler tarafından yapılmalıdır. Türk Ceza Kanununda, delillerin toplanma şekillerinin ve hangi durumlarda geçerliliğinin kaybolacağı gibi durumların ilke ve standartları belirtilmiştir. Adli bilişim sürecinin doğru bir şekilde ilerlemesi için delillerin toplanma şekline ve usullerine dikkat edilmelidir.

Dijital Delillerin Ele Alınabilirliği: Dijital delillerin incelenip en son anlamlı hale getirilme işleminden sonra üçüncü şahısların da anlayabileceği şekilde olması gerekmektedir. Bu ilke inceleme sonuçlarının genellik olması bakımından önemlidir.

1.5.4. Dijital Delillerin Manipüle Edilmemesi

Adli Bilişim sistemlerinde, delillerin korunmasının önemli olduğunu, aksi takdirde delil üzerinde meydana gelecek olası bir değişikliğin ciddi bir sonuç doğuracağı ve delil olmaktan çıkabileceği önceki bölümde ifade edildi. Bir delil üzerinde, dikkat

edilmezse, birçok değişiklik olabilir ve suçun ispat edilmesinde en önemli vasıta olan deliller, değişiklik olması durumunda delil olma özelliğini yitirmiş olur. Bu durumda adaletin tesis edilmesi gecikmiş olur ya da objektif bir değerlendirme yapılamamış olur.

Delillerin, dış etken olmaksızın değiştirilmesinin haricinde, kasten değiştirilebilmeleri de söz konusudur. Bu, adli bilişimde çözülmesi ve aşılması gereken önemli bir konu olup, delillerin kapsam ve özellikleri bölümünde bahsedilen ilkelere uyulması durumunda bu konu önemli ölçüde çözülmüş olacaktır.

Örneğin, bir adli vaka olarak gelen ve disk içerisinde bulunan video görüntülerinin delil olabileceği durumda, delil sahibi video görüntüleri üzerinde kasten değişiklik yapabilir, videonun kayıt edildiği tarihi değiştirebilir ve böyle bir durumda kendini adalet önünde olduğundan daha farklı bir ceza almasına sebebiyet vermiş olabilir. Bu gibi durumların önüne geçmek için, delillerin sahip olduğu ilkelere dikkat edilmelidir.

BÖLÜM 2

DİJİTAL DELİLLERİN ELDE EDİLMESİ VE İNCELENMESİ

Dijital delillerin elde edilmesi, olay yerinden delillerin uzman kişiler tarafından toplanması ile başlar. Usule uygun bir şekilde toplanan deliller kayıt altına alınır ve incelenmek üzere laboratuvar ortamına getirilirler. Laboratuvar ortamında delillerin tasnifi, hasarlı olup olmayışı, içerdikleri veri türü ve büyüklüğü, delilin türü vs. gibi özellikleri göz önünde bulundurularak yapılır. Bu işlemden sonra, dijital deliller inceleme uzmanına gönderilir. İnceleme uzmanı, incelemeyi orijinal veri üzerinden değil, bu verinin bir kopyası üzerinden yapar. Kopya alma işlemi, delil içerisinde mevcut alan ve silinmiş ancak üzerinde veri yazılmamış alanların da bir kopyası olduğu için, bu işleme imaj alma işlemi denilmiştir.

2.1. İMAJ

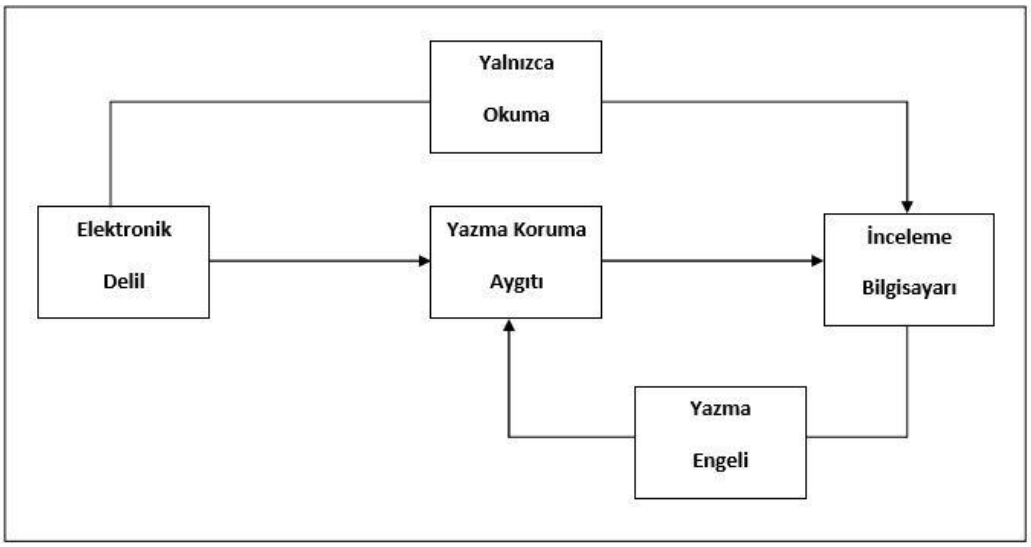
Adli bilişim vakalarında, inceleme yapılabilmesi için ve söz konusu suçun aydınlatılıp sonuca bağlanabilmesi için fiziksel olarak elektronik delil gerekmektedir. İnceleme, delilde bulunan veriler üzerinden yapılacağından dolayı da verilerin açığa çıkarılması gereklidir. İnceleme esasında bu deliller üzerinden de yapılabilir ancak delil üzerinde olası bir değişikliğin meydana gelmemesi için delilin birebir kopyasının alınması ve incelemenin alınan bu kopya üzerinden yapılması gerekmektedir. Bu kopya alma işlemine adli bilişimde imaj alma denilmektedir. Deliller üzerinde bu şekilde birebir kopya alınması işleminde özel yazılımlar ve donanımlar kullanılır. Birebir kopya alma işlemi, hedef diskteki mevcut verileri, silinmiş verileri ve gizli bölümleri de kapsamaktadır. Birebir kopyalama, hedef cihazdan, yani delilden, verilerin her bir sektörünün ayrı ayrı ve tek tek kopyalanması anlamına gelmektedir. Kopya alma işlemi, delil bütünlüğünün bozulmaması için yazma korumalı cihazlar aracılığı ile yapılmalıdır.

Bir adli bilişim uzmanı, inceleme yapmak üzere bir delilin imajını aldığı zaman, bu imajın bir de kopyasını oluşturmalıdır. Nitekim ilk başta oluşturulan orijinal imaj, çeşitli sebeplerden ötürü zarar görebilir. Örneğin; CD ortamına alınan bir imaj, CD'nin çizilmesi ya da bozuk bir CD sürücüsüne takılması ya da zararlı bir virüs programının CD'ye bulaşması sonucu verilere ulaşılamamasına ve dolayısıyla incelemenin yapılamamasına sebep olabilir. Bu yüzden, oluşturulan bir imajın bir de kopyasının oluşturulması gereklidir [12].

2.2. YAZMAYA KARŞI KORUMA (Write Blocker)

Adli bilişimde kopya alma işlemi yapılırken, orijinal veri üzerinde değişiklik yapılmaması ve delil bütünlüğünün sağlanması için yazma korumalı olarak yapılması gerekmektedir. Yazma koruma, bir delilin imaj alma yazılım veya donanımı kullanılarak imajı alınırken, veri akışının tek yönlü, yani; delilden kopyanın oluşturulacağı hedef depolama birimine doğru yapılması demektir. Ancak bu durum, çalışır haldeki delillerden kopya alma işlemi için geçerli değildir. Yazılımsal olarak yazma koruma yapılabilir. Ancak, yazılımsal olarak yapılan yazma koruma, donanımsal olarak yapılacak yazma korumadan daha az güvenlidir. Dolayısıyla yazma koruma işlemini sağlamak için donanımsal cihazın kullanılması daha güvenli bir tercih olacaktır.

Donanımsal olarak kullanılan yazma koruma cihazları daha çok tercih edildiğinden her adli bilişim uzmanının masasında olması gereken bir araçtır. Yazma koruma araçları, genellikle giriş şekillerine göre üretilmektedir. Örneğin, SATA, SCSI, IDE, USB vb. her çeşit giriş için üretilmiş yazma koruma cihazları vardır [13].



Şekil 2.1. Yazma korumanın çalışma şekli.

2.3. İMAJ FORMATLARI

İmaj alma esnasında, her yazılım ve donanım için farklı uzantılara sahip imaj formatları bulunmaktadır. Örneğin bir imaj alma yazılımının kullandığı imaj formatını ya da uzantısını, bir başka yazılım kullanmayabilir. Bunun bazı sebepleri vardır: imaj formatlarından birisi az yer kaplasın diye sıkıştırılarak imaj alırken, bir başka format da imajı ve şifreli bir şekilde alabilir. İmaj alma yazılım ve donanımlarına geçmeden önce, aşağıda bazı imaj formatları anlatılmıştır [14].

2.3.1. E01 Formatı

Bu format, önceki bölümde bahsettiğimiz Encase Forensic programının varsayılan olarak kullanılan formatıdır. Literatürde E01 formatı; Expert Witnes Format adıyla bilinir. İmajı sıkıştırılarak alma seçeneği mevcuttur ve ayrıca metadata bilgisini de tutar. Bu format kullanılarak alınan imajların uzantısı ".e01" dir.

2.3.2. DD/RAW Formatı

Bu imaj formatı, imajı alınan disk ile aynı boyuttadır ve bu yüzden gerçek bit imajı olarak bilinir. Metadata bilgisi tutmayan bu format, imajı sıkıştırma seçeneğine sahip değildir. Ayrıca sahip olduğu imaj uzantısı; ".dd, 001 ya da .img" olabilmektedir.

2.3.3. AFF Formatı

Metadata bilgisi tutma özelliğine sahip olan bu formatta sıkıştırma seçeneği de bulunur ve bu şekilde alınan imajların uzantısı ".AFF" dir.

2.4. İMAJ ALMA/OLUŞTURMA

İmaj alma işlemine adli kopya alma da denilmektedir. İmaj alma işlemi; delillin, imajı alma işlemini yapacak yazılım veya donanıma uygun bağlantı kablosuyla bağlantısı yapıldıktan sonra başlanır. Bir üst bölümde bahsedilen, delilin imajı alınırken, veri yazmayıp sadece okuma yapmasına imkân sağlayan yazma koruma cihazı mutlaka kullanılmalıdır.

İmaj alma/oluşturma işlemi çeşitli yazılım ya da donanımlar kullanılarak yapılabilir. Kullanılan yazılım ve donanımların standartlara uygun olması gerekmektedir. Tezin bu bölümünde yazılım ve donanımların ne oldukları ve nasıl kullanıldıkları anlatılacaktır.

2.4.1 İmaj Alma Yazılımları

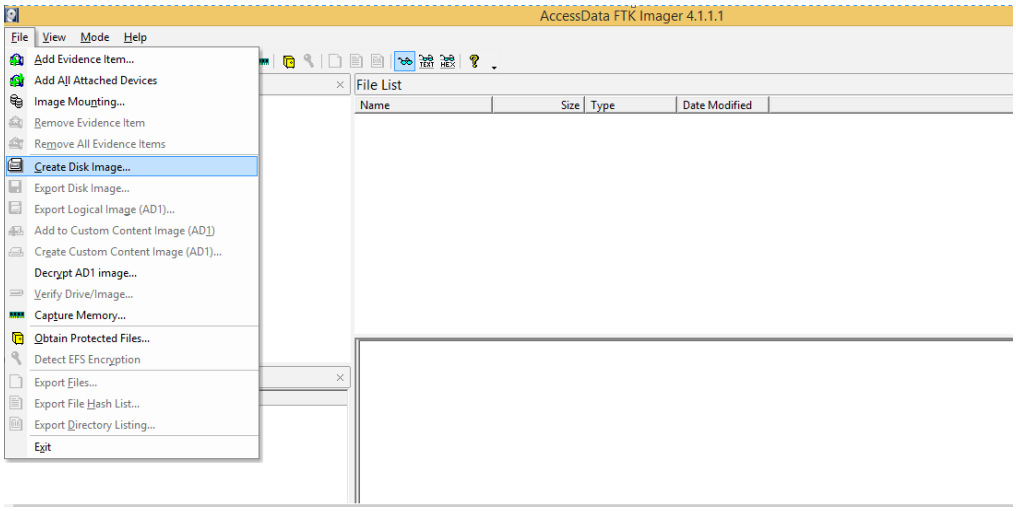
2.4.1.1. FTK Imager

Adli bilişim incelemelerinde en sık yapılan işlemlerden birisi sabit disklerin içerisindeki veriyi herhangi bir değişikliğe uğratmadan imajının alınması işlemidir. Avustralya merkezli AccessData firması tarafından üretilen FTK Imager, delil dosyaları üzerinden inceleme yapma ve imajlarını alma imkânı tanımaktadır. Bu

yazılım sayesinde, sabit disklerin, hafıza kartlarının, flash (taşınabilir) belleklerin, CD ve DVD'lerin ya da seçilen bir dosyanın imajı kolaylıkla oluşturulabilir. Ayrıca bu depolama cihazlarının imajını almadan önce ön izleme yapma imkânı da sağlar.

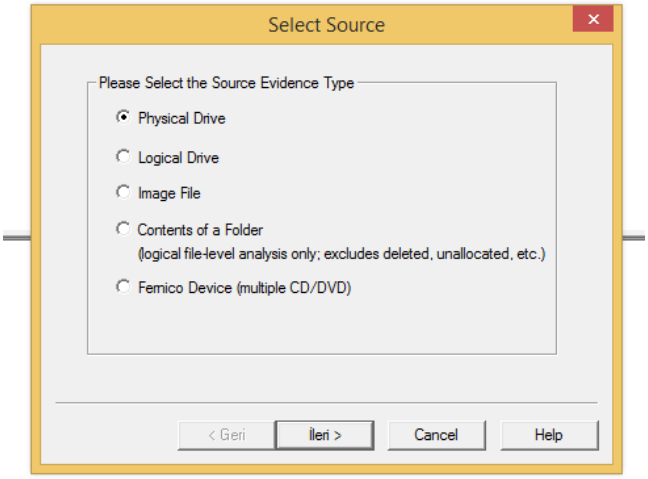
FTK Imager yazılımının bir başka işlevi de, diğer adli bilişim cihazları tarafından oluşturulan imajlar üzerinde inceleme yapabilme ve Windows işletim sistemi aracılığıyla da bu imajların sabit disk gibi kullanılmasına da olanak sağlamaktadır [15].

Yazılım ilk açıldığında File menüsünden "Create Disk Image" seçeneği ile imaj alma işlemine başlanır.



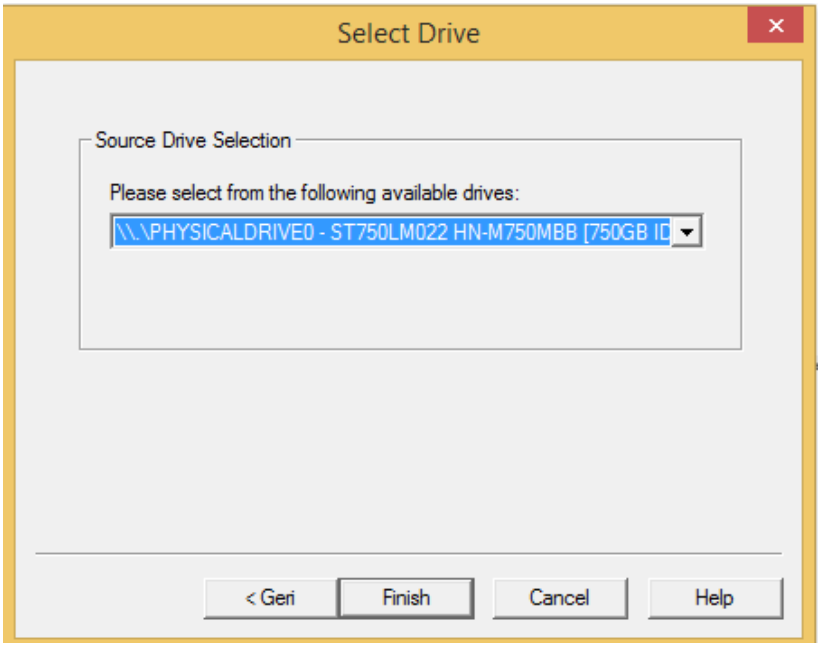
Şekil 2.2. FTK Imagerden imajı başlatma adımı.

Ardından gelen pencerede, "Select Source" menüsünden hangi diskin imajının alınacağının seçilmesi gerekir. Örneğin bilgisayara takılı bir sabit disk ise "Physical Drive", bilgisayardaki bir dosyanın imajı alınacaksa "Contents of a Folder" seçilir.



Şekil 2.3. FTK Imagerden imajı başlatma adımı 2.

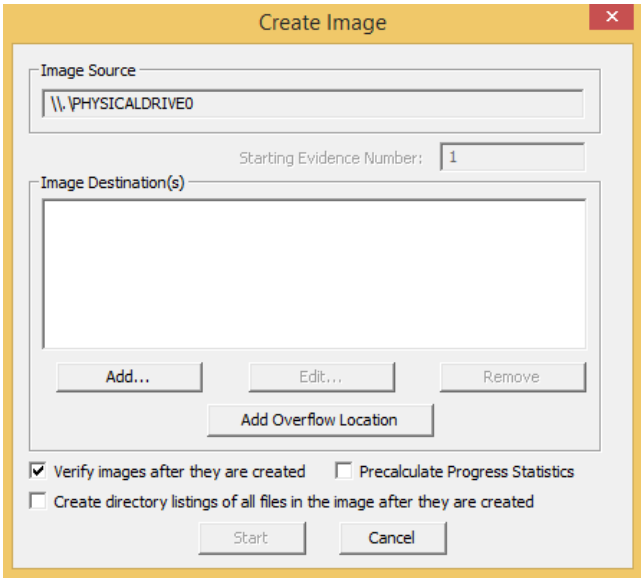
Daha sonra gelen pencerede ise, hangi tip imaj alınacaksa ona göre seçilen menü gelecektir. Yani, bir önceki menüde "Physical Drive" seçilmiş ise bu menüde bilgisayara fiziksel olarak bağlı depolama cihazlarının listesi gelecektir.



Şekil 2.4. FTK Imagerden imajı başlatma adımı 3.

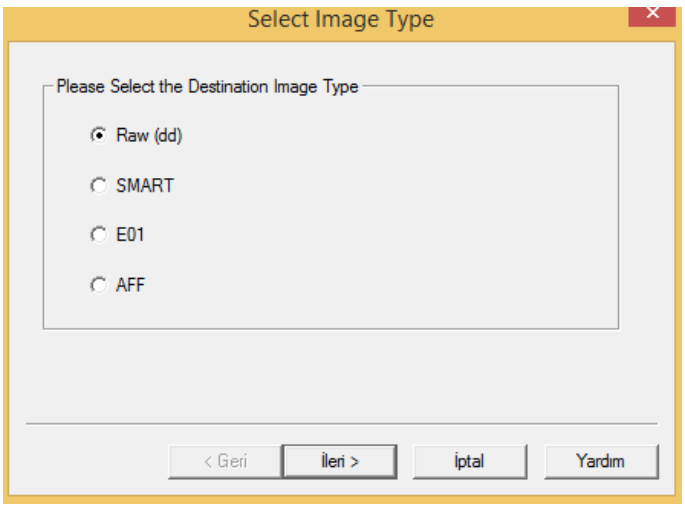
Disk seçildikten sonra gelen pencerede "destinasyon"(hedef) da denilen ve imajın nereye kaydedileceğini gösteren menü gelecektir. Ayrıca bu pencerede, "Verify images after they are created" seçeneği ile doğrulama işlemi ve "Precalculate

Progress Statistics" seçeneği ile de imajın ne kadar süreceği gibi işlemler de yapılabilir.



