T.C.
EGE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

SİMETRİK KRİPTO-SİSTEMLERİN GÜVENLİK ANALİZİ

Melike KARATAY

Danışman: Prof. Dr. Urfat NURİYEV
İkinci Danışman: Dr. Öğr. Üyesi Erdem ALKIM

Matematik Anabilim Dalı
Bilgisayar Bilimleri Yüksek Lisans Programı

İzmir
2019

Melike KARATAY tarafından yüksek lisans tezi olarak sunulan "Simetrik Kripto-Sistemlerin Güvenlik Analizi" başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 18/12/2019 tarihinde yapılan tez savunma sınavında aday oybirliği/~~oyçokluğu~~ ile başarılı bulunmuştur.

Jüri Üyeleri:		**İmza**
Jüri Başkanı	**: Prof. Dr. Urfat NURİYEV**	.....
Raportör Üye	**: Dr. Öğr. Üyesi Arif GÜRSOY**	.....
Üye	**: Dr. Öğr. Üyesi Onur UĞURLU**	.....

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum "Simetrik Kripto-Sistemlerin Güvenlik Analizi" başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

18 / 12 / 2019

Melike Karatay

ÖZET

SİMETRİK KRİPTO-SİSTEMLERİN GÜVENLİK ANALİZİ

KARATAY, Melike

Yüksek Lisans Tezi, Matematik Anabilim Dalı

Tez Danışmanı: Prof. Dr. Urfat NURİYEV

İkinci Tez Danışmanı: Dr. Öğr. Üyesi Erdem ALKIM

Aralık 2019, 80 sayfa

Özet fonksiyonları, seçilen uzunluktaki bir girdiyi işleyerek, sabit uzunluklu eşsiz bir çıktıya ulaştıran kriptografik fonksiyonlardır. Özet fonksiyonlarının güvenlik sınırlarının belirlenmesi oldukça zordur. Bu tezde, simetrik kriptografinin bir elemanı olan özet fonksiyonlarının güvenliklerinin belirlenmesinde kullanılan yöntemler açıklanmıştır. Amerika Standartlar ve Teknoloji Enstitüsü (NIST) tarafından 2018 yılında düzenlenen hafif-siklet kriptografi projesi kapsamında seçilen ve özet fonksiyonu içeren kriptografik sistemler incelenmiştir.

Tezin ilk bölümünde simetrik kriptografi ve özet fonksiyonları hakkında bilgi verilmiştir. Simetrik kriptografide kullanılan matematiksel yapılar tanımlanmıştır. Sonraki bölümlerde, NIST hafif-siklet kriptografi projesinde seçilen ve özet fonksiyonu içeren sistemlerden yararlanarak, özet fonksiyonlarına yapılan genel ataklar araştırılarak, güvenlik tanımları ve kriptanaliz yöntemleri verilmiştir.

Anahtar Sözcükler : Kuantum Sonrası Kriptografi, Simetrik Kriptografi, Özet Fonksiyonları, Kriptanaliz.

ABSTRACT

SECURITY ANALYSIS OF SYMMETRİC CRYPTO-SYSTEMS

KARATAY, Melike

MSc in Mathematics

Supervisor: Prof. Dr. Urfat NURİYEV

Co-Supervisor: Assoc. Prof. Dr. Erdem ALKIM

December 2019, 80 pages.

Hash functions are cryptographic functions that process an arbitrary length input to a unique fixed length output. It is very difficult to determine the security limits of hash functions. In this thesis, the methods used to determine the security of hash functions, which are an element of symmetric cryptography, are explained. Cryptographic systems, which were selected by American Institute of Standards and Technology (NIST) in 2018 within the scope of lightweight cryptography project, were analyzed.

In the first part of the thesis, information about symmetric cryptography and hash functions is given. Mathematical structures used in symmetric cryptography are defined. In the following sections, security definitions and cryptanalysis methods are given by investigating the general attacks on the hash functions by utilizing the systems containing the hash function selected in the NIST lightweight cryptography project.

Key Words: Post Quantum Cryptography, Symmetric Cryptography, Hash Functions, Cryptanalysis.

ÖNSÖZ

Kriptoloji, kısaca şifre bilimidir. Çok eski çağlardan günümüze kadar bilginin saklanması, korunması ve iletilmesinde güvenliği sağlamak amacıyla kriptoloji kullanılmaktadır. Dijitalleşme çağı ile beraber kriptoloji çok daha büyük bir öneme sahip olmuştur. Bu sebeple, akademik çalışmalar yapmaya başladığım günden bugüne kriptoloji ilgimi çekmiştir. Ayrıca, ülkemizde kriptoloji alanında çalışan kişi sayısının yetersizliği ilgimi arttırmıştır.

Yüksek lisans tez konumu belirlerken, literatüre katkı sağlamak ilk hedefim olmuştur. Kuantum sonrası kriptografi alanına da değinmek amacıyla özet fonksiyonları ve güvenliği üzerinde çalışmaya başladım. Değerli hocalarımın desteği ile tez konumun belirlenmesi ve yazım sürecinde yaşadığım tüm zorlukların üstesinden geldim. Bu sebeple danışman hocalarıma sonsuz teşekkürü borç bilirim. Ayrıca tez çalışmam boyunca maddi destek veren TÜBİTAK-BİDEB'e teşekkür ederim.

İZMİR

18/12/2019

Melike Karatay

İÇİNDEKİLER

sayfa

İÇİNDEKİLER (Devam)

sayfa

ŞEKİLLER DİZİNİ

Şekil Sayfa

TABLOLAR DİZİNİ

Tablo Sayfa

1.GİRİŞ

Bilgi güvenliği kavramı gizlilik, bütünlük ve erişilebilirlik öğelerinden oluşur. Çok eski zamanlardan günümüze kadar bilgi güvenliği önemli bir konu olmuştur. Eski zamanlarda insanlar özellikle iletişimlerini güvenli hale getirmek için birçok şifreleme yöntemi kullanmışlardır. Bu şifreleme yöntemleri zamanla çözülerek güvenilirliğini yitirmiştir. Günümüz teknoloji çağında, eski şifreleme yöntemleri yerini dijital yöntemlere bırakmıştır. Ayrıca artık tüm kurum ve kuruluşlar iletişim ağları üzerinde iletişim gerçekleştirdiği için bilginin korunması zorlaşarak, bilgi güvenliği daha önemli bir hal almıştır.

Bilgi güvenliği için yapılan çalışmalar kriptoloji biliminin ortaya çıkmasını sağlamıştır. Kriptoloji, bilginin belirli bir yönteme göre şifrelenmesi ve şifrenin çözülmesi ile ilgilenen bilim dalıdır. Kriptoloji iki ana daldan oluşur; ilk dalı olan kriptografi bilginin şifrelenmesi, ikinci dalı kriptanaliz ise şifrenin analizi ve çözülmesi ile ilgilenir. Kriptografide, bilginin şifrelenmesi için şifreleme anahtarı kullanılmaktadır ve güvenlik bu şifreleme anahtarı ile sağlanmaktadır. Kriptanalizde ise kullanılan anahtar elde edilmeye çalışılır. Şifreleme anahtarının türüne göre kriptografik sistemler, simetrik (gizli anahtarlı) sistemler ve asimetrik (açık anahtarlı) sistemler olmak üzere ikiye ayrılır (Stallings, 2011).

Simetrik sistemler, tek bir gizli anahtar ile şifreleme ve şifre çözme işlemlerini gerçekleştiren kriptografik algoritmalardır. Simetrik kriptografinin bir alt dalı olan kriptografik özet fonksiyonları, anahtarsız simetrik kriptografik algoritmaları kullanarak girdi olarak aldığı açık metinlerin sabit uzunluklu özet değerlerini üreten fonksiyonlardır. Kriptolojide, özet fonksiyonlarının çok geniş bir kullanım alanı vardır. Bu sebeple, kriptografik özet fonksiyonlarının son derece güvenli olması gerekirken böyle bir özet fonksiyonu bulmak oldukça zordur. Kriptografik özet fonksiyonlarının güvenliği kullanıldığı alana göre yeterli görülse bile güvenlikleri aşırı derecede güçlü değildir. NIST tarafından düzenlenen Güvenli Özet Algoritması (Secure Hash Algorithm-3 - SHA-3) yarışmasında, başarılı olan özet fonksiyonlarında dikkat edilen noktalar güvenlik tanımları ve kanıtlarından çok kriptanaliz kısmıdır (Koblitz and Menezes, 2013).

Kuantum bilgisayarların geliştirilmesi ile birlikte günümüzde kullanılan simetrik şifreleme sistemlerinin daha büyük anahtar boyutlarına ihtiyacı olacaktır. Ayrıca Shor'un algoritması göstermiştir ki kuantum bilgisayarlar sonrasında asimetrik şifreleme sistemleri tamamen güvensiz hale gelecektir (Shor, 1994; Shor, 1997). Özet fonksiyonları, hem kullanım alanlarının çokluğu hem de kuantum sonrası güvenlik için gerekli kriptografik yapılar olduğundan üzerine birçok çalışma yapılmaktadır. Özet fonksiyonlarının sağladığı kuantum sonrası güvenliğin yanında, kuantum sonrası güvenlik için önerilen kafes tabanlı ve diğer sistemlerin rastgele sayılardan oluşan çok büyük polinomlara/matrislere ihtiyacı vardır. Bu gibi sistemlerin rastgele sayı ihtiyacı genişletilmiş çıktı fonksiyonları ile

karşılanabilmektedir (Alkim et al., 2016; Güneysu et al., 2012). Rastgele sayı üretimi ise bu tarz sistemlerde en çok zaman harcanan kısımdır (Güneysu et al., 2012; Alkim et al., 2017). Bu sebeple özet fonksiyonları hem simetrik hem de asimetrik sistemlerde özellikle kuantum sonrası güvenlik için önemli bir rol oynamaktadır.

Bu tezde, özet fonksiyonları, güvenlik tanımları ve kriptanaliz üzerinde durulmuştur. 2. bölümde simetrik kriptografi ve özet fonksiyonları tanımlanarak, sistemlerde sıkça kullanılan matematiksel yapıların tanımları verilmiştir. Standart özet fonksiyonları ve SHA-3 özet fonksiyonunun kriptanalizinden bazı noktalar anlatılmıştır. 3. bölümde NIST tarafından 2019 yılında düzenlenen hafif-siklet kriptografi projesi kapsamında NIST'e gönderilen ve özet fonksiyonu içeren şifreleme sistemlerinin kullandığı yapılar, işlemler ve kriptanalizleri incelenmiştir. 4. bölümde kriptografik sistemler için temel güvenlik tanımları verilmiştir. Özet fonksiyonlarına yapılan genel ataklar anlatılmış ve kriptanaliz yöntemlerinden bahsedilmiştir.

2. SİMETRİK KRİPTOGRAFİ

Eski zamanlarda kullanılan birçok şifreleme sistemi vardır. Bu şifreleme sistemleri genellikle metin üzerindeki harflerin yerine başka semboller kullanılması ve bir tablo veya şekil yardımıyla metnin şifrelenmesi gibi yöntemler kullanılarak şifreleme işlemini gerçekleştirmektedir. Gelişen bilim ve teknoloji ile bu yöntemler güvenirliğini yitirmiştir ve ihtiyacı karşılayamaz hale gelmiştir. Çünkü günümüzde, tek amaç bilginin gizlenmesi değildir. Gizliliğin yanında bilgi bütünlüğünün sağlanması, karşı tarafa dijital kanallardan güvenilir şekilde iletimi, kimlik doğrulamaları, dijital imzalar, blok zincir gibi birçok ihtiyaç doğmuştur.

Aslında, eski zamanlarda kullanılan şifreleme yöntemlerinin temel mantığı simetrik şifrelemeye benzemektedir. Haberleşen iki taraf şifreleme yöntemini bilir ve aynı anahtara sahiptir. Şifreleme işlemi tek anahtar ile gerçekleştirilir. Şifreli metni alan karşı taraf aynı anahtarı kullanarak ve yöntemi tersten işleyerek açık metne ulaşır. Bu senaryoda, birbirine güvenen iki taraf olsa da, anahtarın istenmeyen bir kişinin eline geçmesi olasıdır. Şifreleme yönteminde kullanılan algoritma gizli kalmamalıdır. Bu prensibe, Kerckhoffs prensibi denir. Bu prensibe göre, şifreleme yönteminde kullanılan algoritmanın gizli tutulması o sistemin güvenliğini belirlememektedir ve sistemin güvenliği kullanılan anahtara bağlı olmalıdır (Katz and Lindell, 2014).

Klasik kriptografideki terminoloji, modern kriptografide simetrik şifreleme şeması olarak adlandırılır ve ilkel yöntemler yerine matematiksel yöntemler kullanarak şifreleme/şifre çözme işlemini gerçekleştirir. Simetrik şifreleme şemaları üç bölümden oluşur. Bu bölümler; anahtar üretimi, şifreleme ve şifre çözmedir. Anahtar üretimi algoritmaları, belirli matematiksel fonksiyonlar kullanılarak, rastgele görünümlü anahtar üreten algoritmalardır. Şifreleme algoritmaları, açık metni ve anahtar üretimi algoritmasında üretilen anahtarı girdi olarak alarak açık metni şifreli metne dönüştürür. Şifre çözme algoritmaları ise şifreli metni ve şifrelemede kullanılan anahtarı girdi olarak alarak şifreli metni tekrar açık metne dönüştüren algoritmalardır. Alınan açık metni m, anahtarı k, şifreleme algoritmasını E_k ve şifre çözme algoritmasını D_k ile ifade edersek, bir simetrik şifreleme sistemini şu şekilde gösterebiliriz: $D_k(E_k(m))=m$. Başka bir deyişle, c şifreli metin olmak üzere: $E_k(m)=c$ ve $D_k(c)=m$ olur. Özet olarak simetrik şifreleme şemalarında, aynı anahtar ile şifreleme ve şifre çözme işlemleri gerçekleştirilir.

Simetrik şifrelemeye bir örnek olarak ve simetrik şifrelemenin temellerinden bir model tek kullanımlık şerittir (one-time pad). Tek kullanımlık şeritte, mesaj ile aynı uzunluğa sahip rastgele bir anahtar üretilir. Bu anahtar ile girdi mesajı $\oplus$ işlemine tabi tutulur. Mesaj ile aynı uzunlukta bir şifreli metin oluşur. Aynı anahtar ile şifreli metin $\oplus$ işlemine girdiğinde girdi mesajına ulaşmış oluruz. Bu sistemde gizli anahtar yalnızca bir kez kullanılır. Tek kullanımlık şeritte

şifreleme işlemi $E_k(m) = m \oplus k = c$, şifre çözme işlemi $D_k(c) = c \oplus k = m$ şeklinde gösterilir.

Simetrik şifreleme şemaları üç temel yapıdan oluşur. Bu yapılar, akan şifre şemaları, blok şifre şemaları ve özet fonksiyonlarıdır. Akan şifre şemalarında, girdi mesajı olan açık metnin bitleri tek tek şifreleme işlemine tabi tutularak şifreli metin üretilir. Akan şifre şemasına örnek olarak doğrusal geri beslemeli kaydırma kaydı (Linear Feedback Shift Register - LFSR) verilebilir. Blok şifre şemalarında girdi mesajı, bloklara bölünerek, girdi mesajı blokları üzerinde şifreleme işlemi gerçekleştirilir. Blok şifre örneği olarak Gelişmiş Şifreleme Standardı(Advances Encryption Standard - AES) ve Veri Şifreleme Standardı (Data Encryption Standard – DES) sistemleri gösterilebilir (FIPS, 1999; Daemen and Rijmen, 2001). Özet fonksiyonları ise girdi mesajını simetrik şifreleme yöntemi ile işleyerek sabit uzunlukta girdi mesajına özgü bir özet değeri oluşturur. Özet fonksiyonları anahtarsız simetrik şifreleme algoritmalarıdır ve tek yönlü fonksiyonlardır. Özet fonksiyonlarına örnek olarak SHA fonksiyonları verilebilir. Bu şemalar simetrik şifrelemenin pratik modelleridir. Bu şemalar dışında, sözde rastgele sayı üreteçleri gibi teorik modeller de mevcuttur.

2.1 Matematiksel İfadeler ve Tanımlar

Bu bölümde, tezde anlatılmış olan şifreleme sistemlerindeki ve bu şifreleme sistemlerinin kriptanalizinde kullanılan matematiksel yapıların tanımları verilmiştir. Takip eden tanımlar (Cox et al., 1996) kaynağından yararlanılarak yapılmıştır.

Tanım 2.1.1: En az iki elemanlı F kümesi $\oplus$ (XOR) ve & (AND) işlemleri altında aşağıdaki aksiyomları sağlıyor ise cisimdir.

Aksiyom 1 : 0 birim eleman olmak üzere F kümesi $\oplus$ işlemi altında değişmeli gruptur.

Aksiyom 2 : 1 birim eleman olmak üzere F* = F/{0} kümesi & işlemi altında değişmeli gruptur.

Aksiyom 3 : a,b,c∈F olmak üzere $(a \oplus b) \& c = (a \& c) \oplus (b \& c)$ eşitliği sağlanır.

Tanım 2.1.2 : Cisim üzerindeki eleman sayısı cismin mertebesi olarak adlandırılır. Mertebesi sonlu olan cisme sonlu cisim denir.

Tanım 2.1.3 : Elemanları 0 ve 1 sayılarından oluşan sonlu cisim (veya Galois Cismi) F_2 veya GF(2) ile gösterilir. F_2^n (ya da $GF(2)^n$) ise F_2 (GF(2)) cisminin n kez kartezyen çarpımı olarak ifade edilir, yani $F_2^n = F_{2(1)} \times F_{2(2)} \times \ldots \times F_{2(n)}$.

Tanım 2.1.4 : $f: F_2^n \rightarrow F2$ bir fonksiyon olsun. Bu fonksiyonun cebirsel normal formu $\sum_{S \subseteq Zn} t_s x^s$, $t_s \in F_2$ ile gösterilir.

Tanım 2.1.5 : $f: F_2^n \rightarrow F_2$ bir fonksiyon ve t_s cebirsel normal formun katsayısı olmak üzere, $max\{|S|; S \subseteq Z_n \text{ ve } t_S \neq 0\}$, f'fonksiyonunun derecesi olarak tanımlanır ve $deg(f)$ ile gösterilir.

Tanım 2.1.6 : $F: F_2^n \rightarrow F_2^m$, $i \in \{0,...,m-1\}$ olmak üzere f_i bileşenlerinden oluşan bir fonksiyon olsun. $max\{deg f_i \mid i \in \{0,...,m-1\}\}$ değerine F fonksiyonunun derecesi denir. Eğer F sabit fonksiyon ise derecesi 0, doğrusal fonksiyon ise derecesi 1, kuadratik fonksiyon ise derecesi 2 değerindedir.

Tanım 2.1.7 : V, F cismi üzerindeki bir vektör uzayıdır ve elemanları vektör olarak adlandırılır. Öyle ki aşağıdaki şartları sağlar.

$\forall u,v \in V, \forall t,r \in F$

• V toplama işlemi altında değişmeli gruptur.

• $t\cdot(u + v) = t\cdot u + t\cdot v$.

• $(t + r)\cdot v = t\cdot v + r\cdot v$.

• $(t.r)\cdot v = t\cdot(r\cdot v)$.

• $1\cdot v = v$.

Tanım 2.1.8 : $v_1,...,v_k \in F_2^n$ vektörleri lineer bağımsızdır. Öyle ki $i \in \{0,...,k\}$ için bütün t_i skalerleri F_2 cismindedir ve $t_1v_1 + ...t_kv_k = 0$ eşitliği yalnızca $t1 = \cdots = tk = 0$ durumunda sağlanır.

Tanım 2.1.9 : $W \leq V$ şeklinde ifade edilen W yapısı alt uzaydır. W alt vektör uzayı, V vektör uzayının sağladığı tüm özellikleri sağlar.

Tanım 2.1.10 : V ve W, F_2^n vektör uzayının alt uzayları ve $a \in F_2^n$ olsun. $V+a$ ifadesine afin uzay denir ve $V+a = \{v + a \mid v \in V\}$ şeklinde tanımlanır.

Tanım 2.1.11 : V sonlu vektör uzayının boyutu $dim(V)$ ile gösterilir. Bu boyut baz vektörlerinin sayısı ile belirlenir. $V+a$ afin uzayının boyutu V vektör uzayının boyutu ile aynıdır.

Tanım 2.1.12 : Monomial, $a_1,...,a_n$ sıfır olmayan tamsayı kuvvetler olmak üzere $x_1^{a_1} \ldots x_n^{a_n}$ çarpımı şeklinde ifade edilir. Bir monomialin derecesi tüm kuvvetlerin toplamı şeklinde hesaplanır.

Tanım 2.1.13 : $x = x_1,...,x_n$ üzerindeki bir polinom, katsayıları bir sonlu cismin elemanı olan, monomiallerin sonlu bir doğrusal kombinasyonudur. Polinomlar $\sum_a b_a x^a$ ile ifade edilir.

Tanım 2.1.14 : *n* değişkenli bir polinomun katsayıları *K[x]* ile tanımlanan bir değişmeli halkanın elemanı ise bu yapıya *n* değişkenli polinom halkası denir.

Tanım 2.1.15 : *F* bir cisim ve $F[x_1,\ldots,x_n]$ bir polinom halkası olsun. $I \subseteq F[x_1,\ldots,x_n]$ alt kümesi idealdir, ancak ve ancak,

1. $0 \in I$

2. Eğer $f,g \in I$ ise $f + g \in I$

3. Eğer $f \in I$ ve $h \in F[x_1,\ldots,x_n]$ ise $hf \in I$.

Tanım 2.1.16 : *F* bir cisim ve $f_1,\ldots,f_s$ polinomları $F[x_1,\ldots,x_n]$ polinom halkasındaki elemanlar olsun. $V(f_1,\ldots,f_s) = \{(a_1,\ldots,a_n) \in F^n \mid \forall i \in \{1,\ldots,s\} \text{ için } f_i(a_1,\ldots,a_n) = 0\}$ ile tanımlanan $V(f_1,\ldots,f_s)$ vektör uzayına $f_1,\ldots,f_s$ ile tanımlanmış afin çeşitlilik denir.

Tanım 2.1.17 : $V(f_1,\ldots,f_s)$ vektör uzayı, $F_2[x_1,\ldots,x_n]$ cisminde bir afin çeşitlilik olsun. Eğer $f_1,\ldots,f_s$ fonksiyonları sabit veya doğrusal ise $V(f_1,\ldots,f_s)$ vektör uzayına doğrusal afin çeşitlilik denir.

2.2 Kriptografik Özet Fonksiyonları

Kriptografik özet fonksiyonları, simetrik kriptografinin temellerini kullanan matematiksel yapılardır. Mesaj bütünlüğünün doğrulanması, çok büyük dosyalarda karşılaştırma, sözde rastgele sayı üreteci, anahtar üretimi gibi kriptografinin birçok alanında uygulaması mevcuttur. Kriptografik özet fonksiyonları, seçilen uzunlukta bir bit dizisini girdi olarak kabul ederek, sabit uzunluğa sahip çıktı üretir. Üretilen sabit uzunluklu çıktıya özet adı verilir. Matematiksel olarak şu şekilde tanımlanır: H bir özet fonksiyonu ve h sabit bir uzunluk olmak üzere, $H: \{0,1\}^* \rightarrow \{0,1\}^h$. Özet fonksiyonlarında özet değerini hesaplamak hızlı ve kolay olmalıdır.

Kriptografik özet fonksiyonlarının sağlaması gereken üç temel güvenlik özelliği vardır. Bu özellikler; çakışma direnci, ön-görüntü direnci ve ikinci ön-görüntü direncidir. Çakışma, girdi olarak alınan iki farklı mesajın özet değerinin aynı olmasıdır. Ön-görüntü, bir mesaja ait özet değerinden, girdi mesajının tahmin edilmesidir, çünkü özet fonksiyonları tek yönlü fonksiyonlardır. İkinci ön-görüntü ise girdi olan bir mesajı ve bu mesaja ait özet değeri bilinirken, ilk mesaj ile aynı özet değerine sahip, farklı bir mesaj bulunmasıdır. Kriptografik bir özet fonksiyonu, bu açıklıklardan yararlanılarak yapılan ataklara karşı direnç sağlamalıdır. Formal olarak aşağıdaki gibi tanımlanır.

Tanım 2.2.1 : m_1, m_2 iki mesaj ve $H: \{0,1\}^* \rightarrow \{0,1\}^h$ özet fonksiyonu olsun. H özet fonksiyonunda çakışma vardır, öyle ki;

$m_1 \neq m_2$ ve $H(m_1) = H(m_2)$.

Tanım 2.2.2 : m_1 bir mesaj ve H: $\{0,1\}^* \rightarrow \{0,1\}^h$ özet fonksiyonu olsun. Eğer $H(m_1)$ özet değerinden m_1 mesajı bulunabiliyorsa ön-görüntü vardır denir.

Tanım 2.2.3 : m_1, m_2 iki mesaj ve H: $\{0,1\}^* \rightarrow \{0,1\}^h$ özet fonksiyonu olsun. Eğer m_1 mesajı ve $H(m_1)$ özet değeri biliniyorken $H(m_1) = H(m_2)$ olacak şekilde m_1 mesajından farklı bir m_2 mesajı bulunabiliyorsa ikinci ön-görüntü vardır denir.

Kriptografik özet fonksiyonları bu temel özelliklerinin yanında uzunluk genişletme atağı ve özet değerindeki alt küme özelliklerden yararlanılan ataklara karşı da direnç sağlamalıdır.

Özet fonksiyonlarının en çok kullanıldığı alanlardan birisi kriptografik rastgele sayı üreteçleridir. Sözde rastgele sayı üreteci, tohum denilen düzenli bit bit dizisini girdi olarak alır ve etkili matematiksel bir algoritma kullanarak, rastgele görünümlü bit dizisi üretir. Tüm kriptografik sistemler için en temel güvenlik gereksinimidir. Sözde rastgele sayı üreteçlerinin bazı durumlarda çok fazla sayıda rastgele görünümlü bit üretmesi gerekebilir. Bu sebeple çok az miktarda gerçek rastgelelik kullanırlar. Az miktarda kullanmalarının sebebi ise gerçek rastgele sayı üreteçleri çok yavaştır ve bu üreteçler ile sayı üretmek zordur (Katz and Lindell, 2014). Sözde rastgele sayı üreteçlerinin, kriptografik rastgele sayı üreteci olması için bazı gereksinimlere ihtiyacı vardır. Sözde rastgele sayı üretecinde üretilen sayılar istatistiksel olarak güçlü olmalıdır. Genel olarak, üretilen rastgele sayıdaki aktif ve pasif bit oranının %50 olması istenmektedir. Ayrıca üreteçte, o anda üretilmiş rastgele sayıdan yola çıkarak, o sayıdan önce üretilmiş veya sonra üretilecek sayı hakkında tahmin yürütülememelidir (Özkaynak, 2015). Özet fonksiyonları kriptografik rastgele sayı üreteçlerinin özelliklerini sağlamalıdır.

Özet fonksiyonlarının işlem modları olarak Merkle ağacı ve Sponge yapısı sayılabilir. Temel olarak, Merkle ağacı, diğer adı ile özet ağacı, yinelemeli özet fonksiyonları için ilham kaynağı olmuştur. Merkle ağacı, yaprakları veri bloklarının özet değerini, düğümleri ise komşu alt düğümün özet değerini tutan ikili ağaçtır. Büyük verilerde kolayca doğrulama işlemi yapılabilmesini sağlar. Bir blok özetinin, Merkle ağacının bir yaprağı olduğunu ispatlamak log (yaprak) kadar işlem gerektirir. Merkle ağacında, ilk seviyedeki düğüm kök düğümdür. Merkle ağacı zincirleme bir modeldir ve bu model çakışmaya dirençli ise Merkle ağacında kullanılan özet fonksiyonu da çakışmaya dirençlidir (Merkle, 1990; Küçük, 2012). Sponge yapısı ise yinelemeli olarak şifreleme işlemlerini gerçekleştiren matematiksel bir yapıdır. İşlemler tek bir permütasyon temelinde, permütasyon fonksiyonu kullanılarak gerçekleştirir. Permütasyon fonksiyonu sayesinde, özet fonksiyonu seçilen uzunlukta çıktı üretilebilir (Bertoni et al., 2009).

Özet fonksiyonlarının en çok kullanıldığı alanlardan birisi dijital imzalardır. Dijital imza şemalarında daha iyi performans sağlamak ve kuantum sonrası güvenlik oluşturmak amacıyla özet tabanlı imza şemaları geliştirilmiştir. Diğer bir alan parolaların korunmasıdır. Örneğin bir web uygulamasında kullanıcı adlarının

ve parolaların özet değerleri veri tabanında tutulmalıdır. Böylece saldırgan veri tabanına erişebilse dahi kullanıcı şifrelerine ulaşamamaktadır. Ayrıca özet fonksiyonları bilginin doğrulanması için de kullanılmaktadır. Tek bir bit değişiminde bile özet değerindeki büyük değişimlerden dolayı bilginin değiştirildiği kontrol edilebilir. En çok bilinen başka bir kullanım alanı Mesaj Doğrulama Kodu (Message Authentication Code - MAC) algoritmalarıdır. MAC algoritmaları aslında anahtarlı özet fonksiyonlarıdır.

2.2.1 SHA-1 ve SHA-2

SHA-1 özet fonksiyonu, uzunluğu 2^{64} bitten daha az olacak şekilde bit dizisini girdi olarak alır ve 160-bit uzunluğunda özet değeri üretir. SHA-1 fonksiyonu, özet değeri oluşturmak için 80 tur çalışır. $0 \leq i \leq 79$ olmak üzere, fonksiyonlar s_i ile gösterilir. Fonksiyonların tümü mantıksal işlemlerden oluşur ve her turda aynı işlemler uygulanmaz. İşlemler 32-bit kelimeler olan *w,w1,w2* üzerinde gerçekleştirilir ve çıktı olarak 32-bit üretilir.

$s_i(w,w1,w2)$ fonksiyonu;

$0 \leq i \leq 19$ iken, $Ch(w,w1,w2) = (w\&w1) \oplus (w\&w2)$

$20 \leq i \leq 39$ iken, $Parity(w,w1,w2) = w \oplus w1 \oplus w2$

$40 \leq i \leq 59$ iken, $Maj(w,w1,w2) = (w\&w1) \oplus (w\&w2) \oplus (w1\&w2)$

$60 \leq i \leq 79$ iken, $Parity(w,w1,w2) = w \oplus w1 \oplus w2$

şeklindedir. SHA-1 fonksiyonunda kullanılan sabitler ise yukarıdaki her i aralığı için dört farklı değer alır ve K_i ile gösterilir.

Her özet fonksiyonunda olduğu gibi SHA-1 özet fonksiyonunda da tüm işlemlerden önce mesaja dolgulama işlemi yapılır. SHA-1 fonksiyonundaki dolgulama kuralı, mesajın sonuna tek bir 1 biti eklenir ve girdi mesajı boyutu k olmak üzere $k+1+p \equiv 448 \ mod \ 512$ denkliğini sağlayacak şekilde p tane 0 biti eklenir. Dolgulama işlemi tamamlandıktan sonra, dolgulanmış mesaj n tane 512-bit bloğa bölünür. 512-bit girdi bloğu, 16 tane 32-bit kelime anlamına gelir. Toplamda 80 tane 32-bit uzunluğunda kelime vardır ve bu 80 kelime W_i şeklinde gösterilir. Bunun haricinde, 32-bit uzunluğunda beş tane çalışma değişkeni vardır. Çalışma değişkenleri ise a,b,c,d,e şeklinde gösterilir.

SHA-1 fonksiyonu özet değeri üretirken ilk olarak mesaj üzerinde dolgulama işlemini tamamlar ve mesajı belirtilen şekilde bloklara ayırır. Başlangıç olarak bir özet değeri belirlenir. Daha sonra her mesaj bloğu için W_i değerleri aşağıdaki gibi ayarlanır;

Eğer $0 \leq i \leq 15$ ise mesajın o adımdaki bloğu W_i değeridir. ROT ise belirtilen yöne doğru çevrimsel kaydırma işlemidir. Burada L sola doğru ROT işlemini belirtmektedir.

Eğer $16 \leq i \leq 79$ ise $ROTL^1(W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16})$ değeri W_i değeridir ve $0 \leq n \leq 32$ olmak üzere, $ROTL^n(x) = (x<<n)\ OR\ (x>>w-n)$ dir.

a, b, c, d, e değerlerine her mesaj bloğu işlenirken, bir önceki adımın özet değeri 32-bit parçalara ayrılarak sırasıyla atanır. Sonrasında, 80 adım boyunca aşağıdaki işlemler tekrarlanır.

$T=ROTL^5(a) + s_i(b,c,d) + e + K_i + W_i$ olmak üzere;

$e = d,\quad d = c,\quad c=ROTL^{30}(b),\quad b=a,\quad a=T.$

Daha sonra hesaplanan a, b, c, d, e değerleri bir önceki adımın çıktısı olan özet değerinin 32-bit blokları ile $\oplus$ işlemine tabi tutulur. Bu şekilde tüm mesaj blokları işlendikten sonra son oluşan 32-bit uzunluklu parçalar birbiri ardına sırasıyla eklenir ve özet değerini oluşturur.

Özet fonksiyonlarının en önemli özelliği çakışma direnci sağlamasıdır. SHA-1 özet fonksiyonu güvenli bir özet fonksiyonu değildir. Sebebi ise çakışma direncinin düşük olması ve bu zamana kadar yapılmış kriptografik ataklar sonucunda özet değerleri arasında çakışma bulunmasıdır. Yani aynı özet değerini üreten farklı iki mesaj bulunmuştur. 2011 yılında güvensiz olduğu resmi olarak duyurulmuştur.

SHA-1 özet fonksiyonu Merkle-Damgard ailesindendir. Aynı SHA-1 fonksiyonu gibi SHA-2 özet fonksiyonu da Merkle-Damgard ailesindendir. Yani küçük farklılıklar dışında çok benzerlerdir. SHA-2 özet fonksiyonunun 224-bit 256-bit, 384-bit ve 512-bit çıktı veren altı tane çeşidi vardır. Genelde, h özet çıktısı uzunluğu olmak üzere SHA-h şeklinde gösterilir. SHA-256 fonksiyonu 256 bit özet çıktısı verir. Mesaj blokları 512-bit uzunluğundadır ve 16 tane 32-bit kelimeden oluşur. SHA-384 ve SHA-512 fonksiyonlarında ise mesaj blokları 1024-bit uzunluğundadır ve 16 tane 64-bit kelimeden oluşur. SHA-256 fonksiyonu altı tane mantıksal fonksiyonu vardır. Her fonksiyon 32-bit kelime değerlerini işler ve yeni 32-bit kelime değeri üretir. *w, w1, w2* değerleri 32-bit uzunluğunda kelimeler olmak üzere;

$Ch(w,w1,w2) = (w\&w1) \oplus (w\&w2)$

$Maj(w,w1,w2) = (w\&w1) \oplus (w\&w2) \oplus (w1\&w2)$

$0 \leq n \leq w$ iken, $ROTR^n(x) = (x>>n)\ OR\ (x<<w-n)$ ve $SHR^n(x)=x>>n$ şeklinde hesaplanmak üzere;

$\sum_0^{\{256\}}(w) = ROTR^2(w) \oplus ROTR^{13}(w) \oplus ROTR^{22}(w)$

$\sum_1^{\{256\}}(w) = ROTR^6(w) \oplus ROTR^{11}(w) \oplus ROTR^{25}(w)$

$S_0^{\{256\}}(w) = ROTR^7(w) \oplus ROTR^{18}(w) \oplus SHR^3(w)$

$S_1^{\{256\}}(w) = ROTR^{17}(w) \oplus ROTR^{19}(w) \oplus SHR^{10}(w)$

şeklindedir.

SHA-384 ve SHA-512 fonksiyonlarında kullanılan işlemlerin SHA-256 fonksiyonundan farklı kısmı,

$\sum_0^{\{512\}}(w) = ROTR^{28}(w) \oplus ROTR^{34}(w) \oplus ROTR^{39}(w)$

$\sum_1^{\{512\}}(w) = ROTR^{14}(w) \oplus ROTR^{18}(w) \oplus ROTR^{41}(w)$

$S_0^{\{512\}}(w) = ROTR^1(w) \oplus ROTR^8(w) \oplus SHR^7(w)$

$S_1^{\{512\}}(w) = ROTR^{19}(w) \oplus ROTR^{61}(w) \oplus SHR^6(w)$

şeklindedir.

SHA-256 fonksiyonunun mesaj dolgulama kuralı SHA-1 fonksiyonunun mesaj dolgulama kuralı ile birebir aynıdır. SHA-384 ve SHA-512 fonksiyonlarının mesaj dolgulama kuralı ise şu şekildedir; mesajın sonuna tek bir 1 biti eklenir ve girdi mesajı boyutu k olmak üzere $k+1+p \equiv 896 \ mod\ 1024$ denkliğini sağlayacak şekilde p tane 0 biti eklenir. SHA-256 fonksiyonunda da SHA-1 fonksiyonunda olduğu gibi başlangıç özet değeri, sabitler, ve 64 tane W_i değeri vardır. Ayrıca sekiz tane a, b, c, d, e, f, g, h şeklinde gösterilen çalışma değişkenine sahiptir. Mesaj dolgulaması yapılıp, mesaj n tane 512-bit uzunluğunda bloklara ayrıldıktan sonra W_i değerleri her mesaj bloğu için belirli bir kuralla ayarlanır.

Eğer $0 \le i \le 15$ ise mesajın o adımdaki bloğu W_i değeridir.

Eğer $16 \le i \le 63$ ise $S_1^{\{256\}}(W_{i-1}) \oplus W_{i-7} \oplus S_0^{\{256\}}(W_{i-15}) \oplus W_{i-16}$ işleminin sonucu W_i değeridir.

a, b, c, d, e, f, g, h değerlerine her mesaj bloğu işlenirken, bir önceki adımın özet değeri 32-bit parçalara ayrılarak sırasıyla atanır. Sonrasında, 64 adım boyunca aşağıdaki işlemler tekrarlanır:

$T_1 = h + \sum_1^{\{256\}}(e) + Ch(e,f,g) + K_i + W_i$

$T_2 = \sum_0^{\{256\}}(a) + Maj(a,b,c)$

$h = g,\ \ g = f,\ \ f = e,\ \ e = d + T_1,\ \ d = c,\ \ c = b,\ \ b = a,\ \ a = T_1 + T_2.$

Tüm işlemler bittikten sonra elde edilen 32-bit uzunluklu özet değeri parçaları birbiri ardına sırasıyla eklenir ve özet değerini oluşturur. SHA-224 fonksiyonu ise SHA-256 fonksiyonu özet çıktısının 224-bit kesilmiş çıktı vermesi ile üretilir.

SHA-512 fonksiyonu özet değeri üretirken SHA-256 fonksiyonundan küçük farklılıklar içerir. Fark olarak, seksen tane 64-bit uzunluğunda W_i değeri içerir. Ayrıca, sekiz tane çalışma değişkeni de 64-bit uzunluğundadır. Her mesaj bloğu için W_i değeri şu şekilde hesaplanır;

Eğer $0 \leq i \leq 15$ ise mesajın o adımdaki bloğu W_i değeridir.

Eğer $16 \leq i \leq 79$ ise $S_1^{\{512\}}(W_{i-2}) \oplus W_{i-7} \oplus S_0^{\{512\}}(W_{i-15}) \oplus W_{i-16}$ işleminin sonucu W_i değeridir.

80 adım boyunca aşağıdaki işlemler tekrarlanır:

$T_1 = h + \sum_1^{\{512\}}(e) + Ch(e,f,g) + K_i + W_i$

$T_2 = \sum_0^{\{512\}}(a) + Maj(a,b,c)$

$h = g,\ \ g = f,\ \ f = e,\ \ e = d + T_1,\ \ d = c,\ \ c = b,\ \ b = a,\ \ a = T_1 + T_2.$

Tüm işlemler bittikten sonra elde edilen 64-bit uzunluklu özet değeri parçaları birbiri ardına sırasıyla eklenir ve özet değerini oluşturur. SHA-384 fonksiyonu ise SHA-512 ile aynı işlemlerin sonucunda oluşan özet değerinin ilk 384-bitinin kesilmesi ile elde edilir. SHA-2 fonksiyonları günümüzde güvenli kabul edilmektedir (Standard, 2002).

SHA-224 ve SHA-256 fonksiyonlarının çakışma direnci, h özet çıktısı uzunluğu olmak üzere $2^{h/2} = 2^{112}$ şeklinde hesaplanmaktadır. SHA-384 fonksiyonunda çakışma direnci 2^{192}, SHA-512 fonksiyonunda ise 2^{256} şeklindedir. SHA-512/224 ve SHA-512/256 fonksiyonlarında küçük olan değere göre çakışma direnci belirlenir. Fonksiyonların tümünde ön-görüntü direnci ise özet çıktısının uzunluğu kadardır.

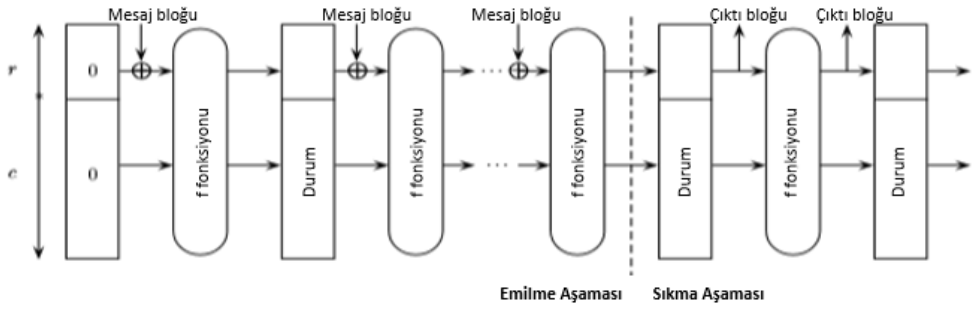
2.2.2 SHA-3

SHA-3, diğer adıyla Keccak, 2015 yılında, NIST tarafından, özet fonksiyonları için yeni standart olarak yayınlanmıştır. Yalnızca özet fonksiyonu değildir, simetrik kriptografinin birçok alanında kullanılabilmektedir. Keccak, Sponge yapısının bir fonksiyonudur. Doğal olarak, Sponge yapısının tüm özelliklerini sağlar ve güvenliği de Sponge yapısına dayanmaktadır. Sponge yapısı kriptografik rastgele permütasyondur.

Sponge, seçilen uzunlukta girdiyi yinelemeli olarak bir f fonksiyonu ile işleyen ve seçilen uzunlukta çıktı veren yapıdır. Tüm işlemlerini tek bir permütasyon temelinde gerçekleştirir. Permütasyon, durum matrisi ile ifade edilir ve adım işlemleri durum matrisi üzerinde gösterilir. Permütasyon, b-bit uzunluğundadır. Diğer parametreler ise r bit oranı yani blok uzunluğu ve c kapasite değerleridir. b uzunluğu r ve c parametrelerinin toplamı ile elde edilir (Bertoni et al., 2011).

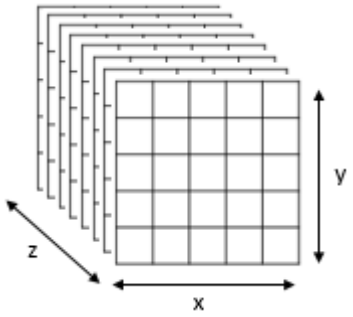
Başlangıçta, durum matrisinin tüm bitleri sıfırdır. Girdi olarak alınan mesaj r-bit bloklara ayrılır ve gerekirse belirli bir kural ile dolgulanır, daha sonra r-bit bloklara ayrılır. Dolgulama kuralı 10*1 ile ifade edilir. Yani boş bit pozisyonlarının birincisine ve sonuncusuna 1 biti, arada kalan diğer boş bit pozisyonlarına ise 0 biti yazılır. Sponge yapısı iki aşamadan oluşur. İlk aşama emilme aşamasıdır. Mesaj, r-bit uzunluğunda bloklara ayrıldıktan sonra durum matrisinin ilk r-biti ile $\oplus$ işlemine girer ve f fonksiyonundan geçirilir. Tüm mesaj blokları bu işlemden geçirildikten sonra ikinci aşama olan sıkma aşamasına geçilir. Emilme aşamasından gelen durum matrisi f fonksiyonundan geçirilir ve ilk r-bit çıktı olarak alınır. Bu işlem istenilen çıktı uzunluğuna göre devam ettirilebilmektedir. Sponge yapısında, son c-bit asla direkt olarak girdi bloklarından etkilenmez ve sıkma aşamasında çıktı olarak verilmez. Fakat f fonksiyonunda yapılan işlemler nedeniyle dolaylı olarak mesajdan etkilenmektedir (Bertoni et al., 2009; Bertoni et al., 2011).

Şekil 2.2.2.1 Sponge Yapısındaki Emilme ve Sıkma Aşamaları



Keccak permütasyonları, permütasyon genişliği açısından b ∈ {25, 50, 100, 200, 400, 800, 1600} olmak üzere 7 tanedir. $l \in \{0,1,2,3,4,5,6\}$ olmak üzere $b = 25 \times 2^l$ ile hesaplanır. SHA-3 fonksiyonunda, Keccak permütasyonunun parametreleri b=1600, r=1088 ve c=512 olarak belirlenmiştir. Yinelemeli olarak uygulanan tur fonksiyonu sayısı ise permütasyon genişliğine bağlı olarak $n_r = 12+2l$ eşitliği ile belirlenir. Keccak sisteminde girdi bitleri, w CPU (işlemci) kelime boyutu olmak üzere, 5x5xw boyutlu bir durum matrisi üzerinde ifade edilir. Bu (x,y,z) koordinatlarına sahip üç boyutlu durum matrisi üzerindeki bit pozisyonları ise a[x][y][z] ile ifade edilir. Pozisyon indislerinin ait olduğu kümeler, $x \in Z_5$, $y \in Z_5$ ve $z \in Z_w$ şeklinde gösterilir.

Şekil 2.2.2.2 Keccak Durum Matrisi



Keccak permütasyonu Keccak-f[b] ile gösterilir. Yinelemeli bir permütasyondur ve n_r tur boyunca R tur fonksiyonu durum matrisine uygulanır. R tur fonksiyonu beş adımdan oluşur. Bu adımlar ι, χ, π, ρ, θ ile gösterilir. Durum matrisi üzerindeki her değer GF(2) cisminin elemanıdır. a[x][y][z], durum matrisinin (x,y,z). koordinatındaki biti belirtir ve $0 \leq i_r \leq n_r\text{-}1$ olmak üzere tur adımları aşağıdaki gibi açıklanır.

$$\theta: a[x][y][z] \leftarrow a[x][y][z] + \sum_{y'=0}^{4} a[x-1][y'][z] + \sum_{y'=0}^{4} a[x+1][y'][z-1],$$

$$\rho: a[x][y][z] \leftarrow a[x][y]\left[z - \frac{(t+1)(t+2)}{2}\right],$$

$$0 \leq t < 24 \; ve \; \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}^t \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \in GF(5)^{2x2},$$

$$eğer \; x = y = 0 \; ise \; t = -1 \, ,$$

$$\pi: \; a[x][y] \leftarrow a[x'][y'], \qquad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}^t \begin{pmatrix} x' \\ y' \end{pmatrix},$$

$$\chi: \; a[x] \leftarrow a[x] + (a[x+1] + 1)a[x+2],$$

$$\iota: \; a \leftarrow a + RC[i_r].$$

$$RC[i_r][0][0]\left[2^j - 1\right] = rc[j + 7i_r], \; \forall \, 0 \leq j \leq l,$$

$$rc[t] = (x^t \; mod \; x^8 + x^6 + x^5 + x^4 + x^1) \; mod \; x \in GF(2)[x].$$

b = τ(a) dönüşümü, a durum matrisinin z ekseni yönünde 1 bit çevirme anlamına gelir. Tüm eksenlerde çevirme işlemi ise $\tau[t_x][t_y][t_z]$ ile gösterilir ve a[(x-

t_x) mod 5][(y-t_y) mod 5][(z-t_z) mod w] şeklinde hesaplanır. $\tau[t_x][t_y][t_z] \circ \alpha = \alpha \circ \tau[t_x][t_y][t_z]$ eşitliğini sağlayan α dönüşümüne çeviri değişmezi adı verilir. a durum matrisi üzerinde a[x][y][(z+d) mod w] = a[x][y][z] şartını sağlayan en küçük d>0 tamsayısına a durum matrisinin z periyodu denir.

χ adımı Keccak tur fonksiyonunun tek doğrusal olmayan adımıdır. Keccak tur fonksiyonundaki diğer tüm adımlar doğrusal işlemlerden oluşur. χ adımında 5w uzunluklu yer değiştirme kutusu (Substitution Box - S-Box) paralel olarak 5-bit uzunluklu satırlar üzerine uygulanır. χ adımı tüm eksenlerde çeviri değişmezidir ve cebirsel derecesi ikidir. Bu adım, durum matrisi üzerinde yayılımı ve korelasyonu sağlar. χ adımındaki fonksiyonun tersi alınabilir fakat χ adımının ters fonksiyonunun doğası χ fonksiyonundan farklıdır. Örneğin tersinin cebirsel derecesi iki değildir. χ adımında, bir adet $\oplus$, bir adet & ve bir adet NOT işlemi kullanılır.

χ adımının cebirsel derecesinin iki olması sayesinde, *a'* ile verilen girdi farkı için mümkün çıktı farklarının uzayı $2^{w_r(a',b')}$ elemanları ile doğrusal afin çeşitlilik oluşturur. Ayrıca *(a',b')* farkının kardinalitesi ya sıfırdır ya da ikinin bir kuvvetidir. Bu farka karşılık gelen $w_r(a',b')$ ağırlığı yalnızca girdi farkı olan *a'* değerine bağlıdır. Tek bir satıra χ adımı uygulandığında, bir girdi farkına karşılık gelen çıktı farklarının afin çeşitliliğini oluşturmak için *i.* pozisyondaki sıfır olmayan bit *S(i), i* ve *j* pozisyonunda sıfır olmayan bit *S(i,j)* ile gösterilir. Ayrıca mümkün çıktı farklarının kümesi *A'= χ(a')* ile gösterilir ve $<c_j>$ baz vektörüdür. Eğer $a'_i a'_{i+1} a'_{i+2} a'_{i+3} \in \{.100, .11, 001.\}$ ise baz *S(i)* ile genişletilir. Eğer $a'_i a'_{i+1} a'_{i+2} a'_{i+3} \in .101$ ise baz *S(i,i+1)* ile genişletilir. Keccak'taki ağırlık ölçümü Hamming ağırlığıdır. Hamming ağırlığı hesaplanırken girdideki sıfır olmayan semboller sayılır. Tüm bir girdi farkı sonuçları tek parite modellerinin afin çeşitliliğidir ve 4 ağırlığındadır. 31 tane sıfır olmayan farkın, beş tanesi 2 ağırlığına, on beş tanesi 3 ağırlığına ve on bir tanesi 4 ağırlığına sahiptir. Ayrıca (a',b') farkı, mutlak değer a sayısının bitleri üzerinde bazı şartlar koyulmasına sebep olur. $B=A' \oplus b' = \chi(a') \oplus b'$ olmak üzere a sayısının i. her bit pozisyonu için aşağıdaki şartlar uygulanır.

Eğer $a'_{i+1} a'_{i+2} = 10$ ise $a'_{i+2} = B_i$,

Eğer $a'_{i+1} a'_{i+2} = 11$ ise $a'_{i+1} \oplus a'_{i+2} = B_i$,

Eğer $a'_{i+1} a'_{i+2} = 01$ ise $a'_{i+1} = B_i$.

χ adımının cebirsel derecesinin iki olması sayesinde, verilen u çıktı maskesi için v girdi maskesi uzayı doğrusal afin çeşitlilik oluşturur. Bu afin çeşitlilik $2^{w_c(v,u)}$ elemana sahiptir. Aslında ağırlık fonksiyonu yalnızca çıktı maskesine bağlıdır ve çift tamsayı değerindedir. χ üzerindeki korelasyon büyüklüğü ya sıfırdır ya da $\pm 2^{w_c(v,u)/2}$ değerindedir. Tek bir satıra uygulanan χ işlemi için çıktı maskesine karşılık gelen girdi maskelerinin afin çeşitliliğin oluşturulması için yöntem (Bertoni et al.,

2009)'de verilmiştir. 1-run ile tanımlanan terim, m adet 1 bitinden önce ve sonra gelen 0 bitinin dizisi olarak tanımlanır. Doğrusal afin çeşitlilik, U' kümesi ve $<c_j>$ bazı yardımıyla ifade edilir. Başlangıçta U' kümesinin tamamı 0 elemanından oluşur ve baz boş kümedir. 1-run dizisi $a_s a_{s+1} \ldots a_{s+m-1}$ ile gösterilirse her bir terimi için bazı işlemler uygulanır. U' kümesinin s. pozisyonuna 1 eklenir ve 1-run dizisinin başlangıç pozisyonu $i=s$ olarak ayarlanır. $a'_i a'_{i+1} = 11$ olduğu sürece *S(i+1,i+3)* ve *S(i+2)* ile baz genişletilir. Eğer $a'_i a'_{i+1} = 10$ ise baz *S(i+1)* ve *S(i+2)* ile genişletilir. Tüm çıktı maskesi sonuçları tek parite modellerinin afin çeşitliliğidir ve 4 ağırlığındadır. 31 tane sıfır olmayan maskenin, on tanesi 2 ağırlığına, yirmi bir tanesi 4 ağırlığına sahiptir.

θ dönüşümü durum matrisi üzerinde yayılım sağlayan ve tüm yönlerde çeviri değişmezi olan doğrusal bir adımdır. Eğer θ dönüşümü adım işlemleri arasında olmazsa Keccak'ta yayılım etkili bir şekilde sağlanamaz. θ dönüşümünün dal sayısı dörttür. Bu düşük dal sayısına rağmen yüksek seviyede yayılım sağlar. θ dönüşümünde yalnızca iki adet $\oplus$ işlemi vardır. Bu adımın χ adımı ile olan etkileşimi sayesinde bir tur girdisindeki her bit 31 çıktı bitini etkiler. Aynı zamanda, bir turun her çıktı biti 31 girdi bitine bağımlı olarak değişmektedir.

θ dönüşümünde $x^i y^j z^k$ monomiali a[i][j][k] pozisyonundaki bit değeri tanımlanır. $\tau[t_x][t_y][t_z]$ çevirisi, $x^{t_x} y^{t_y} z^{t_z}$ monomialinin modunda $1+x^5$, $1+y^5$ *ve* $1+z^w$ polinomlarının çarpımına karşılık gelir. Ayrıca d, z-periyodunu gösteren en küçük sıfır olmayan tamsayı değeri olmak üzere $1+z^d$ ifadesi *a* durum matrisini böler. *a'* polinomu, *a* durum matrisinin z-indirgenmiş haline karşılık gelmek üzere aşağıdaki eşitlik yazılabilir;

$$a = (1 + z^d + z^{2d} + \ldots + z^{w-d}) \times a'' = (1 + z^w) / (1 + z^d) \times a' .$$

Durum matrisi bir polinomla ifade edilirse, θ dönüşümü $\bar{y} = \sum_{i=0}^{4} y^i = (1 + y^5) / (1 + y)$ polinomuyla $1 + \bar{y} (x + x^4 z)$ şeklinde ifade edilebilir. θ dönüşümünün tersi, bu polinomun tersidir. Tüm değerler için θ dönüşümünün tersinin ağırlığı b/2 mertebesindedir. Bunun anlamı, tek bir aktif bit farkı ile θ dönüşümünün tersinin durum matrisine uygulanması ile aktif bitlerin yarısı değişmektedir. Benzer şekilde, θ dönüşümünün tersinin çıktı maskesi, girdi maskesindeki aktif bitlerin yarısını değiştirmektedir.

π dönüşümü, durum matrisi üzerinde uzun süreli bir yayılım özelliği sağlar. π dönüşümünde kullanılan matris M ile gösterilir ve verilen matris haricinde başka birçok matris kullanılabilir. Kullanılan matris elemanları GF(5) cisminin elemanları olan 2x2 boyutlu bir matristir. Bu matris, matris çarpımı altında 1, 2, 3, 4, 5, 6, 8, 10, 12, 20 ve 24 mertebelerine sahip 480 elemanlı bir grubun elemanıdır. π dönüşümünde *u* çıktı maskesinin *v* girdi maskesi üzerine yayılımı $v = \pi^T(u)$ ile gösterilir. Burada π^T ile gösterilen π dönüşümündeki sonuç matrisinin transpozudur.

ρ dönüşümü, durum matrisi üzerindeki dilimler arasında dağılım özelliğini sağlar. Bununla birlikte yayılımın hızlanmasına yardımcı olur. Eğer ρ adımı kullanılmazsa durum matrisi üzerindeki yayılım epey yavaştır. π dönüşümünde de olduğu gibi girdi ve çıktı maskeleri arasındaki ilişki $v = \rho^{T}(u)$ ile gösterilir. Son olarak ι dönüşümü durum matrisi üzerindeki simetrinin kırılması için kullanılan adımdır. Tüm yönlerde çeviri değişmezidir. Tur sabitlerinin sayısı arttıkça simetri daha fazla kırılır. Keccak'ta her turda farklı bir tur sabiti oluşturulur ve bu tur sabitleri, maksimum uzunluklu bir LFSR'nin çıktıları kullanılarak oluşturulur. Keccak'taki bu beş adım dönüşümü uygulanırken öncelikle θ dönüşümü uygulanmalıdır. Diğer tur adımları rastgele sırada uygulanabilir.

Keccak'ta diferansiyel ve doğrusal yolların yayılımı birbirine benzerdir. Diferansiyel yollar Keccak turlarındaki farkların yayılımıdır. Doğrusal yollar ise maskelerin yayılımıdır. χ adımının girdisindeki a farkı için mümkün çıktı farklarının kümesi doğrusal afin çeşitlilik oluşturur. χ adımının çıktısındaki a maskesi için verilen çıktı maskesinin sıfır olmayan korelasyonu ile girdi maskelerinin kümesi doğrusal afin çeşitlilik oluşturur. Bu sebeple girdiden çıktıya diferansiyel yolların yayılımı ve çıktıdan girdiye doğrusal yolların yayılımı benzerdir.

χ adımının i. turdaki girdi farkı a_i ile belirtilsin. χ adımından sonra oluşan çıktı farkı ise b_i ile belirtilsin. Benzer şekilde χ adımının çıktı maskesi a_i, girdi maskesi ise b_i ile gösterilsin. Doğrusal afin çeşitlilik iki durum için de $B(a_i)$ ile gösterilir. χ adımının doğrusal olmayan adım olması sayesinde b_i çıktı farkı a_{i+1} girdi farkına karar vermektedir. Buradan $a_i=\pi(\rho(\theta(b_i)))$ elde edilir. Doğrusal adımlar l ile gösterilmek üzere $l = \pi \circ \rho \circ \theta$ ile hesaplanmaktadır. Benzer şekilde χ adımının doğrusal olmayan adım olması sayesinde b_i maskesi a_{i+1} maskesine karar vermektedir. Bu durumda $a_i= \theta^{T}(\rho^{-1}(\pi^{-1}(b_i)))$ olur ve $l = \theta^{T} \circ \rho^{-1} \circ \pi^{-1}$ ile hesaplanmaktadır. $B(a_i)$ ve l fonksiyonlarının sonuçları diferansiyel veya doğrusal yollara bağlıdır. l fonksiyonunun doğrusallığından dolayı doğrusal afin çeşitlilik tekrar hesaplanır ve $A(a_i)$ ile gösterilir. $0 \le i \le k$ olmak üzere k tur boyunca yollar aşağıdaki gibi tanımlanmaktadır.

$$a_0 \xrightarrow{\chi} b_0 \xrightarrow{l} a_1 \xrightarrow{\chi} b_1 \xrightarrow{l} \dots a_k.$$

Diferansiyel yol Q'nun kısıtlama ağırlığı $w_r(Q) = \sum_{0 \le i < k} w_r(a_i)$ olarak tanımlanmıştır. Permütasyon genişliğine yakın ağırlık değerleri için Q diferansiyel yolunun diferansiyel olasılığı için $2^{-w_r(Q)}$ iyidir. Eğer ağırlık değeri permütasyon boyutuna yakın ise yolun kardinalitesi tamsayı olacağı için artık geçersiz olur. ι adımının ağırlık veya yollar üzerinde bir etkisi yoktur. Doğrusal yolun korelasyon ağırlığı a_0 ve a_k maskeleri ile tanımlanan girdi ve çıktı maskeleri arasındaki korelasyon ile belirlenir. Ayrıca $w_c(Q) = \sum_{0 \le i < k} w_c(a_i)$ ile hesaplanmaktadır. Korelasyon ölçümü ise $2^{-w_c(Q)}$ ile verilir (Bertoni et al., 2011).