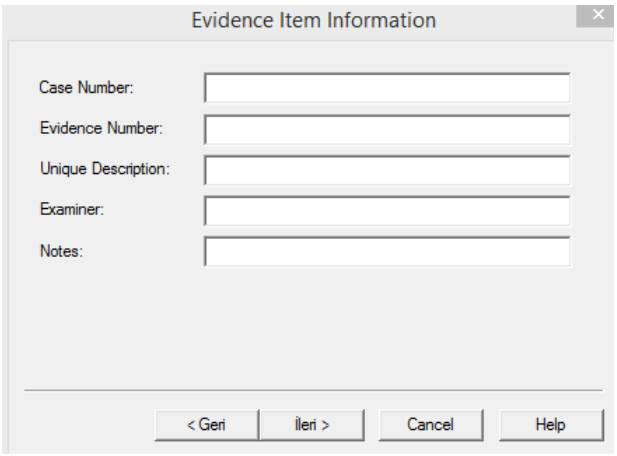
Şekil 2.5. FTK Imagerden imajı başlatma adımı 4.

Bu aşamadan sonra, alınacak imajın türünü belirleme penceresi gelmektedir. Buradan imajın türünün seçilmesi gerekmektedir. FTK Imager 4 farklı biçimde seçenek sunar. Bunlar; raw(dd), E01, SMART ve AFF türünde imajlardır. İstenilen format seçildikten sonra devam edilir.



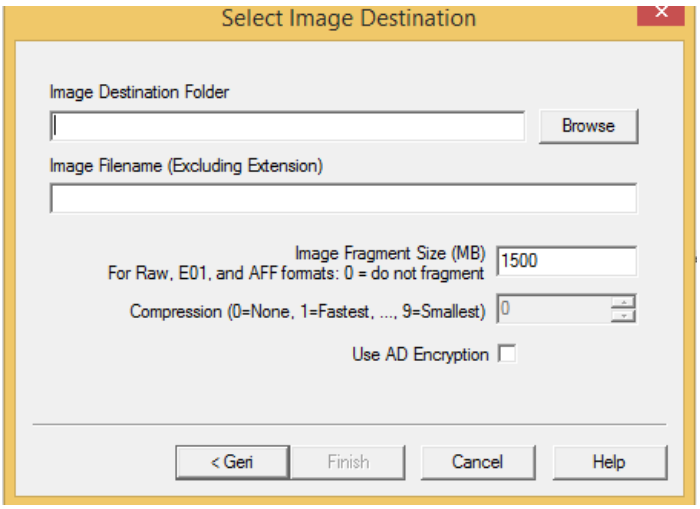
Şekil 2.6. FTK Imagerden imajı başlatma adımı 5.

Ardından gelen pencerede imaj ile ilgili vaka adı, delil numarası incelemeyi kimin yapacağı vs. gibi temel adli bilişim vakalarında gerekli olan kısımlar doldurulur ve devam edilir.



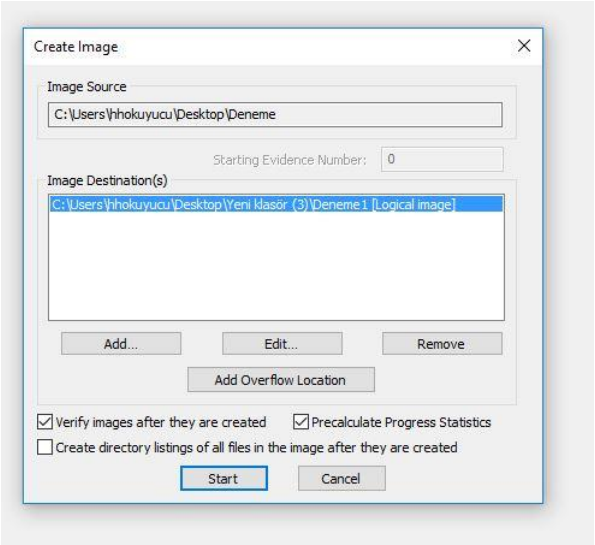
Şekil 2.7. FTK Imagerden imajı başlatma adımı 6.

Daha sonra gelen pencerede imajın kaydedileceği yol belirlenecektir. Ayrıca burada, "Image Fragment Size(MB)" kısmında imajın tek parça olması için "0" yazılması gerekmektedir. Image Fragment Size kısmına yazılacak olan değer imajın parça sayısını belirleyecek olan seçenektir. Örneğin 3000 MB'lık bir imaj ve iki parça şeklinde olması isteniyorsa, bu kısma 1500 MB yazılarak her bir parçanın boyutu da belirlenmiş olacaktır. "Compression" seçeneği ise, imaj sıkıştırılmak istendiği zaman sıkıştırma oranını belirlemek için kullanılacaktır.



Şekil 2.8. FTK Imagerden imajı başlatma adımı 7.

Daha sonra yukarıdaki adımda, Image Destination Folder kısmında, imajın kayıt edileceği klasör seçilir. Image Filaneme kısmında ise imaja verilecek isim yazılır ve Finish'e basılır.



Şekil 2.9. FTK Imagerden imajı başlatma adımı 8.

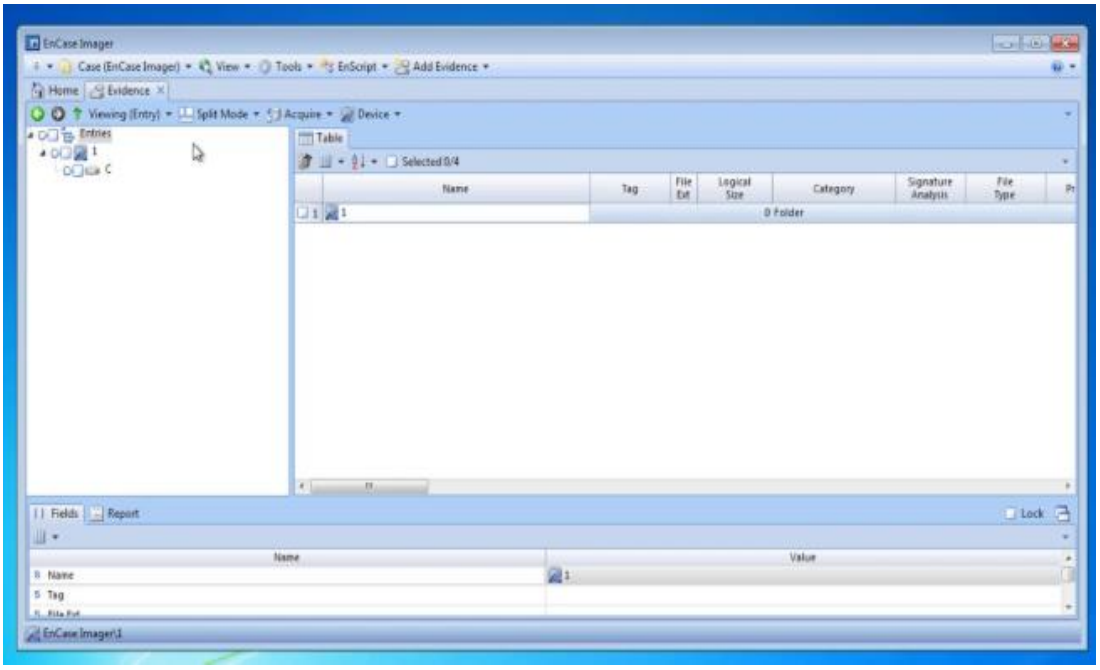
"Finish" butonuna basıldıktan, imaj almanın son aşamasına, yukarıdaki şekilde görüldüğü gibi geçilir. Buradan da "Start"a basıldıktan sonra imaj alma işlemi başlamış olacaktır.

Cihazların tamamına fiziksel disk, bölümlerine ise mantıksal disk bölümü denir. Sabit diskin tamamının imajının alınmasına fiziksel imaj, seçilen bir bölümünün imajının alınmasına da mantıksal imaj denilir.

2.4.1.2. Encase Forensic Image

Guıdance Software'in internet sitesinden indirilebilecek olan program ile bilgisayara fiziksel olarak bağlı olan depolama cihazlarının imajını alma, wipe etme ya da kablo ile başka bir bilgisayara bağlanarak imajını alma gibi işlemler yapılabilir.

Encase Forensic Image programı RAM'ın imajının alınmasına da olanak sağlar. Bu program, E01 ve L01 imaj formatında ve bu formatların daha da sıkıştırılmış hali olan Ex01 ve Lx01 formatlarında alınmış imajların da incelenmesine olanak sağlar.



Şekil 2.10. Encase Forensic Imager yazılımı arayüzü.

Encase Forensic Imager adli bilişim yazılımının, Linux işletim sistemi tabanlı bilgisayarlarda işlem yapmasına da imkan sağlayan "LinEN" isimli sürümü vardır. Linux işletim sistemi, NTFS dosya sisteminde düzgün bir biçimde çalışmadığı için, imajın kaydedileceği depolama cihazının dosya sisteminin NTFS harici, örneğin FAT32 ya da ext2/ext3 gibi bir dosya sistemine sahip olması gerekir. Lınux işletim sisteminde kullanımı ile ilgili olarak bir diğer husus ise autofs seçeneğinin durdurulması gerekliliğidir. Bu seçenek, bilgisayara takılan imajı alınacak cihazları otomatik olarak kullanıma açan bir seçenektir. Bu durum ise, yazma korumalı olmasını engelleyecektir. Dolayısı ile autofs seçeneğinin kapalı olmasına dikkat edilmelidir [16].

2.4.1.3. Forensic Image

Getdata adlı internet sitesinden ücretsiz olarak indirilebilecek olan yazılım, Windows işletim sistemine sahip bilgisayarlar tarafından kullanılabilir. E01, AFF, RAW/DD uzantılı imaj formatlarını destekleyen bu yazılım, ayrıca bu formatlarla alınmış imajları da birbirlerine dönüştürme işlemi de yapabilir. İmaj alma işlemi bittikten

sonra, FTK Imager ve Encase Forensic Imager yazılımlarından farklı olarak SHA1 ve MD5 hash özeti değeri haricinde SHA256 hash özeti değerini de hesaplayabilir. Ancak bu yazılım, depolama cihazlarının HPA ve DCO ile korunan alanlarına erişim sağlayamamaktadır [17].

2.4.1.4. Tableu Image

Tableu adlı internet sitesinden ücretsiz olarak indirilebilecek olan yazılım, Windows işletim sistemine sahip bilgisayarlar tarafından kullanılabilir. E01, RAW/DD ve DMG uzantılı imaj formatlarını destekleyen yazılım, SHA1 ve MD5 hash özeti değerini de hesaplayabilir. Tableu Image yazılımı, yazma korumalı olarak çalışır ve çalışma esnasında herhangi bir sebepten ötürü durdurulan program yeniden çalıştırılarak kaldığı yerden devam edebilir [17].

2.4.1.5. "dd" Komutu

Linux işletim sistemine sahip tüm cihazlarda yer alan bir komuttur. Bu komut linux ile çalışan her bilgisayarda olsa bile, kullanılacağı zaman, sistemde mevcut olan komut yerine güvenilir bir kaynaktan alınmış olan komut kullanılmalıdır. Çünkü imajı alınacak bilgisayardaki komut değiştirilmiş olabilir. Dolayısı ile komut çalıştırıldığında, imajı alınacak veriler değişebilir ya da geri dönülmez biçimde silinebilir.

"dd" komutunun adli bilişimde kullanılan 2 sürümü bulunmaktadır. Bunlar "dc3dd" ve "dcfldd" olarak isimlendirilmiştir.

Bu iki komut kullanılarak;

- Hash değeri hesaplatma
- İmajı alınan verinin sayısını gösterme
- Verileri wipe edebilme
- Adli imajın orijinalinden eksiksiz olduğunu test etme gibi işlemler yapılabilir.

"dcfldd" ve "dc3dd" komutları çalışma bakımından benzerlik gösterebildikleri gibi farklılık da gösterebilmektedir. Örneğin her ikiside imaj alma ve hash hesaplatma tamamlandıktan sonra hash değerini "hashlog" ismiyle kaydeder. Her iki komutta, imaj alma esnasında verinin kaynağı ile ilgili okuma hatasıyla karşılaştıklarında okunamayan yere "0" bilgisi yazar ve her iki komutta sadece RAW/DD formatında imaj alır. Ancak, "dc3dd" komutu ile MD5, SHA, SHA256 ve SHA512 hash türlerini desteklerken, "dcfldd" komutu MD5, SHA, SHA256 ve SHA512 ve SHA384 hash türlerini destekler [18].

2.4.1.6. Guymager

Linux tabanlı işletim sistemine sahip cihazlar tarafından kullanılan adli imaj alma yazılımıdır. Kopyalama süresi ve işlemci kullanımı bakımından FTK Imager yazılımından daha başarılı olduğu tespit edilmiştir.

RAW/DD, AFF ve E01 formatında imaj alınmasına olanak sağlar. Ayrıca diskin birebir aynısının oluşturulması anlamına gelen "disk klonlama" işlemi de yapabilmektedir. MD5 ve SHA256 türünde hash veri özeti değeri hesaplayan ve komut ihtiyacı olmayan kolay kullanıma sahip bir adli imaj alma yazılımıdır [19].

2.4.2. İmaj Alma Donanımları

2.4.2.1.Tableu TD2

Tableu TD2 donanımı, depolama birimlerinden SATA ve IDE/PATA girişe sahip olanlarının imajını alabilen bir donanımdır. İsteğe bağlı olarak da USB ve SAS girişe sahip depolama birimleri için gerekli olan dönüştürücü tedarik edilirse bu cihazların da adli kopyalarını alabilir. Alınan imajların MD5 ve SHA1 türünde hash değerlerini hesaplayabilen cihaz; E01, RAW/DD ve Ex01 türünde imaj alabilir. Tableu TD2 cihazı, alınan imajı iki farklı depolama birimine kaydedebilir ve ayrıca imaj alma esnasındaki değişiklikleri de kaydedebilir. Verileri tamamen geri dönüşümsüz olarak silme anlamına gelen wipe işlemini de yapabilen cihaz kullanımı kolay ve pratik olduğu için adli bilişim uzmanlarının sıkça kullandığı bir donanımdır [20].

Şekil 2.11.Tableu TD2 cihazı görünümü.

2.4.2.2.Tableu TD3

Tableu TD3 donanımı, depolama birimlerinden SAS, SATA, USB 3.0/2.0/1.1, FireWire(1394A/B) ve IDE/PATA girişe sahip olanlarının imajını alabilen bir donanımdır. Dakikada ortalama 9 GBlık veri akışı sağlayabilen cihaz HPA ve DCO gibi korumalı bölgelerinin de imajını alabilmektedir. Alınan imajların MD5 ve SHA1 türünde hash değerlerini hesaplayabilen cihaz; E01, RAW/DD ve Ex01 türünde imaj alabilir. Tableu TD3 cihazının üzerinde bulunan gigabit mertebesinde ethernet girişi vardır ve bu giriş sayesinde delillerin imajını ağ üzerinden paylaşıma açabilir. Tableu TD2 cihazı gibi bu cihaz da alınan imajı iki farklı depolama birimine kaydedebilir ve ayrıca imaj alma esnasındaki değişiklikleri de kaydedebilir. Gene aynı şekilde verileri tamamen geri dönüşümsüz olarak silme anlamına gelen wipe işlemini de yapabilen cihaz kullanımı kolay ve pratik olduğu için adli bilişim uzmanlarının sıkça kullandığı bir donanımdır [20].

Şekil 2.12. Tableu TD3 cihazı görünümü.

2.4.2.3. DİTTO

Ditto imaj alma donanımı, birçok farklı elektronik delilin imajının alınmasında kullanılan bir donanımdır. SAS, SCSI, SATA, PATA, IDE girişlere sahip harddisklerin, USB girişli taşınabilir bellek elemanlarının, SD ve mikro SD hafıza kartlarının imajlarının alınabilmesine olanak sağlar. Bu imajları, imaj boyutuna göre BIN, E01, RAW, DD şeklinde alabilen cihaz 2 farklı elektronik delilin aynı anda imajını alabilmektedir. Ancak bu özellik herhangi bir karışıklık oluşması ihtimaline karşın, adli bilişim uzmanlarınca tavsiye edilmemektedir. Ditto cihazı, imaj almasının yanısıra wipe etme işlemi de yapabilir, delillerin klonunu oluşturabilir ve imaj alma işleminin olmazsa olması hash değeri hesaplama işlemini de yapabilir [20].

Şekil 2.13.Ditto cihazı görünümü.

2.5. İMAJI ANALİZ ETME YAZILIMLARI

2.5.1. Encase Forensic Yazılımı

Dünyada, özellikle imaj inceleme ve analiz etme aşamasında en çok kullanılan yazılımlardan birisi olan Encase Forensic yazılımı Windows işletim sistemi tabanlı bilgisayarlarda kullanılmaktadır. Guidance Software şirketi üzerinden ticari amaçla üretilen yazılımın lisanslı çalışabilmesi için yazılıma ait dongle'ın bilgisayara takılı

olması gerekmektedir. Dongle takılı olmadan yalnızca DOS modda çalışır ve imaj alma işlemi yapar.

İmajı inceleme analiz etme ve raporlandırma konusunda kendisini kanıtlamış bir yazılımdır. Bu sebepten ötürü de dünyada FBI başta olmak üzere birçok kolluk kuvveti tarafından kullanılmaktadır. Ayrıca ABD ve Avrupa'da çoğu mahkeme, bu program tarafından incelenen ve mahkemeye sunulan raporları delil niteliğinde değerlendirip kabul etmektedir.

Göstermiş olduğu performansıyla birlikte, Türkiye'de de adli bilişimde sıkça kullanılan Encase Forensic yazılımının kullanıcısına sunduğu hizmetleri şu şekilde sıralayabiliriz:

- Sabit disk(HDD), SSD, USB Flash Bellek, SD Micro SD Hafıza Kartı, Dosya, Klasör, Telefon, Tablet vs. gibi veri barındıran depolama birimlerinin imajını alma
- Windows ve Unix işletim sistemine sahip dijital deliller üzerinde inceleme yapabilme
- İmajların ve delillerin hash değerini hesaplayabilme
- Veri Kurtarma
- İmza analizi ve Hash analizi yapma
- Hash karşılaştırma yaparak bilinen sistem dosyalarını ayıklayabilme
- Farklı uzantıya sahip dosyaları ayrı ayrı inceleyebilme
- Dizin oluşturma
- Zaman çizelgesi oluşturabilme
- Kelime ve karakter araması yapılabilme gibi olanaklar sağlamaktadır [20].