Tablo 2.2.2.1 Keccak Özet Fonksiyonları Parametre ve Güvenlik Değerleri

	Permütasyon Uzunluğu	**Çıktı Uzunluğu**	**Blok Uzunluğu**	**Kapasite Uzunluğu**	**Çakışma Direnci**	**Ön-Görüntü Direnci**	**2. Ön-Görüntü Direnci**
Keccak-224	1600-Bit	224-Bit	1152-Bit	448-Bit	112-Bit	224-Bit	224-Bit
Keccak-256	1600-Bit	256-Bit	1088-Bit	512-Bit	128-Bit	256-Bit	256-Bit
Keccak-384	1600-Bit	384-Bit	832-Bit	768-Bit	192-Bit	384-Bit	384-Bit
Keccak-512	1600-Bit	512-Bit	576-Bit	1024-Bit	256-Bit	512-Bit	512-Bit

Keccak algoritması ile aynı algoritmayı kullanarak seçilen uzunlukta özet değeri üreten genişletilmiş çıktı fonksiyonuna SHAKE adı verilir. İki adet SHAKE versiyonu vardır. Bunlar SHAKE-128 ve SHAKE-256'dır. Blok işleme aşaması olan emilme aşamasının Keccak algoritmasından hiçbir farkı yoktur. Sıkma aşaması ise istenildiği kadar devam ettirilebilir ve her çalışmadan sonra çıktı blokları alınır. Son olarak bu çıktılar birleştirilir. SHAKE algoritmalarının parametre ve güvenlik değerleri Tablo 2.2.2.2'de verilmiştir (Kelsey et al., 2016).

Tablo 2.2.2.2 SHAKE Algoritmaları Parametre ve Güvenlik Değerleri

	Permütasyon Uzunluğu	**Çıktı Uzunluğu**	**Blok Uzunluğu**	**Kapasite Uzunluğu**	**Çakışma Direnci**	**Ön-Görüntü Direnci**	**2. Ön-Görüntü Direnci**
SHAKE-128	1600-Bit	u-Bit	1344-Bit	256-Bit	mn(u/2,128)	≥mn(u,128)	mn(u,128)
SHAKE-256	1600-Bit	u-Bit	1088-Bit	512-Bit	mn(u/2,256)	≥mn(u,256)	mn(u,256)

Genişletilmiş çıktı fonksiyonu olan SHAKE algoritmasında elde edilen çıktının uzunluğunun çok büyük olması güvenliğin arttığı anlamına gelmemektedir. Kuantum bilgisayarlar sonrası çok uzun rastgele sayı ihtiyacında SHAKE gibi genişletilebilir çıktı fonksiyonlarına ihtiyacımız olacak. Genel ataklara karşı SHAKE direnci, verebileceği en kısa çıktı üzerinden hesaplanmaktadır.

3. NIST HAFİF-SİKLET KRİPTOGRAFİ STANDARTLAŞTIRMA PROJESİ

2018 yılında NIST tarafından hafif-siklet kriptografi için bir standartlaştırma projesi başlatılmıştır. Bu proje kapsamında, bilim insanları standart haline gelmesi gerektiğini düşündükleri doğrulanmış şifreleme algoritmaları ile özet fonksiyonlarını NIST'e göndermiştir. NIST, gönderilen sistemleri inceleyerek eleme usulüyle 2019 yılının Mart ayında standart olarak seçilebilecek sistemleri yayınlamıştır. Bu tez kapsamında, NIST hafif-siklet projesinde ilk elemeyi geçen özet fonksiyonları tanıtılarak, bu özet fonksiyonlarına ait güvenlik tanımlamaları karşılaştırılacaktır.

3.1 ACE

ACE sistemi, doğrulanmış şifreleme algoritması ve özet fonksiyonundan oluşur. ACE sitemi, kriptografik işlemlerinde Sponge-Duplex yapısını kullanır. Sponge yapısının Duplex modu simetrik şifrelemenin birçok alanında kullanılabilmektedir (Bertoni et al., 2012). Sponge yapısının bir modu olduğu için asıl temeli tek bir permütasyon üzerinde tüm işlemlerin gerçekleştirilebilmesidir. Bunun yanında yine Sponge yapısına dayanan birçok avantaja sahiptir (Bertoni et al., 2009). ACE sisteminde kullanılan permütasyon üç parametreden oluşur. Bu parametreler, blok uzunluğunu belirten r değeri, kapasiteyi belirten c değeri ve permütasyon uzunluğu olan $b=r+c$ değeridir. ACE kriptografik sistemindeki tüm işlemler 320-bit permütasyon üzerinde 16 tur boyunca yinelemeli şekilde gerçekleştirir. 320-bit permütasyon beş adet 64-bit kelimeden oluşan permütasyondur ve kelimeler A, B, C, D, E sembolleri ile gösterilir. ACE permütasyonu 288-bit, 320-bit veya 384-bit olabilir. (Aagaard et al., 2019)'a göre 288 bit ile 64-bit işlemci talimatları kullanılamayacağından ve 384-bit uzunluk hafif uygulamalarda donanım için ağır olacağından 320-bit permütasyon uygun görülmüştür.

ACE algoritmasında kullanılan işlemler; $\oplus$, &, sola ROT ve 64-bit kelime değerleri üzerinde karıştırma işlemidir. Algoritmanın içerdiği işlemlerin tümü bit düzeyindedir. ACE sistemindeki doğrusal adım dönüşümü, 64-bit kelime üzerindeki karıştırma işlemidir. Doğrusal adım dönüşümü π ile gösterilir. Örneğin; A kelimesi 0, B kelimesi 1 gibi beş kelime indekslenir ve π(3, 2, 0, 4, 1), sırasıyla bu beş kelimenin yerine gelecek kelimenin indeksini göstermektedir. Doğrusal olmayan adım dönüşümü ise 64-bit blok uzunluğuna sahip ve cebirsel derecesi 36 olan Simeck-Box adı verilen bir yapı sayesinde gerçekleştirilir. Simeck-Box yapısı yalnızca A, C, E kelimelerine yani çift indeksli kelimelere uygulanır. Simeck-box sekiz turdan oluşur ve her tura ait tur sabitleri vardır. ACE fonksiyonunda, tur sabitlerinin yanında adım sabitleri de vardır ve $0 \leq i \leq 7$ olmak üzere sırasıyla (rc_0^i, rc_1^i, rc_2^i), (sc_0^i, sc_1^i, sc_2^i) şeklinde gösterilir. Bu sabitlerin her biri 8-bit

uzunluğundadır. Simeck-Box yapısında tur sabitleri kullanılırken, adım sabitleri B, D, E kelimeleriyle $\oplus$ işlemine girer. ACE sisteminde anahtar toplaması yerine sekiz tur boyunca tur sabiti toplaması yapılır ve bu işlem γ_i ile gösterilir. $\gamma_i = 1^{31} \,||\, rc_i$ şeklinde tanımlanır ve yinelemeli bir yapıya sahiptir. Simeck-Box yapısını matematiksel olarak aşağıdaki şekilde tanımlanır.

$x \in \{0,1\}^n$ olmak üzere $L^i(x) = (x_i, x_{i+1}, \ldots, x_{n-1}, x_0, x_1, \ldots, x_{i-1})$,

$f_{(5,0,1)} : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$ olmak üzere, $f_{(5,0,1)}(x) = (L^5(x) \,\&x) \oplus L^1(x)$,

$2 \leq i \leq 9$ olmak üzere, $x_j \leftarrow f_{(5,0,1)}(x_{j-1}) \oplus x_{j-2} \oplus \gamma_{j-2}$,

$(x_9 \,||\, x_8) \leftarrow$ Simeck-Box-64$(x_1 \,||\, x_0, rc)$.

A[4], A[5], A[6], A[7], C[4], C[5], C[6], C[7] baytları permütasyon üzerindeki *r* parametresini oluşturur. Yani *r* parametresi 64-bit değerindedir. ACE permütasyonundaki *r*-bit kısım doğrulanmış şifreleme ve özet değeri üretme işlemlerini gerçekleştirmek için emilme ve sıkma aşamalarına girecek olan mesaj bloğunun uzunluğunu gösterir. ACE sisteminde işlemlerin tamamı durum matrisi üzerinde ifade edilir.

ACE özet fonksiyonu, 24-bit uzunluğunda IV başlangıç vektörü ve seçilen uzunluktaki mesajı girdi olarak alır, 256-bit uzunluğunda çıktı üretir. Matematiksel olarak; $ACE\text{-}Hash : \{0,1\}^* \times \{0,1\}^{24} \rightarrow \{0,1\}^{256}$ şeklinde gösterilir. Öncelikle girdi mesajı 64-bit uzunluğunda parçalara bölünür ve eğer son bloktaki tüm bit pozisyonları dolu değil ise 10* kuralı ile boş bit pozisyonlarına dolgulama yapılır. Durum matrisinin ilk 64-bit kısmı emilme ve sıkma aşamalarından geçecek bölümdür. *b* permütasyon uzunluğu olmak üzere $b = r + c$ eşitliğinden *c* parametresi 256-bit uzunluğundadır ve Sponge yapısına göre güvenliği belirleyen gizli kısımdır.

ACE özet fonksiyonunda ilk olarak durum matrisi üzerindeki B[7], B[6] ve B[5] baytları başlangıç vektörünün bitleri ile doldurulur. Yani, bu bayt değerlerine sırasıyla 0x80, 0x40 ve 0x40 değerleri yerleştirilir. Durum matrisi üzerindeki diğer bit pozisyonları 0 biti ile doldurulur ve durum matrisi bir kez ACE permütasyonundan geçirilir. Daha sonra dolgulanmış mesaj ile beraber emilme aşaması başlar. Emilme aşamasında ilk olarak 64-bit uzunluğundaki her mesaj bloğu, durum matrisinin ilk 64-bit kısmı ile $\oplus$ işleminden geçirilir. Daha sonra durum matrisi ACE permütasyonu ile işlenir. Tüm mesaj blokları bu şekilde işlendikten sonra emilme aşaması biter. Bu aşama bittikten sonra oluşan durum matrisi, sıkma aşamasına girdi olarak verilir. Sıkma aşaması özet çıktısının alındığı aşamadır. Emilme aşamasından gelen durum matrisinin ilk 64-biti çıktı olarak alınır ve durum matrisi ACE permütasyonundan geçirilir. Yeni oluşan durum matrisinin ilk-64 biti çıktı olarak alınır. Bu işlem toplamda dört kez tekrar eder. Sıkma aşaması bittikten sonra 256-bit uzunluğunda özet değeri oluşur (Aagaard et al., 2019).

Tablo 3.1.1 ACE Özet Fonksiyonu Güvenlik Değerleri

ACE Özet Fonksiyonu	Çakışma	Ön-Görüntü	2. Ön-Görüntü
	128-bit	192-bit	128-bit

ACE özet fonksiyonunun sağladığı güvenlik tabloda gösterilmiştir. (Aagaard et al., 2019)'a göre ACE sisteminin yayılım gücü yüksektir. ACE permütasyonu 11. turda Simeck-Box ile istenilen yayılımı sağlar fakat her adımda kelime değerleri güncellendiğinden adım fonksiyonu 5 kez uygulanmış olmalıdır. Diferansiyel ve doğrusal kriptanalizi ACE permütasyonu modellenerek en az sayıdaki aktif Simeck-Box sayısı kullanılarak yapılmıştır. Bu durumda maksimum diferansiyel karakteristik olasılığı en az 2^{-320} olarak sınırlandırılmıştır. Tur sayısı 8 olduğunda, tam yayılım için gerekli adım fonksiyonu sayısı en az 15 olur. Tur sayısı 8, adım fonksiyonu sayısı 16 kabul edildiğinde; doğrusal adım $\pi = (3,2,0,4,1)$ kullanılsın. 21 tane aktif Simeck-Box olmak üzere 8 tur Simeck-Box için maksimum diferansiyel olasılık sınırı $p = 2^{-15,8}$ olsun. Bu durumda $|21 \times (-15.8)| \approx 331.8 > 320$ elde edilir. Yani maksimum diferansiyel olasılık sınırı $p^{21} = 2^{-331.8}$ olur. Bir doğrusal karakteristiğin maksimum kare korelasyonunun hesaplanmasında $\gamma = 2^{-15.6}$ kullanılır ve sonuç $2^{-327.6}$ dır. Bu sonuç 8 tur Simeck-Box'a ait maksimum kare korelasyondur. ACE permütasyonunda cebirsel derecenin hesaplanmasında bit tabanlı bölme işlemi kullanılmıştır. Ayrıca ACE permütasyonunda kelime sayısı tek olmasının bileşen fonksiyonlarının cebirsel derecesi üzerinde etkisi vardır.

Kriptografik permütasyonlardaki turların simetri özelliklerinden yararlanarak yapılan birçok atak vardır. Bu atakların önüne geçmek amacıyla tur ve adım sabitlerinin iyi seçilmesi gerekmektedir. ACE permütasyonu tur ve adım sabitlerini oluşturmak için 7-bit LFSR kullanır. (Aagaard et al., 2019)'a göre doğrusal olmayan adım olarak Simeck-box seçilmesinin sebebi neredeyse tüm platformlarda donanım verimliliği sağlaması ve yüksek performansıdır. Ayrıca yalnızca $\oplus$, & ve kaydırma işlemleri ile çalışır. Diferansiyel ve doğrusal kriptanalize karşı beklenen sınırları Simeck-Box karşılayabilmektedir. Doğrusal adım seçimi, ara adımların diferansiyel ve doğrusal özellikleri yansıtması açısından çok önemlidir. Az miktarda aktif Simeck-Box kullanarak istenenlerin sağlanması için iki adet uygun permütasyon bulunmuştur. Sadece $\pi = (3,2,0,4,1)$ ve $\pi_0 = (3,4,1,2,0)$ permütasyonları ACE sistemini 21 aktif Simeck-Box ile 5 turda istenilen bit yayılımına ulaştırmaktadır. Tüm güvenlik tanımları Sponge yapısına dayanılarak yapılabilmektedir. Bunların haricinde ACE özet fonksiyonunun güvenlik sınırları da yine Sponge yapısı ile belirlenir. Böylece;

Çakışma: $minimum(2^{h/2},2^{c/2})$

Ön-Görüntü: $minimum(2^{min(h,b)},maksimum(2^{min(h,b)-r},2^{c/2}))$

2. Ön-Görüntü: $minimum(2^h, 2^{c/2})$

şeklindedir (Bertoni et al., 2009; Aagaard et al., 2019).

3.2 ASCON

ASCON sistemi hem doğrulanmış şifreleme hem de özet fonksiyonu algoritmalarına sahiptir. Doğrulanmış şifreleme algoritmaları ASCON-128 ve ASCON-128a, özet fonksiyonu ASCON-HASH ve uzatılmış çıktı fonksiyonu ASCON-XOF ile adlandırılır. ASCON tüm işlemlerini 320-bit permütasyon üzerinde gerçekleştirir. (Dobraunig et al., 2016)'e göre tüm sistemleri 128-bit güvenlik sağlamaktadır. Permütasyon işlemlerinin tümü bit düzeyindedir. Ayrıca cebirsel derecesi 2 olan bir S-Box kullanır. Bu sebeplerle, ASCON donanım için elverişlidir (Dobraunig et al., 2016).

ASCON, (Bertoni et al., 2012)'de anlatılan Monkey-Duplex yapısını kullanır ve bu sebeple iki adet permütasyon işlemi vardır. Bu permütasyon işlemlerinin içeriği, tur sayıları haricinde tamamen aynıdır ve üç adet tur dönüşümünden oluşur. ASCON özet fonksiyonu bu permütasyonların yalnızca ilkini kullanır. 320-bit permütasyon bir durum matrisi üzerinde gösterilir ve ASCON permütasyonu üzerindeki kelime değerleri 64-bit uzunluğundadır. Birinci tur dönüşümü sabitler ile toplama işlemidir. Her turda, tur sabiti durum matrisi üzerindeki üçüncü kelimeye eklenir. Diğer bir tur dönüşümü ise yer değiştirmedir. Bu adımda 5-bit S-Box kullanılarak durum matrisi üzerinde 64 tane paralel işlem yapılır. Son tur dönüşümü ise doğrusal yayılımdır. Bu adımda, w, w1, w2, w3, w4 durum matrisi üzerindeki kelimeler ve >> gösterimi ROT işlemi olmak üzere;

$$w \leftarrow \sum_0(w) = w \oplus (w >> 19) \oplus (w >> 28),$$

$$w1 \leftarrow \sum_1(w1) = w1 \oplus (w1 >> 61) \oplus (w1 >> 39),$$

$$w2 \leftarrow \sum_2(w2) = w2 \oplus (w2 >> 1) \oplus (w2 >> 6),$$

$$w3 \leftarrow \sum_3(w3) = w3 \oplus (w3 >> 10) \oplus (w3 >> 17),$$

$$w4 \leftarrow \sum_4(w4) = w4 \oplus (w4 >> 7) \oplus (w >> 41),$$

işlemleri uygulanır.

ASCON-HASH ve ASCON-XOF fonksiyonları Sponge yapısı temelli özet fonksiyonlarıdır. (Dobraunig et al., 2016)'te ASCON özet fonksiyonu için önerilen parametre değerleri blok uzunluğu 64-bit, kapasite uzunluğu ise 256-bittir. İlk permütasyon işlemi 12 tur boyunca çalışır ve 256-bit uzunluğunda özet değeri üretir. Özet değerinin uzunluğu h ile gösterilebilir. Hem özet fonksiyonu hem de genişletilmiş çıktı fonksiyonu aynı algoritmayı kullanır. Eğer özet değerinin

uzunluğu, yani h değeri, sıfır olarak belirtilirse, bu durum, fonksiyonun genişletilmiş çıktı fonksiyonu olduğunu gösterir.

ASCON özet fonksiyonu emilme ve sıkma adı verilen iki aşamadan oluşur. Bu aşamalara geçilmeden önce girdi mesajı boyutunun 64-bit blok uzunluğunun katı olup olmadığı kontrol edilir. Eğer son mesaj bloğunda boş bit pozisyonları var ise dolgulama işlemi uygulanır. Mesaj üzerine uygulanan dolgulama kuralı, eksik bitlerin ilk bitine 1, geri kalan bit pozisyonlarına ise alabildiğince 0 biti koyulmasıdır ve 10* ile gösterilir. Amaç girdinin 64-bit uzunluğunda bloklara tam olarak ayrılabilmesidir. ASCON özet fonksiyonunda ilk olarak 320-bit durum matrisi üzerine başlangıç vektörü yerleştirilir. Durum matrisi 12 tur boyunca ASCON tur işlemleri ile işlenir. Dolgulanmış mesaj 64-bit bloklara ayrılır. Her mesaj bloğu, emilme adı verilen aşamada öncelikle durum matrisi ile $\oplus$ işlemine tabi tutulur. Daha sonra, elde edilen durum matrisi ASCON permütasyon işlemlerinden geçirilir. Tüm mesaj blokları işlendikten sonra sıkma aşamasına geçilir. Bu aşama, özet çıktısının üretildiği aşamadır ve genişletilmiş çıktı fonksiyonu sayesinde sıkma aşaması tekrar tekrar çalıştırılarak, istenilen uzunlukta özet değeri üretilebilir. Sıkma aşamasında, emilme aşamasından alınan durum matrisinin ilk 64-biti çıktı olarak alınır ve durum matrisi ASCON permütasyonu ile işlenir. Elde edilen durum matrisinin ilk 64-biti çıktı olarak alınarak önceki çıktının sonuna eklenir. Bu işlem istenilen uzunlukta özet değeri elde edilene kadar tekrar eder (Dobraunig et al., 2016).

Tablo 3.2.1 ASCON Özet ve Genişletilmiş Çıktı Fonksiyonu Güvenlik Değerleri

	Çakışma	**Ön-Görüntü**	**2. Ön-Görüntü**
ASCON-HASH	128-bit	128-bit	128-bit
ASCON-XOF	Min(128,h/2)	Min(128,h)	Min(128,h)

ASCON özet fonksiyonu ve genişletilmiş çıktı fonksiyonunun güvenlik analizleri Sponge yapısına dayandırılmıştır ve aynı şekilde yapılmaktadır. ASCON permütasyonu ideal olmayan diferansiyel ve doğrusal özelliklere sahiptir. ASCON sisteminin doğrusal olmayan adımında kullanılan S-Box yapısının maksimum diferansiyel olasılığı 2^{-2} olarak hesaplanmıştır ve S-Box yapısı 3 adet diferansiyel dal sayısına sahiptir. Ayrıca maksimum doğrusal sapması 2^{-2} ve doğrusal dal sayısı 3 değerindedir. Doğrusal fonksiyonların toplamının diferansiyel ve doğrusal dal sayısı 4 değerindedir. Diferansiyel karakteristik için 3 turda aktif S-Box yapısının minimum sayısı 15, doğrusal karakteristik için aktif S-Box yapısı 13 olarak hesaplanmıştır. Ayrıca Tablo 3.2.2'de 5. Tura kadar olan minimum aktif S-Box sayıları gösterilmiştir.

,

Tablo 3.2.2 ASCON Diferansiyel ve Doğrusal Karakteristik İçin Mininmum Aktif S-Box

	1 Tur	2 Tur	3 Tur	4 Tur	5 Tur
Diferansiyel Karakteristik İçin	1	4	15	≤ 44	≤ 78
Doğrusal Karakteristik İçin	1	4	13	≤ 43	≤ 67

ASCON sistemi 2 cebirsel derecesine sahiptir. (Dobraunig et al., 2016)'e göre bu cebirsel derece donanım uygulamaları için oldukça iyidir fakat cebirsel ataklara karşı dirençli olması için tur sayısı iyi belirlenmelidir. Cebirsel derece k tur sonra 2^k değeri ile üstten sınırlıdır. $x_{0,i}$, $x_{1,i}$, $x_{2,i}$, $x_{3,i}$, $x_{4,i}$ 64-bit uzunluğunda 5 kelime değerini ve i bit pozisyonlarını göstersin. Bunun yanında, $y_{0,i}$, $y_{1,i}$, $y_{2,i}$, $y_{3,i}$, $y_{4,i}$ sembolleri kelime değerlerindeki bit pozisyonlarına S-Box uygulandıktan sonraki yeni kelime değerlerini göstersin. Bu durumda S-Box yapısının kelime değerleri üzerine uygulanmasının cebirsel normal formu aşağıdaki gibidir.

$$y_{0,i} = x_{4,i}\, x_{1,i} \oplus x_{3,i} \oplus x_{2,i}\, x_{1,i} \oplus x_{2,i} \oplus x_{1,i}\, x_{0,i} \oplus x_{1,i} \oplus x_{0,i},$$

$$y_{1,i} = x_{4,i} \oplus x_{3,i}\, x_{2,i} \oplus x_{3,i}\, x_{1,i} \oplus x_{3,i} \oplus x_{2,i}\, x_{1,i} \oplus x_{2,i} \oplus x_{1,i} \oplus x_{0,i},$$

$$y_{2,i} = x_{4,i}\, x_{3,i} \oplus x_{4,i} \oplus x_{2,i} \oplus x_{1,i} \oplus 1,$$

$$y_{3,i} = x_{4,i}\, x_{0,i} \oplus x_{4,i} \oplus x_{3,i}\, x_{0,i} \oplus x_{3,i} \oplus x_{2,i} \oplus x_{1,i} \oplus x_{0,i},$$

$$y_{4,i} = x_{4,i}\, x_{1,i} \oplus x_{4,i} \oplus x_{3,i} \oplus x_{1,i}\, x_{0,i} \oplus x_{1,i}.$$

Bu sefer $y_{0,i}$, $y_{1,i}$, $y_{2,i}$, $y_{3,i}$, $y_{4,i}$ sembolleri kelime değerlerindeki bit pozisyonlarına S-doğrusal adım uygulandıktan sonraki yeni kelime değerlerini göstersin. Bu durumun cebirsel normal formu aşağıdaki gibidir.

$$y_{0,i} = x_{0,i} \oplus x_{0,i+19} \oplus x_{0,i+28},$$

$$y_{1,i} = x_{1,i} \oplus x_{1,i+61} \oplus x_{1,i+39},$$

$$y_{2,i} = x_{2,i} \oplus x_{2,i+1} \oplus x_{2,i+6},$$

$$y_{3,i} = x_{3,i} \oplus x_{3,i+10} \oplus x_{3,i+17},$$

$$y_{4,i} = x_{4,i} \oplus x_{4,i+7} \oplus x_{4,i+41}.$$

3.3 Gimli

Gimli sistemi hem doğrulanmış şifreleme hem de özet fonksiyonu yapısından oluşur. 384-bit uzunluğunda permütasyona sahiptir ve 24 turdan

oluşmaktadır. Permütasyon 3x4x32 boyutunda bir durum matrisi ile ifade edilir. w=32-bit, (x,y,z) eksenlerinden oluşan üç boyutlu durum matrisinin z eksenini belirtir ve kelime olarak adlandırılır. Gimli permütasyonunda $\oplus$, &, OR, sola ROT ve sola kaydırma işlemleri kullanılır ve tümü bit düzeyindedir.

Gimli, üç adım fonksiyonundan oluşur. Bu adım fonksiyonlarından biri doğrusal olmayan adımdır. Doğrusal olmayan adımda SP-Box adı verilen yapı Gimli durum matrisinin sütunlarına paralel olarak uygulanmaktadır. SP-Box yapısı ise üç adımdan oluşmaktadır. w,w_1,w_2 durum matrisinin bir sütunundaki kelimeler, $<<<<$ sola ROT işlemi ve $<<$ sola kaydırma olmak üzere;

$w \leftarrow w <<<< 24$
$w_1 \leftarrow w_1 <<<< 9$
$w \leftarrow w \oplus (w_2 << 1) \oplus ((w_1 \& w_2) << 2)$
$w_1 \leftarrow w_1 \oplus w \oplus ((w \text{ OR } w_2) << 1)$
$w_2 \leftarrow w_2 \oplus w_1 \oplus ((w \& w_1) << 3)$
$w \leftarrow w_2$
$w_1 \leftarrow w$

işlemleri SP-Box yapısını oluşturur. İkinci adım doğrusal adımdır. Bu adım, küçük ve büyük olmak üzere iki adet yer değiştirme işleminden oluşur. Küçük yer değiştirme işlemi, durum matrisinin y ekseni üzerindeki 1. ile 2. bitin ve 3. ile 4. bitin yer değiştirmesi işlemidir. Büyük yer değiştirme, durum matrisinin y ekseni üzerindeki 1. ile 3. bitin ve 2. ile 4. bitin yer değiştirmesi işlemidir. Küçük yer değiştirme işlemi birinci turdan başlayarak her dört turda bir kere, büyük yer değiştirme işlemi üçüncü turdan başlayarak her dört turda bir kere gerçekleşir. Son adım ise tur sabiti ile toplama işlemidir. Gimli'de, turlar 1,2,…,24 sayıları ile belirtilir. O anda bulunulan tur r ile gösterilir ve tur sabiti $0x9e377900 \oplus r$ ile hesaplanır. Her turda bu değer ile durum matrisi üzerindeki ilk kelime $\oplus$ işlemine tabi tutulur (Bernstein et al., 2017).

Gimli özet fonksiyonu ile özet değeri oluşturulurken başlangıçta 384-bit durum matrisinin tamamı 0 biti ile doludur. Özet fonksiyonu, girdi mesajını 128-bit bloklar halinde işlemektedir. Bu yüzden girdi mesajı 128-bit yani 16-bayt bloklara ayrılır. Gimli'de dolgulama kuralı şu şekilde gösterilmektedir; $0 \leq s \leq 15$ olmak üzere s değeri mesajın sonra 16 baytının kaçının dolu olduğunu göstersin. Bu durumda, durum matrisindeki ilk s-bayt ile belirtilen mesaj bloğu $\oplus$ işlemine tabi tutulur. Durum matrisindeki bir sonraki bayt ve durum matrisindeki son bayt 1 biti ile $\oplus$ işlemine tabi tutulur. Son olarak, bu durum matrisine Gimli permütasyonu uygulanır. 16-bayt uzunluğunda bölünmüş girdi mesajı blokları sırasıyla, öncelikle durum matrisinin ilk 16-baytı ile $\oplus$ işleminden geçer daha sonra durum matrisine Gimli permütasyonu uygulanır. Bu işlem tüm mesaj bloklarına uygulanır. Daha sonra 32-bayt uzunluğunda özet çıktısı elde etmek için öncelikle durum matrisindeki ilk 16-bayt alınır ve durum matrisi tekrar Gimli permütasyonundan

geçirilir. Yeni oluşan durum matrisinin ilk 16-baytı, bir önceki 16-baytın devamına eklenir ve özet değeri oluşturulmuş olur.

Gimli özet fonksiyonunda anlatılan işlemler iki aşamaya ayrılır. Başlangıçta girdiyi bloklara ayırarak Gimli permütasyonunun her bloğa uygulanmasına emilme aşaması, özet değerinin oluşturulduğu aşama ise sıkma aşaması denir. (Bernstein et al., 2017), hem özet fonksiyonu hem de doğrulanmış şifre için tüm ataklara karşı güvenlik direncini 2^{128} olarak belirtmiştir.. Gimli sisteminin kriptanalizinde iki güvenlik ihtiyacı üzerinde durulmuştur. Bunlardan biri çığ etkisi olarak da bilinen girdideki bir bit değişiminin çıktıda çok büyük farklılıklara yol açmasıdır. İkincisi ise girdide yalnızca bir bit ayarlandığında, çıktı olarak tamamen 1 biti ile dolu bir durum matrisine ulaşmak için kaç tur gereklidir. Bu işlem için $\oplus$ işlemi yerine & ve OR işlemleri kullanılır.

Gimli sistemindeki bit yayılımını göstermek için SP-Box yapısının girdi boyutu dikkate alınarak rastgele bir girdi üzerinde 1 bit değiştirilir. Bu bit değişiminde çıktıdaki bitlerin en az yarısının değişmesi için 10 tura ihtiyaç olduğu görülmüştür. Yapılan başka bir işlem ise SP-Box yapısında kullanılan işlemler aşağıdaki gibi değiştirilmiştir.

$x' \leftarrow x \text{ OR } (z << 1) \text{ OR } ((y \text{ OR } z) << 2)$,

$y' \leftarrow y \text{ OR } x \text{ OR } ((x \text{ OR } z) << 1)$,

$z' \leftarrow z \text{ OR } y \text{ OR } ((x \text{ OR } y) << 3)$.

Bu işlemler 384-bit pozisyonu için test edilmiştir ve tüm durum matrisini 1 biti ile doldurmak için maksimum 8 tura ihtiyaç olduğu kanıtlanmıştır.

Gimli sisteminin diferansiyel kriptanalizinin yapılması için doğrusal olmayan $x' \leftarrow y \,\&\, z$, $y' \leftarrow x \text{ OR } z$, $z' \leftarrow x \,\&\, y$ işlemlerine bakılmıştır. Ayrıca farkların yayılımın sağlanması için SP-Box işlemlerine yanına bazı doğrusal fonksiyonlar eklenmiştir. Girdi farkı (x, y, z), çıktı farkı (x', y', z') ile tanımlansın. Ayrıca T diferansiyel yolunun diferansiyel olasılığı $DP(T)$ ile gösterilirken, T diferansiyel yolunun ağırlığı $w=-log_2(DP(T))$ şeklinde hesaplanır. Gimli sisteminin 8 tura kadar olan diferansiyel yolları Tablo 3.3.1'de verilmiştir. Gimli'de 7 tur için diferansiyel kullanılarak, 2^{-188} olasılık ile 12 tur için diferansiyel yol oluşturulabilir.

Yardımcı Teorem 3.3.1 : f fonksiyonu ile elde edilen her mümkün olasılık için aşağıdaki eşitlikler sağlanır;

$x' \,\&\, (y \text{ OR } z) = 0$,

$y' \,\&\, (x \text{ OR } z) = 0$,

$z' \,\&\, (x \text{ OR } y) = 0$,

(x & y & NOT(z)) & NOT(x' ⊕ y') = 0,

(x & NOT(y) & z) & (x' ⊕ z') = 0,

(NOT(x) & y & z) & NOT(x' ⊕ y') = 0,

(x & y & z) & NOT(x' ⊕ y' ⊕ z') = 0.

Diferansiyel olasılık $DP((x, y, z) \xrightarrow{f} (x', y', z')) = 2^{-2.w(x\ OR\ y\ OR\ z)}$ ile hesaplanır.

Tablo 3.3.1 Gimli İndirgenmiş Tur Sayısı İçin Diferansiyel Yollar

Turlar	1	2	3	4	5	6	7	8
Ağırlık	0	0	2	6	12	22	36	52

Gimli sisteminin cebirsel derecesi 2 değerindedir fakat yinelemeli tur fonksiyonu ile bu derece sürekli artar. Örneğin 9. Turdaki cebirsel derece 266 değerindedir. Cebirsel bir atakta çıktıdaki bir bite odaklanılırsa cebirsel derecenin artışı çok daha yavaştır (Bernstein et al., 2017).

3.4 XOODYAK

XOODOO, özet fonksiyonu, MAC ve doğrulanmış şifreleme gibi kriptografik alanlarda kullanılabilir. 48-bayt XOODOO permütasyonunu kullanır. Beklenen güvenlik seviyesini sağlaması için 12 tur boyunca çalışmalıdır. Merkez fonksiyonu XOOFFF olarak adlandırılır. XOOFFF, XOODOO tabanlı Farfalle olarak da bilinir. Farfalle, genişletilmiş bit dizisini girdi olarak alan ve seçilen uzunlukta çıktı üreten anahtarlı kritografik fonksiyondur. Farfalle maliyet açısından kazançlıdır. Çünkü bir kez hesapladığı hiçbir yapıyı tekrar hesaplamaz, direkt kullanır. Ayrıca tanımlanması gereken diğer bir fonksiyon ise Deck fonksiyonudur. Deck fonksiyonu, anahtarlı kriptografik fonksiyonlar için tanımlanmıştır. Deck fonksiyonu seçilen uzunlukta bit dizisini girdi olarak alır ve seçilen uzunlukta rastgele sayı üretir. Anahtarsız kriptografik fonksiyonlar için kullanılan fonksiyon ise Dec ile adlandırılır. XOODYAK ise XOODOO permütasyonunun hafif-siklet uygulaması olarak sunulmuştur. XOODOO permütasyonunun kullanılabildiği tüm alanlarda uygulanabilmektedir (Daemen at al., 2018).

XOODOO permütasyonu tur sayısı k olmak üzere XOODOO[k] şeklinde gösterilir. Permütasyon *3x4x32* boyutlarına sahip bir durum matrisi ile ifade edilir. Yani permütasyon uzunluğu 384-bittir. Yinelemeli tur fonksiyonu işlemlerinden oluşur ve işlemlerini 384-bit uzunluğundaki tek bir permütasyon temelinde gerçekleştirir. XOODOO sisteminde tüm bit diziler Z_2^* ile tanımlanır. Boş dizi e ile gösterilir. M bit dizisinin uzunluğu m olmak üzere bit pozisyonları $M^{(0)}…M^{(m-1)}$ ile

gösterilir. XOODOO tur fonksiyonu beş tane adım işleminden oluşur. Bu adım işlemleri; karıştırma adımı θ, düzlemde sağa kaydırma ρ_r, düzlemde sola kaydırma ρ_l, tur sabitleri ile toplama adımı ι ve doğrusal olmayan adım χ olarak gösterilir. $i \in \{0,1,2\}$, S_i durum matrisi üzerindeki düzlemler, << sembolü x ve z eksenleri üzerinde sola ROT işlemi ve " ¯ " bit düzeyinde tümleyen olmak üzere;

$\theta : P \leftarrow S_0 \oplus S_1 \oplus S_2,$

$E \leftarrow P << (1,5) \oplus P << (1,14),$

$S_y \leftarrow S_y \oplus E, y \in \{0,1,2\}.$

$\rho_r : S_1 \leftarrow S_1 << (1,0),$

$S_2 \leftarrow S_2 << (0,11).$

$\iota : S_0 \leftarrow S_0 \oplus Tur\ Sabiti.$

$\chi : B_0 \leftarrow \bar{S}_1 \& S_2,$

$B_1 \leftarrow \bar{S}_2 \& S_0,$

$B_2 \leftarrow \bar{S}0 \& S1,$

$S_y \leftarrow S_y \oplus B, y \in \{0,1,2\}.$

$\rho_l : S_1 \leftarrow S_1 << (0,1),$

$S_2 \leftarrow S_2 << (2,8).$

XOODYAK, Duplex yapısına dayanmaktadır. 8-bit uzunluklu bit dizileri üzerinde çalışır. Bu sebeple tüm bit dizilerinin uzunluğu 8-bit olmalıdır. Motorist ve Cyclist olmak üzere iki adet modu vardır. Cyclist modu, kriptografik permütasyon ile ilişkili durumsal bir moddur. Cyclist modu; f permütasyonu, özet uzunluğu, girdi uzunluğu ve anahtarlı modda kullanılan çıktı uzunluğu ile parametrelendirilir. Cyclist modu, 2-bayt uzunluğunda bayrak bitleri kullanır. b permütasyon uzunluğu olmak üzere, max(özet uzunluğu, girdi uzunluğu, çıktı uzunluğu)+2 ≤ b olmalıdır. Cyclist modunun içinde fonksiyonlar bulunmaktadır. Özet fonksiyonu bu fonksiyonlardan yalnızca emilme ve sıkmayı kullanır. Keccak fonksiyonunda olduğu gibi emilme, blokların adım işlemleriyle yinelemeli olarak işlendiği, sıkma ise 128-bit özet değerinin üretildiği aşamadır.

Özet fonksiyonu aynı zamanda genişletilmiş çıktı fonksiyonudur. Yani istenilen uzunlukta özet değeri elde edilene kadar sıkma aşaması defalarca çalıştırılabilir. Ayrıca sıkma aşaması her çalıştırıldığında farklı uzunlukta çıktılar alınarak birleştirilebilir. XOODYAK özet fonksiyonunun güvenlik iddiası SHAKE128 ile aynıdır. h, özet çıktısının uzunluğu olmak üzere genel ataklara karşı

çakışma direnci min(8h/2,128)-bit, ön-görüntü ve ikinci ön-görüntü direnci min(8n,128)-bittir (Daemen et al., 2019).

3.5 Yarara ve Coral

Yarara, 128-bit anahtar, 128-bit genel mesaj numarası ve 128-bit etiket parametreleri ile beraber doğrulanmış şifreleme algoritmasıdır. Coral ise 256-bit çıktıya sahip özet fonksiyonudur. İki algoritmada da girdi ve çıktılar bayt olarak değerlendirilir. Yani girdi ve çıktı değerleri mutlaka 8-bitin tam katı olmalıdır. Permütasyon tabanlı bir sistemdir ve Sponge yapısı ile Sponge-Duplex yapısını kullanır. Permütasyonunu bir durum matrisi ile ifade eder ve durum matrisi *1x4x64* uzunluğunda üç-boyutlu bit matristir. Buradan, permütasyon uzunluğu 256-bit olup, kelime uzunluğu 64-bittir. Sponge yapısına dayandırıldığından dolayı $b = r + c$ eşitliği bu sistemde de geçerlidir. Coral özet fonksiyonunda r değeri 32-bit, c değeri 224-bit olarak kabul edilir.

Coral özet fonksiyonu, donanım ve yazılım için uygundur çünkü kullandığı işlemler yalnızca bit düzeyinde $\oplus$, NOT, & ve kaydırma işlemleridir. (Montes and Penazzi, 2019)'e göre durum matrisinin boyutu küçüktür ve çoğu işlemcide verimlilik gösterecek şekilde tasarlanmıştır. Permütasyonun tersinin bulunmasına hiçbir zaman ihtiyaç yoktur. Bu zaman açısından avantaj sağlamaktadır. Bunların yanında paralelleştirilemez bir yapıdadır.

Coral özet fonksiyonunun başlangıcında 256-bit durum matrisi 0 biti ile doldurulmuştur. Dolgulama kuralı mesajın son bloğunda en başa tek bir 1 biti ve r-bit uzunluğa tamamlamak için gerektiği kadar 0 biti eklenir. Sponge yapısını kullandığı için işlemler iki aşamadan meydana gelir. Bu aşamalar; emilme ve sıkma aşamasıdır. Girdi mesajına dolgulama uygulandıktan sonra r-bit bloklara ayrılır. Daha sonra her mesaj bloğu emilme aşamasına girer. Her emilme aşamasından sonra durum matrisine π_I permütasyonu uygulanır. Tüm mesaj blokları emilme aşamasından geçtikten sonra sıkma aşamasına başlanır. Sıkma aşaması özet çıktısının alındığı aşamadır. Coral, 256-bit özet çıktısı sağladığından ve her seferinde 32-bit çıktı alınabileceğinden dolayı π_I permütasyonu sıkma aşamasında sekiz kez uygulanmalıdır.

Coral özet fonksiyonunda tanımlanan dört permütasyon vardır. Bu permütasyonlar; π_I permütasyonu 10 kez, π_{AD} permütasyonu 6 kez, π_{AE} permütasyonu 6 kez ve π_F permütasyonu 6 kez olmak üzere uygulanmaktadır. Bu permütasyonların tümü 3 adımdan oluşan bir tur fonksiyonundan oluşur. İlk adım tur sabiti ile toplama adımıdır. Tüm işlemler bit düzeyinde mantıksal işlemler olduğundan dolayı toplama işlemi $\oplus$ işlemidir. İkinci adımı yer değiştirme adımıdır. Yer değiştirme adımı aynı zamanda sistemin doğrusal olmayan adımıdır ve 64 tane 4-bit S-Box yapısı kullanılarak uygulanır. S-Box yapısının cebirsel normal formu aşağıdaki şekilde tanımlanır;

w0, w1, w2, w3 durum matrisi üzerindeki dört kelime olmak üzere;

$w0 = w0 \oplus w1w2 \oplus w1w3 \oplus w2w3 \oplus w0w2w3 \oplus w1w2w3$,

$w1 = 1 \oplus w1 \oplus w2 \oplus w0w2 \oplus w0w3 \oplus w2w3 \oplus w0w1w2 \oplus w0w2w3$,

$w2 = w1 \oplus w3 \oplus w0w2 \oplus w2w3 \oplus w0w1w3 \oplus w0w2w3$,

$w3 = w0 \oplus w2 \oplus w3 \oplus w1w3 \oplus w0w1w2 \oplus w0w1w3.$

Kullandığı S-Box yapısına göre daha düşük cebirsel dereceli S-Boxlar vardır. Yarara ve Coral sisteminde kullanılan S-Box, diğer düşük dereceli S-Box yapılarına göre daha karmaşıktır. Son adım ise doğrusal yayılım adımıdır. Bu adım kendi içinde üç adımdan oluşur. Birinci ve üçüncü adım satır karıştırma işlemidir. ASCON sistemindeki yapıyı kullanır ve w, w1, w2, w3 kelimeler olmak üzere matematiksel olarak;

$w \leftarrow \sum_0(w) = w \oplus (w >> 19) \oplus (w >> 28),$

$w1 \leftarrow \sum_1(w1) = w1 \oplus (w1 >> 61) \oplus (w1 >> 39),$

$w2 \leftarrow \sum_2(w2) = w2 \oplus (w2 >> 1) \oplus (w2 >> 6),$

$w3 \leftarrow \sum_3(w3) = w3 \oplus (w3 >> 10) \oplus (w3 >> 17),$

şeklinde ifade edilir. İkinci adım ise sütun karıştırma işlemidir. Aşağıda verilen matris ile durum matrisinin soldan çarpılması ile yapılır. İşlemlerin tümü bit düzeyinde yani çarpma işlemi &, toplama işlemi $\oplus$ olarak kabul edilmelidir.

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Coral özet fonksiyonu Sponge yapısındaki güvenlik parametresi c ile hesaplanan c/2 yani 112-bit güvenlik sağlar. 4x4 S-Box yapısı kullandığı için güçlü kriptografik özelliklere sahiptir. Maksimum diferansiyel olasılığı ¼ olmakla birlikte diferansiyel kriptanalize karşı uygun güvenliği sağlamaktadır. Doğrusal kriptanalizde de yine bu oran ¼ olarak belirtilmiştir. Sistemin cebirsel derecesinin doğrusal olmayan adım ile beraber 3 olduğu görülmektedir. Ayrıca 3 dal sayısına sahiptir. Yayılım aktif S-Box sayısını arttırır ve 5 aktif S-Box'a kadar olan tüm girdiler için iki turda en az 20 aktif S-Box üretilir (Montes and Penazzi, 2019).

. Yarara ve Coral sisteminde kullanılan S-Box ile diferansiyel ve doğrusal olasılıklar 2^{-2} olarak belirtilmiştir. Diferansiyel ve doğrusal ataklara karşı direncin sağlanması için 64 tane aktif S-Box ihtiyacı vardır. 64 aktif S-Box ile çok düşük bir turda çığ etkisini gerçekleştirir. Cebirsel ataklar, kullanılan S-Box yapısının

derecesi ve tur sayısı ile ilgilenir. Bu sistemde S-Box yeterince büyük bir dereceye sahiptir. Bu sebeple permütasyonun cebirsel derecesi yeterlidir. Ayrıca tur sayısı ataklara karşı direnç sağlayacak şekilde seçilmektedir (Montes and Penazzi, 2019).

3.6 PHOTON-Beetle

PHOTON-Beetle sisteminin içerisinde hem doğrulanmış şifreleme hem de özet fonksiyonu vardır. İki işlemi de Sponge yapısını kullanmaktadır ve 256-bit Sponge permütasyonu üzerinde işlemlerini gerçekleştirmektedir. 256-bit permütasyon bir durum matrisi üzerinde gösterilir ve gösterim şekli 64 tane 4-bit uzunluğunda eleman şeklindedir. Bu matris *8x8* gibi düşünülmüştür. Sponge yapısına dayandığından dolayı blok uzunluğu olan r ve kapasite değeri c ile parametrelendirilir. Ayrıca dolgulama olarak *10** kuralını kullanmaktadır. Dört işlemden oluşan tur fonksiyonu ile yinelemeli şekilde çalışmaktadır ve toplam 12 turdan oluşmaktadır.

Tur fonksiyonu dört adımdan oluşur ve AES tur fonksiyonunun adımlarına benzemektedir (Daemen and Rijmen, 2001). Bu adımlar tur sabiti ile toplama, yer değiştirme işlemi, satırlarda kaydırma ve sütun karıştırmadır. Tur sabiti ile toplama adımında, her adım için daha önceden belirlenmiş tur sabiti değeri ile durum matrisi $\oplus$ işlemine tabi tutulur yani toplanır. Yer değiştirme adımında, 4-bit S-Box yardımı ile 64 tane 4-bit uzunluğundaki yapılar üzerinde yer değiştirme işlemi yapılır. Satırlarda kaydırma adımında, her satırda ROT işlemi yapılır. Son adım olan sütun karıştırma adımında ise, belirlenen bir matris ile durum matrisi çarpılır ve sütundaki elemanlar üzerinde doğrusal şekilde karıştırma işlemi yapılır. Bu çarpım, x^4+x+1 indirgenemez polinomuyla beraber GF(2^4) cismi üzerindedir.

PHOTON-Beetle sisteminde kullanılan ve ρ ile gösterilen bir matematiksel bileşen vardır. ρ doğrusal fonksiyonu girdi olarak, durum matrisi üzerindeki r-biti ve r-bit veya daha kısa uzunluktaki U girdi verisini alır. Eğer U verisi r-bit uzunluktan kısa ise r-bit uzunluğa tamamlanana kadar 0 biti eklenir. U girdi verisinin uzunluğuna sahip çıktı üretir. Çıktı üretirken karıştırılmış durum matrisinin basit $\oplus$ işlemini ve dolgulanmış girdi verisini kullanır. ρ matematiksel bileşeninin tersi alınabilir.

PHOTON-Beetle özet fonksiyonunu girdi olarak seçilen uzunluktaki mesajı ve 256-bit etiket değerini girdi olarak alır. Dolgulanmış girdi mesajının ilk bloğu 128-bit diğer blokları r-bit bloklara ayrılır ve başlangıçta 0 bitleri ile dolu olan durum matrisinin ilk 128-biti ile $\oplus$ işlemine tabi tutulur daha sonra durum matrisi PHOTON permütasyonundan geçirilir. Diğer bloklar r-bit uzunlukta olduğundan durum matrisinin 32-biti ile $\oplus$ işlemine tabi tutularak PHOTON permütasyonuna girer. Tüm mesaj blokları için bu işlem uygulanmaktadır. Bu işlemlerin uygulandığı aşama emilme aşaması olarak adlandırılır. Sponge yapısından dolayı

ikinci aşama olan sıkma aşaması iki kez çalışmaktadır. Çünkü blok uzunluğu 128-bittir. 256-bit özet değeri üretmektedir (Bao et al., 2019).

(Bao et al., 2019)'da Photon-Beetle özet fonksiyonu için blok uzunluğu değeri 32-bit önerilmiştir. PHOTON-Beetle sisteminin özet fonksiyonu için çakışma güvenliği 112-bit, ön-görüntü için 128-bittir. Çakışma güvenliği hesaplanırken, saldırganın q adet permütasyon çağırdığı varsayılır. Saldırganın amacı, tamamı 0 bitleri ile dolu olan ilk durum matrisine ulaşmaktır. Saldırgan girdi ve çıktı üzerinde sorgular oluşturabilir. Ayrıca özet değerinde bir

çakışma yakalamak için blok kısmındaki bitleri kendisi ayarlayabilir. Bu durumun olasılığı $q^2 / 2^{256-r}$ ile sınırlandırılmıştır. Ön-görüntü olasılığı ise $q / 2^{128}$ ile sınırlandırılmıştır (Bao et al., 2019).

3.7 Sneiken ve Sneikha

Sneik sistemi, Sneiken doğrulanmış şifrelemesi ve Sneikha özet fonksiyonundan oluşur. Üç tane Sneiken tipi vardır. Bunlar; Sneiken128, Sneiken192 ve Sneiken256'dır. Özet fonksiyonu Sneikha ise çıktı boyutuna göre Sneikha256 ve Sneikha384 olmak üzere ikiye ayrılır. Bunların haricinde genişletilmiş çıktı fonksiyonuna sahiptir ve Sneigen olarak adlandırılır. Sneik sisteminde permütasyon uzunluğu 512-bittir. Permütasyon durum matrisi ile ifade edilir ve durum matrisinin boyutu *1x16x32* şeklindedir. Sneik durum matrisinin kelime uzunluğu 32-bittir. Sneik tüm işlemlerini bu 512-bit permütasyon üzerinde gerçekleştirir. (Saarinen, 2019)'e göre hiçbir işlem 64-bayttan fazla rastgele erişimli bellek (Random Access Memory - RAM) gerektirmez.

Sneik permütasyonu π ile gösterilir. Sneik permütasyonu yüksek yayılım özelliğine sahiptir. (Saarinen, 2019)'e göre esnek tasarımı sayesinde birçok donanımda verimli şekilde çalışabilmektedir. Permütasyonun kolayca tersi alınabilir fakat bu hiçbir zaman gerekmemektedir. Sneik permütasyonu doğrusal olmayan geri beslemeli kaydırma kaydı (non-Linear Feedbeck Shift Register - NLFSR) gibi görünür. $|w|$ kelime sayısı, s_i aritmetik adımlar, $\boxplus$ mod 2^{32} tabanında toplama işlemi, << sola ROT, d alan ayırıcı, $i \geq |w|$ ve A durum matrisi olmak üzere;

$s_1 = A[i-|w|] \oplus d[i]$,

$s_2 = s_1 \boxplus A[i-1]$,

$s_3 = s_2 \oplus (s_2 \lll 24) \oplus (s_2 \lll 25)$,

$s_4 = s_3 \oplus A[i-2]$,

$s_5 = s_4 \boxplus A[i-|w|+2]$,

$s_6 = s_5 \oplus (s_5 \lll 9) \oplus (s_5 \lll 17)$,

$s_7 = s_6 \oplus A[i-|w|+1]$,

$A[i] = s_7$,

işlemleri Sneik permütasyonunu oluşturur. Sneik sistemi her turda farklı bir değer olmak üzere tur sabitlerine sahiptir. Tur sabitleri alan ayırıcı olarak kullanılır. Sneik sistemi matematiksel Sponge yapısına dayanmaktadır. 512-bit permütasyon uzunluğu b ile gösterilir. Blok uzunluğu r ve kapasite c değerinin toplamı permütasyon uzunluğunu vermektedir. Sneik işlemleri emilme ve sıkma aşamalarından meydana gelir (Saarinen, 2019). Sneikha özet fonksiyonunun parametre ve güvenlik değerleri Tablo 3.7.1'de verilmiştir.

Tablo 3.7.1 Sneikha Özet Fonksiyonlarının Parametre ve Güvenlik Değerleri

Özet Fonks.	Özet Boyutu	Blok Boyutu	Turlar	Güvenlik
Sneikha256	256	256	8 tur	2^{128}
Sneikha384	384	128	8 tur	2^{128}

3.8 SIV-TEM-PHOTON

SIV-TEM-PHOTON sistemi, PHOTON sistemine benzemektedir. TEM-PHOTON sisteminin anahtar ve ince ayar (tweak) değeri sabit alınarak, Sponge yapısı temelli bir permütasyon ile özet fonksiyonu elde edilir. (Bao et al., 2019)'a göre hem doğrulanmış şifreleme hem de özet fonksiyonu donanım açısından PHOTON kadar hafif ve güvenlidir.

TEM (Tweakable Even-Mansour) yapısı sabit bir permütasyondan blok şifre oluşturmak için kullanılan bir yapıdır. K_1, K_2 anahtarlar ve P permütasyon olmak üzere, x metninin bu yapı ile şifrelenmesi $K_2 \oplus P(x \oplus K_1)$ şeklinde gösterilir. Bu tek bir adımın gösterimidir. Güvenliği arttırmak için r kere yinelenen bir yapı kullanılır. Bu ise $(K_{r+1} \oplus P_r (K_{r-1} \oplus P_{r-1}(\cdots(P_1(x \oplus K_1))))$ ile gösterilir. Aynı yapının tek anahtarlı ve üç kez yinelemeli versiyonu da vardır. TEM-PHOTON tasarımında ihtiyaçlar 256-bit blok uzunluğu ile beraber Tweakable blok şifre ve 128-bit anahtardır. TEM yapısında 128-bit aynı anahtar uç uca eklenir ve yeni anahtar oluşturulur. Ayrıca tweak değeri, 128-bit 0 bitinin sonuna etiket değerinin birleştirilmesi ile oluşturulur. Aynı anahtarın art arda eklenmesi ile yeni anahtar oluşturulması, güvenlik açısından herhangi bir saldırıda kullanılamaz.

PHOTON permütasyonu, AES sistemindeki fonksiyonlara benzer fonksiyonlar kullanır (Daemen and Rijmen, 2001). Beş tane permütasyona sahiptir. Bu permütasyonların yalnızca boyutları birbirinden farklıdır ve 100, 144, 196, 256 ve 288 bittir. Genelde kullanılan permütasyonu 256-bit uzunluğundadır. Permütasyon durum matrisi olarak ifade edilir ve 256-bit durum matrisi üzerinde

8x8x4 şeklinde gösterilmiştir. Yinelemeli tur fonksiyonuna sahiptir ve orijinal PHOTON yapısında tur sayısı 12 iken SIV-TEM-PHOTON 20 turdan oluşur. Her turda; tur sabiti ile toplama, yer değiştirme, satırda kaydırma ve sütunlar üzerinde karıştırma işlemi yapılır.

SIV-TEM-PHOTON özet fonksiyonu Sponge yapısına dayanmaktadır. Emilme ve sıkma aşamalarından oluşmaktadır. Emilme aşamasının ilk adımında blok uzunluğu 128-bit, diğer adımlarında blok uzunluğu 32-bittir. Sıkma aşamasında ise blok boyutu 32-bittir. Permütasyon boyutu, blok uzunluğu ve kapasite değerlerinin toplamından oluşur. Özet fonksiyonun dolgulama kuralı, 10* ile gösterilir.

SIV-TEM-PHOTON yapısı AES işlemleri ile permütasyona dayalı çalıştığından güvenlik sınırları (Daemen and Rijmen, 2001)'de sunulan AES sistemine de dayanır. Çok hafif bir permütasyon kullandığından donanım için hafiftir ve ek maliyete gerek kalmadan tur sayısı arttırılabilecek şekilde tasarlanmıştır. (Bao et al., 2019)'da özet fonksiyonunun çakışma güvenliği 112-bit, ön-görüntü güvenliği 128-bit ve ikinci ön-görüntü güvenliği 112-bit olarak belirtilmiştir. SIV-TEM-PHOTON, TEM şifresi altında dört tane 5 tur permütasyona ayrılmıştır. AES benzeri fonksiyonlara göre PHOTON'un 256-bit permütasyonunun dört turu 81 aktif S-Box sağlar (Daemen and Rijmen, 2001). Bunun anlamı, her dört tur için diferansiyel olasılıklar ve doğrusal olasılıkların üst sınırının 2^{-128} olmasıdır. Bu sebeple, beş turda kesinlikle daha fazla S-Box aktif olacağından olasılıkların hiçbiri 2^{-128} değerinden yüksek olamaz.

SIV-TEM-PHOTON sistemine diferansiyel kriptanaliz yapıldığında, 2 tura indirgenmiş permütasyon ile diferansiyel yollar elde edilmiştir. Mümkün 2 ardışık tur için diferansiyel yollara dayanarak aktif S-Box yapısının minimum sayısına ulaşılabilmektedir. Ayrıca, 4 tane 5 tur permütasyon için aktif S-Box sayıları sırasıyla en az 30, 29, 30, 29 olarak hesaplanmıştır. Toplamda en az 118 tane aktif S-Box oluşmaktadır. Bu ise yeterli bir sayıdır (Bao et al., 2019).

3.9 CLX

CLX sistemi, hem doğrulanmış şifreleme hem de özet fonksiyonuna sahiptir. Doğrulanmış şifrelemesi Duplex-Sponge tabanlı iken, özet fonksiyonu Sponge yapısına dayanmaktadır. CLX permütasyonu $P_{160+x,n}$ ile gösterilir. Permütasyon durum matrisi (160+x)-bit uzunluğundadır ve n turdan oluşur. Permütasyonun i. turunda (160+x)-bit NLFSR ile durum matrisi güncellenir. 160-bit ve (160+x)-bit permütasyon için s_i durum matrisinin bitleri olmak üzere güncelleme işlemi aşağıdaki şekilde yapılır.

160-bit için;

for x=0

geribesleme=$s_0 \oplus s_{35} \oplus(NOT(s_{93}\&s_{106}))\oplus s_{127}$

for i=0 to 158

$s_i=s_{i+1}$

s_{159}=*geribesleme*

end

(160+x)-bit için;

for x=0

geribesleme=$s_0 \oplus s_x \oplus s_{35+x} \oplus(NOT(s_{93+x}\&s_{106+x}))\oplus s_{127+x}$

for i=0 to (160+x)-2

$s_i=s_{i+1}$

$s_{(160+x)-1}$=*geribesleme*

end

CLX özet fonksiyonunun girdi mesajı uzunluğu 2^{50} bitten kısa olmalıdır. Özet fonksiyonu 256-bit uzunluğunda özet değeri üretir. Özet fonksiyonunun permütasyonu $P_{288,n}$ şeklinde gösterilir. Bu gösterimde 288 permütasyon uzunluğu, n ise tur sayısıdır. Aynı zamanda permütasyon uzunluğu (160+288)-bite genişletilir. Başlangıçta $P_{288,n}$ permütasyonu ile durum matrisi güncellenir. Bu işlem aşağıdaki algoritma ile açıklanabilir.

for x>0

geribesleme=s0 $\oplus$ s19 $\oplus$ s128 $\oplus$ s163 $\oplus$ (NOT (s221&s234)) $\oplus$ s255

for i=0 to 286

si=si+1

s287=geribesleme

end

Başlangıç aşamasında durum matrisinin tamamı 0 biti ile doldurulur. Durum matrisinin 196. biti 1 biti ile değiştirilir. Son olarak durum matrisi $P_{288,1024}$ permütasyonu ile güncellenir. Girdi mesajı 32-bit bloklara ayrılır ve ilk mesaj bloğu durum matrisi ile $\oplus$ işlemine tabi tutulur. Daha sonra durum matrisi $P_{288,2560}$ permütasyonu ile güncellenir. Bu işlem tüm mesaj blokları için uygulanır. Son

aşama özet değerinin üretildiği aşamadır. Sekiz kez tekrarlanan işlemler ile özet değeri üretilir. Durum matrisinin son otuz iki biti çıktı olarak alınır. Sonra yüz doksan altıncı biti 1 biti ile değiştirilir ve $P_{288,256}$ permütasyonu ile durum matrisi güncellenir. 32-bit çıktılar birleştirilerek özet değeri üretilir(Wu and Huang, 2019).

CLX özet fonksiyonu, 112-bit çakışma, 112-bit ön-görüntü ve 112-bit ikinci ön-görüntü direncine sahiptir. CLX özet fonksiyonunun güvenlik analizinde seçilen girdi farkları ve seçilen çıktı farkları incelenmiştir. CLX permütasyonu uygulandığında optimum fark, durum matrisi üzerindeki tek bit değişiminde meydana çıkmaktadır. Bu tip farklar için en büyük fark olasılıkları hesaplanmış ve Tablo 3.9.1'de gösterilmiştir. Ayrıca permütasyonun doğrusal sapmaları da Tablo 3.9.2'de gösterilmiştir.