2.5.2. Forensic ToolKit (FTK) Yazılımı

AccesData Software şirketi tarafından ücretli olarak piyasaya sürülen Forensic Toolkit yazılımı Encase Forensic Yazılımı gibi adli bilişimde çok kullanılan bir yazılımdır. Forensic Toolkit(FTK) yazılımı, mahkemelerde güvenilirliği kabul edilmiş ve Türkiye'de sıkça kullanılan bir yazılımdır.

Bu yazılım kullanıcıya birçok olanaklar sağlar. Bu olanakları kısaca şöyle sıralayabiliriz:

- Sabit disk(HDD), SSD, USB Flash Bellek, SD Micro SD Hafıza Kartı, Dosya, Klasör, Telefon, Tablet vs. gibi veri barındıran depolama birimlerinin imajını alma
- İmajların ve delillerin hash değerini hesaplayabilme
- Veri Kurtarma
- İmza analizi ve Hash analizi yapma
- Hash karşılaştırma yaparak bilinen sistem dosyalarını ayıklayabilme
- Farklı uzantıya sahip dosyaları ayrı ayrı inceleyebilme
- Şifre kırabilme
- Devam eden prosesi durdurma, devame ettirme
- Raporlama
- Kapsamlı olarak uçucu veri analizi
- Dizin oluşturma
- Kelime ve karakter araması yapılabilme
- Zaman çizelgesi oluşturabilme
- E-posta içeriklerini başka bir programa gerek kalmadan önizleme yapma
- SQLite veri tabanı ile çalışabilme
- Apple DMG ve DD/DMG türünden olan imajlar üzerinde inceleme yapma
- Credant, SafeBoot, Utimaco, SafeGuard Enterprise and Easy, EFS, PGP, GuardianEdge, Pointsec and S/MIME gibi şifreli alanların şifrelerini çözebilme gibi özellikleri vardır [21].

2.5.3. Cellebrate UFED

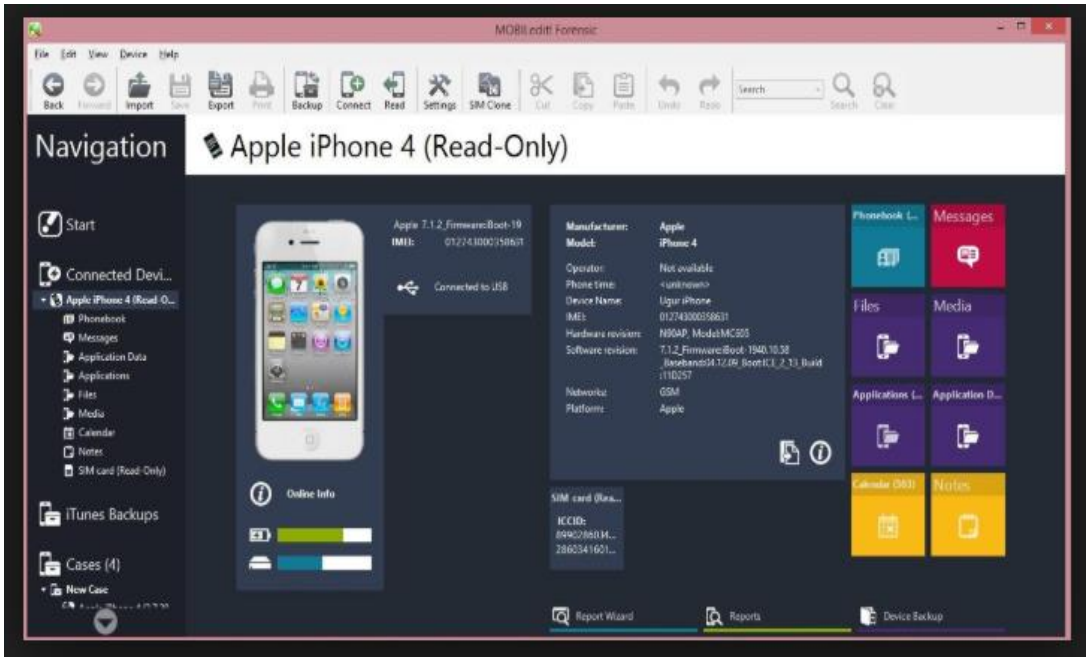
1999 yılında kurulan bir telekom şirketi tarafından piyasaya sürülen Cellebrate UFED yazılımı, mobil cihazları incelemeye yönelik olarak kullanılmaktadır. The Cellebrate "Universial Forensic Extraction Device (UFED)" yazılımı hem donanımsal hem de yazılımsal olarak inceleme yapmaya imkân sağlar. Bu yazılım,

2010 Eylül tarihinden itibaren 2500'ün üzerinde mobil cihaz türünü, GSM, TDMA, CDMA, IDEN de dahil olmak üzere desteklemektedir.

Cellebrate UFED adli bilişim yazılımı, adli bilişim uzmanları tarafından cep telefonu imajı alınmasında ve incelenmesinde en çok kullanılan programdır. Öyleki pazarda çok önemli bir yeri olan bu yazılım/donanım, sektör uzmanları tarafından 2009, 2010, 2011 ve 2012 yıllarında piyasadaki "En İyi Telefon Adli Bilişim İnceleme Yazılım/Donanım" ünvanına layık görülmüştür. Ayrıca bu yazılımı piyasada söz sahibi yapan bir başka özelliği de yazma korumalı olarak çalışması ve delil bütünlüğünün sağlaması özelliğidir. 200den fazla Android cihazda, fizikel ve dosya sisteminden pin ve şifre kilidini by-pass ederek analiz etme özelliği de bulunmaktadır.

Facebook, Twitter, WhatsApp, Viber, Fring, Tiger Metin, Google+ gibi birçok uygulamanın analizini gerçekleştirebilmektedir. Cellebrate UFED adli bilişim yazılım/donanımını diğer adli bilişim yazılımlarından ayıran bir diğer özelliği de, cep telefonları için neredeyse tamamı Çinde üretilen chipsetlerin tamamına uyumlu olmasıdır.

Cellebrate UFED yazılımı imaj alındıktan sonra alınan imajı inceleme aşamasında, kullanıcıya anahtar kelime araması yapma, fiziksel imajlardan resim kazıması yapma, incelenen imaj cep telefonuna aitse, bu cep telefonunun ilk etkinleştirme tarihini bulma ve bluetooth ile alınan/gönderilen verileri tespit etme gibi özellikleri ile Türkiye'de adli bilişim uzmanları tarafından en çok kullanılan adli bilişim programıdır [22].



Şekil 2.14. UFED yazılımı arayüz görünümü.

2.5.4. XRY Yazılımı

XRY yazılımı; "GSM, CDMA, UMTS, IDEN ve 3G" telefonları da dahil olmak üzere 5 binden fazla marka ve model mobil cihazı destekleyen bir adli bilişim yazılımıdır. XRY yazılımına, sahip olduğu Infrared(IR) bağlantı noktası üzerinden bluetooth veya kablo ile bir cep telefonu bağlanabilir. Bağlantı kurulduktan sonra ekranda, cep telefonun modeli, cihaz adı, üreticisi, seri numarası, IMEI numarası, cihaz saati vs. cihazla ilgili bütün bilgileri ekranda gösterir. Cihaz tanımlanması yapıldıktan sonra da cep telefonundan alınan veriler XRY formatında saklanır ve istenildiği zaman export edilerek inceleme yapılabilir. SIM/USIM kartların da incelemesini yapan program bu kartlardan veri çıkarımı ile fiziksel ve mantıksal imaj alabilmektedir.

Başarılı bir incelemenin ardından XRY adli bilişim yazılımı ile bir cep telefonunda mevcut olan bütün veriler tespit edilebilir. Rehber, kişi listesi, resimler, aramalar, yüklü ve kaldırılan uygulamalar, SMS, MMS, ses, video, ağ gibi bütün içerikler görülebilir ve export edilebilir. XRY yazılımı, kullanıcısına sağlamış olduğu bu

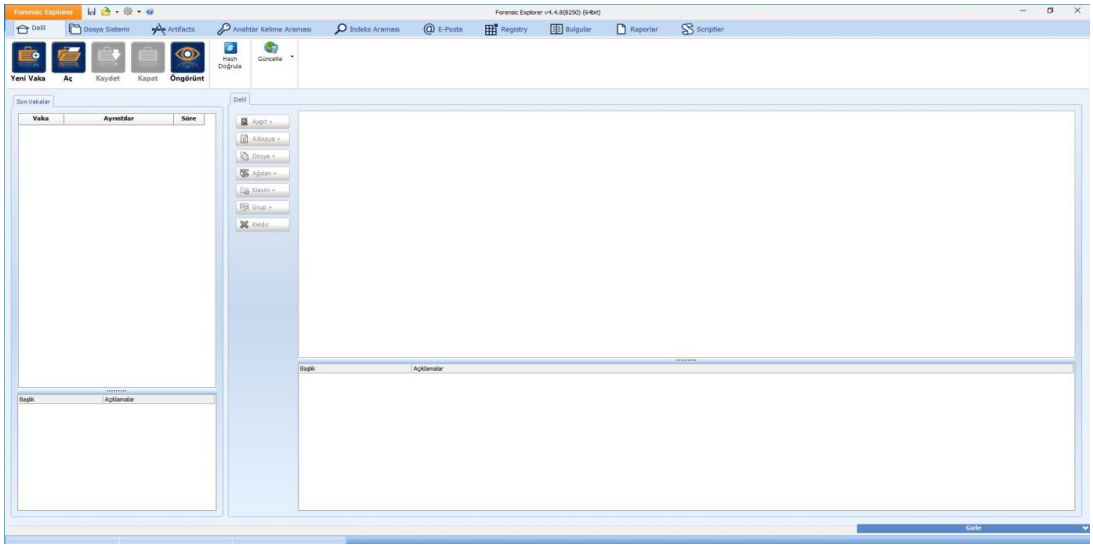
kadar olanaktan sonra dünyada yaklaşık olarak 60 ülke tarafından tercih edilen bir adli bilişim yazılımı olmuştur [23].

2.5.5. Forensic Explorer Yazılımı

Forensic Explorer yazılımı, imajı inceleme yazılımı olup dongle aracılığıyla lisanslı sürümü kullanılabilir. Farklı formatlardaki imaj türlerini programa bir vaka klasörü açıp yükleyebilme ve bu imaj içerisinde kullanıcının istediği şekilde inceleme yapılmasına imkan sağlayan bu yazılım, en çok kullanılan analiz yazılımlarındandır. İmaj üzerinde; dosya analizi, dosya genişletme, silinen verileri geri getirmeye yönelik klasör kurtarma, anahtar kelime araması yapma, tespit edilen verilerden istenilenleri export edebilme, export edilen verilerin bilgilerinin tutulduğu bir tablo oluşturma gibi birçok işlem için Forensic Explorer yazılımı kullanılmaktadır. Hatta, bir imaj üzerinden vaka klasörü oluşturmadan da ön izleme menüsü ile imaj içerisinde verilere görüntüleyebilme ve export edebilme özelliğini de barındırmaktadır [20].

Ayrıca bu yazılımın sahip olduğu öteki özellikleri de şu şekilde sıralayabiliriz:

- Sayısal imza hesaplama (hashing) ve sayısal imza karşılaştırma (Hash Match)
- Tekil ve çoğul anahtar kelime arama (Keyword Search)
- İndeksleme yapma (Index Search)
- Sık kullanılanlara ekleme (Bookmarks)
- Özel şablonla raporlama (Reporting)
- Windows Registry Analizi (Registry)
- E-posta analizi (Email)
- Disk shadow kopyası incelme (Volume Shadow Copy)
- Veri kazıma (Data carving)
- Hardware ve Software RAID sistem kopyalar



Şekil 2.15. Forensic Explorer programının arayüz görünümü.

2.5.6. The Sleuth Kit ve Autopsy Yazılımları

The Sleuth yazılımı Unix ve Windows işletim sistemlerinde çalışabilen ücretsiz bir yazılımdır ve komut satırı üzerinden bağlanılmaktadır. Komut satırı üzerinden inceleme yapmak çok tercih edilen bir yöntem olmadığı için bu sepeten dolayı içerisinde Autopsy isimli arayüz üzerinden yazılıma ait komutları kullanmaya izin veren bir program mevcuttur.

The Sleuth Kit ve Autopsy yazılımları da diğer yazılımlar gibi kullanıcısına birçok olanak sağlamıştır. Bu olanaklardan bazıları,

- DD, Encase ve AFF formatına sahip imajlar üzerinde inceleme yapabilme
- NTFS, FAT, UFS 1/2, EXT2FS, EXT3FS, HFS ve ISO 9660 gibi birçok dosya sistemini destekleme
- Kelime ve karakter araması yapılabilme
- Hash hesaplayabilme
- Hash karşılaştırma yaparak bilinen sistem dosyalarını ayıklayabilme
- Zaman çizelgesi oluşturabilme
- Veri Kurtarma
- Steganografi kontrolü yapma gibi birçok olanak mevcuttur [24].

BÖLÜM 3

KRİPTOGRAFİK HASH FONKSİYONLARI

3.1. KRİPTOLOJİ VE KRİPTOGRAFİ NEDİR?

3.1.1. Kriptoloji

En basit tanımıyla şifreleme bilimi olarak adlandırılan kriptoloji, gönderilen mesajların güvenliğini sağlamak amacıyla, gönderici ve alıcı arasına üçüncü kişilerin girmesini engelleyen ve böylece iletilen mesajın güvenliğini bu şekilde sağlayan bir bilim dalıdır. Daha teknik bir ifadeyle kriptoloji, anlamlı halde olan bir verinin şifreleme türleri kullanılarak üçüncü kişilerin araya girmemesi için anlamsız hale getirilmesidir. Gönderilin mesajların güvenilirliği ise, devletlerarası iletişimin korunması, özel kurum ve kuruluşların bilgilerinin güvenilirliği, bankaların müşteriler için sunduğu hizmet vs. gibi durumlar göz önüne alındığında oldukça önem arz etmektedir.

Kriptoloji ayrı bir bilim dalı olarak gözükse de, matematik bilimi ile çok fazla benzerlikleri bulunmaktadır. Sayı teorisi, asal sayılar, matrisler, çarpanlara ayırma gibi konular matematik bilimimde oldukça fazla kullanıldığı gibi kriptolojinin de en temel konularıdır [25].

3.1.2. Kriptoloji Tarihi

Kriptolojinin tam olarak ne zaman ortaya çıktığı bilinmese de, yazının bulunmasıyla veri iletiminde güvenilirliği sağlamak maksadıyla kullanılmaya başlanmıştır. Tarihte ilk kriptolog 4000 yıl önce Mısırda yaşamış Mısırlı bir kâtiptir. Kriptoloji eski medeniyetlerde sıklıkla kullanılan bir güvenlik sistemi olmuştur. Ancak eski medeniyetlerin birbiri ile haberleşmesi fazla olmadığı için, bir medeniyetim

kriptolojiye sağladığı katkıyı öteki medeniyet takip edememiş ve bu yüzden kriptoloji biliminin gelişimi yavaş olmuştur.

Kriptolojinin ilerleyişi askeri alanda daha hızlı olmuştur. M.Ö. 5. yy'da ilk defa askeri alanda Spartalılar tarafından kullanılmıştır ve Roma İmparatoru Julius Caesar tarafından da askeri alanda kullanılmıştır. Hatta Julius Caesar, kendi adıyla bilinen bir şifreleme yöntemini de litaratüre kazandırmıştır.

Caesar şifrelemesi olarak bilinen bu yöntem şu şekilde çalışmaktadır: Alfabede bulunan her harf kendinden sonra gelen 3. harf ile yer değiştirilip metinler bu şekilde hazırlanmakta ve alıcı taraf mesajı okuyacağı zaman, şifreli olduğunu bildiği için, her harften 3 geriye giderek tekrar yeni metini oluşturmaktadır. Böylece şifreli gelen mesajı çözmüş olmaktadır.

Casesar Şifrelemenin kullandığı tablo şu şekildedir:

Çizelge 3.1. Sezar şifrelemesi giriş-çıkış tablosu.

Giriş Değeri	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	W	X	Y	Z
Çıkış Değeri	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	W	X	Y	Z	A	B	C

Örnek: KARABÜK ÜNİVERSİTESİ
Burada şifreleme, herbir harften 3 sonra gelen harfi bulup yazma şeklinde olacaktır.
Şifrelenmiş Metin: NÇTÇDXN XPLYĞTULVĞUL

Kriptolojiye katkı sağlayan bir başka bilim adamı da El Kindi'dir. 9. yy'da Bağdatta yaşamış; tıp, matematik, astronomi ve felsefe alanlarında eserler vermiştir. Kriptoloji alanında tek alfabeli yerine koyma yöntemini geliştirmiş ve frekans analizini bulmuştur. El Kindi, frekans analizini şu şekilde geliştirmiştir: Bütün alfabelerde en çok kullanılan harflerin bir istatistiğini hazırlamış ve doğal olarak, normal metinde en çok kullanılan harf şifreli metinde de en çok kullanılan harf olmuştur. Böylece, bir metnin şifresi tam olarak çözülmese de yüksek oranda Caesar ile şifrelenmiş metinler

çözümlenmiştir. Örneğin Türkçe'de en çok kullanılan harflerin yüzdelik olarak kullanım sıklıkları aşağıda gösterilmiştir. Tabloya göre Türkçe'de en çok "A" harfi kullanılmakta ve kullanım sıklığı %11,92 olarak tespit edilmiştir. Aynı şekilde tabloya bakıldığı zaman en az kullanılan harf ise %0,034 ile "J" olmuştur.

Çizelge 3.2. Harflerin kullanılma sıklıkları.

1	A	11,92
2	B	2,844
3	C	0,963
4	Ç	1,156
5	D	4,706
6	E	8,912
7	F	0,461
8	G	1,253
9	Ğ	1,125
10	H	1,212
11	I	5,114
12	İ	8,6
13	J	0,034
14	K	4,683
15	L	5,922
16	M	3,752
17	N	4,487
18	O	2,476
19	Ö	0,777
20	P	0,886
21	R	6,722
22	S	3,014
23	Ş	1,78
24	T	3,014
25	U	3,235
26	Ü	1,854
27	V	0,959
28	Y	3,336
29	Z	1,5

Kriptolojinin şifreleme bilimi olduğunu ilk paragrafta belirtildi. Şifreleme bilimi ise, temelde iki alanın birleşmesinden meydana gelmiştir. Bu alanlar şifreleme anlamına gelen "kriptografi" ve şifre çözme anlamına gelen "kriptoanaliz"dir.

Tezimizde bahsetmiş olduğumuz hash fonksiyonları da bir tür şifreleme olduğu için doğrudan kriptografi ile ilgilidir. Tezin konusu olan hash fonksiyonları, şifreleme yaptığı için kriptografik hash fonksiyonları ismiyle de kullanılmaktadır [25].

3.2. HASH FONKSİYONU NEDİR?

Hash fonksiyonu, verilerin bütünlüğünü kontrol etmek için kullanılan ve ait olduğu verinin ilk sektöründen son sektörüne kadar bütün bitlerin özel bir algortimik işleme tabi tutulması sonucu eşşiz bir sabit değer oluşturan matematiksel fonksiyonlardır. Burada eşsiz olması en kritik noktadır. Bir başka ifadeyle, oluşan matematiksel değer, sadece ve sadece ait olduğu veriye özgüdür ve başka hiçbir şekilde aynı değer üretilemez. Hash fonksiyonunun işleme tabi tutulması sonucunda ürettiği değere hash değeri ya da özetleme fonksiyonu da denilmektedir.