Tablo 3.9.1 CLX Permütasyonunun Diferansiyel Özellikleri

Permütasyon	Tur	Olasılık
$P_{160,n}$	1280	2^{-160}
$P_{192,n}$	1280	2^{-180}
$P_{256,n}$	1280	2^{-95}
$P_{320,n}$	1920	2^{-99}

Tablo 3.9.2 CLX Permütasyonunun Doğrusal Sapmaları

Permütasyon	Tur	Sapma
$P_{160,n}$	480	2^{-31}
$P_{192,n}$	480	2^{-33}
$P_{256,n}$	448	2^{-31}
$P_{320,n}$	448	2^{-27}

CLX özet fonksiyonu 288-bit permütasyondur. Bu permütasyon üzerinde yukarıda anlatılan fark hesaplanmıştır. 2048 tur için bu farkın olasılığı 2^{-307} olarak belirtilmiştir. Bu sebeple her 32-bit uzunluğundaki mesaj bloğu için umulan çakışma direncinin sağlanması için 2560 tura ihtiyaç vardır (Wu and Huang, 2019).

3.10 DryGASCON

DryGASCON sistemi, DrySponge ve ASCON tabanlı doğrulanmış şifreleme ve özet fonksiyonundan oluşur. DrySponge yapısı Duplex-Sponge yapısına benzeyen bir yapıdır. Duplex-Sponge yapısından farklı noktaları, girdi

mesajını durum matrisi ile birleştirme ve durum matrisinden çıktı oluşturmaktır. DryGASCON yapısı fiziksel atakların algoritmik düzeye indirilmesini engellemek için oluşturulmuştur. Sponge yapısı uygulanabilen tüm alanlarda DrySponge yapısı da uygulanabilir. Sponge yapısında olduğu gibi parametre olarak r blok uzunluğu ve c kapasite değerlerini kullanır. Ayrıca emilme ve sıkma aşamaları ile şifreleme işlemlerini gerçekleştirir.

DrySponge yapısının ana fikri kullandığı kullanılan *F* fonksiyonunu üç aşamaya bölmektir. Bu aşamalar karıştırma, çekirdek ve biriktirme olarak adlandırılır. *F* fonksiyonu girdinin bitlerinden oluşan durum matrisini ve alan ayırıcıyı girdi olarak alır. Çıktı olarak ise durum matrisindeki *r*-bit uzunluğundaki kısmı verir. Alan ayırıcıyı ve mesaj bloklarını durum matrisine dahil etme işlemi karıştırma aşamasında gerçekleşir. Çıktı üretmek için ise çekirdek ve biriktirme aşamaları gerçekleşir. Çıktı üretmek için gerçekleşen çekirdek ve biriktirme aşamalarına G fonksiyonu denir.

Çekirdek aşaması, doğrusal ve doğrusal olmayan işlemlerden oluşan rastgele bir permütasyon olarak belirtilmektedir. Bu aşama yeteri kadar uygulandığında, yayılımda büyük etkileri olmaktadır. Girdideki tek bir bit değişiminde çıktının en az yarısının değiştiği görülmektedir. Biriktirme aşamasında temel olarak yapılan işlem belirlenen bir değer ile tüm girdi bloklarının $\oplus$ işlemine tabi tutulmasıdır. Karıştırma aşamasında ise kendi S-Box yapısını kullanmaktadır.

DryGASCON özet fonksiyonunda, öncelikle girdi mesajı 10* kuralı ile dolgulanır. Daha sonra dolgulanmış mesaj belirlenen uzunlukta bloklara ayrılır. Daha sonra F ve G fonksiyonları ile özet değeri elde edilir. Seçilen uzunlukta özet değeri elde edilebilir. Girdi mesajının uzunluğu 2^{64}-bitten kısa olmalıdır. Bu durumda DryGASCON özet fonksiyonu 128-bit güvenlik sağlar. Eğer mesaj uzunluğu 2^{64}-bitten uzun, 2^{128}-bitten kısa ise 256-bit güvenlik sağlar. DryGASCON sisteminde kullanılan S-Box yapısının hem diferansiyel hem de doğrusal dal sayısı 3 değerindedir. Cebirsel derecesi yalnızca 2 değerindedir. Yayılım karakteristiği ise 0 değerindedir (Riou, 2019).

3.11 KNOT

KNOT sistemi tüm işlemlerini bir permütasyon üzerinde yinelemeli olarak gerçekleştirir. KNOT permütasyon işlemleri tur sabiti ile toplama, sütunlarda yer değiştirme ve satırlarda kaydırma olmak üzere üç tanedir. Kullanılan permütasyon P_b ile gösterilir. b ile tanımlanan permütasyon genişliğidir ve 256-bit, 284-bit veya 512-bit olabilir. Permütasyon bir durum matrisi üzerinde gösterilir ve bu durum matrisinin boyutu 4 x b/4 olarak belirtilmiştir. Tur sabitleri d-bit uzunluğundadır ve d, 6-bit, 7-bit veya 8-bit olabilir. Tur sabitlerinin başlangıç değeri RC[0]=0x1 ile ifade edilir. d değerinin uzunluğuna göre değişen üç farklı LFSR kullanır. d=6 için her turda 6-bit uzunluğundaki tur sabiti 1-bit sola kaydırılır. Tur sabitinin ilk biti, beşinci bit ile dördüncü bitin $\oplus$ işlemine tabi tutulması ile oluşur. d=7 için her turda

7-bit uzunluğundaki tur sabiti 1-bit sola kaydırılır. Tur sabitinin ilk biti, beşinci bit ile altıncı bitin $\oplus$ işlemine tabi tutulması ile oluşur. d=8 için her turda 8-bit uzunluğundaki tur sabiti 1-bit sola kaydırılır. Tur sabitinin ilk biti, yedinci bit, beşinci bit, dördüncü bit ve üçüncü bitin $\oplus$ işlemine tabi tutulması ile oluşur. d değerinin uzunluğu tur sayısına bağlıdır.

Sütunlarda yer değiştirme işleminde sütunlar üzerinde paralel olarak 4-bit S-Box kullanılır. KNOT sisteminde kullanılan S-Box S:$F_2^4 \rightarrow F_2^4$ ile gösterilir. Satırlarda kaydırma işleminde, her satır üzerinde ROT işlemi yapılır. İlk satır üzerinde kaydırma işlemi yapılmaz. İkinci satır üzerinde c_1-bit, üçüncü satır üzerinde c_2-bit, son satır üzerinde c_3-bit ROT yapılır. c uzunluğu, b uzunluğuna bağlı olarak değişmektedir. b değeri 256-bit olduğunda c_1,c_2,c_3 sırasıyla 1,8,25; b değeri 384-bit olduğunda c_1,c_2,c_3 sırasıyla 1,8,55; b değeri 512-bit olduğunda c_1,c_2,c_3 sırasıyla 1,16,25 değerlerine sahiptir.

KNOT özet fonksiyonu, girdi mesajını alır ve özet çıktısı üretir. Genişletilmiş Sponge yapısında, sıkma aşaması için blok uzunluğundan farklı bir r değeri seçili ve r' olarak gösterilir. Bu seçim r'>r şeklinde yapılır. Bu yapının avantajı sıkma aşamasında harcanan performansın azaltılmasıdır. Fakat bununla beraber ön-görüntü direnci düşer (Guo et al., 2011). KNOT özet fonksiyonu genişletilmiş Sponge yapısını kullanır. Başlangıçta girdi mesajı r-bit bloklara ayrılır. Eğer son mesaj bloğunun tüm bitleri dolu değilse son bloğa dolgulama işlemi yapılır. Dolgulama, boş bit pozisyonlarının ilkine 1 biti diğer pozisyonlara ise 0 biti koyularak yapılır. Daha sonra özet değeri üretmek için emilme ve sıkma aşamalarını geçilir. Emilme aşamasında, dolgulanmış mesaj r-bit uzunluklu bloklara ayrılır. Daha sonra başlangıçta tamamen 0 biti ile doldurulmuş durum matrisi ile ilk mesaj bloğu $\oplus$ işlemine tabi tutulur ve durum matrisi KNOT permütasyonu ile belirlenen tur sayısı kadar işlenir. Bu işlemler her mesaj bloğu için tekrarlanır. Tüm mesaj blokları işlendikten sonra sıkma aşamasına geçilir. Bu aşamada n-bit özet değeri üretilir ve sıkma aşamasının her adımında r'-bit çıktı olarak alınır.

Tablo 3.11.1 KNOT Özet Fonksiyonu Parametre Değerleri

İsim	Çıktı	Permütasyon	Kapasite	R	r'	Tur Sayısı
KnotÖzet(256,256,32,128)	256-bit	256-bit	224-bit	32-bit	128-bit	68
KnotÖzet(256,384,128,128)	256-bit	384-bit	256-bit	128-bit	128-bit	80
KnotÖzet(384,384,48,192)	384-bit	384-bit	336-bit	48-bit	192-bit	104
KnotÖzet(512,512,64,256)	512-bit	512-bit	448-bit	64-bit	256-bit	140

KNOT özet fonksiyonunun güvenliği genişletilmiş Sponge yapısının güvenliğine dayandırılmaktadır (KNOT: algorithm specifications and supporting document, 2019; Guo et al., 2011; Bertoni et al, 2011). Bu sebeple ön görüntü direnci $\min\{2^{\min\{\text{çıktı},b\}},\max\{2^{\min\{\text{çıktı},b\}-r'},2^{c/2}\}\}$, ikinci ön görüntü direnci $\min\{2^{\text{çıktı}},2^{c/2}\}$ ve çakışma direnci $\min\{2^{\text{çıktı}},2^{c/2}\}$ ile hesaplanmaktadır. Bu şartlarda KNOT özet fonksiyonlarının sağladığı güvenlik değerleri tabloda gösterilmiştir.

Tablo 3.11.2 KNOT Özet Fonksiyonu Güvenlik Değerleri

İsim	Ön-Görüntü	2 Ön-Görüntü	Çakışma
KnotÖzet(256,256,32,128)	128-bit	112-bit	112-bit
KnotÖzet(256,384,128,128)	128-bit	128-bit	128-bit
KnotÖzet(384,384,48,192)	192-bit	168-bit	168-bit
KnotÖzet(512,512,64,256)	256-bit	224-bit	224-bit

x_i girdi bitleri, y_i çıktı bitleri olmak üzere, KNOT sisteminde kullanılan S-Box yapısının cebirsel normal formu aşağıdaki gibidir. KNOT sistemindeki S-Box yapısının cebirsel derecesi yalnızca 2 değerindedir.

$y_0 = x_0x_1 + x_2 + x_0x_2 + x_3 + x_1x_3 + x_0x_1x_3 + x_2x_3,$

$y_1 = x_1 + x_2 + x_0x_3 + x_2x_3 + x_1x_2x_3,$

$y_2 = 1 + x_0 + x_1 + x_2 + x_1x_2 + x_3,$

$y_3 = x_1 + x_0x_1 + x_2 + x_3.$

(Matsui, 1994)'de sunulan DES algoritmasının diferansiyel ve doğrusal yollarını arama metodu kullanılarak 3 KNOT permütasyonu için en iyi diferansiyel yollar hesaplanmıştır. i = 1, 2, 3… için b-bit KNOT permütasyonunda en iyi diferansiyel yolların ağırlığı $w_{b,i}$ ile gösterilir. Bu durumda aşağıdaki sonuçlara ulaşılmıştır.

$w_{256,i} = w_{384,i} = w_{512,i},\ 1 \le i \le 11.$

$w_{256,i} = w_{256,i-3} + 16,\ 12 \le i \le 49.$

$w_{384,i} = w_{384,i-3} + 16,\ 12 \le i \le 73.$

$w_{512,i} = w_{512,i-3} + 16,\ 12 \le i \le 97.$

256-bit KNOT permütasyonu için 49-tur diferansiyel yolun en iyi olasılığı 2^{-258}, 384-bit KNOT permütasyonu için 73-tur diferansiyel yolun en iyi olasılığı 2^{-386} ve

512-bit KNOT permütasyonu için 97-tur diferansiyel yolun en iyi olasılığı 2^{-514} olarak belirtilmiştir.

KNOT permütasyonunun doğrusal kriptanalizinde 3 KNOT permütasyonu için en iyi doğrusal yollar hesaplanmıştır. i = 1, 2, 3… için b-bit KNOT permütasyonunda en iyi doğrusal yolların ağırlığı $w_{b,i}$ ile gösterilir. Bu durumda aşağıdaki sonuçlara ulaşılmıştır.

$w_{256,i} = w_{384,i} = w_{512,i}$, $1 \leq i \leq 8$.

$w_{256,i} = w_{256,i-1} + 3$, $9 \leq i \leq 40$.

$w_{384,i} = w_{384,i-1} + 3$, $9 \leq i \leq 40$.

$w_{512,i} = w_{512,i-1} + 3$, $9 \leq i \leq 40$.

256-bit KNOT permütasyonu için 40-tur doğrusal yolun en iyi olasılığı 2^{-113}, 384-bit KNOT permütasyonu için 40-tur doğrusal yolun en iyi olasılığı 2^{-113} ve 512-bit KNOT permütasyonu için 40-tur doğrusal yolun en iyi olasılığı 2^{-113} olarak belirtilmiştir. Yani permütasyon uzunluğu ne olursa olsun olasılık aynı kalmaktadır. Tur sayısı arttıkça olasılık azalmaktadır. Bu kriptanalizlerin dışında permütasyon uzunluğuna göre ayırt edici uzunlukları Tablo 3.11.3'de verilmiştir (KNOT: Algorithm Specifications and Supporting Document,2019).

Tablo 3.11.3 KNOT Sistemi Permütasyon Uzunluğuna Göre Ayırt Edici Uzunlukları

	256-bit için	384-bit için	512-bit için
Diferansiyel Kriptanaliz	48	72	96
Doğrusal Kriptanaliz	44	66	87

3.12 HERN ve HERON

HERN doğrulanmış şifreleme ve HERON özet fonksiyonu bit tabanlı sistemlerdir. İki sistem de her seferinde 1 bit işlenecek şekilde paralel 32 işlem yaparak şifreleme işlemlerini gerçekleştirir. HERON özet fonksiyonu girdi olarak seçilen uzunluktaki mesajı alır ve 256-bit uzunluğunda özet değeri üretir. HERON özet fonksiyonunda girdi mesajı en fazla 2^{64}-1-bit uzunluğunda olmalıdır.

HERN durum değişkenleri iki bileşenden oluşur. Bu bileşenlerin ilki, dört 64-bit değişkene LFSR uygular. Bu değişkenler s^0, s^1, s^2 ve s^3 ile gösterilir. Diğer bileşen ise doğrusal olmayan geri besleme bitlerini işleyen iki tane 1-bit uzunluğundaki a ve b değişkenleridir. Bu değişkenler sırasıyla girdi bitini işlemek ve çıktı bitini üretmek için kullanılır. HERN sistemi tek başına özet fonksiyonu üretmek için çok küçüktür. Bu yüzden kendi bileşenlerinin haricinde 512-bit ara durum değişkenleri kullanılır. Yeni oluşan bu yapıya ise HERON adı verilir. s^0, s^1,

s^2 ve s^3 işlemleri çekirdek adımıdır. a ve b ile yapılan işlemler ise adda ve addb ile gösterilir. Çekirdek adımında iki doğrusal olmayan işlem kullanılarak 8-bit 1-bite düşürülür. Bu işlemler SB ve SB' ile gösterilir.

$SB(x_0,y_0,x_1,y_1,x_2,y_2,x_3,y_3) = 1 \oplus x_0\&y_0 \oplus x_1\&y_1 \oplus x_2\&y_2 \oplus x_3\&y_3,$

$SB'(x_0,y_0,x_1,y_1,x_2,y_2,x_3,y_3) = x_0\&y_2 \oplus y_0\&y_3 \oplus x_1\&x_3 \oplus y_1\&x_2.$

adda işlemindeki a değeri s^0 değerinin son biti ile s^2 değerinin son bitinin $\oplus$ işlemine tabi tutulması ile elde edilir. addb işlemindeki b değeri ise s^1 değerinin son biti ile s^3 değerinin son bitinin $\oplus$ işlemine tabi tutulması ile elde edilir.

Tur fonksiyonları, beş adımda açıklanmıştır. İlk üç adım HERN doğrulanmış şifrelemesine aittir. Diğer iki adım ise HERON özet fonksiyonunun adımlardır. HERON özet fonksiyonunun adımlarından biri ara durum matrisinin güncellenmesidir. Diğeri ise girdi mesajı bitlerinin işlenmesidir. HERON özet fonksiyonunda başlangıçta durum matrisi belirlenen başlangıç vektörünün bitleri ile doldurulur. Girdi mesajı işlenmeden önce 32-bit uzunluklu bloklara ayrılır. Eğer son mesaj bloğunda eksik bit pozisyonları var ise bu pozisyonlar 0 biti ile doldurulur. Bu işleme dolgulama denir. Dolgulanmış mesaj bloklarının her birine HERON özet fonksiyonu işlemleri yinelemeli olarak uygulanır.

Tablo 3.12.1 HERON Özet Fonksiyonu Güvenlik Değerleri

	Ön-Görüntü	Çakışma
HERON	256-Bit	128-Bit

HERON özet fonksiyonunun güvenliği, çakışma problemine indirgenmiştir. Yani m ve m' iki farklı mesaj olmak üzere, bu iki farklı mesajın işlenmiş durum matrisi olarak çıktısı aynı olursa çakışma meydana gelmiş demektir. Bu durum bir denklem sistemi olarak ifade edilebilmektedir. Elde edilen denklem sistemini çözmek için bilinen tek yöntem ise diferansiyel kriptanalizdir. Diferansiyel kriptanaliz, en düşük maliyetli diferansiyel yolu bulmak için kullanılan bir yöntemdir. (Ye et al., 2019)'de adda fonksiyonunun girdisi a ile gösterilmiştir. *A*, tüm mesaj işlem basamakları için a değerlerinden oluşan dizidir. Diferansiyel yol ise *(A,A')* ikilisi ile gösterilir. *A'* ile belirtilen değerler *A* dizisinin SB' işleminin çıktılarını ifade etmektedir ve SB', aktif olmayan S-Box yapıları için 0 değerine sahiptir. Bir yolun maliyeti, bu yolu gerçekleştirmek için iş faktörünü belirtirken, maliyet q olduğunda, iş faktörünün 2^q olduğu anlamına gelir.

Bir diferansiyel yolun maliyeti hesaplanırken dikkat edilen noktalar şu şekilde belirtilmiştir: Sabit pozisyondaki her aktif S-Box, maliyeti iki katına çıkarır ve bu mesaj pozisyonundaki aktif olmayan her S-Box, maliyeti bir azaltır. Farklı

bir izin maliyeti, tüm alt izlerinin maliyetlerinin maksimumudur. Diferansiyel yolun maliyeti, tüm alt yolların maliyetinin maksimumudur. *A* akışı, *A'* = 0 olduğunda, *B* doğrusal matrisinin en küçük polinomunun küçük bir katı olarak seçilir.

$$B = \begin{pmatrix} B0 & Z0 & 0 & 0 \\ 0 & B1 & Z1 & 0 \\ 0 & 0 & A2 & Z2 \\ Z3 & 0 & 0 & B3 \end{pmatrix}$$

e_j ile tanımlanan eleman j. elemanı 1 olan birim vektördür. B_i ve Z_i, F_2 cismi üzerindeki 64x64 boyutuna sahip matrislerdir. i = 0,…,3 olmak üzere B_i matrisindeki tüm satırların j. elemanı e_{j+1}, B_i matrisinin 63. Satırı ise $e_0 \oplus e_{31} \oplus e_{32}$ ($e_0 \oplus e_{28} \oplus e_{30}$, $e_0 \oplus e_{22} \oplus e_{27}$, $e_0 \oplus e_8 \oplus e_{19}$) şeklinde ifade edilir. Z_i matrisinin ilk 6 satırı sıfır vektörüdür. 63. satırı ise e_{13} (e_1, e_{26}, e_{31}) şeklindedir. Durum matrisinin güncellenmesi ise S^T, 6x1 boyutlu *S* vektörünün transpozu olmak üzere $S^T \leftarrow B \,.\, S^T$ ile gösterilmiştir.

Küçük kat, neredeyse tüm S-Box yapılarını aktif hale getirir. Böylece, tüm alt yollar geçerli olur. Bu alt yollardaki aktif S-Box sayısı yaklaşık olarak *(l-c)* ile hesaplanır. Burada *l*=256 değerindedir, *c* ise sabit bir değeri ifade eder. Bu alt yolların maliyeti, |M| ilk kodlanan mesaj bloğu ve $p = |M| / (|M|+32)$ olmak üzere $2 \,.\, (l - c) \,.\, (1 - p)$ şeklinde hesaplanır. Herhangi bir M mesajının uzunluğu *l* değerinin yaklaşık yarısı kadardır bu sebeple maliyet $2 \,.\, (l / 2) \,.\, (1 - p)$ şeklinde hesaplanabilir. Buradan elde edilen maksimum maliyet tüm yolların maliyetini gösterdiğinden maliyet $2 \,.\, (l - c) \,.\, (1 - p)$ şeklinde ifade edilir. Ayrıca HERON özet fonksiyonu, Sponge yapısına dayandığından uzunluk genişletme atağına karşı duyarlı değildir (Ye et al., 2019).

3.13 Subterranean 2.0

Subterranean şifreleme şeması hem özet fonksiyonu hem de akan şifre tekniği olarak 1992 yılında önerilmiştir (Claesen et al., 1993; Daemen at al., 2019). Subterranean sistemi, Sponge yapısına dayanır ve hem Keccak hem de Xoodoo sistemini içermektedir. Kullandığı tur fonksiyonu ile yinelemeli şekilde şifreleme işlemlerini gerçekleştirir. Tur fonksiyonu Keccak ve Xoodoo sisteminden farklıdır. Permütasyon tabanlı bir sistemdir ve girdi bitlerini bir durum matrisi üzerine yerleştirir. Bu durum matrisi Keccak ve Xoodoo sisteminin aksine bir boyutludur ve 257-bit uzunluğundadır. 257-bit durum matrisi kullanıldığı için yazılım uygulamalarında verimli kullanılamaz. Buna rağmen küçük değişiklikler ile donanımda çok etkili çalışmaktadır (Daemen at al., 2019).

Subterranean tur fonksiyonu dört adımdan oluşur. Bu adımlar π, θ, ι ve χ ile gösterilir. π adımı bitler arasında yayılımı, θ adımı karıştırmayı, ι adımı simetriyi kırmayı ve χ adımı doğrusallığı bozmayı sağlar. Tur adımları içerisinde tek doğrusal olmayan adım χ adımıdır. Durum matrisi S üzerindeki bit pozisyonları $0 \leq i < 257$ olmak üzere S_i ile gösterilir. δ_i Kronecker deltayı belirtir ve eğer i=0 ise 1 değerine

sahiptir, diğer durumlarda ise 0 değerini alır. Bu parametreler ile adım işlemleri aşağıdaki gibi tanımlanır.

$\chi : S_i \leftarrow S_i \oplus (S_{i+1} \oplus 1) \& S_{i+2}$

$\iota : S_i \leftarrow S_i \oplus \delta_i$

$\theta : S_i \leftarrow S_i \oplus S_{i+3} \oplus S_{i+8}$

$\pi : S_i \leftarrow S_{12i}$.

Subterranean sistemi, ilk tur fonksiyonunda durum matrisi üzerine Duplex yapısını uygular. Daha sonra en fazla 32-bit uzunluğunda bit dizisini durum matrisine işler. Bu işlem gerçekleşmeden önce, eğer bit dizisi 32-bit uzunluktan daha kısa ise 10* kuralı ile dolgulama işlemi yapılır ve bit dizisi 33-bit uzunluğa tamamlanır. Bu durumda durum matrisine bu 33-bit işlenir. Subterranean sisteminde uygulanan iki Duplex yapısı arasında durum matrisinden 32-bit, yani blok uzunluğu kadar, uzunlukta çıktı verilebilir. Her 32-bit uzunluğundaki çıktı, iki durum matrisi bitlerinin toplamını oluşturur. Bunlar, mod 257 tarafından oluşturulan 64 mertebesinin çarpımsal alt grubunun elemanlarından alınmıştır. Bu alt grup G_{64} ile tanımlanır. Dolgulanmış 33-bit uzunluklu bit dizisi durum matrisine işlendiğinde, bit pozisyonları G_{64}'de 12^4 değerinin ilk 33 kuvvetini gösterir. Subterranean sisteminin özet fonksiyonunda ise girdi uzunluğu sınırı 8-bit olarak ayarlanmıştır. Bunun anlamı, dolgulanmış girdi bitlerinin sadece ilk 9 biti sıfır olmayabilir. Bu durumda, bitleri etkili şekilde durum matrisine işleme hızı 9-bit olarak belirtilir. Durum matrisine işlenen bu 9 bitin bit pozisyonları G_{64}'de 12^4 değerinin ilk dokuz kuvvetini gösterir.

Subterranean sisteminin Duplex yapısı, iki fonksiyondan oluşur. Bu fonksiyonlar; yapının çağırılması ve çıktı vermedir. Duplex yapısının çağırılması fonksiyonunda durum matrisine tur fonksiyonu uygulanır ve girdi işlenir. Çıktı verme fonksiyonu ise bir önceki paragrafta anlatılmıştır. Bu fonksiyonlar Sponge yapısındaki emilme ve sıkma aşamalarını destekler niteliktedir. Subterranean sisteminde, 256-bit sabit uzunluklu özet fonksiyonu dışında genişletilmiş çıktı fonksiyonu da vardır. Yani, özet fonksiyonu kullanılarak seçilen uzunlukta çıktı elde edilebilir. Subterranean sisteminin genişletilmiş çıktı fonksiyonu Sponge yapısındaki genişletilmiş çıktı fonksiyonu gibi çalışır ve kapasite değeri 224-bit uzunluğundadır. (Daemen et al., 2019)'a göre bu sistemin genişletilmiş çıktı fonksiyonu için güvenlik iddiası 112-bittir. Aslında özet fonksiyonunda 9-bit girdi bloğu olduğundan kapasite değeri 257-9=248 bit olmalıdır. Uzatılmış çıktı fonksiyonunda bu değer 24-bit kısaltılmıştır.

Subterranean sistemindeki χ doğrusal olmayan adımı, Keccak sistemindeki doğrusal olmayan adım işleminden de bilinmektedir. Bu işlemdeki her çıktı biti, çok az sayıda girdi bitine bağlıdır. Ayrıca bu adımın cebirsel derecesi yalnızca 2

değerindedir. Düşük cebirsel derecenin avantajı diferansiyel güç analizine karşı önlem alınmasını kolaylaştırmaktadır. Ayrıca χ işleminin tersinin cebirsel derecesi 128 olarak belirtilmiştir. Bu derece de yine kriptanalizde direnci yükseltmektedir. θ adımının çıktısındaki tek bir bit ise yalnızca 3 adet girdi bitinin toplamından etkilenmektedir. θ adımının cebirsel özellikleri durum matrisi *s*, $\sum_i s_i x^i$ polinomu ile ifade edilirse daha kolay görünmektedir. θ işlemi modüler çarpma işlemi ile oluşur. Yani; $\theta(s(x)) = s(x)(x^8+x^3+1) \bmod (x^{257}+1)$ şeklinde ifade edilir. x^8+x^3+1, θ doğrusal kaydırma değişmezi dönüşümünün çarpım polinomudur.

$x^{257}+1$ ile aralarında asal, derecesi 257'den küçük polinomlar $p(x)$ ile gösterilir ve Q ile gösterilen bir grup oluşturur. $x^{257}+1$ polinomu, 16 tane 16 dereceli indirgenemez polinomun $x+1$ ile çarpımından oluşur. Yani; $p^{2^n}(x) \bmod (x^{257}+1) = p(x^{2^{n \bmod 257}})$. *n*, $(Z/257Z^*,x)$'de 2 mertebesine sahipse $p^{2^n}(x) \bmod (x^{257}+1) = p(x)$ olur. θ adım işleminin tersinin Hamming ağırlığı 127 olarak belirtilmiştir.

Subterranean tur fonksiyonu 2 cebirsel derecesine sahiptir. Eğer k tur boyunca çalışırsa cebirsel derece 2^k ile belirlenir. Subterranean genişletilmiş özet fonksiyonunda ise çakışma atağında sıfır olmayan girdi farkının, durum matrisi üzerinde oluşturduğu çıktı farkları incelenir. Anahtarsız yapısında, emilme aşamasında çakışmaya sahip durum matrisi bulmak oldukça zordur. Subterranean sisteminde girdi mesajı 8-bit uzunluklu bloklara ayrılır ve eğer son mesaj bloğu 8-bit uzunluktan daha kısa ise 10* kuralı ile dolgulama işlemi yapılır. Saldırgan iki emilme aşaması arasında boş bir tur olduğundan ve girdi bit dizileri olarak alınıyorsa art arda iki Duplex yapısı çağrısının arasında 9-bit seçebilir. Eğer girdi bayt dizileri olarak alınıyorsa 2^8+1 girdi değerinden seçebilir. Bu durumda bir Duplex yapısı çağrısı başına seçilen bit e küçük bir sayı olmak üzere 8+e-bite indirgenebilir. Durum matrisi üzerindeki çakışma, eşit uzunluklu iki farklı girdi veya farklı uzunluklu iki farklı girdi için oluşabilir. Eşit uzunluklu girdilerdeki Duplex yapısı için yüksek olasılık potansiyeline sahip diferansiyeller bulunmaya çalışılabilir. Farklı uzunluklu girdilerde ise sabit nokta üretimi ile diferansiyeller bulunmaya çalışılabilir. Çakışma direnci doğum günü paradoksuna bağlanmıştır ve rastgele girdiler verilerek denenir.

Eşit uzunluktaki girdilerdeki Duplex yapıları için, yüksek olasılık potansiyeli olan diferansiyel ya da yoldan yararlanarak bir iç çakışma oluşturmaya çalışılabilir. Durum matrisi üzerindeki 248-bit uzunlukta bir çakışma elde edilmesi gerekmektedir. Saldırgan her 2 girdi bloğu için bir Duplex yapısı kullanırken 9-bit seçmelidir. Belirli bir durum matrisinden başlandığında, $248/(2*9) \approx 14$ bloktan daha az sayıda çakışan girdi dizisi bulmak mümkün değildir. Subterranean sistemindeki özet fonksiyonunun tüm ataklara karşı direnci 128-bittir (Daemen et al., 2019)

3.14 TRIAD

TRIAD sistemi, doğrulanmış şifreleme ve özet fonksiyonu içeren akan şifredir. TRIVIUM sistemin benzer bir yapı kullanır. TRIVIUM sistemi 80-bit güvenlik sağlarken, TRIAD sistemi 112-bit/128-bit arasında güvenlik sağlar. TRIVIUM sisteminin 288-bit durum matrisi ile sağladığı güvenliği, TRIAD sistemi 256-bit durum matrisi ile sağlar. TRIAD özet fonksiyonu, Sponge tabanlı permütasyon ile işlemlerini gerçekleştirir. Sponge tabanlı bir sistem olduğundan durum matrisi blok ve kapasite değerlerinden oluşur. TRIAD sistemindeki işlemlerin girdi boyutu en fazla 2^{50} bayt olabilir.

TRIAD permütasyonu TRIAD-P ile gösterilir. TRIAD sisteminde girdi mesajının her biti permütasyon tarafından işlenir. Her bit için 1024 kez tur fonksiyonu uygulanır. TRIAD tur fonksiyonu işlemleri aşağıda gösterilmiştir.

$x=(x_1||x_2||\ldots||x_{80})$, $y=(y_1||y_2||\ldots||y_{88})$ ve $z=(z_1||z_2||\ldots||z_{88})$ değişkenleri 256-bit durum matrisinin bitleri, m ise girdi mesajının bir biti olmak üzere;

$t_1 \leftarrow x_{68} \oplus x_{80} \oplus y_{85} \& z_{85}$,

$t_2 \leftarrow y_{64} \oplus y_{88}$,

$t_3 \leftarrow z_{68} \oplus z_{88}$,

$z \leftarrow t_1 \oplus t_2 \oplus t_3$,

$t_1 \leftarrow t_1 \oplus x_{73} \& x_{79} \oplus y_{66} \oplus m$,

$t_2 \leftarrow t_2 \oplus y_{65} \& y_{87} \oplus z_{84} \oplus m$,

$t_3 \leftarrow t_3 \oplus z_{77} \& z_{87} \oplus x_{74} \oplus m$,

$(x_1,x_2,\ldots,x_{80}) \leftarrow (t_3, x_1,x_2,\ldots,x_{79})$,

$(y_1,y_2,\ldots,y_{88}) \leftarrow (t_1, y_1,y_2,\ldots,y_{87})$,

$(z_1,z_2,\ldots,z_{88}) \leftarrow (t_2, z_1,z_2,\ldots,z_{87})$.

TRIAD özet fonksiyonunda 256-bit permütasyonun girdi blok uzunluğu 32-bit, kalan 224-bit ise kapasite değeridir. Çıktı blok uzunluğu ise 128-bittir ve toplamda 256-bit özet değeri oluşturulur. Başlangıçta özet değeri alınacak mesaj 32-bit bloklara ayrılır. Eğer son bloktaki tüm bit pozisyonları dolu değilse 10* dolgulama kuralı ile dolgulama işlemi yapılır. Emilme aşamasında girdi mesajının blokları durum matrisinin ilk 32-biti ile $\oplus$ işlemine tabi tutulur. Sonrasında durum matrisi TRIAD permütasyonundan geçirilir. Tüm bloklar için bu işlem yapıldıktan sonra sıkma aşamasına geçilir. Sıkma aşamasında, ilk olarak emilme aşamasından alınan durum matrisi TRIAD permütasyonundan geçirilir ve yeni durum matrisinin

ilk 128-biti çıktı olarak alınır. Bu durum matrisi tekrar TRIAD permütasyonundan geçirilir ve 128-biti çıktı olarak alınarak önceki çıktının sonuna eklenir. Sonuçta 256-bit özet değeri oluşturulur.

TRIAD özet fonksiyonunun çakışma direnci h özet boyutu olmak üzere $\min(2^{h/2},2^{c/2})$, ikinci ön-görüntü direnci $\min(2^{h/2},2^{c/2})$ ve ön-görüntü direnci $\min(2^{\min(h,t)}, \max(2^{\min(h,t)-r'},2^{c/2}))$ ile hesaplanır. h=256, c=224, r=32, r'=128 ve t=256 olmak üzere sırasıyla 2^{112}, 2^{112} ve 2^{128} güvenlik sağlar (Banik et al., 2019).

3.15 SYCON

SYCON sistemi, 320-bit kriptografik permütasyon üzerinde doğrulanmış şifreleme ve özet fonksiyonu işlemlerini gerçekleştirir. SYCON doğrulanmış şifelemesi MonkeyDuplex yapısına dayanırken, özet fonksiyonu Sponge yapısına dayanmaktadır. SYCON sistemi yinelemeli olarak çalışan tur fonksiyonuna sahiptir. Tur fonksiyonu S-Box, permütasyon katmanı, alt bloklarda yayılım ve tur sabiti ile toplama işlemi olmak üzere dört işlemden oluşur. SYCON sisteminde bitler bir durum matrisi üzerine yerleştirilir ve tur fonksiyonu bu durum matrisi üzerinde çalışır.

S-Box işlemi, 320-bit durum matrisi üzerine yerleştirilmiş bitler üzerinde yapılır. Durum matrisi 64 tane 5-bit uzunluğunda kelimeye sahiptir ve 5x5 S-Box bu kelime değerleri üzerine uygulanır. Bu adım doğrusal olmayan adımdır ve cebirsel derecesi 2 değerindedir. Permütasyon katmanı iki işlemden oluşur. İlk permütasyon katmanı işlemi, S-Box uygulanmış durum matrisini girdi olarak alır. Sonrasında belirli bir kural ile durum matrisi üzerindeki tüm bitlerin yerini değiştirir. İkinci permütasyon katmanı işleminde ise durum matrisi üzerindeki ilk 64-bit üzerinde ROT ve karıştırma işlemleri yapılır.

Alt bloklarda yayılım adımında durum matrisindeki 320-bit beş tane 64-bit uzunluğunda bloklara ayrılır. << ROT işlemi ve X_0, X_1, X_2, X_3, X_4 64-bit uzunluğundaki değerler olmak üzere alt bloklarda yayılım işleminde aşağıdaki adımlar uygulanır;

$X_0 \leftarrow X_0 \oplus (X_0 << 11) \oplus (X_0 <<22)$,

$X_1 \leftarrow X_1 \oplus (X_1 << 13) \oplus (X_1 <<26)$,

$X_2 \leftarrow X_2 \oplus (X_2 << 31) \oplus (X_2 <<62)$,

$X_3 \leftarrow X_3 \oplus (X_3 << 56) \oplus (X_3 <<60)$,

$X_4 \leftarrow X_4 \oplus (X_4 << 6) \oplus (X_4 <<12)$.

Tur sabiti ile toplama işlemi durum matrisi üzerindeki simetriyi dağıtmak için kullanılır. Tur sabitleri, F_2 cismi üzerindeki x^5+x^3+1 polinomu ile tanımlanan 5-bit LFSR ile üretilir. Başlangıç tur sabiti (1,0,1,0,1) değeridir ve 14 tane tur sabiti üretilir.

SYCON özet fonksiyonu, 64-bit blok uzunluğu ve 256-bit kapasite değerine sahiptir. 14 tur boyunca durum matrisi üzerinde girdi mesajını işleyerek 256-bit uzunluğunda özet değeri oluşturur. SYCON özet fonksiyonunda girdi mesajı 64-bit bloklara ayrılır. Eğer mesajın son bloğunda veya birleştirilmiş verinin son bloğunda bit pozisyonlarında boş kısımlar var ise 10* dolgulama kuralı ile dolgulama işlemi yapılır.

SYCON özet fonksiyonu üç aşamadan oluşur. İlk aşama başlangıç vektörünün durum matrisine yüklenme aşamasıdır. Başlangıç vektörü 6 bit uzunluğundadır ve durum matrisi üzerine $0^{128}||$başlangıç vektörü$||0^{128}$ şeklinde yerleştirir. Daha sonra dolgulanmış girdi mesajını 64-bit bloklara bölünerek öncelikle mesajın işlendiği aşama olan emilme aşaması uygulanır. Emilme aşamasında her mesaj bloğu durum matrisinin ilk 64-biti ile $\oplus$ işlemin tabi tutulur. Sonra durum matrisi SYCON permütasyonu ile 14-tur boyunca işlenir. Tüm mesaj bloklarına bu işlemler uygulandıktan sonra emilme aşamasının çıktısı olan durum matrisinin ilk 64-biti özet değerinin ilk 64-bitini oluşturur. Sıkma aşamasında bloklar halinde toplam 256-bit uzunluğunda özet değeri oluşturulur. 256-bit oluşturulması için sıkma aşaması toplamda 4 tez çalıştırılmış olur.

SYCON özet fonksiyonunun çakışma direnci 128-bit, ön-görüntü direnci 192-bit ve ikinci ön-görüntü direnci 128-bittir. SYCON sisteminde kullanılan S-Box yapısının maksimum diferansiyel olasılığı 2^{-2} olarak hesaplanmıştır. Diferansiyel dal sayısı 3 olmak üzere yayılım S-Box adımında başlar. 4 turdaki aktif S-Box sayıları sırasıyla 1, 4, 11 ve 51 olmak üzere, diferansiyel ataklara karşı karmaşıklık 2^{102} mertebesindedir. SYCON S-Box yapısının maksimum doğrusal olasılığı yine 2^{-2} olarak belirtilmiştir. Dal sayısının 3 olması aktif S-Box sayısının artmasına yardımcı olur. 4 turdaki aktif S-Box sayıları sırasıyla 1, 4, 9 ve 39 olmak üzere, doğrusal ataklara karşı karmaşıklık 2^{78} mertebesindedir (Sarkar et al., 2019).

3.16 SPARKLE

SPARKLE sistemi, özet fonksiyonu ve doğrulanmış şifreleme yapılarından oluşur. Üç farklı uzunlukta permütasyona sahiptir. Bu permütasyonlar, $n_s \in N$ adım sayısı olmak üzere, SPARKLE256n_s, SPARKLE384n_s ve SPARKLE512n_s ile gösterilir. SPARKLE sistemindeki özet fonksiyonu ESCH ile adlandırılır. ESCH özet fonksiyonu, seçilen uzunluktaki mesajı girdi olarak alır ve sabit uzunluklu özet değeri üretir. Özet çıktısı boyutuna göre ESCH256 ve ESCH384 olmak üzere iki faklı fonksiyona sahiptir. SPARKLE permütasyonları Sponge tabanlıdır ve ESCH özet fonksiyonu işlemlerini SPARKLE permütasyonları üzerinde gerçekleştirir.

SPARKLE permütasyonları iki bileşenden oluşur. İlk bileşeni ARX-Box ile adlandırılır ve A_c ile gösterilir. ARX-Box 64-bit blok şifredir. ARX-Box yinelemeli 4 turdan oluşur ve her turda iki 32-bit kelime değeri üzerine farklı miktarda ROT işlemi yapar. Ayrıca her turda 32-bit bir sabit ile 32-bit ilk kelime $\oplus$ işlemin tabi tutulur. 32-bit uzunluklu sabit değer anahtar olarak da adlandırılır. (x,y) $\in F_2^{32}$, + modüler toplama ve $\gg$ ROT işlemi olmak üzere, ARX-Box işlemlerinin matematiksel gösterimi aşağıdaki gibidir.

$x \leftarrow x + (y \gg 31)$ $\quad y \leftarrow y \oplus (x \gg 24)$ $\quad x \leftarrow x \oplus c,$

$x \leftarrow x + (y \gg 17)$ $\quad y \leftarrow y \oplus (x \gg 17)$ $\quad x \leftarrow x \oplus c,$

$x \leftarrow x + (y \gg 0)$ $\quad y \leftarrow y \oplus (x \gg 31)$ $\quad x \leftarrow x \oplus c,$

$x \leftarrow x + (y \gg 24)$ $\quad y \leftarrow y \oplus (x \gg 16)$ $\quad x \leftarrow x \oplus c.$

SPARKLE permütasyonunun diğer bileşeni ise doğrusal yayılım katmanıdır ve n_b tur sayısı olmak üzere Ln_b ile gösterilir. Ln_b, h=n_b/2 olmak üzere $(F_2^{64})^h$'nin bir permütasyonu olan doğrusal Feistel fonksiyonunun bir turudur. Ln_b katmanında M_h Feistel fonksiyonu ve iki 32-bit kelime arasında yer değiştirme işlemi uygulanır. w>1 ve (x,y) 32-bit uzunluklu iki kelime değeri olmak üzere $(F_2^{32})^w$'nin bir permütasyonu olan doğrusal Feistel fonksiyonu $M_w((x_0,y_0),...,(x_{w-1},y_{w-1}))=((u_0,v_0),...,(u_{w-1},v_{w-1}))$ şeklinde tanımlanır. (u_i,v_i) değerleri aşağıdaki denklemlerle elde edilir.

l: $F_2^{32} \rightarrow F_2^{32}$, $l(x)=(x \ll 16) \oplus (x \,\&\, 0xffff)$ olmak üzere,

$t_x \leftarrow \bigoplus_{i=0}^{w-1} x_i, \quad t_y \leftarrow \bigoplus_{i=0}^{w-1} y_i,$

$u_i \leftarrow x_i \oplus l(t_y), \forall i \in \{0,...,w-1\},$

$v_i \leftarrow y_i \oplus l(t_x), \forall i \in \{0,...,w-1\}.$

ESCH özet fonksiyonunda, SPARKLE permütasyonu r blok uzunluğu ve c kapasitesi ile parametrelendirilir. ESCH256 özet fonksiyonu r değeri 128-bit uzunlukta, c değeri 256-bit uzunlukta olmak üzere 384-bit uzunluklu SPARKLE384n_s permütasyonunu kullanır. ESCH384 özet fonksiyonu ise r değeri 128-bit uzunlukta, c değeri 384-bit uzunlukta olmak üzere 512-bit uzunluğa sahip SPARKLE512n_s permütasyonunu kullanır. ESCH256 ve ESCH384 özet fonksiyonlarında blok uzunluğu değeri aynıdır. Bunun anlamı her iki özet fonksiyonunda da girdi mesajı 128-bit uzunluklu bloklara ayrılarak işlem görür. Eğer son mesaj bloğunda boş bit pozisyonları var ise 10* dolgulama kuralı ile bit pozisyonları doldurulur.

ESCH özet fonksiyonu emilme ve sıkma olmak üzere iki aşamadan oluşur. ESCH256 özet fonksiyonunda bitler 384-bit durum matrisi üzerine yerleştirilir.

Başlangıçta, durum matrisinin tamamı 0 bitleri ile doldurulur. ESCH özet fonksiyonunda hiçbir mesaj bloğu direkt olarak durum matrisi ile ⊕ işlemine girmez. Son mesaj bloğu hariç, tüm mesaj bloklarının sonuna 64-bit uzunluğunda 0 biti dolgulanır. 192-bit uzunluklu dolgulanmış mesaj blokları öncelikle durum matrisi ile ⊕ işlemine tabi tutulur. ⊕ işleminden sonra durum matrisi $SPARKLE384_7$ permütasyonundan geçirilir. Son mesaj bloğu ise öncelikle 64-bit 0 biti ile dolgulandıktan sonra bir sabit ile ⊕ işlemine tabi tutulur. Bu işlemden sonra ise elde edilen sonuç durum matrisi ile ⊕ işleminden geçirilir ve durum matrisi $SPARKLE384_{11}$ permütasyonu ile işlenir. Bu işlemlerin tümü emilme aşamasında gerçekleşir. Kullanılan 192-bit uzunluklu sabit, son mesaj bloğunun dolgulanmış olup olmamasına bağlı olarak, en anlamsız iki bitinden biri 1 diğer bitlerinin ise 0 olması ile belirlenir.

ESCH384 özet fonksiyonu 512-bit durum matrisi üzerinde işlemlerini gerçekleştirir. Başlangıçta, durum matrisinin tamamı 0 bitleri ile doldurulur. Son mesaj bloğu hariç, tüm mesaj bloklarının sonuna 128-bit uzunluğunda 0 biti dolgulanır. 256-bit uzunluklu dolgulanmış mesaj blokları öncelikle durum matrisi ile ⊕ işlemine tabi tutulur. ⊕ işleminden sonra durum matrisi $SPARKLE512_8$ permütasyonundan geçirilir. Son mesaj bloğu ise öncelikle 128-bit 0 biti ile dolgulandıktan sonra bir sabit ile ⊕ işlemine tabi tutulur. Bu işlemden sonra ise elde edilen sonuç durum matrisi ile ⊕ işleminden geçirilir ve durum matrisi $SPARKLE512_{12}$ permütasyonu ile işlenir. Bu işlemlerin tümü emilme aşamasında gerçekleşir. Kullanılan 256-bit uzunluklu sabit, eğer son blok dolgulanmış ise 000…01, eğer son blok dolgulanmamış ise 000…10 değerine sahiptir.

Emilme aşaması bittikten sonra özet çıktısı alınan sıkma aşamasına geçilir. ESCH256 özet fonksiyonunun sıkma aşamasında öncelikle durum matrisinin ilk 128-biti çıktı olarak verilir. Daha sonra durum matrisi $SPARKLE384_7$ permütasyonundan geçirilir. Yeni durum matrisinin ilk 128-biti önceki 128-bit uzunluklu çıktının devamına eklenir. Sonuçta 256-bit özet değeri elde edilir. ESCH384 özet fonksiyonunun sıkma aşamasında ise emilme aşamasından gelen durum matrisinin ilk 128-biti çıktı olarak verilir. Daha sonra durum matrisi $SPARKLE512_8$ permütasyonundan geçirilir. Yeni durum matrisinin ilk 128-biti önceki 128-bit uzunluklu çıktının devamına eklenir. Bu işlem bir kez daha tekrarlanır ve yeni durum matrisinin ilk 128-biti önceki 256-bit uzunluklu çıktının devamına eklenir. Sonuçta 384-bit özet değeri elde edilir.

ESCH256 ve ESCH384 için güvenlik seviyesi c / 2 ile belirlenir. Tam çığ etkisini SPARKLE permütasyonlarında yayılım ölçüsü olarak tanımlanmıştır. Tam çığ etkisi, i. girdi biti her değiştirildiğinde j. çıktı bitinin değişme olasılığının ½ olması ile tanımlanır. Diferansiyel kriptanaliz için dal düzeyinde tanımlanan tüm kesilmiş yollar doğrusal adımla uyumlu olarak numaralandırılır. ARX-Box için oluşturulmuş olan olasılığı kullanarak bu kesilen yolların olasılığına bağlı bir sınır elde edilir. Pratikte diferansiyel yol, $n \in \{4,6,8\}$ olmak üzere, n uzunluklu d_i ikilik

vektörlerinin dizisidir. Her adımda dal sayısının en fazla $2^{n/2} \in \{8,16,32\}$ ile çarpacağını bilerek tüm kesilmiş yolların bulunması bir ağaç araması olarak uygulanmaktadır. Ayrıca (Matsui, 1994)'de sunulan algoritma kullanılarak ve belirli eşik değerleriyle sınırlama işlemi yapılarak bu ağaç arama işlemi daha da basit hale indirgenmiştir. 4 tur ARX-Box yapısı ve sınırlandırmalar kullanılarak elde edilen permütasyon sınırları Tablo 3.16.1'de verilmiştir.

Tablo 3.16.1 SPARKLE Sisteminde Bazı Adımlardaki Diferansiyel Sınırlar

n/adımlar	3	4	5	6	7	8	9	10	11	12	13
256	64	88	140	168	192	216	≥256	≥256	≥256	≥256	≥256
384	70	100	178	200	230	260	326	356	≥384	≥384	≥384
512	76	112	210	232	268	276	295	424	433	496	≥512

Simetrik bir şifreleme sisteminin diferansiyel ve doğrusal kriptanalize karşı direnci, maksimum olasılıklı diferansiyel/doğrusal özelliklerle belirlenir. Bunun nedeni, diferansiyel/doğrusal bir saldırının başarı olasılığının, atağın gerçekleşmesi için gereken girdi miktarına bağlı olmasıdır. SPARKLE sisteminin tüm güvenlik analizi (Beierle et al., 2019)'de açıklanmıştır. Sponge yapısına dayandığından sistemin güvenliğinde Sponge yapısının güvenlik özelliklerinden faydalanılmıştır. Bunların yanında SPARKLE sisteminin özet fonksiyonları olan ESCH256 ve ESCH384 fonksiyonlarının parametre değerleri aşağıdaki Tablo 3.16.2'de verilmiştir. Ayrıca genel ataklara karşı sağladıkları dirençler de Tablo 3.16.2'de verilmiştir (Beierle et al., 2019).

Tablo 3.16.2 ESCH256 ve ESCH38 Özet Fonksiyonlarının Parametre Değerleri ve Güvenliği

	Perm. Uzunluğu	Blok Uzunluğu	Kapasite	Çakışma Direnci	2. Ön-Görüntü Direnci	Ön-Görüntü Direnci
ESCH256	384	128	256	128	128	128
ESCH384	512	128	384	192	192	192

3.17 SKINNY

SKINNY sistemi doğrulanmış şifreleme ve özet fonksiyonundan oluşur. SKINNY özet fonksiyonu iki üyeden oluşur. Bu üyeler SKINNY-128-384 ve SKINNY-128-256 ile gösterilir. İki SKINNY özet fonksiyonu da Sponge yapısına dayanmaktadır ve 256-bit uzunluğunda özet değeri üretmektedir. SKINNY-128-384 özet fonksiyonu F_{384} : $F_2^{384} \rightarrow F_2^{384}$ fonksiyonu kullanılarak oluşturulur. SKINNY-128-256 özet fonksiyonu ise F_{256} : $F_2^{256} \rightarrow F_2^{256}$ fonksiyonu kullanılarak oluşturulur. SKINNY-128-384 özet fonksiyonundaki 384-bit permütasyonun 128-

bit kısmı blok uzunluğu, 256-bit kısmı ise kapasite olarak belirtilmiştir. SKINNY-128-256 özet fonksiyonundaki 256-bit permütasyonun 32-bit kısmı blok uzunluğu, 224-bit kısmı ise kapasite olarak gösterilir.

SKINNY-128-384 ve SKINNY-128-256 permütasyonları, sıkma aşamasında 128-bit blok uzunluğuna sahiptir. Her ikisi de başlangıçta girdi mesajlarını 8-bit uzunluğunda kelimelere ayırarak 4x4 durum matrisi üzerine yerleştirir. Ayrıca iki permütasyonun farklı uzunlukta olmak üzere tweak değeri vardır. Permütasyonlar tur fonksiyonlarından oluşur. Tur fonksiyonu bitler arasında yer değiştirme, sabitlerle toplama, tur tweak değeri ile toplama, satırlarda kaydırma ve sütunlarda karıştırma olarak beş adımdan oluşur. Kullanılan tweak değeri ile bağımlı olarak tur sayısı değişmektedir. SKINNY-128-256 toplamda 48 turdan oluşurken, SKINNY-128-384 56 turdan oluşur.

Bitler arasında yer değiştirme işleminde, 8-bit PICCOLO S-Box yapısı kullanılır. Kullanılan S-Box yapısının yaptığı işlem *x0,…,x7* S-Box girdileri olmak üzere; $(x7,x6,x5,x4,x3,x2,x1,x0) \rightarrow (x7,x6,x5,x4 \oplus (\overline{x7\ or\ x6}),\ x3,x2,x1,x0 \oplus (\overline{x3\ or\ x2}))$ şeklinde gösterilir. Bu işlem üç kez tekrar eder. Son bir adım olarak ise x1 ve x2 bitleri yer değiştirir. Sabitlerle toplama adımında ise 6-bit LFSR kullanılarak elde edilen tur sabitleri durum matrisi üzerindeki bitler ile $\oplus$ işlemine tabi tutulur. Tur tweak değeri ile toplama adımında durum matrisinin birinci ve ikinci satırındaki bitler, pozisyonlarına karşılık gelen tweak değerleri ile $\oplus$ işlemine girer ve durum matrisi üzerinde LFSR uygulanır. Satırlarda kaydırma adımında, ilk satır hariç diğer satırlara, sırasıyla 1,2,3 bit sağa doğru ROT işlemi uygulanır. Sütunlarda karıştırma işleminde ise durum matrisi ile aşağıdaki matris çarpılır.

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

$F_{384}: F_2^{384} \rightarrow F_2^{384}$ fonksiyonu SKINNY-128-384 permütasyonu tabanlıdır. F_{384} fonksiyonu 384-bit uzunluğunda girdiyi bir t tweak değeri altında SKINNY-128-384$_t$ permütasyonu ile işler. F_{384} fonksiyonunun çıktısı $F_{384}(t)$=SKINNY-128-384$_t(0^{128})$||SKINNY-128-384$_t(0^7||1||0^{120})$||SKINNY-128-384$_t(0^6||1||0^{121})$ ile hesaplanır. $F_{256}: F_2^{256} \rightarrow F_2^{256}$ fonksiyonu SKINNY-128-256 permütasyonu tabanlıdır. F_{256} fonksiyonu 256-bit uzunluğunda girdiyi bir t tweak değeri altında SKINNY-128-256$_t$ permütasyonu ile işler. F_{256} fonksiyonunun çıktısı $F_{256}(t)$=SKINNY-128-256$_t(0^{128})$||SKINNY-128-256$_t(0^7||1||0^{120})$ ile hesaplanır.

SKINNY-128-384 özet fonksiyonu, 384-bit durum matrisi üzerindeki tüm bitleri başlangıçta 0 olarak ayarlar. Girdi mesajı 128-bit uzunlukta bloklara ayrılır. Eğer son blokta boş bit pozisyonları var ise bu bit pozisyonları 10* kuralı ile dolgulanarak doldurulur. Sponge yapısına dayandığından SKINNY özet

fonksiyonları emilme ve sıkma aşamasından oluşur. SKINNY-128-384 özet fonksiyonunun emilme aşamasında, mesaj blokları öncelikle 256-bit 0 değeri ile dolgulanır ve durum matrisi ile $\oplus$ işlemine girer. Sonra durum matrisi F_{384} fonksiyonu ile işlenir. Son mesaj bloğu işlenene kadar bu işlem devam eder. Sonra sıkma aşamasına geçilir. Sıkma aşaması özet değerinin çıktı olarak alınacağı aşamadır. Sıkma aşamasına geçildiğinde, öncelikle durum matrisinin ilk 128-biti çıktı olarak alınır. Daha sonra durum matrisi F_{384} fonksiyonu ile işlenir ve yeni durum matrisinin ilk 128-biti önceki çıktının devamına eklenerek 256-bit özet çıktısı elde edilir.

SKINNY-128-256 özet fonksiyonu, SKINNY-128-384 özet fonksiyonu ile aynı çalışma mantığına sahipken kullandığı fonksiyon F_{256} fonksiyonu olarak değişir ve emilme aşamasında mesaj blokları 32-bit parçalar halinde işlenir. Sıkma aşaması ise SKINNY-128-384 özet fonksiyonu ile aynıdır. Yalnızca iki kez tekrar eder ve her seferinde 128-bit çıktı alınarak 256-bit özet değeri oluşturulur (Beierle et al., 2019).

SKINNY özet fonksiyonlarında çakışma ve ön-görüntü dirençleri seçilen üyeye bağlı olarak 112-bit ve 128-bit olarak değişmektedir. SKINNY özet fonksiyonları SHA-3 yapısına dayandırıldığından güvenliği de Sponge yapısına dayanmaktadır. SKINNY-128-256 için bakılırsa 256-bit kapasite, 128-bit değişmezlik sağlar. Bu nedenle, 2^{128} işlem maliyeti altında belirli bir atak gerçekleştirilemez. Emilme aşamasında kapasite değerinin 224-bit olması 112-bit güvenlik sağlar. İşlemlerin daha hızlı gerçekleştirilmesi için SKINNY özet fonksiyonlarındaki emilme ve sıkma aşamalarındaki blok uzunluğu ve kapasite değerleri farklıdır (Naito and Ohta, 2014). c ve c' sırasıyla emilme ve sıkma aşamalarının kapasite değerleri olsun. Bu durumda c' kapasitesi $O(2^{c/2})$ karmaşıklığı korunarak $c' \geq c/2+\log_2 c$ değerine kadar büyütülebilir. Hafif-siklet bir uygulama ve 112-bit güvenlik için $c' \geq 224/2+\log_2 224 \approx 119.8$ değerindedir. Aslında SKINNY özet fonksiyonun analizinde (Naito and Ohta, 2014)'da permütasyon tabanlı sistemler için yapılan analiz fonksiyon tabanlı sistemlere genişletilmiştir.

3.18 SIV-Rijndael256

SIV-Rijndael256 sistemi doğrulanmış şifreleme ve özet fonksiyonundan oluşur. Özet fonksiyonu Sponge yapısına dayandırılmaktadır ve SIV-Rijndael256-Hash ile gösterilir. Özet fonksiyonu Rijndael256 fonksiyonunun gizli anahtarının 0 olarak ayarlandığı permütasyonu kullanır. Rijndael256 fonksiyonu, 256-bit durum matrisi üzerinde işlemlerini gerçekleştirir. Durum matrisi 4x8 boyutundadır ve her hücresi 1-bayt yani 8-bit uzunluğundadır.

Rijndael256 fonksiyonu dört adımdan oluşur. Bu adımlar baytlarda yer değiştirme, satırlarda kaydırma, sütunlarda karıştırma ve tur anahtarı ile toplamadır. Baytlarda yer değiştirme adımı, tek doğrusal olmayan adımdır. Bir 8-bit S-Box tablosu yardımı ile durum matrisi üzerindeki baytlarda yer değiştirme işlemi yapılır.

Satırlarda kaydırma adımında, ilk satır hariç diğer satırlarda sırasıyla 1, 2, 3 bayt sola ROT işlemi yapılır. Sütunlarda karıştırma işleminde aşağıdaki matris ile her sütun çarpılarak yeni sütun durum matrisi üzerine yerleştirilir. Tur anahtarı ile toplama adımında ise tur gizli anahtarı ile durum matrisi $\oplus$ işlemine girer. Rijndael256 fonksiyonu 14 turdan oluşmaktadır ve son turda sütunlarda karıştırma işlemi uygulanmaz.