Hash değerinin eşsiz oluşu birçok önemli kullanım amacını doğurmuştur. Örneğin; hash değeri hesaplandıktan sonra, o veri üzerinde bir değişiklik yapılırsa ve tekrar hash hesaplatılırsa, önceki hesaplanan değer ile uyuşmayacaktır ve veri üzerinde değişiklik yapıldığı ortaya çıkmış olacaktır. Bu durum da verinin bütünlüğünü bozacak ve dolayısıyla verinin bütünlüğü ilkesini çiğnemiş olacaktır.

Hash fonksiyonlarının bir başka kullanım amacı da, büyük boyutlardaki verinin tarifi yapılırken hash değeri üzerinden yapılması ve büyük boyuttaki veriyi küçük boyutlara indirgemiş olmasıdır.

Hash değerinin uygulamada çok önemli oluğunu sık sık vurgulamak gerekir. Çünkü adli bilişim sistemlerinde incelemeler deliller üzerinden yapılır. Deliller üzerinde inceleme yapılırken de önceki bölümde bahsettiğimiz “Delil Bütünlüğü” ve “İnkâr Edilememesi” ilkelerinin sağlanmış olması gerekir. Bunu sağlamak için de önceden hash değeri hesaplatılmalıdır. Bu denli öneme sahip hash fonksiyonlarının bazı önemli özelliklerini de şu şekilde sıralanabilir:

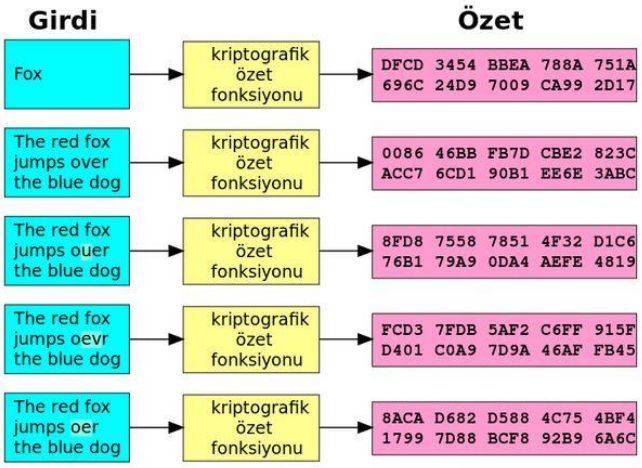
- Hash fonksiyonları tek yönlüdür, özetlenen veriden asıl veri elde edilemez.
- Hash Fonksiyonlarında anahtar kullanılmaz.
- Hash fonksiyonlarında simterik ya da asimetrik gibi sınıflandırma bulunmaz.
- Blok uzunluğu ne kadar uzun olursa, bütünlük o kadar güvenli olacaktır.

En çok kullanılan hash fonksiyonları MD ailesi, SHA ailesi, HAVAL ve WHIRPOOL algoritmalarıdır.

MD serisi, Ron Rıvest tarafından geliştirilmiştir ve en çok kullanılan algoritmalardan birisidir. Girilen verinin boyutundan bağımsız olarak 128 bitlik çıktı(hash değeri) üretir. SHA(Secure Hash Algorithm) serisi, Amerika Birleşik Devletlerinde, ulusal güvenlik ajansı(NSA) tarafından geliştirilmiştir ve girilen verinin boyutundan bağımsız olarak 160 bitlik çıktı üretmektedir. Bu fonsiyonlarda, farklı veriler için aynı sonucu verme ihtimali de vardır. Litaretürde bu duruma çakışma (collision) denir ve istenmeyen bir durumdur. Bu durum fonksiyonların güvenilirliğini zedeleyen bir durumdur.

Örneğin aşağıda rastgele metinler yazılmış ve bu metinlerin kriptografik özet fonksiyonu ile hash değeri hesaplatılmış ve hepsinin de birbirinden farklı olduğu görülmüştür.

Çizelge 3.3. Hash değerlerinin hesaplanması.



Hash fonksiyonlarının matematiksel olarak ifade edilişi bir sonraki bölümde ifade edilecektir.

3.3. HASH FONKSİYONLARININ MATEMATİKSEL ALGORİTMASI

Tezimizin bu bölümünde adli bilişim sistemlerinde en sık kullanılan hash fonksiyonlarından MD5 ve SHA1 algoritmalarının matematiksel algoritmaları ve bu algoritmalarının güvenilirliği anlatılacaktır.

3.3.1. MD5 Algoritması ve Güvenilirliği

3.3.1.1. MD5 Algoritması

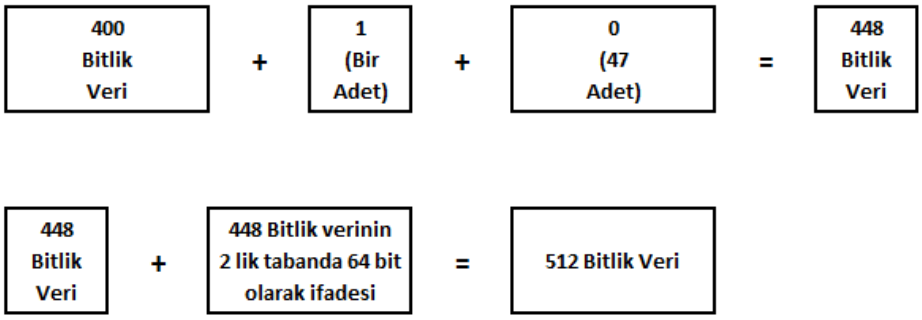
MD5 Algoritması, özeti çıkarılacak olan verinin her birini 512 bitlik parçalara bölerek şifreleme yapmaktadır. Bu durumda mesaj özeti çıkarılacak verinin 512 bit veya katları olması gerekmektedir. Verinin bu özelliği sağlamadığı durumlarda algoritma, veri üzerinde ekleme(padding) işlemi yapar ve 512 bitin katı haline getirir.

Algoritma bu işlemi şu şekilde yapar:

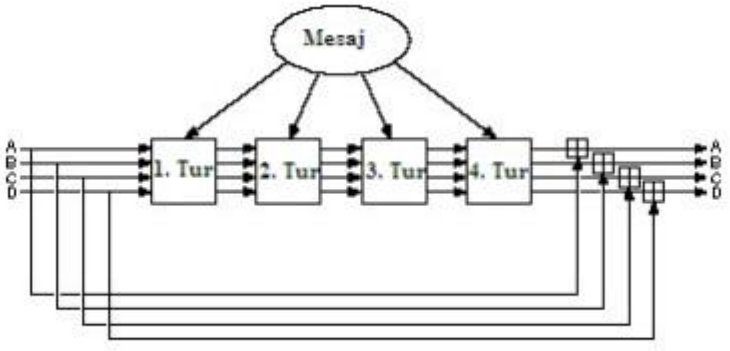
Veri uzunluğu, 512 bitin en yakın değerinin 64 eksiği olacak şekilde, ilk bite 1, ondan sonrakine 0 olacak şekilde eklenir. Geri kalan 64 bit ise veri uzunluğunun belirlenmesinde kullanılır.

Bu durumu bir örnekle açıklayacak olursak;

Elimizde 400 bit uzunluğunda veri olsun. Bu verinin, 512 bitin 48 bit eksiği olan 448 bite tamamlanması gerekir. Dolayısıyla da verinin en sonuna 1 adet 1, 47 adet 0 yazılır ve 448 bitlik veri oluşturulmuş olunur. Daha sonra edilen bu 448 bitlik veri, 2 lik tabanda 64 bit ile ifade edip 448 bitin yanına yazılır ve böylece 512 bitlik veri elde edilmiş olunur. 512 bite tamamlama işlemi aşağıda görsel olarak anlatılmıştır.



Şekil 3.1.Örnek olarak alınmış 400 bitin 512 bit olarak ifade edilmesi.



Şekil 3.2.MD5 Algoritması Çalışma Mantığı.

Yukarıdaki şekilde görüldüğü gibi, her biri 32 bit olan dört adet A, B, C ve D değişkenleri bulunur ve bu değişkenleri değeri her defasında işleme tutuldukça değişir. 4. turun sonunda elde edilen A, B, C ve D değerleri yanyana yazılarak 128 bitlik mesaj özeti değeri oluşturulur. Algortimada şekilde görüldüğü gibi 4 adet tur bulunmakta ve bu turların hepsinin kendine özgü matematiksel işlemi mevcuttur. Bu matematiksel işlem her defasında 16 kez tekrar edilir ve elde edilen sonuç bir sonraki tura aktarılır.

Birinci turda yapılan işlem:

$for\ (i = 0\ to\ 15)$ olmak üzere,
$A = B + ((A + F(B, C, D) + Mi + Ki) <<< s$ şeklindedir.

$F(X,Y,Z) = (X \wedge Y) V (X' \wedge Z)$

Buradaki Mi değeri her dögü için farklılık gösteren 32 bitlik mesaj bloğudur. "Ki" değeri ise, her döngü için sabit olan 32 bitlik sabit bir değeri temsil etmektedir. "$<<< s$" ise, s noktasından sola kaydırma işlemi yapar.

İkinci turda yapılan işlem:

$for\ (i = 0\ to\ 15)\ A = B + ((A + G(B,C,D) + Mi + Ki) <<< s$

$G(X,Y,Z) = (X \wedge Z) V (X \wedge Z')$

Üçüncü ve dördüncü turda yapılan işlem ise,
aşağıda verilen H ve I fonksiyonlarıdır.

$H(X,Y,Z) = X \oplus Y \oplus Z$

$I = (X,Y,Z) = Y \oplus (X\ V\ Z')$

Bu algortimanın sonucunda 128 bitlik özetleme değeri oluşmaktadır [26].

3.3.1.2. Algoritmanın Güvenilirliği

Bütün hash fonksiyonlarında olduğu gibi, MD5 fonksiyonu da, girişin büyüklüğünden tamamen bağımsız olarak bütün veriler için 128 bitlik bir özet fonksiyonu oluşturur. Bu yüzden teorik olarak çözümlenmeleri zordur. Ancak yine de, çözümlenmesi için bazı ataklara maruz kalmış ve bu yüzden güvenilirliği nispeten zedelenmiştir.

Bu ataklardan bazılarını şu şekilde sıralayabiliriz:

1993 yılında, BOSSELAERS ve Bert DEN BOER adında iki kişi, iki farklı girdi için aynı MD5 özet değerini üreten bir çakışma bulmuşlar ve bu MD5 algoritmasına olan güveni azaltmıştır.

2008 yılında da bir grup hacker sahte SSL sertifikasını doğrulamak için MD5'i kullanmışlardır. Yine aynı şekilde, 2012 yılında İran tarafından üretilen Windows işletim sistemi tabanlı Flame adında bir bilgisayar, Microsoft dijital imzası üretmek amacıyla MD5 algoritmasının zayıf yönlerini kullanmıştır.

Bu gibi ataklara ve saldırılara karşı korunmak ve zayıf yönlerinin kötü amaçlar için kullanılmasını engellemek amacıyla MD5 ailesi, özet değerini birkaç defa özetlemek gibi çözümler üretmiştir. Örneğin MD5(MD5(MD5(MD5(kullanıcı şifresi))))) gibi [26].

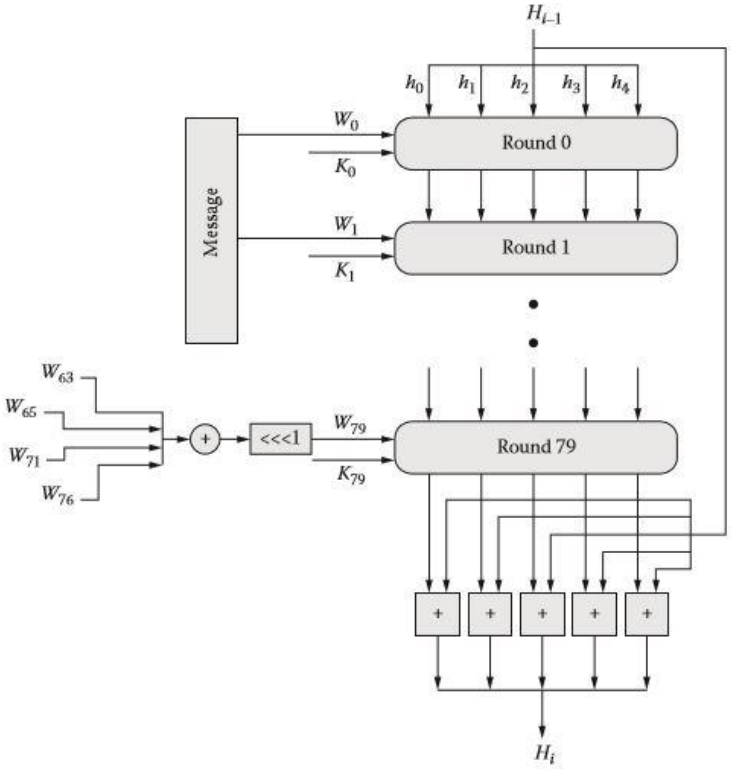
3.3.2. SHA1 Algoritması ve Güvenilirliği

3.3.2.1. SHA1 Algoritması

SHA1 Algoritması, MD5'ten farklı olarak 160 bitlik veri özeti değeri oluşturmaktadır. Bu yüzden algoritmasında her biri 32 bit olan 5 farklı değişkeni vardır. Mesaj özeti değeri, giriş değeri, 512 bit ve katları olmalıdır ve bu 512 bite tamamlama işlemi MD5 algoritmasında olduğu gibi yapılmaktadır.

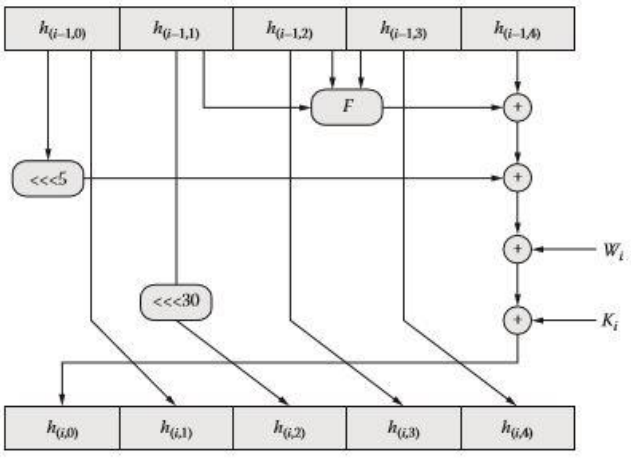
Bu 5 değişken şu şekildedir:

A = 0x67452301
B = 0xefcdab89
C = 0x98badcfe
D = 0x10325476
E = 0xc3d2e1f0



Şekil 3.3.SHA1 algoritması çalışma mantığı.

Algoritma çalışmaya bu şekilde başlar ve mesajlar 512 bitlik bloklar halinde işlenmeye başlar. Her bir işlem, MD5 te 16 defa yapılırken SHA1 algoritması, 20 kez tekrar eder ve kaydırma ve toplama işlemi MD5'te olduğu gibi yapılır.



Şekil 3.4.Algoritmasının ilk adımı.

Algoritmada kullanılan doğrusal olmayan fonksiyonlar ise şu şekildedir:

$f(i,A,B,C) = (A \wedge B) \vee (-A \wedge C) \longrightarrow$ t:0-19 arasında

$f(i,A,B,C) = A \oplus B \oplus C \longrightarrow$ t:20-39 arasında

$f(i,A,B,C) = (A \wedge B) \vee (A \wedge C) \vee (B \wedge C) \longrightarrow$ t:40-59 arasında

$f(i,A,B,C) = A \oplus B \oplus C \longrightarrow$ t:60-79 arasında

Algoritmada kullanılan 4 sabit değer ise şöyledir:

*K*t = 0x5a827999, *t* = 0 ile 19 arasındaki iken.
*K*t = 0x6ed9eba1, *t* = 20 ile 39 arasındaki iken.
*K*t = 0x8f1bbcdc, *t* = 40 ile 59 arasındaki iken.
*K*t = 0xca62c1d6, *t* = 60 ile 79 arasındaki iken [27].

3.3.2.2. Algoritmanın Güvenilirliği

Çakışma, bir hash algoritmasında olması istenmeyen bir durum olup, farklı girdiler için aynı mesaj özeti değerinin çıkması anlamına gelmektedir. Çakışmalar hash algoritmalarının güvenliğini zedeleyen bir durumdur.

SHA1 algoritmasının zayıf yönünü bulmak, çözümlemek ve çakışmaları bulmak için kaba kuvvet saldırısıyla, olabilecek tüm ihtimallerin denenmesi durumunda 2^{80} adet işlemin yapılması gerekmektedir. 2005 yılının Şubat ayında, SHA1 algoritması üzerinde çakışma olup olmadığı üzerinde çalışmalar yapan YIQUN LISA LIN, XIAOYUN WANG ve HONGBO YU adlı kişiler 2^{69} adet işlem sonucu çakışma olabileceğini ileri sürmüşlerdir. 2005 yılında yapılan bir konferansta ise, çakışma bulmak için yapılması gereken işlem sayısının 2^{69} olduğu duyurulmuştur.

SHA1 algoritmasının çakışmasının olup olmadığını bulmak için yapılan saldırılar içinde en etkili olanı, 2012 yılında MARC STEVEN tarafından yapılmış ve 2^{61} adet saldırı ile çakışmanın bulunabileceğini ileri sürmüştür.

SHA1 algoritmasının çakışmasını ve başka zayıf yönlerini bulmak için yapılan saldırıların tamamı teorikte kalmış ve pratik olarak hiçbir zaman kırılamadığı için, güvenliğini her zaman korumaktadır [27].

3.3.3. MD5 ve SHA1 Algoritmalarının Karşılaştırılması

- MD5 Algoritması 32 bitlik 4 farklı değişkene sahipken, SHA1 Algoritması 32 bitlik 5 farklı değişkene sahiptir. Bu yüzden MD5 128 bitlikk çıktı (mesaj özeti) üretirken, SHA1 160 bitlik veri özeti değeri üretir.
- Her iki algoritmada 512 bitlik mesaj blokları üzerinden işlem yapar.
- MD5 Algoritmasında herbir işlem basamağı 16 kez yapılırken, SHA1'de bu sayı 20dir.
- Her iki algoritmada da giriş değerinin en büyük olabileceği değer $2^{64} - 1$ bit olarak kabul edilir.
- SHA1 Algoritması saldırılara ve kaba kuvvet ataklarına karşı daha güvenlidir [26].

3.4. HASH FONKSİYONLARININ KULLANIM ALANLARI VE UYGULAMADA KARŞILAŞILAN PROBLEMLER

3.4.1. Hash Fonksiyonlarının Kullanım Alanları

Kriptografik hash fonksiyonları, diğer adıyla özetleme fonksiyonları adli bilişimde birçok alanda kullanılmaktadır. Bu alanların en başında daha önceki bölümlerde bahsettiğimiz dosya bütünlüğü ve sayısal(dijital) imza gelmektedir. Dosya bütünlüğü sayesinde gönderilen verinin iletim esnasında bozulmadığı ve sayısal imza sayesinde de mesajı (veriyi) ileten kişinin kimliği bilinen bir gönderici tarafından gönderildiği belirlenmiş olur.