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

SIV-Rijndael-Hash özet fonksiyonu Sponge tabanlı bir fonksiyon olduğundan 256-bit permütasyon uzunluğu, blok uzunluğu ve kapasite değerleri ile belirlenir. Ayrıca özet değerinin oluşturulması emilme ve sıkma aşamalarından oluşur. Sponge yapısından tek farkı emilme aşamasındaki blok uzunluğu ile sıkma aşamasındaki blok uzunluğu birbirinden farklıdır. Emilme aşamasında blok uzunluğu 32-bit, kapasite değeri 224-bit uzunluğundayken, sıkma aşamasında blok uzunluğu 128-bit, kapasite değeri ise 128-bit uzunluğundadır. Girdi mesajı emilme aşamasına geçilmeden önce 32-bit bloklara ayrılır ve eğer boş bit pozisyonları var ise 10* dolgulama kuralı ile dolgulanır. Emilme aşamasında, ilk mesaj bloğu doğrudan durum matrisinin ilk 32-bitine yerleştirilir. Durum matrisindeki kalan boş bit pozisyonları ise 0 biti ile doldurulur. İlk olarak durum matrisi Rijndael256 fonksiyonu ile işlenir. Diğer mesaj blokları öncelikle durum matrisi ile $\oplus$ işlemine girer sonra Rijndael256 fonksiyonu ile işlenir. Sıkma aşamasına geçildiğinde öncelikle durum matrisinin ilk 128-biti çıktı olarak alınır. Sonra durum matrisi Rijndael256 fonksiyonu ile işlenir ve ilk 128-biti önceki çıktının sonuna eklenerek 256-bit uzunluğunda özet değeri elde edilir (Bao et al., 2019).

SIV-Rijndael-Hash için çakışma direnci 112-bit, ön-görüntü direnci 128-bit, ikinci ön-görüntü direnci ise 112-bittir. SIV-Rijndael-Hash özet fonksiyonunun kriptanalizinde Rijndael üzerindeki bilinen-anahtar atakları dikkate alınmıştır. Bilinen-anahtar atakları, 192-bit blok uzunluğu ile 8 tur Rijndael ve 256-bit blok uzunluğu ile 9 tur Rijndael üzerinde çalışır. 256-bit blok uzunluklu Rijndael'e yapılan atakta, bir tanesi 128-bit farklılığa sahip ikili bulunabilir. İdeal durum 2^{64} gerekliliğe sahipken, girdi ve çıktının ikisi de 2^{48} hesaplama ve 2^{32} bellek gerektirir. 256-bit blok uzunluğuna sahip 10 tur Rijndael'de bu özellikler çok farklıdır. Bunun yanında, 192-bit blok uzunluklu Rijndael için, kesilmiş diferansiyel ataklara karşı direnç sağlamak için satırlarda kaydırma işleminde daha iyi parametre değerleri önerilmektedir. Bununla birlikte, 256 bitlik blok uzunluklu Rijndael için, satırlarda kaydırma işlemi için orijinal olandan daha iyi parametreler önerilmemiştir. Bu parametrelerin optimum değerlerine minimum aktif S-Box yapısının sayısı dikkate alınarak karar verilmiştir (Bao et al., 2019).

3.19 Shamash ve Shamashash

Shamashash, Shamash doğrulanmış şifrelemesi ile bağlantılı 256-bit çıktı veren özet fonksiyonudur. (Bertoni et al., 2012)'de sunulan Sponge-Duplex yapısını kullanır ve permütasyon tabanlıdır. Bitler 320-bit durum matrisi üzerinde ifade edilir. Bu 320-bit durum matrisi, beş adet 64-bit uzunluğunda kelimeden oluşur. Bu 64-bit uzunluğundaki kelimeler w_0, w_1, w_2, w_3, w_4 ile gösterilir. Başlangıçta durum matrisi üzerindeki kelime değerlerinden w_0 ve w_1 yerine gizli anahtar bitleri, w_2 ve w_3 yerine genel mesaj numarası değeri ve son kelime değeri olan w_4 yerine w_0 kelimesinin 32-bit çevrimsel kaydırması, w_1 kelimesinin 32-bit çevrimsel kaydırması ve 0xff sabitinin $\oplus$ işleminden geçirilmiş değeri yazılır. Shamashash özet fonksiyonunda anahtar değeri 0 olarak belirlenmiştir. Özet değerinin üretilmesi için Shamash doğrulanmış şifrelemesinde de kullanılan bir tur fonksiyonu kullanılır. Bu tur fonksiyonu ilk bitler yüklendikten sonra 12 tur boyunca durum matrisi üzerinde çalıştırılır.

Girdi mesajının durum matrisi üzerine işlenmesi işleminde, öncelikle girdi mesajı 128-bit uzunluklu bloklara ayrılır. Girdi mesaj bloğunun ilk 64-biti durum matrisi üzerindeki w_3 kelimesi ile $\oplus$ işleminden geçirilir. Sonraki 64-biti ise durum matrisi üzerindeki w_2 kelimesi ile $\oplus$ işleminden geçirilir. $\oplus$ işleminden geçirilmiş bloklar durum matrisi üzerinde $\oplus$ işleminin gerçekleştiği kelimelerin yerine yazılır. Bu işlem her 128-bit mesaj bloğu için tekrar ettikten sonra, durum matrisi her seferinde 9 kez tur fonksiyonundan geçirilir. Çıktı alma aşamasında ise durum matrisine fazladan 3 tur daha uygulanır.

Bu tur fonksiyonu üç adımdan oluşur. Bu adımlar w_0 kelimesi ile tur sabitinin $\oplus$ işleminden geçirilmesi, yer değiştirme ve yayılım adımlarıdır. i bulunulan turu göstermek üzere, tur sabiti (11*i) $\oplus$ 0xff ile hesaplanır. Yer değiştirme adımında 64 tane 5x5 S-Box kullanılır. Bu S-Boxların hepsi aynı değerlere sahiptir. S-Box yapısı aşağıdaki gibi anlatılabilir.

x, y, z, u, v bitler ve $0 \leq i \leq 31$ olmak üzere $i = 24y + 23u + 22z + 2y + x$ denklemi ile ifade edilirken, $S[i]=2^4f_4 + 2^3f_3 + 2^2f_2 + 2f_1 + f_0$ ile tanımlanan ve + işlemi GF(2) cismi üzerinde toplama işlemi ise S-Box yapısının fonksiyonlarının cebirsel normal formu;

$$f_4 = 1 + zv + xv + u + z + xy + y + x,$$

$$f_3 = uv + xv + v + yu + z + y + x,$$

$$f_2 = uv + v + zu + u + xz + y + x,$$

$$f_1 = yv + v + zu + u + yz + z + x,$$

$$f_0 = v + xu + u + yz + z + xy + y.$$

Daha basit bir şekilde,

$f_4 = 1 + u + (v \text{ OR } z) + (x \text{ OR } (y + v))$,

$f_3 = z + (u \text{ OR } y) + (v \text{ OR } (x + u))$,

$f_2 = y + (z \text{ OR } x) + (u \text{ OR } (v + z))$,

$f_1 = x + (y \text{ OR } v) + (z \text{ OR } (u + x))$,

$f_0 = v + (x \text{ OR } u) + (y \text{ OR } (z + x))$.

Yayılım adımı, $\oplus$ ve ROT işlemlerini içeren 3 adımdan oluşur. İlk adımda, a ve b sabit değerler olmak üzere durum matrisi üzerindeki her kelime değerine öncelikle a-bit ROT işlemi uygulanır daha sonra b-bit ROT işlemi uygulanır. Kelime değeri üzerindeki ROT işlemleri ile elde edilen iki değer $\oplus$ işlemine tabi tutularak, durum matrisi üzerinde kelime değerinin yerine yazılır. İkinci adımda durum matrisi üzerindeki kelimeler $i=0,1,2$ için $w_i=w_i \oplus w_3 \oplus w_4$ ve $i=3,4$ için $w_i=w_i \oplus w_0 \oplus w_1 \oplus w_2$ işlemleri ile güncellenir. Son adımda ise ilk dört kelime $2i+1$ bayt ROT işleminden geçirilir. Son kelime değerine kaydırma işlemi uygulanmaz.

Shamashash özet fonksiyonunun güvenlik analizleri, Shamash ile aynıdır. Shamash sisteminde 5x5 boyutlu S-Box en iyi diferansiyel ve doğrusal olasılığa sahip olacak şekilde seçilmiştir ve bu olasılık 2^{-4} olarak belirtilmiştir. Eğer turlar boyunca en az 32 tane aktif S-Box var ise ispatlanabilir güvenlik $(2^{-4})^{32} = 2^{-128}$ seviyesindedir. Örneğin, eğer 9 tur boyunca bir turda ortalama 4 aktif S-Box varsa herhangi bir diferansiyel karakteristik veya doğrusal yol olma olasılığı $(2^{-4})^{4x9}=2^{-144}$ değerinden ya azdır ya da bu değere eşittir. Yayılımdaki değişimler için S-Box adımı kullanılmadan yalnızca yayılım adımı 4 tur için incelenirse, 1 tane aktif S-Box ile başlandığında 4-tur sonunda toplamda 171 aktif S-Box oluşur. Eğer 2 tane aktif S-Box ile başlanırsa 171 aktif S-Box, eğer 3 tane aktif S-Box ile başlanırsa 176 aktif S-Box oluşur. Shamash permütasyonu 2 cebirsel derecesine sahiptir. 9 tur sonunda cebirsel derece $2^9=320$ değerine ulaşmış olacaktır. Güçlü yayılım için permütasyon yüksek cebirsel dereceye sahiptir (Penazzi and Montes, 2019).

3.20 SATURNIN

SATURNIN bir blok şifredir. Şifreleme işlemlerini her biri 4-bitlik 4x4x4 boyutuna sahip üç boyutlu ifade edilen bir durum matrisi üzerinde gerçekleştirir, yani 256-bit uzunluğundadır. Yinelemeli tur fonksiyonuna sahiptir. Tur fonksiyonu üç işlemden oluşur. İlk tur işlemi S-Box adımıdır. Bu adımda iki adet S-Box kullanılır. İlk S-Box çift indeksli turlarda uygulanırken, ikinci S-Box tek indeksli turlarda uygulanır. İki S-Box da 4-bit elemanlardan oluşur. Bu iki S-Box sırasıyla s_0 ve s_1 ile gösterilmek üzere değerleri aşağıdaki tabloda verilmiştir. S-Box adımı

doğrusal olmayan adımdır ve S-Box tablosundaki değerler durum matrisi üzerindeki değerler ile yer değiştirilir.

Tablo 3.20.1 SATURNIN S-Box Değerleri

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s_1(x)$	0	6	14	1	15	4	7	13	9	8	12	5	2	10	3	11
$s_2(x)$	0	9	13	2	15	1	11	7	6	4	5	3	8	12	10	14

Tur fonksiyonunun ikinci adımı tur sayısına bağımlı gerçekleşen bir permütasyondan oluşur. SR ile gösterilir. Tüm çift indeksli turlarda SR kimlik fonksiyonu uygulanır. Eğer tur sayısının mod 4 değeri 1 ise satırlar üzerinde paralel olarak SR uygulanır. Eğer tur sayısının mod 4 değeri 3 ise kelime değerleri üzerinde SR uygulanır. Son adım işlemi ise doğrusal adım olarak adlandırılır ve tüm sütunlara paralel olarak M işlemi uygulanır. M doğrusal işlemi *x, y, z, t* dört bitlik yapıları göstermek üzere aşağıda gösterilmiştir.

$$\text{M:}\begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} \rightarrow \begin{pmatrix} \alpha(x) \oplus \alpha^2(y) \oplus \alpha(y) \oplus \text{z} \oplus \text{t} \\ x \oplus \alpha(y) \oplus \text{y} \oplus \alpha^2(z) \oplus \text{z} \oplus \alpha^2(t) \oplus \alpha(t) \oplus \text{t} \\ x \oplus \text{y} \oplus \alpha^2(z) \oplus \alpha^2(t) \oplus \alpha(t) \\ \alpha^2(x) \oplus \text{x} \oplus \alpha^2(y) \oplus \alpha(y) \oplus \text{y} \oplus \text{z} \oplus \alpha(t) \oplus \text{t} \end{pmatrix}$$

M işlemindeki α dört bitlik yapıların bitleri $a_0, \ldots, a_3$ olmak üzere aşağıdaki çarpma işlemi ile ifade edilir.

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

Bu işlem LFSR yapısına karşılık gelir. Geri besleme polinomu ise x^4+x^3+1 ile gösterilir.

SATURNIN özet fonksiyonu Merkle-Damgard yapısını kullanarak 256-bit özet değeri üretir. 16 tur boyunca çalışır (Merkle, 1990). Özet değeri üretmek için 256-bit zincir değeri ve 256-bit mesaj bloğu kullanır. Başlangıçta tüm bloklar 0 bitleri ile doludur. Son blok hariç tüm bloklara 16 tur boyunca SATURNIN işlemleri 16 tur boyunca uygulanır. Her adımda mesaj bloğu ile SATURNIN işlemlerinden geçen blok $\oplus$ işlemine girer. Son mesaj bloğunda eğer boş bit pozisyonları var ise 10* kuralı ile dolgulama kuralı uygulanır ve dolgulanmış mesaj bloğu SATURNIN işlemlerinden geçer daha sonra çıktı bloğu ile dolgulanmış mesaj bloğu $\oplus$ işlemine girer. Bu zincirleme yapı sonucunda 256-bit özet değeri oluşturulmuş olur.

SATURNIN 16 tur S-Box yapısının diferansiyel düzenliliği 80 olarak belirtilmiştir. Bunun anlamı en yüksek diferansiyel olasılığının 80 x $2^{-16} = 2^{-9.68}$ olmasıdır. AES benzeri adım işlemleri kullanması 4 turda 25 aktif S-Box yapısı olacağını gösterir (Daemen and Rijmen, 2001). 4 ve 8 turda sağladığı en iyi diferansiyel karakteristiği sırasıyla $2^{-241.9}$ ve $2^{-483.9}$ olarak belirtilmiştir. SATURNIN S-Box yapısının doğrusallığı Walsh dönüşümü ile ölçülmüştür. S-Box yapısının doğrusallığı 3072 olarak gösterilmiştir. Doğrusal yollar için en yüksek kare korelasyon ise 4 ve 8 tur için sırasıyla $2^{-220.7}$ ve $2^{-441.5}$ olarak belirtilmiştir. SATURNIN 4-bit uzunluklu S-Box yapılarının cebirsel derecesi 9 olarak hesaplanmıştır. Bu S-Box yapılarının terslerinin de cebirsel derecesi yine 9 olarak hesaplanır. Bu kriptanalizlerin yanında SATURNIN özet fonksiyonunun çakışma direnci 2^{128} olarak ölçülmüştür. SATURNIN özet fonksiyonu için klasik bir ön-görüntü ve ikinci ön-görüntü atağı yoktur (Canteaut at al, 2019).

3.21 ORANGE

ORANGE sistemi doğrulanmış şifreleme ve özet fonksiyonundan oluşur. ORANGE sistemindeki özet fonksiyonu ORANGISH ile adlandırılır. ORANGISH özet fonksiyonu Sponge yapısı tabanlı 256-bit permütasyon kullanarak 256-bit özet değeri üretir. Sponge yapısına dayandığından parametreleri olan blok uzunluğu 128-bit, kapasite değeri ise 128-bit olarak ayarlanmıştır. ORANGISH sisteminin kullandığı permütasyon 256-bit PHOTON permütasyonudur. 256-bit PHOTON permütasyonu, işlemlerini 8x8x4 boyutuna sahip üç boyutlu bir durum matrisi üzerinde gerçekleştirir. Bu durum matrisi aslında 64 tane 4-bit uzunluklu kelime değerlerinden oluşur (Guo et al., 2011).

PHOTON permütasyonu, tur fonksiyonundan ve yinelemeli 12 turdan oluşur. Tur fonksiyonu ise dört adım işleminden meydana gelir. Bu adım işlemleri tur sabitleri ile toplama, bitlerde yer değiştirme, satırlarda kaydırma ve sütunlarda karıştırmadır. Tur sabitleri ile toplama işleminde, belirlenmiş sabit değerler ile durum matrisi $\oplus$ işlemine girer. Bitlerde yer değiştirme işleminde 4-bit S-Box kullanılır. Satırlarda kaydırma adımında, belirli bir kural ile satırlarda ROT işlemi yapılır. Son olarak sütunlarda karıştırma işleminde, durum matrisi üzerindeki tüm sütunlar aşağıdaki matris ile çarpılır ve elde edilen yeni sütun durum matrisinde ilgili yere yazılır (Chakraborty and Nandi, 2019).

Tablo 3.21.1 ORANGISH Özet Fonksiyonu S-Box Yapısı

x		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S-Box		C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ a0 & a1 & a2 & a3 & a4 & a5 & a6 & a7 \end{pmatrix}$$

ORANGISH özet fonksiyonunun çakışma direnci 112-bit, ön-görüntü direnci 128-bittir. ORANGISH özet foksiyonuna yapılan bir çakışma atağında, saldırgan q tane permütasyon çağırır. İlk durum matrisis bitlerinin tamamının 0 biti olarak tanımlandığını bilelim. Bu durumda saldırgan varsayımlar kullanarak bu başlangıç durum matrisinden farklı durum matrislerine ulaşır. Saldırganın q tane permütasyon ile nq tane durum matrisine ulaşabildiği düşünülmüştür. Dolayısıyla çakışma bulma olasılığı en fazla $n^2q^2/2^{256}$ olarak belirtilmiştir. Ön-görüntü olasılığı ise q değeri permütasyonun ve permütasyonun tersinin çağırılma sayısı olmak üzere $q/2^{128}$ ile sınırlandırılmıştır. Kullanılan PHOTON permütasyonunun analizi ise orijinal PHOTON dokümanında yapılmıştır (Guo et al., 2011).

3.22 GAGE

GAGE, Sponge tabanlı özet fonksiyonudur. Üzerinde işlem yapılacak bitler boyutu 232-bitten 576-bite kadar değişebilen bir durum matrisi üzerine yerleştirilir. Girdi mesajı permütasyon fonksiyonu ile işlenmeden önce bloklara ayrılır. Girdi mesaj bloklarının uzunluğu olan r değeri ise 8-bit, 16-bit, 32-bit veya 64-bit olabilir. Son bloklarda eksik baytlar var ise dolgulama işlemi yapılır. Dolgulama işlemi bayt olarak yapılır. Dolgulama kuralı ise girdi mesajındaki ilk boş bayt pozisyonuna 0x80, diğer bayt pozisyonlarına ise 0x00 baytı yerleştirilerek yapılır.

GAGE permütasyonu tur fonksiyonundan oluşur. GAGE tur fonksiyonu iki işlemden oluşur. Bu işlemler doğrusal olmayan yer değiştirme ve bit karıştırma olarak adlandırılır. Doğrusal olmayan yer değiştirme adımında Q ile isimlendirilen bir 4→2 S-Box kullanır. Q S-Box öncelikle durum matrisi bitlerinin en sağına veya en soluna iki bit sabit ekler. Q S-Box, durum matrisi üzerindeki bitleri iki alt kümeye bölerek bu iki alt küme üzerinde paralel olarak yer değiştirme işlemi yapar. Q S-Box yapısının b-bit durum matrisi üzerindeki bu uygulama tarzı, onu (b+2)→b S-Box yapar. Q S-Box yapısının cebirsel normal formu x_1, x_2, x_3, x_4 girdi değişkenleri olmak üzere, $Q(x_1, x_2, x_3, x_4) = (f_1(x_1, x_2, x_3, x_4), f_2(x_1, x_2, x_3, x_4))$ şeklindedir. f_1 ve f_2 fonksiyonları, + sembolü $\oplus$ işlemi olmak üzere, matematiksel olarak,

$f_1 = x_1 + x_3 + x_2x_3 + x_2x_4$ ve $f_2 = 1 + x_1 + x_2 + x_2x_3 + x_4 + x_2x_4$

şeklinde ifade edilir. GAGE tur fonksiyonunun ikinci adımı bit karıştırmadır. Bit karıştırma adımı doğrusal işlemlere sahiptir ve π_8 ile gösterilir. Durum matrisi

üzerindeki bitler öncelikle bayt olarak gruplandırılır. Daha sonra her baytın bitleri üzerinde aşağıdaki kural ile yer değiştirme işlemi yapılır.

$$\pi_8 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 5 & 8 & 7 & 6 & 2 & 4 \end{pmatrix}$$

GAGE özet fonksiyonunda, durum matrisi üzerine yerleştirilen bitler blok uzunluğu ve kapasite değerinin toplamı kadardır. Girdi mesajı, blok uzunluğu ve kapasite değeri 8-bitin katı olmalıdır çünkü işlemler bayt olarak yapılmaktadır. Girdi mesajına dolgulama işlemi yapıldıktan sonra girdi mesajı bloklara ayrılır. Durum matrisi başlangıçta 0 bitleri ile doldurulur. Emilme aşamasında, mesaj blokları sırasıyla durum matrisinin ilk r-biti ile $\oplus$ işlemine girer daha sonra tur fonksiyonu ile belirlenen tur sayısı kadar işlenir. Tüm mesaj blokları bu aşamadan geçirildikten sonra sıkma aşamasına geçilir. Sıkma aşamasında, emilme aşamasından alınan durum matrisi tur fonksiyonu ile belirli bir tur kadar işlenir ve ilk r-bit kısmı çıktı olarak alınır. Yeni oluşan durum matrisine aynı işlemler uygulanarak durum matrisinin ilk r-biti diğer çıktıya eklenir. İstenilen boyuta ulaşana kadar bu işlem devam ettirilerek özet değeri oluşturulur.

GAGE özet fonksiyonunda önerilen parametreler 232-bit permütasyon, 8-bit blok uzunluğu ve 224-bit kapasite değeridir. Bu parametreler altında GAGE özet fonksiyonu 112-bit çakışma direnci, 112-bit ikinci ön-görüntü direnci ve 223-bit ön-görüntü direnci sağlar. Parametrelerin değiştirilmesi durumunda, h özet değeri uzunluğu ve c kapasite olmak üzere çakışma direnci min(h,c)/2, ikinci ön-görüntü direnci min(h,c/2) ve ön görüntü direnci min(h,c-1) ile hesaplanmaktadır.

GAGE özet fonksiyonunun en küçük doğrusal olmayan yapısı olan 4→2 S-Box Q yapısının diferansiyel karakteristiği incelenmiştir. S-Box Q yapısının diferansiyel karakteristiği oldukça zayıftır. Dört farklı 1 olasılığına sahip hücre bulunmaktadır. Bu hücrelere karşılık gelen dört tane dört bit sırasıyla {0,0,0,0}, {0,0,1,1}, {1,0,0,0} ve {1,0,1,1}olarak gösterilmiştir. GAGE Q S-Box'ta aralıklı biçimde S-Box boyutu (b+2)→b boyutuna genişletilir. b-bit uzunluklu durum matrisi için i = 0, 1, …, b/2 olmak üzere $p_{out} = 2^{-i}$ olasılıkla çıktı diferansiyeline sahip $N_{p=2^{-i}}$ tane girdi diferansiyeli vardır ve $N_{p=2^{-i}} = 4 \times 3^i \times \binom{b/2}{i}$ şeklinde hesaplanır. Sonuç olarak, $\sum_{i=0}^{b/2} N_{p=2^{-i}} = 2^{b+2}$ olmaktadır. Bu durumda GAGE b-bit durum matrisinde 2^{-1} olasılığa sahip hücre sayısı $N_{2^{-1}} = 6\,x\,b$ ile hesaplanmaktadır.

GAGE sisteminin doğrusal karakteristiği diferansiyel karakteristiğine benzemektedir. Doğrusal karakteristiğin en yüksek değeri -8 olarak hesaplanmıştır. GAGE özet fonksiyonunun doğrusal karakteristiğini analiz etmek için 8-bit boyutlu bir durum matrisi için 8→8 S-Box yapısı kullanılmıştır. Farklı tur sayıları için tur fonksiyonu çağırılmıştır ve doğrusal katman kullanılmıştır (Gligoroski et al., 2019).

4. GÜVENLİK TANIMLARI VE KRİPTANALİZ

Kriptografik sistemlere yapılacak ataklar, saldırganın gücü olabildiğince arttırılarak modellenmiştir. Bu modellerden bir tanesi yalnızca şifreli metin atağıdır (ciphertext-only attack). Bu atakta, saldırganın bir veya daha fazla şifreli metni elde ettiği düşünülür. Saldırgan, elindeki şifreli metinleri kullanarak açık metin hakkında bilgi sahibi olmaya çalışır. Başka bir model bilinen açık metin atağıdır (known-plaintext attack). Bilinen açık metin atağında, saldırgan bir veya daha fazla açık metne ve bu açık metinlerin bazı anahtarla şifrenmiş şifreli metinlerine sahiptir. Saldırgan aynı anahtarı kullanarak başka bir şifreli metinden açık metne ulaşmaya çalışır. Seçilmiş açık metin atağı (chosen-plaintext attack) ise bilinen açık metin atağı ile aynıdır. Tek farkı saldırgan açık metinleri kendisi seçebilir. Son bir atak modeli ise seçilmiş şifreli metin atağıdır (chosen-ciphertext attack). Bu atak modelinde, saldırgan seçtiği şifreli metinleri çözmeye çalışır. Saldırganın amacı, açık metni tam olarak elde edemese bile, şifreleme işleminde kullanılan anahtar hakkında bilgi edinmektir.

Her şifreleme algoritmasının güvenliği matematiksel olarak kanıtlanmalıdır. Güvenlik kanıtları, hem şifreleme algoritmasının güvenliğinin matematiksel olarak tanımlanmasına yardımcı olur hem de şifreleme sisteminin matematiksel güvenliğini garantiler. Diğer taraftan, güvenlik kanıtları varsayımlara dayandırılarak yapılmaktadır. Eğer üzerinde durulan varsayımlar geçersiz ise güvenlik kanıtının da bir anlamı kalmaz. Bu sebeple güvenlik kanıtları yapılırken, varsayımlar iyi seçilmelidir. Özellikle saldırganın gücü yüksek tutulmalı ve her durum değerlendirilmelidir. Varsayımlara dayandırılarak yapılan matematiksel güvenlik ispatları hesaplamalı güvenlik alanında incelenir.

Hesaplamalı güvenlikte iki yaklaşım vardır. Bu yaklaşımların ilki somut yaklaşımdır. Somut yaklaşımda, bir kriptografik sisteme atak yapan saldırganın belirli bir süre için veya belirli bir hesaplama gücü ile maksimum başarı olasılığını kesin sınırlarıyla belirleyerek kriptografik sistemin güvenliği ölçülmektedir. Diğer bir yaklaşım ise asimptotik yaklaşımdır. Asimptotik yaklaşım, somut yaklaşımın zorlukları sebebiyle ortaya çıkmıştır. Asimptotik yaklaşım karmaşıklık teorisine dayanmaktadır. Bu sebeple, kriptografik sistem ve saldırgan dahil olmak üzere tüm taraflar parametreleştirilir. Bu yaklaşımda, asimptotik olarak belirlenen n güvenlik parametresi, somut güvenliğe çevrilmeye çalışılır. Somut yaklaşımda hiçbir ihmal durumu olamazken, asimptotik yaklaşımda saldırgan için ihmal edilebilir (negligible) başarı olasılıkları olabilir (Katz and Lindell, 2014).

Başka bir güvenlik alanı ise semantik güvenliktir. Semantik güvenlikte, saldırgan iki açık metinden birini seçer ve o açık metinlerden birinin şifreli metnini alır. Eğer saldırgan verilen şifreli metnin hangi açık metnin şifrelenmiş hali olup olmadığını 1/2 oranından daha iyi olasılıkla tahmin edemez ise şifreleme şeması semantik olarak güvenlidir denir. Aslında bu tanım, kriptografik sistemlerin

rastgele görünüm sağlamaları gibi kriptografik sistemlerle ilgili birçok gereklilik ile bağlantılıdır (Sako, 2011).

4.1 Güvenlik Tanımları

Anlatıldığı gibi, şifreleme sistemlerinin güvenliği belirli yaklaşımlarla ölçülebilir. Bu yaklaşımlara ek olarak oluşturulan şifreleme sisteminin sağlaması gereken özellikler temel tanımlarla da açıklanmıştır. Bu zamana kadar sözel olarak, şifreleme sisteminin güvenliği belirlenirken saldırganın gücünün yüksek tutulması, sistem üzerine yapılacak ataklardaki varsayımların iyi değerlendirilmesi ve şifreleme sistemlerindeki her turda verilecek çıktının rastgelelik özelliğini sağlaması gerektiği anlatılmıştır. Bu bölümde, güvenli şifreleme sistemleri oluşturmak için bazı tanımlar verilecektir.

Bir şifreleme şeması, anahtar üreteci algoritması G, şifreleme algoritması E ve şifre çözme algoritması D ile parametrelendirilir. Anahtar üreteci algoritması, K sonlu anahtar uzayı olmak üzere, $k \in K$ anahtarının oluşturulduğu olasılık algoritmasıdır. Ayrıca, M mesaj uzayı olmak üzere, $m \in M$ açık metni; C şifreli metin uzayı olmak üzere $c \in C$ şifreli metni ifade etmek için kullanılır. $k \in K$ iken $P[Ky = k]$, anahtar üreteci tarafından üretilmiş bir anahtarın k değerine eşit olma olasılığını belirtir. $m \in M$ iken $P[Mg = m]$, şifrelenecek açık metnin m mesajı olmasının olasılığını gösterir. Son olarak $c \in C$ iken $P [Cp = c]$, şifreli metnin sabit bir c değerine eşit olma olasılığını göstermektedir.

Bir şifreleme şeması, sınırsız hesaplama gücüne sahip bir saldırgana karşı dahi güvenli ise bu şifreleme şemasına mükemmel gizli (perfectly secret) denir. Mükemmel gizlilik tanımı Shannon tarafından sunulmuştur (Shannon, 1949). Formal olarak mükemmel gizlilik tanımı Tanım 4.1.1'de verilmiştir.

Tanım 4.1.1 : Bir şifreme şeması (G, E, D), M mesaj uzayında mükemmel gizlidir. Öyle ki; $\forall m \in M$ ve $P[Cp = c]>0$ durumunda $\forall c \in C$ olmak üzere M mesaj uzayı üzerindeki her olasılık dağılımı için $P[Mg = m \mid Cp = c] = P[Mg = m]$ eşitliği sağlanır.

Bir şifreme sistemine atak yapan saldırgan sonsuz hesaplama gücüne sahip olsun. Saldırgana M mesaj uzayından seçilmiş iki adet açık metin verilsin. Daha sonra anahtar üreteci ile üretilmiş rastgele bir anahtar ile bu açık metinlerden bir tanesi şifrelensin ve saldırgana verilsin. Saldırgan şifreli metnin hangi mesaja ait olduğunu tahmin etmeye çalışır. Doğru tahminleri için 1, yanlış tahminleri için 0 yazılır. Eğer saldırgan ½ olasılıktan daha iyi bir ihtimalle doğru metni tahmin edemez ise şifreleme şeması mükemmel ayırt edilemezdir (perfectly indistinguishable). Yapılan deney Prv ile gösterilirse, bu tanım formal olarak Tanım 4.1.3'de verilmiştir.

Tanım 4.1.2 : Bir şifreleme şeması (*G, E, D*), *M* mesaj uzayında mükemmel ayırt edilemezdir. Öyle ki; P [Prv = 1] = ½ .