Kriptografik hash fonksiyonlarının kullanıldığı öteki alanlar ise şu şeklide sıralanabilir:

- Şifre Koruması (Password Protection)
- Kimlik Doğrulama Protokolleri(Authentication Protocols)
- Rootkit Tespiti(Rootkit Detection) [28].

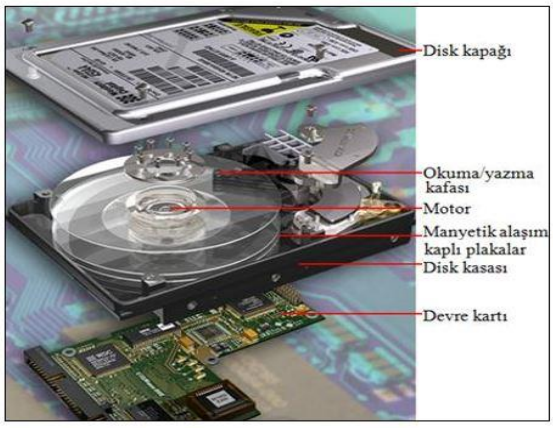
3.4.2. Hash Fonksiyonlarının Uygulamadaki Problemleri

Adli bilşim sitemlerinde dijital(elektronik) delillerin elde edildiği ve depolandığı aygıtlar genellikle, CD/DVD, HDD, SSD ve taşınabilir flash diskler olarak karşımıza çıkmaktadır. Bu depolama aygıtları; içinde muhafaza ettiği veriyi bozulmadan tutabilmek için belirli bir ömre sahiptirler. Örneğin CD ve DVDler optik teknolojiye, HDDler manyetik alan teknolojisine, SSD diskler flash yonga teknolojisine sahip oldukları için, bu aygıtların maruz kaldığı dış etkinin türüne, şiddetine ve süresine bakarak değişen kullanım ömrüne sahiptirler.

3.4.2.1. Sabit Disklede (HDD)'lerde karşılaşılan Hash Problemler

Sabit diskler, diğer bir adıyla HDDler (Hard Disk Driver) içerisinde bulunan mekanik aksam ve manyetik alan teknolojisiyle beraber hızlı bir şekilde veri yazılabilen ve veri okunabilen aygıtlardır. Bu aygıta yazılan veriler kalıcı bir şekilde yazılmakta ve yazma esnasında meydana gelecek olası bir elektrik kesintisi olması durumunda yazılan veriler silinmeyecektir.

Sabit diskler veri depolama işlemini içerisinde bulunan; seramik, cam ve metal bir plaka ile kaplı olan bir yüzey üzerine yapmaktadırlar. Günümüzde genellikle boyut olarak 2,5" ve 3,5" olan türleri bulunur ve bu diskler genellikle SATA, SAS, SCSI ve IDE girişli olan türleri bulunmaktadır.



Şekil 3.5.Bir sabit diskin içyapısı.

Sabit disklerin üretimi yapılıp son testleri de yapıldıktan sonra depolama yüzeyindeki bazı noktaların işlevsiz yani veri depolamaya müsait olmadığı görülmektedir. Bunu tespit etmeye yarayan yönteme yüzey testi denilmektedir. Yüzey testi esnasında bozuk olan sektörlerin hangilerinin olduğu, sabit disklerin kullanıcının ulaşamayacağı bir bölümüne yazılır ve buralara veri kaydedilmez. Ayrıca, disk yüzeyinde fabrikasyon olarak hasarlı olmayıp sonradan hasarlı halen gelen noktalar da bulunmaktadır. Üretim esnasında yapılan yüzey testinde tespit edilen bozuk sektörlere litaratürde P-list, fabrikasyondan sonra meydana gelen bozulma sonucu üzerine veri yazılamayacak durumda olan sektörlere ise G-list denilmektedir.

Her sabit diskte P-list ve G-list bulunmaktadır. P-listler; bozuk sektörler olduğu için disk okuma yazma esnasında bu sektörleri daima atlayarak adresleme yapar. Adresleme, bir diske veri yazma demektir. Bir diskin P-list değeri bilinmiyorsa ve bu diske veri yazılacaksa, P-list bilinmediği ve dolayısıyla hangi sektörlerin bozuk olduğu da bilinemeyeceği için, bütün sektörlere veri yazılır. Dolayısı ile de bozuk olan sektörlere yazılan veriler kaybolacaktır. Bu yüzden diskte hangi sektörlerin bozuk olduğu ve veri yazılmaması gerektiği bilgisini tutan P-list ve G-list oldukça önemlidir.

Adli imaj alma işlemi sırasında disk içindeki okuma yazma işlemini yapan kafalar, sektörleri okumaya çalışırken bozuk olan sektörleri G-liste ekler, onu yerine başka bir sektör atar ve okuma yapmaya devam eder. Bu durum bütün diskler için geçerlidir. Böylece, diskin kullanımı sırasında bozuk olduğu tespit edilen sektörün yerine yenisi atandığı için, imaj alma sırasında hesaplanan hash değeri, aynı diskten tekrar imaj alındığında hesaplatılan hash değeri ile uyuşmayacaktır.

Bütün bu durumlar göz önüne alındığı zaman, dışardan hiçbir müdahalede bulunulmadığı halde bir diskten farklı zamanlarda alınan imajların, hesaplatılan hash değerlerinin farklı olmasını sebebi, okunamayan veya kısmen okunan sektörün bir başka denemede okunabilecek olması, sektörün ilk başta yanlış okunmuş fakat sonraki denemede doğru okunmuş olması gibi diskten veya imaj alma yazılımından kaynaklanan sebeplerdir [29].

3.5. DELİL BÜTÜNLÜĞÜ VE HASH FONKSİYONU

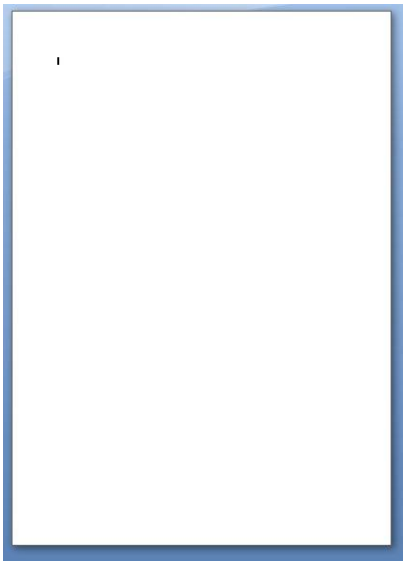
Teknolojik gelişmeler ile birlikte internet; dünya çapında çok fazla kullanılan bir kanal olmuştur. İnternetin çok fazla kullanılmasının doğal sonucu olarak da internet aracılığıyla dünya genelinde dolaşan verinin doğru olup olmadığının ya da tasdikinin tespit edilmesi gerekliliği gibi bir gereklilik ortaya çıkmıştır. Bilişim sistemlerinin tam ve en verimli şekilde kullanılması için de bu gerekliliğin ve buna bağlanacak hüküm ve sonuçların ispat edilmesi gerekmektedir. Bu sayede de bilişim sistemlerinden tam olarak yararlanılmış olunacaktır.

Bu duruma ilişkin olarak, bir elektronik delilin üzerinde durulması gereken ve daha önceki bölümde de bahsetmiş olduğumuz hususlardan birisi de delil bütünlüğü ilkesidir. Delil bütünlüğü ilkesi, olay yerinden alınan ve üzerinde inceleme yapılıp sonuçlandırılan delillerin, hiçbir değişikliğe uğramaksızın herhangi bir zamanda herhangi bir kişi tarafından da incelendiğinde aynı sonucu vermesi gerektiği ilkesidir. Burada bahsedilen bütünlük; imaj esnasında hash değerinin belirlenmesi ve yazılı bir şekilde kayıt altına alınmasıyla sağlanır. Daha sonraki işlemler, bu hash değerine bağlı kalınarak yapılacağı için meydana gelebilecek olası itirazlar bu sayede önlenecektir.

Hash değeri, ait olduğu delile özgü olduğu için, bu delilde meydana gelebilecek en ufak bir değişiklik hash değerinde de değişikliğe sebep olacaktır. Bu durumda da bir delil üzerinde hash değeri hesaplanıp incelemesi yapıldıktan sonra bu delilde bir değişikliğin olup olmadığı hash değerine bakılarak kontrol edilebilecektir [30].

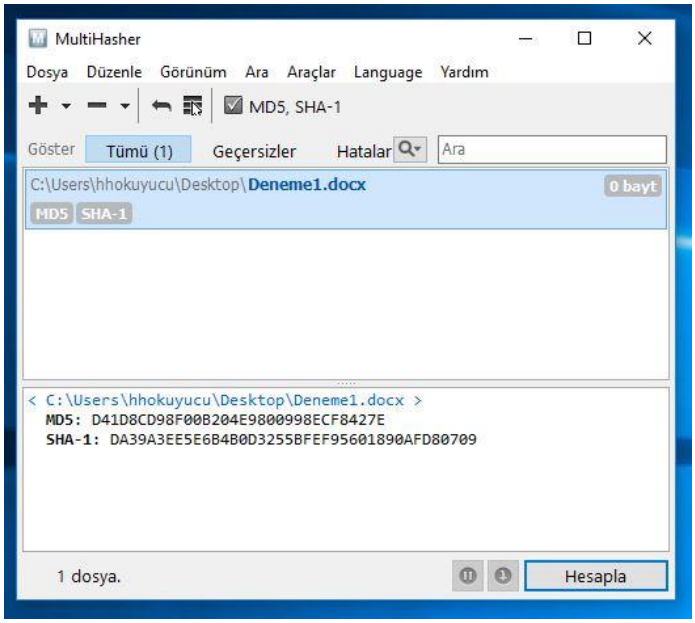
Bir örnek yardımıyla, delil veya bir dosya üzerinde yapılacak ufak bir değişikliğin hash değerini de değiştirileceğini açıklamaya çalışalım:

Öncelikle bilgisayar üzerinde "Deneme1" adında boş bir word dökümanı oluşturalım.



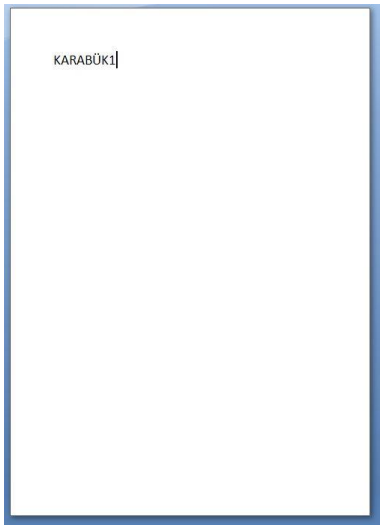
Şekil 3.6. İçine herhangi bir veri yazılmamış boş word belgesi.

Bu dökümanın sahip olduğu hash değerini “Multihasher” adlı hash değeri hesaplayan bir programla hesaplayalım:



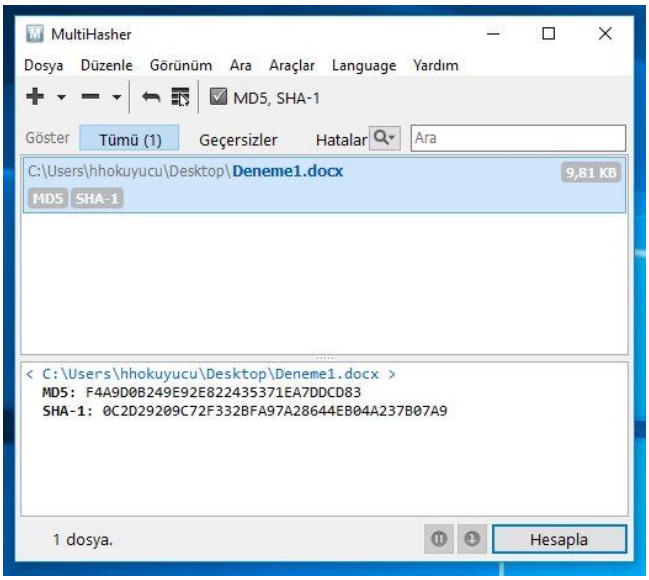
Şekil 3.7. Boş dokümanın hash değeri.

Şimdi de, hash değerini değiştirmeye yönelik, "Deneme1" adlı dosyanın daha önceden boş olan içeriğine herhangi bir metin yazalım ve kaydedelim:



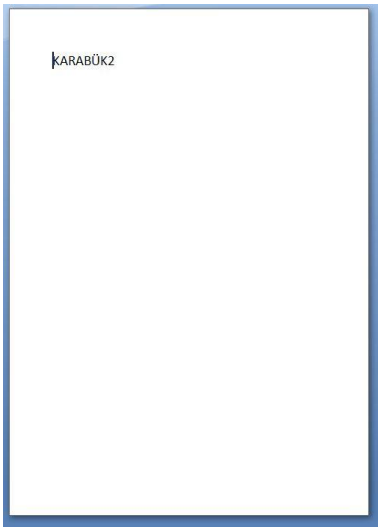
Şekil 3.8. Karabük1 yazısı eklenmiş word dokümanı.

Daha sonra bu dokümanın hash değerini gene aynı şekilde "Multihasher" programı ile hesaplayalım:



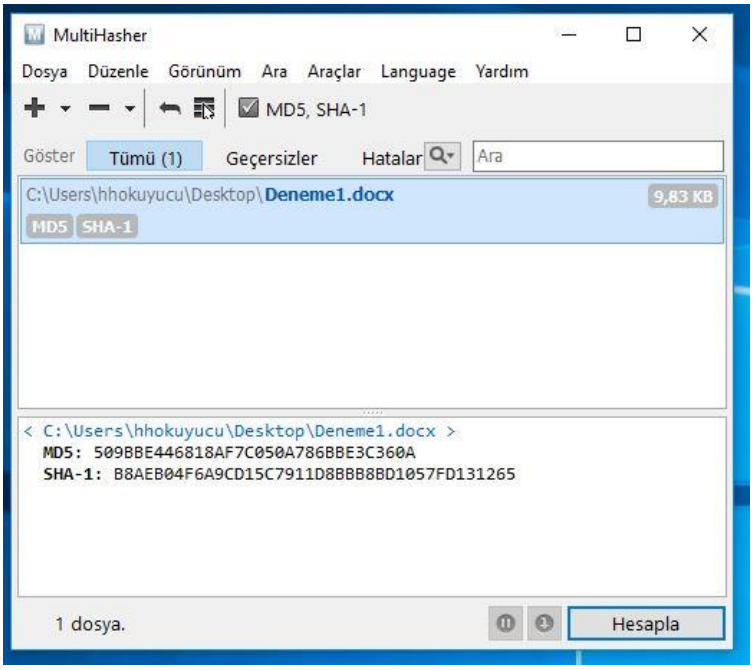
Şekil 3.9. Karabük1 yazılmış word dokümanının hash değeri.

Şimdi de aynı örnek üzerinden küçük bir değişiklik daha yapalım ve hash değerini hesaplatalım:



Şekil 3.10. Karabük2 yazılmış word dokümanı.

Yapılan bu değişikliği kaydedip "Multihasher" ile bu dokümanın hash değerini hesaplayalım:



Şekil 3.11. Karabük2 yazılmış word dokümanının hash değeri.

Yukarıdaki örnekte yaptığımız işlemleri bir excel tablosu ile açıklayacak olursak,

Çizelge 3.4. Oluşan hash değerlerinin karşılaştırılması.

Dosya Adı	İçerik	MD5 Hash Değeri	SHA1 Hash Değeri
Deneme1	BOŞ	D41D8CD98F00B204E9800998ECF8427E	DA39A3EE5E6B4B0D3255BFEF95601890 1890AFD80709
Deneme1	KARABÜK1	F4A9D0B249E92E822435371EA7DDCD83	0C2D29209C72F332BFA97A28644EB04A237B07A9
Deneme1	KARABÜK2	509BBE446818AF7C050A786BBE3C360A	B8AEB04F6A9CD15C7911D8BBB8BD1057FD131265

Tabloya göre, bir belge üzerinde değişiklik yapıldıktan sonra, adli bilişim sistemlerinde belgenin ya da delilin kimliği olarak adlandırılan hash değerinin değiştiği görülmüştür. En çok kullanım alanı "Dosya bütünlüğü" olan hash fonksiyonlarının, belge üzerinde ufak bir değişiklik yapıldıktan sonra değiştiğini örnekle görmüş olduk. Bu yüzden bir delilin, delil olarak kalması ve bütünlüğünün bozulmaması isteniyorsa delil üzerinde asla değişiklik yapılmamalıdır. Ayrıca üçüncül kişiler tarafından delil üzerinde değişiklik yapılıp yapılamadığı da bu yöntemle öğrenilebilir.

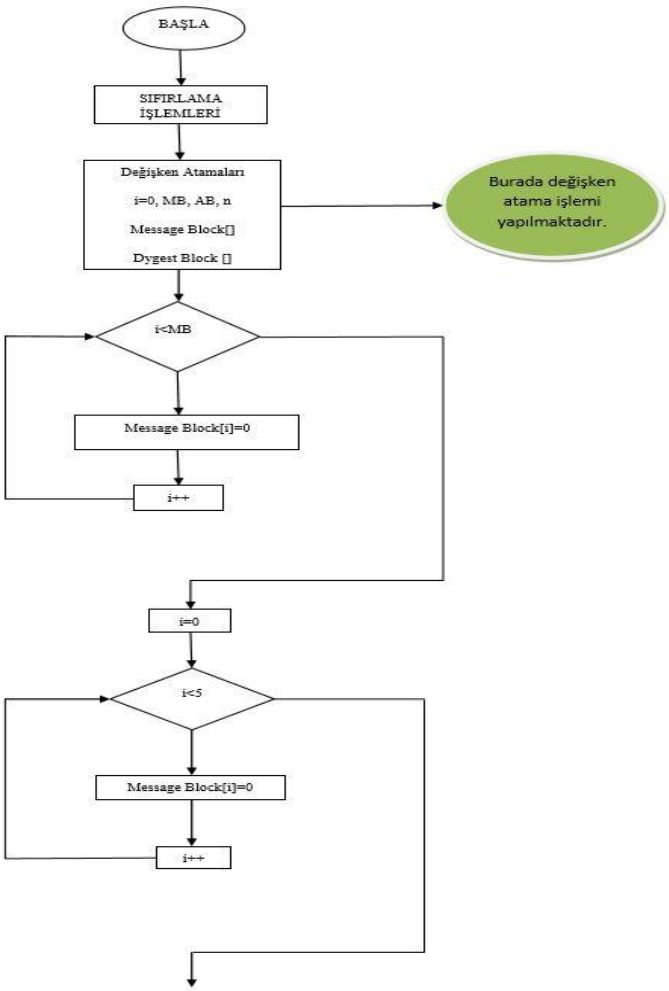
BÖLÜM 4

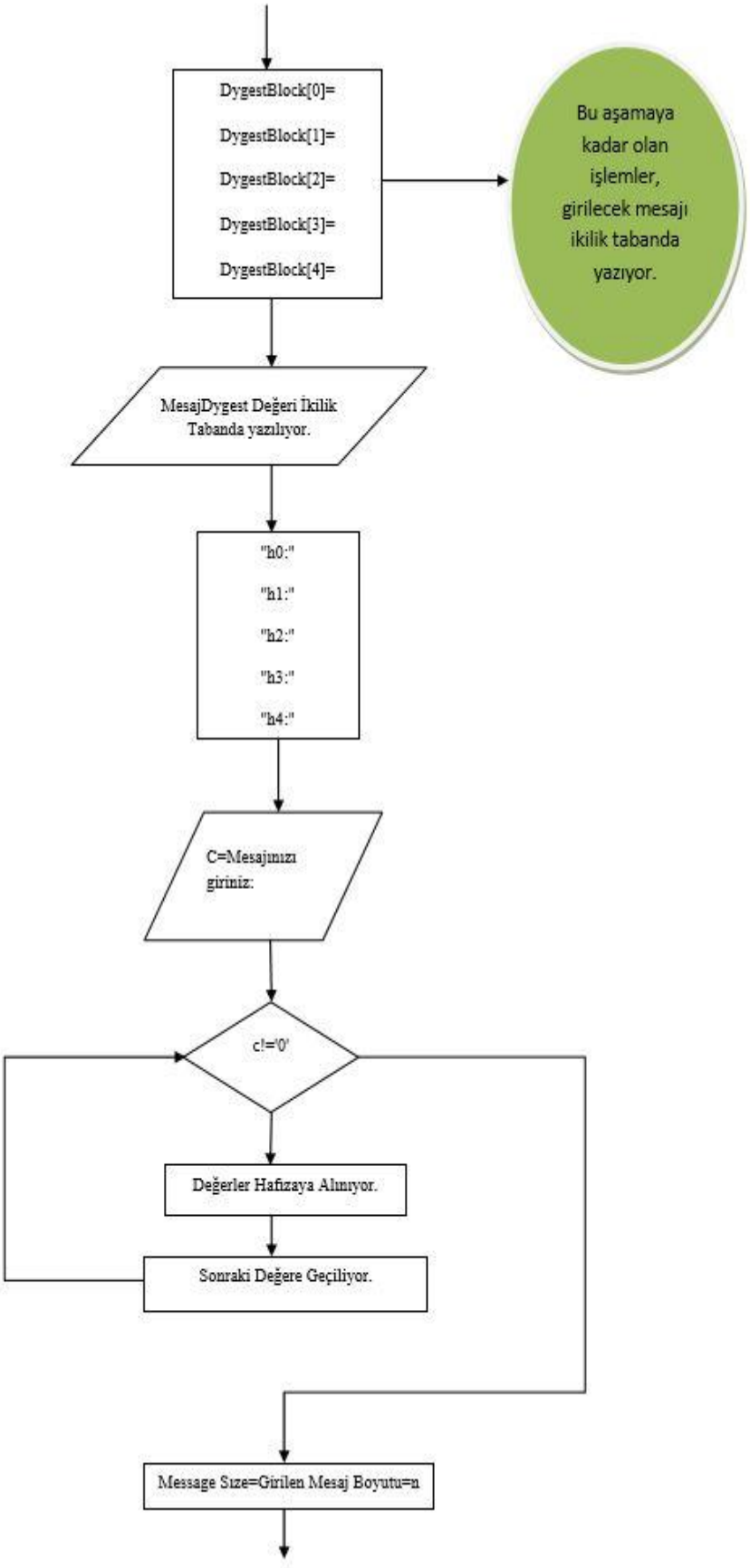
C++ PROGRAMLAMA DİLİ İLE BİR VERİNİN HASH DEĞERİNİ HESAPLAYAN PROGRAM ALGORİTMASI

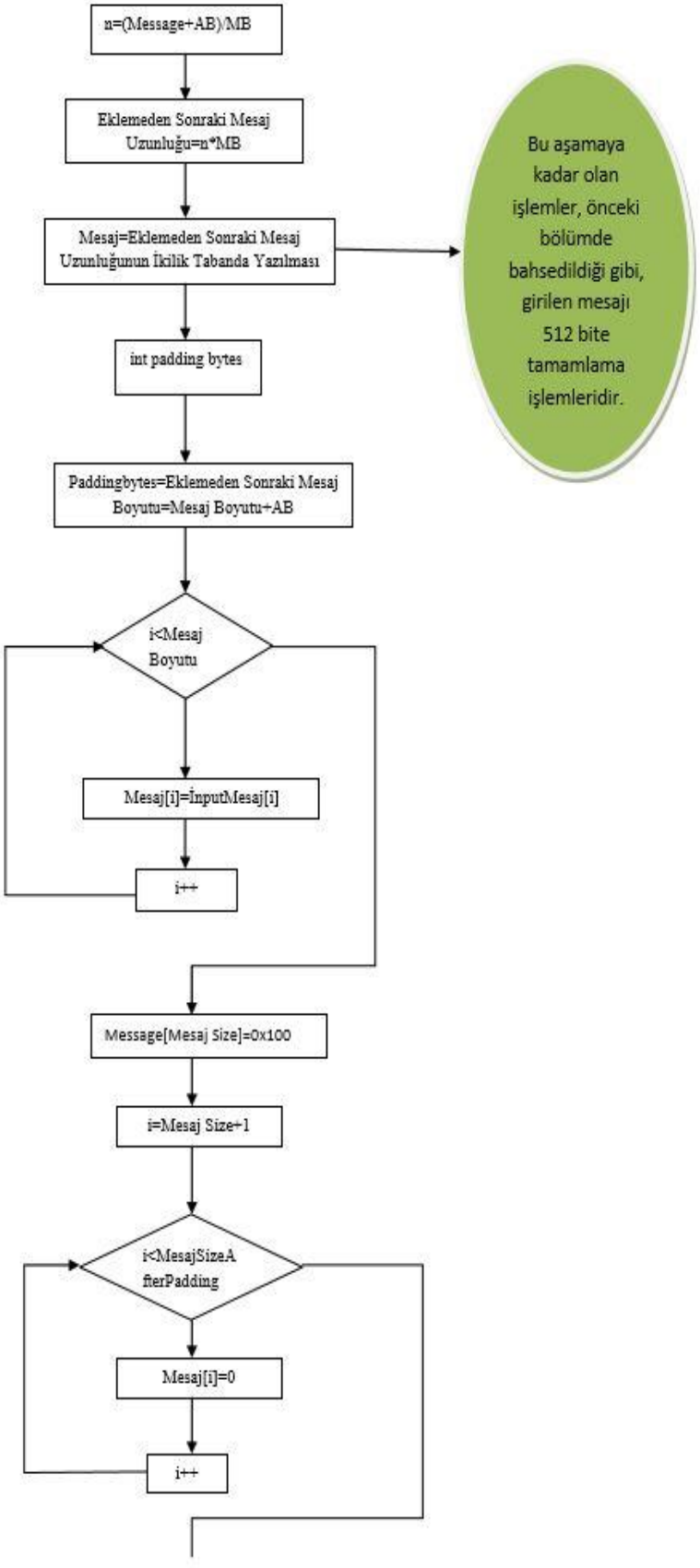
Bu bölümde, tezin ilk bölümlerinde bahsedilen hash fonksiyonlarından olan SHA1 özetleme fonksiyonunun C++ kodları yardımıyla gerçekleştirilen uygulamasının algoritması yazılacak ve güvenilirliğinden bahsedilecektir.

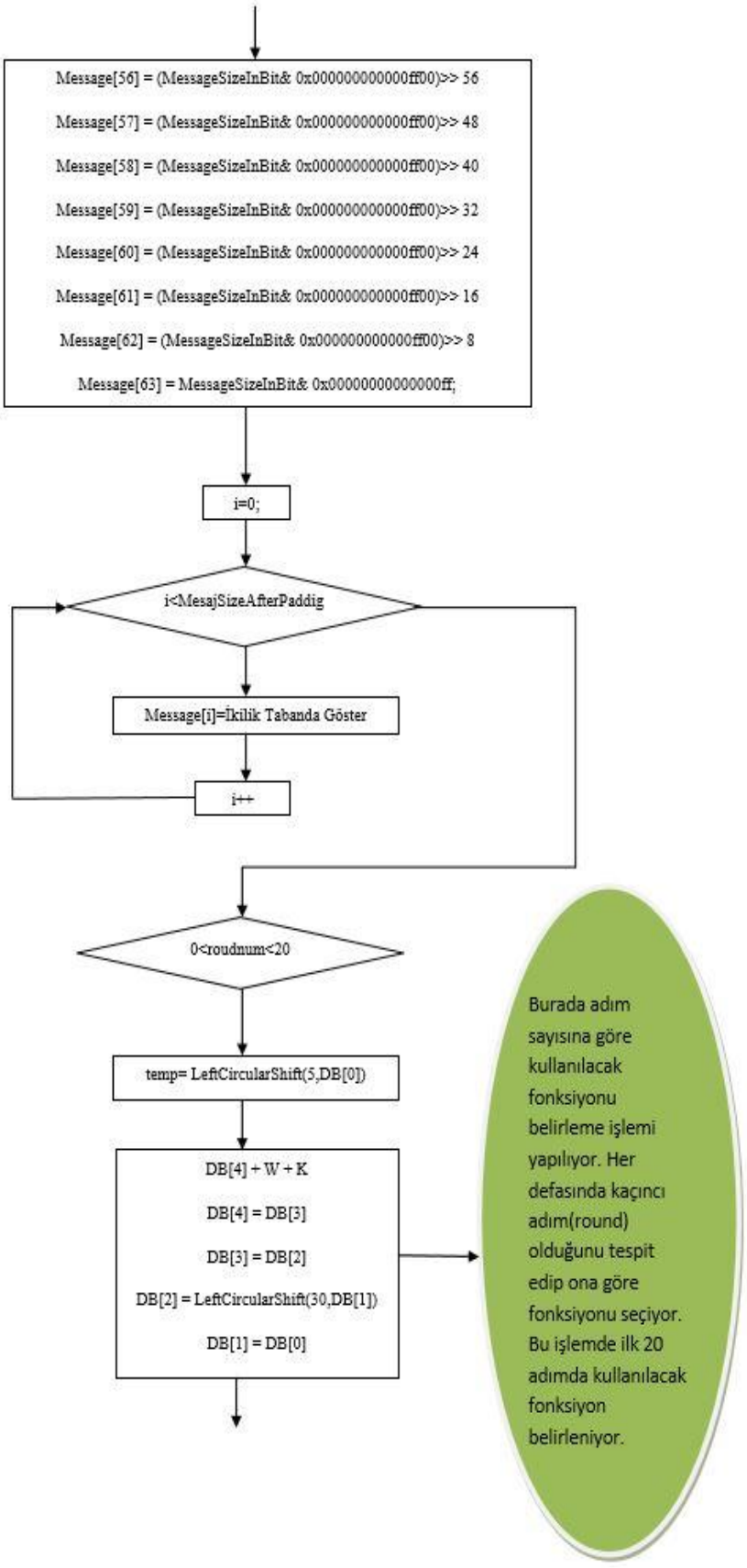
4.1. PROGRAMIN ALGORİTMASI ve ÇALIŞMASI

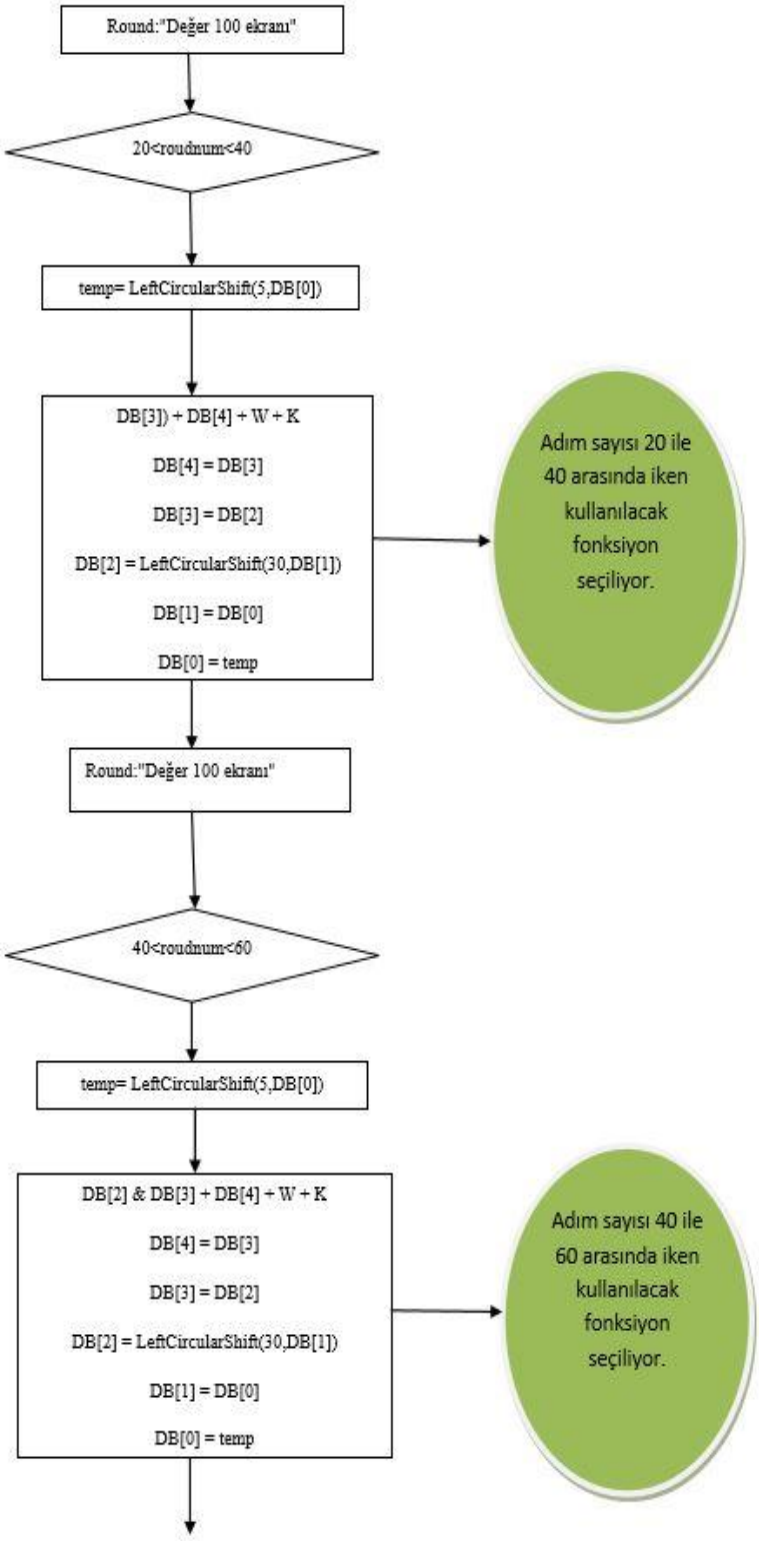
4.1.1.Programın Algoritması

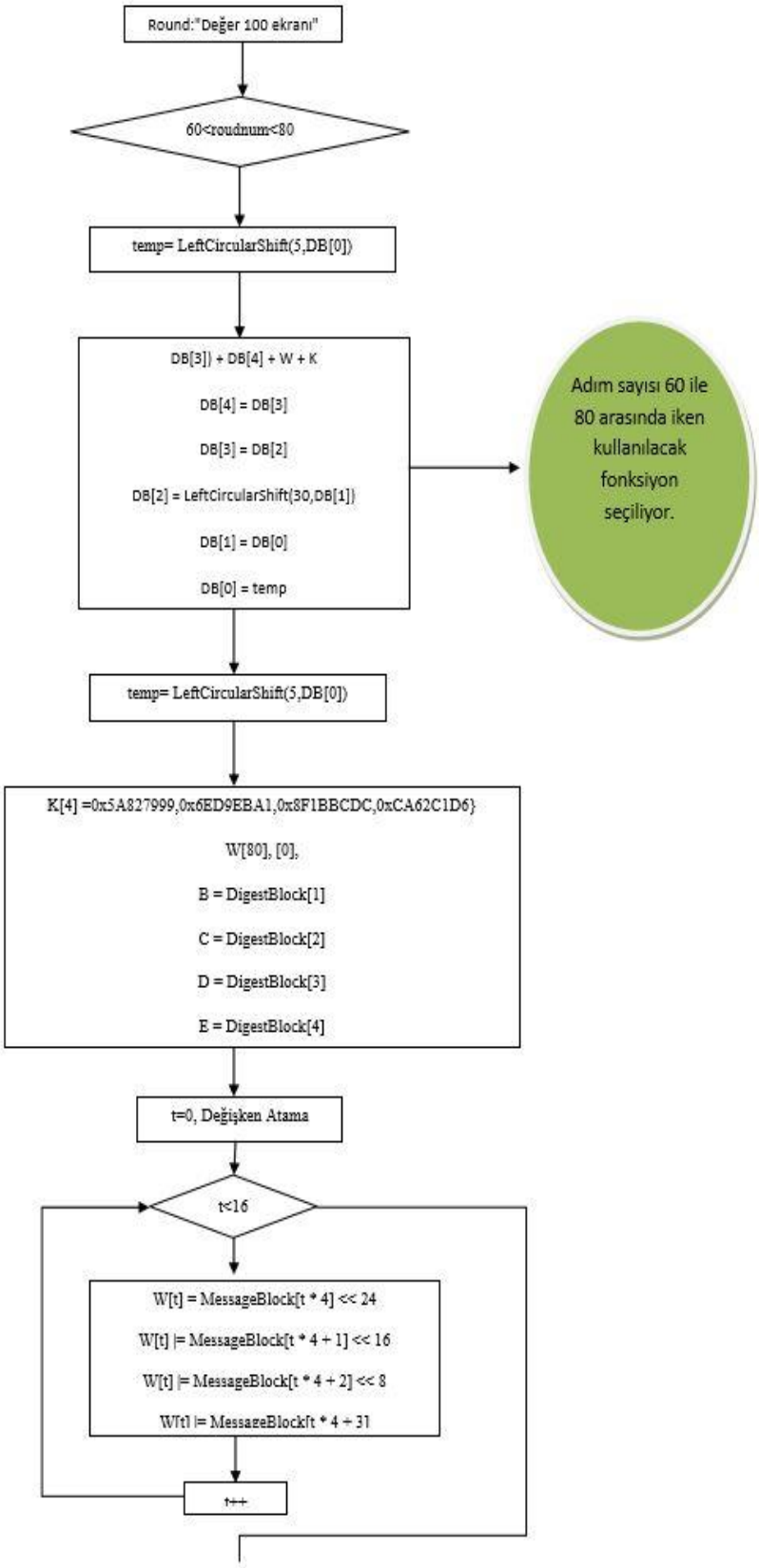


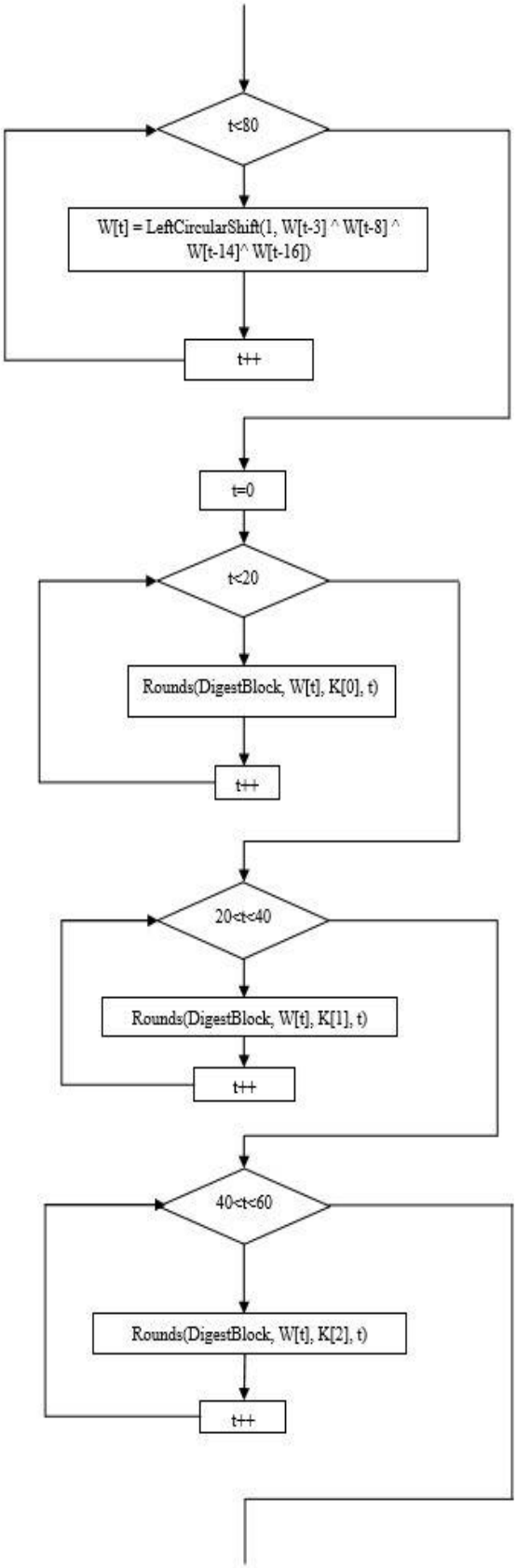

DygestBlock[0]=
DygestBlock[1]=
DygestBlock[2]=
DygestBlock[3]=
DygestBlock[4]=
Bu aşamaya kadar olan işlemler, girilecek mesajı ikilik tabanda yazıyor.
MesajDygest Değeri İkilik Tabanda yazılıyor.
"h0:"
"h1:"
"h2:"
"h3:"
"h4:"
C=Mesajınızı giriniz:
c!='0'
Değerler Hafizaya Alınıyor.
Sonraki Değere Geçiliyor.
Message Size=Girilen Mesaj Boyutu=n

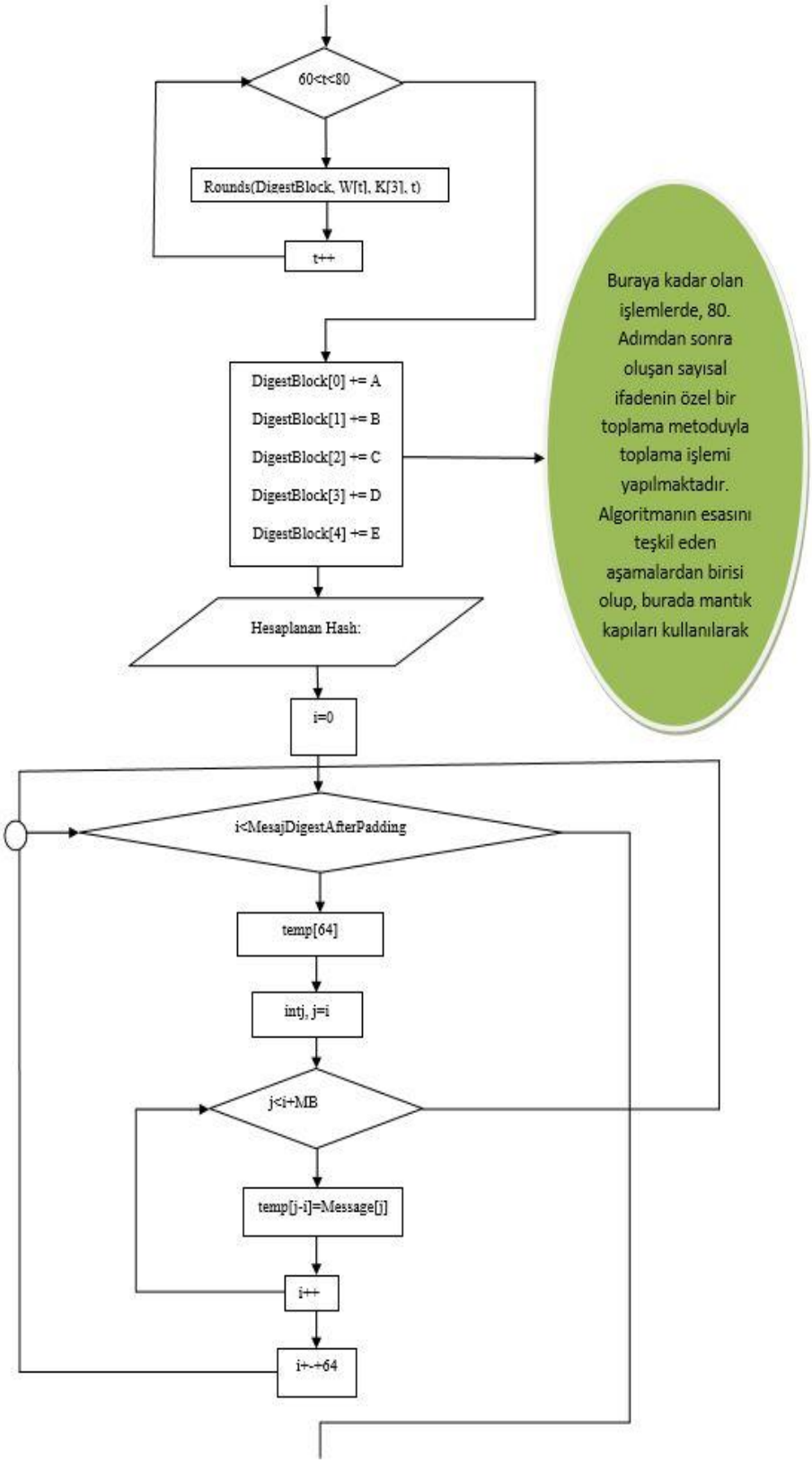

n=(Message+AB)\MB
Eklemeden Sonraki Mesaj Uzunluğu=n*MB
Mesaj=Eklemeden Sonraki Mesaj Uzunluğunun İkilik Tabanda Yazılması
Bu aşamaya kadar olan işlemler, önceki bölümde bahsedildiği gibi, girilen mesajı 512 bite tamamlama işlemleridir.
int padding bytes
Paddingbytes=Eklemeden Sonraki Mesaj Boyutu=Mesaj Boyutu+AB
i<Mesaj Boyutu
Mesaj[i]=İnputMesaj[i]
i++
Message[Mesaj Size]=0x100
i=Mesaj Size+1
i<MesajSizeAfterPadding
Mesaj[i]=0
i++

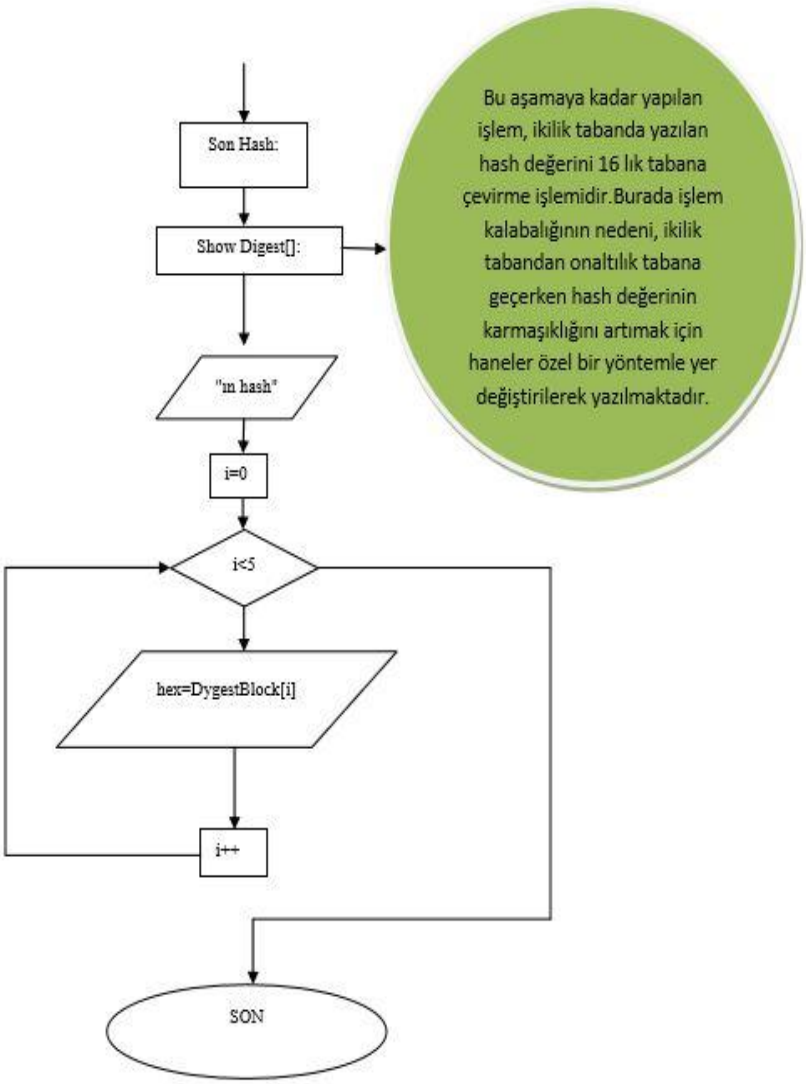

Message[56] = (MessageSizeInBit& 0x000000000000ff00)>> 56
Message[57] = (MessageSizeInBit& 0x000000000000ff00)>> 48
Message[58] = (MessageSizeInBit& 0x000000000000ff00)>> 40
Message[59] = (MessageSizeInBit& 0x000000000000ff00)>> 32
Message[60] = (MessageSizeInBit& 0x000000000000ff00)>> 24
Message[61] = (MessageSizeInBit& 0x000000000000ff00)>> 16
Message[62] = (MessageSizeInBit& 0x000000000000ff00)>> 8
Message[63] = MessageSizeInBit& 0x00000000000000ff;
i=0;
i<MesajSizeAfterPaddig
Message[i]=İkilik Tabanda Göster
i++
0<roudnum<20
temp= LeftCircularShift(5,DB[0])
DB[4] + W + K
DB[4] = DB[3]
DB[3] = DB[2]
DB[2] = LeftCircularShift(30,DB[1])
DB[1] = DB[0]
Burada adım sayısına göre kullanılacak fonksiyonu belirleme işlemi yapılıyor. Her defasında kaçıncı adım(round) olduğunu tespit edip ona göre fonksiyonu seçiyor. Bu işlemde ilk 20 adımda kullanılacak fonksiyon belirleniyor.


Round:"Değer 100 ekranı"
20<roudnum<40
temp= LeftCircularShift(5,DB[0])
DB[3]) + DB[4] + W + K
DB[4] = DB[3]
DB[3] = DB[2]
DB[2] = LeftCircularShift(30,DB[1])
DB[1] = DB[0]
DB[0] = temp
Adım sayısı 20 ile 40 arasında iken kullanılacak fonksiyon seçiliyor.
Round:"Değer 100 ekranı"
40<roudnum<60
temp= LeftCircularShift(5,DB[0])
DB[2] & DB[3] + DB[4] + W + K
DB[4] = DB[3]
DB[3] = DB[2]
DB[2] = LeftCircularShift(30,DB[1])
DB[1] = DB[0]
DB[0] = temp
Adım sayısı 40 ile 60 arasında iken kullanılacak fonksiyon seçiliyor.


Round:"Değer 100 ekranı"
60<roudnum<80
temp= LeftCircularShift(5,DB[0])
DB[3]) + DB[4] + W + K
DB[4] = DB[3]
DB[3] = DB[2]
DB[2] = LeftCircularShift(30,DB[1])
DB[1] = DB[0]
DB[0] = temp
Adım sayısı 60 ile 80 arasında iken kullanılacak fonksiyon seçiliyor.
temp= LeftCircularShift(5,DB[0])
K[4] =0x5A827999,0x6ED9EBA1,0x8F1BBCDC,0xCA62C1D6}
W[80], [0],
B = DigestBlock[1]
C = DigestBlock[2]
D = DigestBlock[3]
E = DigestBlock[4]
t=0, Değişken Atama
t<16
W[t] = MessageBlock[t * 4] << 24
W[t] |= MessageBlock[t * 4 + 1] << 16
W[t] |= MessageBlock[t * 4 + 2] << 8
W[t] |= MessageBlock[t * 4 + 31
t++


t<80
W[t] = LeftCircularShift(1, W[t-3] ^ W[t-8] ^ W[t-14]^ W[t-16])
t++
t=0
t<20
Rounds(DigestBlock, W[t], K[0], t)
t++
20<t<40
Rounds(DigestBlock, W[t], K[1], t)
t++
40<t<60
Rounds(DigestBlock, W[t], K[2], t)
t++


60<t<80
Rounds(DigestBlock, W[t], K[3], t)
t++
DigestBlock[0] += A
DigestBlock[1] += B
DigestBlock[2] += C
DigestBlock[3] += D
DigestBlock[4] += E
Hesaplanan Hash:
i=0
i<MesajDigestAfterPadding
temp[64]
intj, j=i
j<i+MB
temp[j-i]=Message[j]
i++
i+-+64
Buraya kadar olan işlemlerde, 80. Adımdan sonra oluşan sayısal ifadenin özel bir toplama metoduyla toplama işlemi yapılmaktadır. Algoritmanın esasını teşkil eden aşamalardan birisi olup, burada mantık kapıları kullanılarak



Şekil 4.1. Algoritmanın akış diyagramı.

4.1.2. Programın Çalışması

Tezin bu bölümünde yazılan algoritmanın çalışma mantığından bahsedilecektir. Temel olarak hash değeri hesaplayan bütün özetleme fonksiyonlarının kullandığı algoritma yukarıdaki şekildeki gibi olsa da, rastgele atanan değişkenlerden dolayı, hash değeri hesaplayan farklı programlarda aynı verinin hash değerinin farklı çıkması gayet olağan bir durumdur. Böyle bir durumda yapılması gereken, hash değerinin hesaplatıldığı yazılımın, öteki hesaplamalarda da temel alınması gereklidir.

Önceki bölümde de bahsedildiği üzere, hash değeri tek yönlü bir fonksiyondur. Oluşturulan veri özeti değerine bakılarak esas veri hakkında herhangi bir çıkarımda bulunulamaz. Tek yönlü fonksiyonlarda bu özelliği ve oluşturulan değerin yalnızca ait olduğu veriye ait olduğu özelliğini sağlamak için, hash fonksiyonları oldukça fazla matematiksel işleme tabii tutulurlar.

Tezde yazılan algoritmayı bu şekilde açıklamaya çalışalım:

Program öncelikle, her defasında çalıştığında önceki verileri yok etmek adına sıfırlama işlemi yapmaktadır. Bu işlemin ardından, fonksiyonda kullanılacak olan ve rastgele seçilen değişken atama işlemi yapılmakta ve her değişken atama işleminden sonra atanan değişkeni ikilik tabanda yazılmaktadır. Bu işlemlerden sonra, hash değerini hesaplatacağımız veriyi ekrana girmemizi isteyen program, biz veriyi girdikten sonra, veriyi kontrol edip uzunluğunu tespit etmektedir. Sonraki aşamada, uzunluk, ikilik tabanda yazılmaktadır.

MD5 algoritmasında ve SHA1 algoritmasında, mesajın hash değerinin hesaplatılması için, mesaj ve mesaj uzunluğunun toplamda 512 bit olması gerektiği önceki bölümde belirtildi. Mesaj uzunluğunun 64 bit olarak ifade edileceğini düşünürsek, mesajın 512-64=448 bit olması gerektiği sonucu ortaya çıkar. Bu durumda, mesajın boyutu ikilik tabanda ne kadar uzunlukta olursa olsun, program 512 bite tamamlama işlemi yapar ve daha sonra matematiksel işlemlerin olduğu aşamaya geçilir.

Matematiksel işlemler oldukça karmaşık bir yapıya sahiptir. Önceki aşamada 512 bit şeklinde ifade edilen mesaj, matematiksel işlemler aşamasında, 32 bitler halinde 16 parçaya ayrılırlar ve her defasında 16 bitlik gruplar halinde işleme tabii tutulurlar. Toplamda 80 defa 3. Bölüm Şekil 5'te belirtilen algoritmadaki işleme tabii tutulan mesaj, çıkıştan sonra şekildeki gibi toplanır. Herbir adıma round denir ve toplam 80 round(işlem adımı) vardır. Üretilen mesajın çözümünü imkânsız hale getirmek için ve algoritmanın analizinin yapılmasını engellemek için matematiksel ifadeler oldukça karmaşık bir yapı haline getirilmiştir. Burada anlatılmak istenen, toplamda 80 round var iken, kullanılan "f" fonksiyonu her 20 roundda değiştirilmiştir.

Örneğin,

1 ile 20. Round arasında; $f(i,A,B,C) = (A \wedge \mathrm{B}) \vee (-\mathrm{A} \wedge \mathrm{C})$

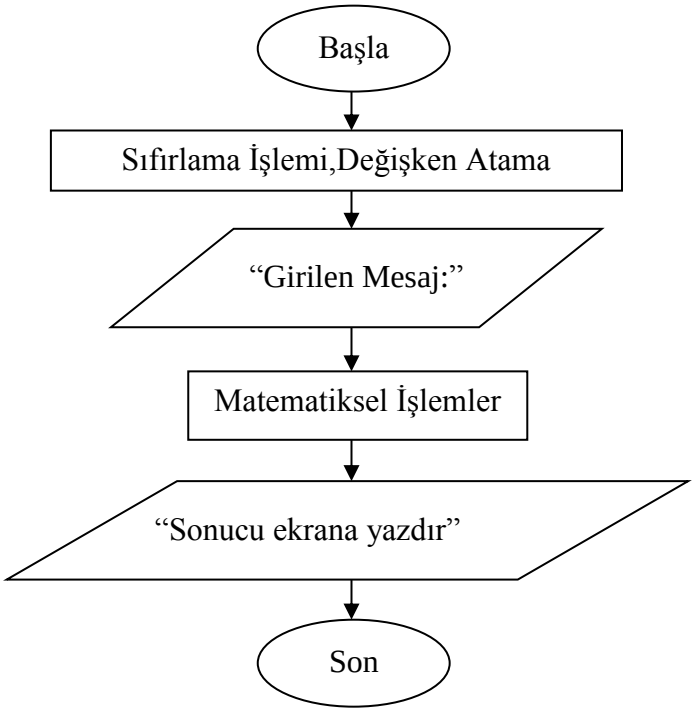
20 ile 40. Round arasında; $f(i,A,B,C) = A \oplus \mathrm{B} \oplus \mathrm{C}$

40 ile 60. Round arasında; $f(i,A,B,C) = (A \wedge \mathrm{B}) \vee (\mathrm{A} \wedge \mathrm{C}) \vee (\mathrm{B} \wedge \mathrm{C})$

60 ile 80. Round arasında $f(i,A,B,C) = A \oplus \mathrm{B} \oplus \mathrm{C}$ fonksiyonları kullanılmıştır.

80. rounddan sonra ikilik tabanda 160 bitlik veri oluşmaktadır.Oluşan bu veri hash değeridir ve 16 lık tabana dönüştürülüp ekrana yazılır ve program sonlanmış olur.

Bütün bu algoritmayı ise en basite indirgeyecek olursak;



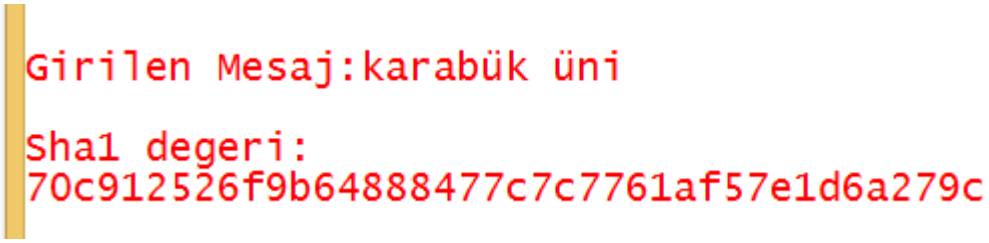
Şekil 4.2. C++ yardımıyla yazılan algoritmanın basite indirgenmiş hali.

4.2. PROGRAMIN DOĞRULUĞU ve SONUÇLAR

Bir üstteki bölümde, program çalıştırılıp gerekli işlemler yapıldıktan sonra, oluşan 160 bitlik ve onaltılık tabanda 40 haneli olan veriye "mesaj özeti değeri" ya da "hash değeri" denilmiştir. Tezin ilk bölümlerinde bir imaj ya da herhangi bir veri üzerinde yapılan en ufak bir değişikliğin bile, mesaj özeti değerini değiştireceğini, bu yüzden, verilerde herhangi bir değişikliğin yapılmaması gerektiği aksi takdirde, mesaj bütünlüğünün bozulacağı ve delil bütünlüğünün zarar görebileceği konuları anlatıldı. Şimdi ise, C++ yardımıyla yazılmış olan ve yukarıda algoritması verilen program aracılığıyla, veriler üzerinde değişiklikler yapılarak sonuçlar tartışılacaktır.

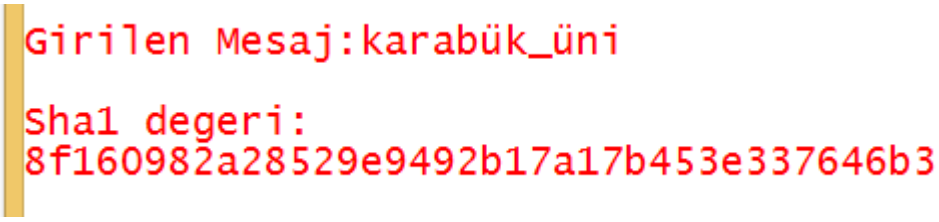
Bir uygulama yapacak olunursa;

"karabük üni" kelimesinin mesaj özeti değerini hesaplayalım:



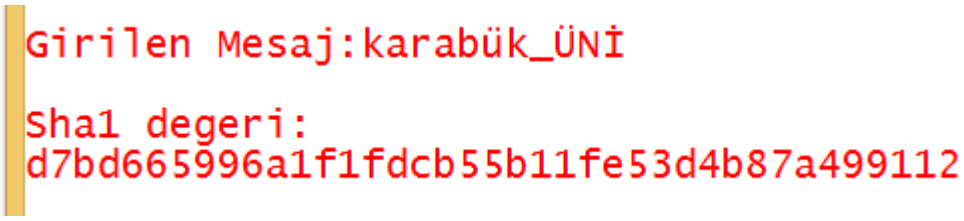
Şekil 4.3. Orijinal metin ve Hash değeri.

Örnek 1: İlk kelime üzerinde bir değişiklik yaparak "karabük_üni" kelimesinin mesaj özeti değerini hesaplayalım:



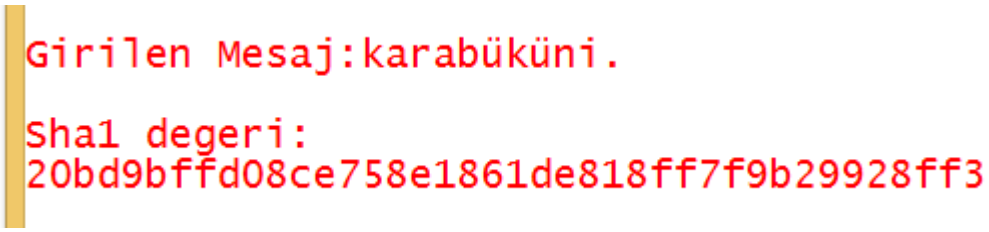
Şekil 4.4. Örnek 1 ve Hash değeri.

Örnek 2: İlk kelime üzerinde başka bir değişiklik yaparak "karabük_ÜNİ" kelimesinin mesaj özeti değerini hesaplayalım:



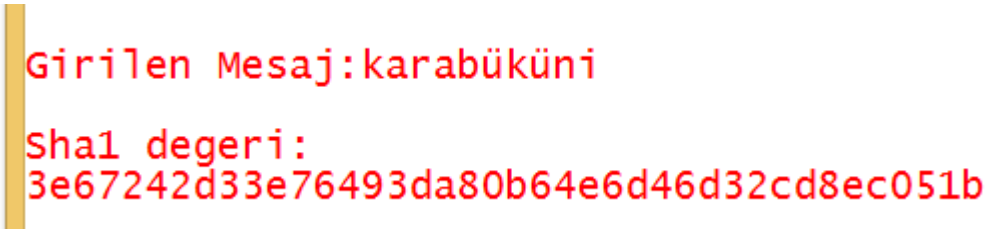
Şekil 4.5. Örnek 2 ve Hash değeri.

Örnek 3: İlk kelime üzerinde başka bir değişiklik yaparak "karabüküni." kelimesinin mesaj özeti değerini hesaplayalım:



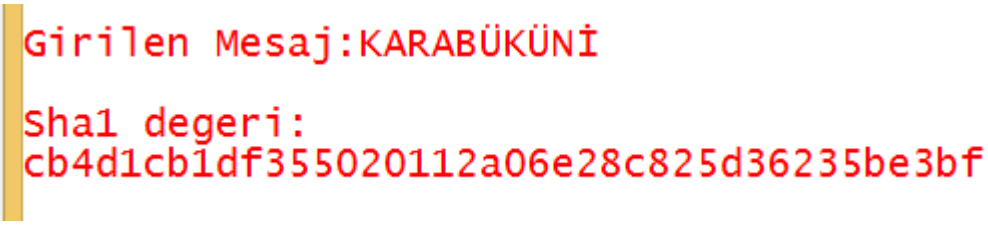
Şekil 4.6. Örnek 3 ve Hash değeri.

Örnek 4: İlk kelime üzerinde başka bir değişiklik yaparak "karabüküni" kelimesinin mesaj özeti değerini hesaplayalım:



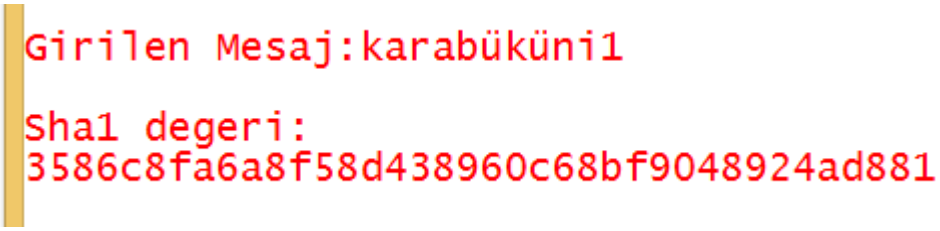
Şekil 4.7. Örnek 4 ve Hash değeri.

Örnek 5: İlk kelime üzerinde başka bir değişiklik yaparak "KARABÜKÜNİ" kelimesinin mesaj özeti değerini hesaplayalım:



Şekil 4.8. Örnek 5 ve Hash değeri.

Örnek 6: İlk kelime üzerinde başka bir değişiklik yaparak "karabüküni1" kelimesinin mesaj özeti değerini hesaplayalım:



Şekil 4.9. Örnek 6 ve Hash değeri.

Yukarıda yapılan örneklerde görüldüğü gibi, bir veri üzerinde yapılan ufak bir değişiklik sonucu, orijinal veri üzerinde hesaplanan mesaj özeti değeri de değişmektedir.

Çizelge 4.1. Hash değerlerinin karşılaştırılması.

	Orijinal Mesaj	**Hash(Mesaj Özeti) Değeri**
	karabük üni	70c912526f9b64888477c7c7761af57e1d6a279c
Örnek	**Değişiklik Sonucu Oluşan Mesaj**	**Hash(Mesaj Özeti) Değeri**
1	karabük_üni	8f160982a28529e9492b17a17b453e337646b3
2	karabük_ÜNİ	d7bd665996a1f1fdcb55b11fe53d4b87a499112
3	karabüküni.	20bd9bffd08ce758e1861de818ff7f9b29928ff3
4	karabüküni	3e67242d33e76493da80b64e6d46d32cd8ec051b
5	KARABÜKÜNİ	cb4d1cb1df355020112a06e28c825d36235be3bf
6	karabüküni1	3586c8fa6a8f58d438960c68bf9048924ad881

Tabloya bakıldığı zaman, bir mesaj üzerinde yapılan değişikliğin türü farketmeksizin, orijinal mesaj üzerinden hesaplanan “hash(mesaj özeti) değeri” yapılan değişiklikten dolayı değişmektedir.

Örnek 1’de orijinal mesaja yalnızca “alt tire” eklenmesine rağmen, Örnek 2’de büyük harf kombinasyonu yapılmasına rağmen, Örnek 3’te yalnızca “nokta” eklenmesine rağmen, Örnek 4’te orijinal mesajdan “boşluk” karakterinin kaldırılmasına rağmen, Örnek 5’te orijinal mesajın tamamen büyük harf kombinasyonu kullanılarak yazılmasına rağmen ve son olarak Örnek 6’da yalnızca sayısal karakter eklenmesine rağmen “hash (mesaj özeti) değeri” değişmiştir. Bu değişiklik, tezde iddia edilen durumu ispatlamasının yanı sıra yazılan programın da düzgün çalıştığını göstermektedir.

BÖLÜM 5

SONUÇLAR

Türk hukuk sisteminde, bir vakanın objektif şekilde sonuçlanmasını sağlayan en önemli faktörlerden birisi delillerdir. Deliller, vakaların türlerine ve bulundukları yere göre adli bilişim sistemlerinin konusu olabilmektedir. Adli bilişim sistemlerine konu olan deliller elektronik delil olarak adlandırılmakta ve değerlendirilmesi de adli bilişim sisteminin gerektirdiği ilke ve standartlara göre yapılmaktadır.

Elektronik delillerin değerlendirilmesi yapılırken, delil bütünlüğü ilkesi en önemli ilkelerden birisidir. Bu ilke gereğince elektronik delillerin sahip olduğu "hash(mesaj özeti)" değerinin hiçbir şartta ve durumda değişmemesi gerekmektedir. Aksi takdirde, elektronik delil, delil olma kabiliyetini yitirmiş olabilir ve mahkemece dikkate alınamayabilir. Bu durumda objektif değerlendirme sekteye uğramış olacaktır.

Bu çalışmada bir veri üzerinde yapılan değişikliğin, bu verinin sahip olduğu hash değerini de değiştirdiği ve bu durumun aynısının elektronik deliller için de geçerli olduğu C++ programlama dili yardımıyla yazılan program ile anlatılmıştır. Bu program ayrıca var olan öteki programlardan farklı olarak; giriş mesajının uzunluğuna ve girilen karakterlerin türüne bakılmaksızın bütün verilerin hash değerini hesaplamakta ve sonucu ekrana yazdırmaktadır.

Adli bilişim vakalarında, delil bütünlüğünü bozulduğu zaman delillerin mahkemece geçerliliğinin yitirilmesinin haricinde, bu bütünlüğü bozan kişiler de bir yargılamaya tabii tutulacağı için delil bütünlüğünü korumak önemlidir.

Bu tez, elektronik delillerin incelenmesi yapılırken; delil bütünlüğünün korunması gerektiğinin önemini vurgulamak, hash (veri özeti) değerini hesaplayan matematiksel

yapının daha anlaşılır olabilmesi için algoritmasını anlaşılır düzeyde ifade etmek, iki kişi arasında güvenli haberleşme yapılırken kontrol mekanizması olarak kullanılabilecek bir program yazmak ve en önemlisi de bir veri üzerinde yapılacak ufak bir değişikliğin bu verinin hash değerini değiştirilebileceğini göstermek amacıyla hazırlanmıştır.

KAYNAKLAR

1. Altheide, C. and Harlan C. ***Digital forensics with open source tools***. Elsevier, (2011).

2. Casey, E. ***Handbook of digital forensics and investigation***. Elsevier, (2010).

3. İnternet; Organization, US Goverment**.** Computer Forensics. **http://www.us-cert.gov/reading_room/forensics.pdf** (2012).

4. Whitcomb, C. M. "An historical perspective of digital evidence: A forensic scientist's view." ***International Journal of Digital Evidence***,1.1, 7-15. (2002).

5. Kaya, Y**.** "Bulut temelli adli bilişim", Yüksek Lisans Tezi, ***İstanbul Bilgi Üniversitesi Sosyal Bilimler Enstitüsü***, İstanbul (2016).

6. Ukşal, M. "Mobile Forensics", Yüksek Lisans Tezi, ***İstanbul Bilgi Üniversitesi Sosyal Bilimler Enstitüsü***, İstanbul (2015).

7. İnternet: *Açık Anahtar Altyapısı Destekli Dijital Delilleri Doğrulama Modeli.* Uzunay, Y. and Bıçakçı, K.. **http://www.emo.org.tr/ekler/4843973f9b66701_ek.pdf**. (2014).

8. İnternet: Ekizer, A. H.; "Adli Bilişim", **http://www.ekizer.net/content/view/16/1/** (2014).

9. Henkoğlu, T.. Adli bilişim: Dijital delillerin elde edilmesi ve analizi. ***Pusula***, İstanbul, (2011).

10. Değirmenci, O. "Ceza Muhakemesinde Sayısal (Dijital) Delil." ***Seçkin Yayıncılık***, Ankara, (2014).

11. Özocak, G. "Ceza Muhakemesinde Elektronik Delillerin Tespiti ve Toplanması" ***İzmir 2. Uluslararası Bilişim Hukuku Kurultayı, Bildiriler Kitabı*** 17-19 Kasım, (2011).

12. Tipton, H. F. and Micki K., eds. *Information Security Management Handbook, Volume 3*. Vol. 3. ***CRC press***, (2006).

13. Cardwell, K.. The Best Damm Cybercrime and Digital Forensics Book Period. Elsevier Inc, (2007).

14. İnternet; Öztürkci, H., "Adli Bilişim ve Bilişim Güvenliği Günlüğü", **https://halilozturkci.com/adli-bilisim-incelemelerinde-sabit-disk-imajlari-dd-raw-imaj-formati/** (2014).

15. **Okuyucu, H. H..** *Adli Tıp Kurumu Adli bilişim İhtisas Dairesi Raporu, (*2020).

16. **Bolat, M.,** *Encase Nedir? Özel Bilirkişilik ve Uzman Mütaalası,* Ankara, (2013).

17. **Altheide, C.** Digital Forensics with Open Source Tools. Waltham, USA, (2011).

18. İnternet: **Gürel, A.** Linux'ta DD ile İmaj Alma ve İmajın Canlandırılması. *Information Security Blog,* **https://gurelahmet.com/linuxta-dd-ile-imaj-alma-ve-imaj%C4%B1n-canland%C4%B1r%C4%B1lmas%C4%B1/.**

19. Şirikçi, A. S. and Nergis C., "Adli Bilişim İncelemelerinde Birebir Kopya Alınmasının (İmaj Almak) Önemi." ***Internatıonal Journal Of Informatıcs Technologıes*** 5.3, 29-34, (2012).

20. Adli Tıp Kurumu Adli Bilişim İhtisas Dairesi Metodolojisi, (2020).

21. İnternet: (FTK®), Forensic Toolkit®. Recognized around the World as the Standard in Computer Forensics Software. 2013 Mayıs. **http://www.accessdata.com/products/digital-forensics/ftk**, (2013).

22. Security, U.S. Department of Homeland. Best Practices For Seizing Electronic Evidence. A.B.D.: 4, ***United States Secret Service***, (2006).

23. Security U.S. Department of Homeland. Best Practices For Seizing Electronic Evidence, 4, *(*2006).

24. İnternet: *The Sleuth Kit (TSK) & Autopsy: Open Source Digital Investigation Tools.* Kit, The Sleuth**.. http://www.sleuthkit.org/sleuthkit/index.php**, (2013).

25. Ural, N. and Örenç, Ö., Kriptoloji. Uygulamalı Şifreleme ve Şifre Çözme Yöntemleri, ***Pusula***, (2018).

26. Bodur, H. Özetleme Fonksiyonları. ***Java Diliyle Kriptoloji Uygulamaları***. Vol. 2, p. 21, İstanbul, (2018)

27. İnternet: Kaya, M. Kullanıcı Uzayında Dinamik Boyutlu ve Şifreli bir Dosya Sisteminin Gerçekleştirilmesi.. p. 17. **http://www.enderunix.org/metin**, (2008).

28. Cihat, E. "Kriptografik hash fonksiyonlarının incelenmesi", Yüksek Lisans Tezi ***Trakya Üniversitesi Fen Bilimleri Enstitüsü***, Edirne (2012).

29. Al Mamun, A., GuoXiao Guo, and Chao Bi. Hard disk drive: mechatronics and control. CRC press, (2017).

30. Başlar, Y., "Elektronik Delilin Toplanması ve Muhafazası." ***Hacettepe Hukuk Fakültesi Dergisi*** 10.1: 77-107, (2020).

ÖZGEÇMİŞ

Hacı Hasan OKUYUCU, 1992 yılında, Yozgat'ın Yenifakılı ilçesinde doğdu. İlk ve orta eğitimini bu ilçede, liseyi Boğazlıyan ilçesinde bulunan Boğazlıyan Lisesi'nde tamamladı. 2010 yılında lisans eğitimine Atatürk Üniversitesi (Erzurum) Elektrik-Elektronik Mühendisliği bölümünde başladı ve 2014 yılında mezun oldu. Çeşitli özel sektör deneyimlerinin ardından 2017 yılının Nisan ayında, Adalet Bakanlığına bağlı Adli Tıp Kurumuna Adli Bilişim Uzmanı olarak atandı. Evli ve bir çocuk babası olup, halen burada çalışmaktadır.

ADRES BİLGİLERİ

Adres : Adli Tıp Kurumu Ankara Grup Başkanlığı
Şevkat Mah. Dr. Besim Ömer Cd. No:45
Keçiören / ANKARA
E-posta :okuyucuhacihasan@gmail.com