Lemma 4.1.1 : Şifreleme şeması ES = (*G, E, D*) mükemmel gizlidir ancak ve ancak ES mükemmel ayırt edilemezdir.

Teorem 4.1.1 : Bir şifreleme şeması semantik güvenlidir ancak ve ancak şifreleme şeması ayırt edilemezdir (Goldreich, 2009).

Aslında ayırt edilemezlik kavramı istatistiksel bir kavram olarak düşünülebilir. Bir sözde rastgele sayı üretecinin çıktısının belirli istatistiksel testleri geçmesi gerekir. Örneğin, rastgele görünümlü çıktıdaki 0 ve 1 biti sayısının ½ olması beklenir. Ayırt edilemezlik bunun gibi istatistiksel testlerin beklenen sonuçları verdiği durumlarda kullanılır.

Tanım 4.1.3 : Eğer her pozitif polinom *p* için $i > I$ şartını sağlayan tüm tamsayı *I* değerleri için $f(i) < 1 / p(i)$ ise $f : N \rightarrow R^{+} \cup \{0\}$ fonksiyonu ihmal edilebilirdir.

Tanım 4.1.4 : Herhangi bir *r* ve $t \in \{0,1\}^{r}$ girdisi için *G(t)* fonksiyonunun sonucu *p(r)* uzunluklu bit dizisi olmak üzere *p* bir polinomdur ve aşağıdaki şartlar sağlandığında *G* kararlı polinom zamanlı algoritma (deterministic polynomial-time algorithm) sözde rastgele sayı üretecidir.

- $\forall r$ için $p(r) > r$ olmalıdır ve *p* genişleme faktörüdür.
- Herhangi bir olasılıklı polinom zamanlı algoritma (probabilistic polynomial-time algorithm) *T* için *ng* ihmal edilebilir fonksiyon ve $u \in \{0,1\}^{p(r)}$ olmak üzere; $|P[T(G(t))=] - P[T(u)=1]| \leq ng(r)$.

Bir sözde rastgele sayı üreteci, mesajdan daha kısa bir anahtarla (veya tohum) güvenli, sabit uzunlukta bir şifreleme şeması oluşturmak için kullanılabilmektedir. Tek kullanımlık şeritte girdi mesajı ile aynı uzunlukta bir gizli anahtar $\oplus$ işlemine girer. Fakat gizli anahtar değeri mesajın uzunluğuna bağlı olarak uzayacağından bir tohum değeri ile bu gizli anahtarı üretmek ve tohum değerini karşı tarafla paylaşmak daha mantıklı olacaktır.

Tanım 4.1.5 : G sözde rastgele sayı üreteci ve p genişleme faktörü olmak üzere; p uzunluklu bir mesajın gizli anahtarlı şifreleme şeması aşağıdaki şekilde tanımlanır.

- Anahtar üretimi algoritması, 1^{r} ile $k \in \{0,1\}^{r}$ değerlerini girdi olarak alır ve anahtar değerini çıktı olarak üretir.
- Şifreleme algoritması, $k \in \{0,1\}^{r}$ anahtarı ile $m \in \{0,1\}^{p(r)}$ mesajını girdi olarak alır ve $c = G(k) \oplus m$ işlemi ile şifreli metni çıktı olarak verir.
- Şifre çözme algoritması, $k \in \{0,1\}^{r}$ anahtarı ile $c \in \{0,1\}^{p(r)}$ şifreli metnini girdi olarak alır ve $m = G(k) \oplus c$ girdi mesajını çıktı olarak verir.

Teorem 4.1.2 : Eğer *G* sözde rastgele sayı üreteci ise Tanım 4.1.5'teki sabit uzunluklu gizli anahtarlı şifreleme şeması istenmeyen dinleyici varlığında ayırt edilemezdir.

Çoklu şifreleme güvenliği, bir istenmeyen dinleyici varlığında taraflar birbirine birden çok şifreli mesaj gönderdiğinde oluşan sistemin güvenliği ile ilgilenir. *n* güvenlik parametresi olarak gösterilir ise; Saldırgana 1^n girdisi ve eşit uzunluklu iki farklı mesajın çıktısı verilir. Bu iki mesaj 0 ve 1 ile indekslenmiştir. Daha sonra anahtar üretimi algoritması olan *G* kullanılarak $G(1^n)$ ile anahtar üretilir ve $b \in \{0,1\}$ biti seçilir. Seçilen bite karşılık gelen indeksli mesaj gizli anahtar ile şifrelenir. Şifreli metin saldırgana verilir. Saldırgan şifreli metnin karşılık geldiği mesajın indeksini belirtmek üzere bir $b' \in \{0,1\}$ biti seçer. Eğer $b' = b$ ise saldırgan başarılıdır. Değilse, başarılı olamamıştır.

Tanım 4.1.6 : Eğer olasılıklı polinom zamanlı tüm saldırganlar için $P[Prv(n)=1] \leq ½+ng(n)$ ise (G,E,D) şifreleme seması, istenmeyen dinleyici varlığında çoklu şifrelemelerde ayırt edilemezdir.

Rastgele kahin, şifreleme şemalarının güvenlik ispatlarında kullanılan soyut bir modeldir. Kara kutu olarak da bilinir. Şifreleme şeması normal işlemlerini yaparken, rastgele kahin modeli şifreleme şemasına sorgular gönderen ve sorgu karşılığını çıktı kümesinden bir elemana eşleyen matematiksel fonksiyondur. Asıl amacı özet fonksiyonlarını modellemektir çünkü özet fonksiyonlarının rastgele sayı üreteçleri gibi çalışması beklenmektedir. Saldırgan bir şifreleme sistemine atak yaparken rastgele kahine erişimi olduğu varsayılır. Bu durumda saldırganın başarı olasılığı ihmal edilebilir bir fonksiyon ile ifade edilebilirse şifreleme sistemi ideal bir sistemdir (Canetti et al., 2004).

Seçilmiş açık metin atağının modeli oluşturulurken, saldırganın bilmediği bir anahtar ile saldırganın seçtiği açık metinler şifrelenir. Daha sonra saldırganın şifreleme kahinine ve şifreli mesajlara erişim sağladığı varsayılır. Saldırgan bu rastgele kahine istediği kadar açık metni girdi olarak vererek sonuçlar elde edebilir. Seçilmiş açık metin atağı altında ayırt edilemezlik senaryosunda anahtar üretimi algoritması *G* ile gizli anahtar $k=G(1^n)$ üretilir. Burada *n* güvenlik parametresidir. Saldırgana 1^n girdisi verilir ve şifreleme kahinine erişimi sağlanır. Saldırgan aynı uzunluğa sahip iki mesajı çıktı olarak verir. Mesajlar sırasıyla 0 ve 1 ile indekslenmiştir. Daha sonra bir bit *b* (0 veya 1) seçilir ve bu bite karşılık gelen indeksteki mesaj gizli anahtar ile şifrelenerek, şifreli metin saldırgana verilir. Saldırgan şifreleme kahinini kullanmaya devam eder ve *b'* bitini çıktı olarak döndürür. Eğer $b=b'$ ise saldırgan başarılı olmuştur.

Tanım 4.1.7 : Eğer her olasılıklı polinom zamanlı saldırgan için *ng* ihmal edilebilir fonksiyon olmak üzere $P[Prv(n)=1] \leq 1/2+ng(n)$ ise simetrik şifreleme şeması seçilmiş açık metin atağı altında ayırt edilemezdir veya CPA-güvenlidir denir.

Teorem 4.1.3 : Herhangi bir simetrik şifreleme şeması CPA-güvenli ise çoklu şifrelemeler için de CPA-güvenlidir.

Tanım 4.1.8 : $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ verimli, uzunluk koruyan ve anahtarlanmış bir fonksiyon olsun. Eğer, her olasılıklı polinom zamanlı ayırt edici D için $P[D^F(1^n)=1] - P[D^f(1^n)=1] \leq ng(n)$ ise F sözde rastgele fonksiyondur. İlk olasılık anahtarın düzgün dağılımlı olması, ikinci olasılık ise f fonksiyonunun düzgün dağılımlı olmasıdır.

Tanım 4.1.9 : $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ verimli, uzunluk koruyan ve anahtarlanmış bir permütasyon olsun. Eğer, her olasılıklı polinom zamanlı ayırt edici D için $|P[D^{F,F^{-1}}(1^n)=1] - P[D^{f,f^{-1}}(1^n))=1] \leq ng(n)$ ise F sözde rastgele permütasyondur. İlk olasılık anahtarın düzgün dağılımlı olması, ikinci olasılık ise f permütasyonunun düzgün dağılımlı olmasıdır.

Teorem 4.1.4 : F sözde rastgele fonksiyon olsun. n uzunluklu bir m mesajının $(m \in \{0,1\}^n)$ sabit uzunluklu çıktı veren simetrik şifreleme şeması (G,E,D) ile gösterilsin. G anahtar üretimi algoritması 1^n girdisi ile $k \in \{0,1\}^n$ gizli anahtarını üretsin. Şifreleme algoritması anahtarı, mesajı ve $r \in \{0,1\}^n$ düzgün dağılımlı bit dizisini girdi olarak alsın. Şifreleme işlemi $c=(r,F_k(r) \oplus m)$ ile gösterilsin. Şifre çözme algoritması $m = F_k(r) \oplus (F_k(r) \oplus m)$ ile gösterilir. Bu şifreleme şeması F sözde rastgele fonksiyonu ile birlikte n uzunluklu mesajlar için CPA-güvenlidir.

Seçilmiş şifreli metin atağında, saldırgan seçilmiş açık metin atağına göre daha güçlüdür. Seçilmiş şifreli metin atağında, saldırgan hem kendi seçtiği mesajların şifrelenmiş haline sahiptir hem de seçtiği şifreli metinlerin şifresini çözebilir. Seçilmiş açık metin atağında saldırgan şifreleme kahinine sahipken, seçilmiş şifreli metin atağında saldırgan hem şifreleme hem de şifre çözme kahinine sahiptir. Seçilmiş şifreli metin atağı altında ayırt edilemezlik senaryosunda öncelikle G anahtar üretimi algoritması ile $k=G(1^n)$ anahtarı üretilir. Saldırgana 1^n girdisi verilir ve saldırganın şifreleme ile şifre çözme kahinine erişimi sağlanır. Saldırgan ise aynı uzunluğa sahip iki mesajı çıktı olarak gönderir. Mesajlar 0 ve 1 ile indekslenmiştir. Daha sonra bir b biti seçilir ve seçilen bite karşılık gelen indeksteki mesaj şifreleme algoritması ile şifrelenir. Şifreli metin saldırgana verilir. Saldırgan şifreleme ve şifre çözme kahinine erişim sağlayabilir fakat şifreli metin üzerinde ikinci bir sorgu yapmasına izin verilmemektedir. Son olarak saldırgan da bir b' biti gönderir. Eğer $b=b'$ ise saldırgan başarılı olmuştur.

Tanım 4.1.10 : Eğer her olasılıklı polinom zamanlı saldırganlar için $P[Prv(n)=1] \leq \frac{1}{2} + ng(n)$ ise simetrik şifreleme şeması seçilmiş şifreli metin atağı altında ayırt edilemezdir veya CCA-güvenlidir. (Katz and Lindell, 2014)

4.2 Özet Fonksiyonlarına Yapılan Genel Ataklar

Özet fonksiyonlarına yapılan genel ataklar, özet fonksiyonlarının yapılarından bağımsız olarak gerçekleştirilebilen ataklardır. Özet fonksiyonlarına yapılan genel ataklar çakışma, ön-görüntü ve ikinci ön-görüntü ataklarıdır. Bu atak türleri kaba-kuvvet saldırısı olarak da adlandırılır. Özet fonksiyonlarına yapılan kaba-kuvvet saldırıları, algoritmadan bağımsızdır. Yalnızca özet değerinin boyutuna bağımlı olarak saldırı gerçekleştirilir. Kaba-kuvvet saldırısı, deneme yanılma yoluyla yapılan saldırılardır (Stallings, 2011).

Bir özet fonksiyonunda, iki farklı mesajın aynı özet değerine sahip olması çakışma olarak adlandırılır. Özet fonksiyonlarındaki çakışma problemi güvercin yuvası prensibine benzemektedir. Güvercin yuvası prensibi şu şekilde açıklanabilir; n adet güvercin yuvası, i ≥ 1 tamsayı olmak üzere n+i güvercin var iken ve her güvercini mutlaka bir yuvaya yerleştirmemiz gerekiyorsa en az bir güvercin yuvasında birden fazla güvercin vardır. Eğer bir özet fonksiyonu kullanarak, iki farklı girdiden aynı özet değeri elde edilebilirse, bu durum çakışma olarak adlandırılır. Güvercin yuvası prensibine dayanarak özet fonksiyonlarındaki çakışma direnci hesaplanabilir. Bu problem genişletilerek, özet fonksiyonlarındaki çakışma direncinin hesaplanması doğum günü paradoksuna (Birthday Paradox) bağlanmıştır.

Doğum günü paradoksu veya problemi, rastgele seçilen kişilerden oluşan bir kümedeki bazı ikililerin doğum günlerinin aynı olması ihtimali ile ilgili paradoksu tanımlar. Bu paradoksta her kişinin yılın herhangi bir gününde doğmuş olma olasılığı eşittir. Doğum günü paradoksunun matematiksel olasılık modeli kullanılarak özet fonksiyonlarında çakışma karmaşıklığı düşürülmüştür. Özet fonksiyonlarına yapılan bu atağa doğum günü atağı (Birthday Attack) denir. Doğum günü atağı, kaba-kuvvet saldırısıdır.

Doğum günü paradoksu matematiksel olarak aşağıdaki gibi tanımlanabilir.

$\forall x \geq 0$ için $(1-x) \leq e^{-x}$ eşitsizliği doğrudur. *1* ve *n* sayıları arasında düzgün dağılıma sahip tamsayılar verilsin. Bu tamsayılar arasında bir $k \leq n$ sayısının en az bir kere tekrar etme olasılığı $P(n,k)$ ile gösterilsin. Bu durumda;

$$P(n,k) = 1 - n! / ((n - k)! \ n^k),$$

$$= 1 - ((n \times (n - 1) \times \dots \times (n - k + 1)) / n^k),$$

$$= 1 - [((n-1) / n) \times ((n-2) / n) \times \dots \times ((n-k+1) / n)],$$

$$= 1 - [(1 - 1/n) \times (1 - 2/n) \times \dots \times (1 - (k-1)/n)].$$

Yukarıda verilen eşitsizlik kullanılarak aşağıdaki eşitsizlik elde edilir.

$P(n,k) > 1 - [(e^{-1/n}) \times (e^{-2/n}) \times \dots \times (e^{-(k-1)/n}),$

$> 1 - e^{-(k \times (k-1))/n}.$

Bu eşitsizlik kullanılarak $P(n,k) > ½$ olması için k değeri,

$½ = 1 - e^{-(k \times (k-1))/n}$,

$2 = e^{(k \times (k-1))/2n}$

$ln(2) = (k \times (k-1)) / 2n.$

Büyük bir k değeri için $(k \times (k-1))$ yerine k^2 yazılırsa,

$k = \sqrt{2(ln(2))n} = 1.18\sqrt{n} \approx \sqrt{n}.$

Doğum günü atağı herhangi bir özet fonksiyonu üzerinde ifade edilebilir. H bir özet fonksiyonu ve sabit t değeri özet çıktısı uzunluğu olmak üzere 2^t özet fonksiyonunun çıktı kümesinin eleman sayısı olsun. x ve y değerleri bu özet fonksiyonunun iki farklı girdisi olmak üzere $H(x)=H(y)$ olma olasılığı yukarıdaki denklemden, $k = \sqrt{2^t} = 2^{t/2}$ şeklinde hesaplanır (Bellare and Kohno, 2004; Stallings, 2011). Doğum günü saldırısına göre özet fonksiyonları, verdiği özet değeri çıktısının uzunluğunun yarısı kadar çakışma direnci sağlamaktadır.

Özet fonksiyonlarına yapılar başka bir genel atak ön-görüntü atağıdır. Ön-görüntü atağında, bir özet fonksiyonundan elde edilen özet çıktısından girdi mesajı elde edilmeye çalışılır. H bir özet fonksiyonu $H(m)=h$ ise m mesajının sabit t uzunluğa sahip özet çıktısı olsun. Bu özet fonksiyonunun ön-görüntü atağına karşı sağladığı güvenlik direnci matematiksel olarak 2^t ile ifade edilir. Yani özet fonksiyonlarının ön-görüntü direnci özet çıktısı boyutu kadardır. Bunun sebebi ön-görüntü atağının bir kaba-kuvvet saldırısı olmasıdır. İkinci ön-görüntü atağı ise çakışma atağına çok benzerdir. Çakışma atağında, aynı özet değerine sahip iki farklı mesaj aranırken, ikinci ön-görüntü atağında saldırgan bir mesaja ve o mesajın özet değerine sahiptir. Saldırgan ikinci ön-görüntü atağı yaparken elinde bulunan mesaj ile aynı özet değerine sahip farklı bir mesaj arar. Özet fonksiyonlarının ikinci ön-görüntü direnci, çakışma dirençleri ile aynı olup, $2^{t/2}$ olarak ifade edilir (Katz and Lindell, 2014).

4.3 Kriptanaliz Yöntemleri

Kriptanaliz, şifreleme yapılırken kullanılan yapıları inceleyerek, şifreleme sisteminin güvenlik açıklarının araştırılmasıdır. Kaba-kuvvet saldırısında kullanılan algoritma ile ilgilenilmezken, kriptanalizde tam olarak algoritmadaki açıklar aranmaktadır. 56-bit kısa bir anahtar kullandığı için DES simetrik blok şifre yöntemi kaba-kuvvet saldırısı ile kırılmıştır. Daha sonra ortaya çıkan 3-DES simetrik blok şifre yöntemi ile kriptanaliz teknikleri de popüler hale gelmiştir çünkü

kaba-kuvvet saldırısı uzun anahtar boyutlarında pratikliğini kaybetmiştir (Stallings, 2011; Curtin and Dolske, 1998). Kriptanaliz yöntemleri temel olarak, diferansiyel ve doğrusal olmak üzere iki alanda incelenebilir.

4.3.1 Doğrusal kriptanaliz

Doğrusal kriptanaliz, girdi mesajı ile şifreli mesaj arasındaki doğrusal ilişkileri inceleyen yöntemdir. Açık metin ve şifreli metin arasındaki doğrusal ilişkileri incelemenin yanında, şifreleme şemasındaki doğrusal olmayan adımlar doğrusal olarak ifade edilmeye çalışılır. Doğrusal kriptanaliz, (Matsui, 1994)'de teorik bir çalışma olarak sunulmuştur. İlk olarak DES algoritmasına uygulanmıştır ve başarılı olunmuştur. Doğrusal kriptanalizde, saldırganın elinde şifreleme algoritmasının, açık metnin ve şifreli metnin olduğu varsayılır. Saldırganın amacı, şifreleme algoritmasını kullanarak, açık metin ve şifreli metin arasında yüksek olasılıkla doğrusal ifadeler bulmaktır.

Simetrik şifreleme sistemleri doğrusal olmayan fonksiyonlar içermektedir. Simetrik sistemlerdeki bu doğrusal olmayan fonksiyonlar genelde S-Box yapıları ile sağlanmaktadır. Doğrusal kriptanaliz, şifreleme sistemindeki S-Box yapısının özelliklerini kullanır. Bu sebeple sistemlerdeki aktif S-Box yapıları arttırılır. Aktif S-Box, küçük bit değişimleri ile iki farklı mesajın, algoritmanın her turunda farklı girdileri oluşturmasını sağlar. S-Box yapıları olabildiğince doğrusallıktan uzak seçilir. Böylelikle şifreleme sistemleri doğrusal kriptanalize karşı daha dayanıklı hale gelir. Diğer taraftan, doğrusal kriptanalizde şifreleme algoritmasındaki doğrusal yapılar birleştirilerek de sistem analiz edilebilir.

Doğrusal kriptanalizde önemli olan başka bir nokta ise şifreleme algoritmasındaki tur girdileri ve çıktıları arasındaki doğrusal sapmaların incelenmesidir. Büyük sapma değerleri sistemde analiz yapılmasına izin verir. Bu sebeple şifreleme sistemleri düşük sapma değerlerine sahip olmalıdır. Kriptanalist bir sistemi analiz ederken eğer büyük bir sapma değeriyle karşılaşırsa, büyük ihtimalle sistemde bir güvenlik açığı bulmuş demektir. Sonuç olarak şifreleme sistemlerinin doğrusal kriptanalize dayanıklı olması için aktif S-Box sayısı olabildiğince yüksek tutulmalıdır ve bu durum yayılım işlemleri ile desteklenmelidir. Ayrıca turlarda girdi-çıktı arasındaki sapma değeri 0 değerine yakın olmalıdır.

Doğrusal kriptanalizde bahsedilen doğrusal ifade $x_1 \oplus x_2 \oplus .. \oplus x_n = 0$ gibi gösterilebilir. Bir şifreleme sistemi için bir doğrusal ifadenin doğru olma olasılığı Yardımcı Teorem 4.3.1.1.'de verilmiştir. Yine (Matsui, 1994)'de gösterilmiştir ki; şifreleme sisteminde doğrusal sapmalar, algoritmanın sondan bir önceki turu ile son turu arasında incelenir. Bu kriptanalizi gerçekleştirmek için ise gerekli açık metin ve şifreli metin çiftlerinin sayısı doğrusal sapma ε ile gösterilirse yaklaşık olarak $1/\varepsilon^2$ olarak belirtilir.

Yardımcı Teorem 4.3.1.1 (Pilling-up Lemma): n S-Box sayısı ve $1 \leq i \leq n$ olmak üzere x_i değeri p_i olasılıkla 0, $1-p_i$ olasılıkla 1 değerindedir. Buradan, seçilen $x_1 \oplus x_2 \oplus \ldots \oplus x_n = 0$ doğrusal denklemin doğru olma olasılığı aşağıdaki gibi hesaplanır (Matsui, 1994).

$$\frac{1}{2} + 2^n \prod_{i=1}^{n} (p_i - 1/2)$$

4.3.2 Diferansiyel kriptanaliz

Diferansiyel kriptnaliz ilk olarak Eli Biham ve Adi Shamir tarafından önerilmiştir (Biham and Shamir, 1991). Girdi mesajındaki farklar ile oluşan şifrelenmiş mesajdaki farkları karşılaştıran ve girdi ile çıktı farkları arasındaki ilişkiyi tanımlamaya çalışan yönteme diferansiyel kriptanaliz denir. Blok şifre kriptanalizlerinde kullanılır. Diferansiyel kriptanalizde asıl aranan özellik, girdide belirlenen herhangi bir değişiklikte çıktıdaki farkların rastgele olup olmamasıdır. Eğer rastgele bir değişim gözlenmez ise bu büyük bir güvenlik açığı olarak belirtilir.

Diferansiyel kriptanalizde, şifreleme algoritmasında kullanılan her yapı için girdi ile çıktı farkları kontrol edilir. Simetrik şifreleme algoritmalarının doğrusal olmayan S-Box yapısının girdi ile çıktı farkları öncelikli olarak incelenen kısımdır. S-Box yapısının her girdi farkı için mümkün olan çıktı farkları incelenir ve çıktı farklarının belirli bir olasılıkla dağılımları elde edilir. $\Delta x \rightarrow \Delta y$ gösterimi x girdisinin farklarının y çıktısındaki farklarını ifade etmek için kullanılır. Bu durumun p olasılıkla doğruluğu kabul edilir.

Diferansiyel kriptanalizde girdideki her fark için çıktı farkları çok iyi takip edilmelidir. Bu takip sonrasında şifreleme algoritmasının diferansiyel karakteristiği elde edilebilir. Diferansiyel karakteristik, algoritmanın her turunda girdi farkının oluşturduğu çıktı farkı yollarının her birine verilen isimdir. Bu yolların tümünün kendi olasılık değerleri vardır. Çünkü girdide oluşturulan her fark, şifre yapısını ve aktif S-Box sayısını etkileyebilmektedir. (Biham and Shamir, 1991)'de yapılan karakteristik tanımı Tanım 4.3.2.1'de verilmiştir.

Tanım 4.3.2.1 : n tur karakteristik $\mu = (\mu_p, \mu_\Delta, \mu_t)$ ile gösterilir. $\mu = (\mu_p, \mu_\Delta, \mu_t)$ ifadesindeki μ_p ve μ_t k-bit uzunluklu sayılar ve $\Delta_i = (\alpha_I^i, \alpha_O^i)$ şeklinde ifade edilmek üzere $\mu_\Delta = (\Delta_1, \Delta_2, \ldots, \Delta_n)$ n-elemanlı listedir. Burada k-bit şifreleme sistemindeki blok uzunluğunu temsil etmek üzere α_I^i, α_O^i sayıları k/2-bit uzunluğundadır. Bu durumda karakteristik için aşağıdakilere ihtiyaç duyulur.

$\alpha_I^1 = \mu_p$ değerinin sağdaki bitlerinin yarısı,

$\alpha_I^2 = \mu_p \oplus \alpha_O^1$ değerinin soldaki bitlerinin yarısı,

α_I^n = μ_t değerinin sağdaki bitlerinin yarısı,

α_I^{n-1} = $\mu_t \oplus \alpha_O^n$ değerinin soldaki bitlerinin yarısı.

Ayrıca 2 ≤ i ≤ n-1 olmak üzere $\alpha_O^i = \alpha_I^{i-1} \oplus \alpha_I^{i+1}$ eşitliği sağlanır.

Tanım 4.3.2.2 : Eğer $\alpha_I^i \rightarrow \alpha_O^i$ F fonksiyonunun girdi ve çıktısı olmak üzere p_i^{μ} olasılığına sahip ise μ karakteristiğinin i tur olasılığı p_i^{μ} ile gösterilir.

Bir şifrenin diferansiyel özelliklerinin belirlenmesi için birçok yol vardır. En temel yol, girdi sayısına göre tüm mümkün çıktı farklarının belirlenmesidir. Kesin bir tahmin yapılmak istenildiğinde, bu yol pek kullanışlı değildir çünkü test edilmesi gereken çok fazla sayıda girdi ile çıktı farkı ilişkisi vardır. Başka bir yol, tüm girdi farkları için en yüksek olasılıklı çıktı farklarının belirlenmesidir. Bu olasılıklar belirlenirken, kullanılan aktif S-Box yapılarına dikkat edilmelidir. Aktif S-Box sayısının artma olasılığı p_i, başlangıçtaki S-Box sayısı n ile gösterilirse diferansiyel karakteristik olasılığı aşağıdaki gibi hesaplanır.

$$DP = \prod_i^n p_i$$

Diferansiyel kriptanaliz gizli anahtarlı blok şifreler için kullanılıyor olsa da özet fonksiyonlarında da uygulanabilmektedir. Özet fonksiyonlarına yapılan çakışma atağında aslında aranan şey girdi ve çıktı farkının sıfır olmasıdır. Diferansiyel kriptanaliz kullanılması özet fonksiyonlarına yapılan genel atakların karmaşıklığının azaltılmasına yardımcı olabilir (Biham and Shamir, 1991).

5. SONUÇ

Bu tezde, simetrik kriptografi başlığı altındaki özet fonksiyonları ve genel atakları anlatılmıştır. Öncelikle SHA-3 özet fonksiyonunun daha sonra NIST hafif-siklet kriptografi projesinde ilk elemeyi geçen ve özet fonksiyonu içeren sistemlerin kullandığı yapılar, işlemleri ve kriptanalizi incelenmiştir. Bu incelemeler sonucunda sistemlerin güvenliğinin dayandırıldığı temel tanımlar ve kriptanaliz yöntemleri araştırılmıştır. NIST hafif-siklet kriptografi projesinde ilk elemeyi geçip özet fonksiyonu içeren sistemlerden ve kriptanalizlerinden faydalanarak, yeni oluşturulacak özet fonksiyonları için yol gösterecek bir kaynak oluşturulmuştur.

5.1 Güvenlik

Özet fonksiyonlarının kullanım alanlarının çokluğu ve güvenlik seviyelerinin belirlenmesinin çok zor olması göz önüne alındığında, öncelikle özet fonksiyonunu oluşturan yapıların ve işlemlerin güvenliğine dikkat edilmelidir. Özet fonksiyonuna yapılan genel ataklara karşı direnç hesaplamasında kullandığı yapıların kriptanalizinin büyük katkısı vardır. Diferansiyel ve doğrusal kriptanaliz yöntemleri indirgenmiş tur sayıları üzerinden uygulanır. Bu işlem ile istenen güvenliği sağlamak için optimum tur sayısı ve sistemin cebirsel derecesi belirlenir.

Her şifreleme sisteminde olduğu gibi özet fonksiyonlarında da rastgelelik çok önemli bir özelliktir. Öyle ki özet fonksiyonlarına yapılan genel ataklar rastgelelikle bağlantılıdır. Şifreleme sistemlerinde, rastgele görünümü sağlamak amacıyla şifrelenmiş metinlerdeki 0 ve 1 biti oranının aynı olması istenir. Ayrıca şifrelenmiş metinde tekrar eden bit grupları bulunmamalıdır (Nuriyeva and Karatay, 2017). Aynı özellikler özet fonksiyonları için de geçerlidir. Hem şifrelemede hem de özet fonksiyonlarında çıktının rastgeleliğinin önemli olması sebebi ile ikisi de rastgele kahin olarak kullanılabilir.

Şifreleme sistemlerinde çok önemli olan CCA ve CPA güvenlik kavramları özet fonksiyonlarında kullanılamaz. Bunun sebebi özet fonksiyonları tek yönlü fonksiyonlardır yani şifre çözme işlemi bulunmamaktadır. Bunun yanında özet fonksiyonlarının güvenliğinin belirlenmesindeki en önemli nokta çakışma direncidir. Çakışma direnci için belirtilen değer doğum günü paradoksu ile hesaplanır ve çakışma bulunan bir özet fonksiyonu kesinlikle güvenli değildir.

5.2 Hafif-Siklet Kriptografi

Hafif-siklet kriptografi algoritmaları, enerji tüketimi az olan sensör, çip gibi donanımlarda güvenliği sağlamak için oluşturulmaktadır. Hafif-siklet kriptografide genelde blok şifreler tercih edilirken blok boyutu ve tur sayısı küçük donanımlarda verimli çalışabilecek şekilde seçilmektedir. Bu amaçla başlatılmış NIST Hafif-Siklet Kriptografi Projesi'nde ilk elemeyi 56 adet şifreleme sistemi geçmiştir. Bu sistemler içinde özet fonksiyonu bulunduran 22 sistem Tablo 5.1.1'de listelenmiştir.

Tablo 5.2.1 NIST hafif-siklet kriptografi projesi kapsamında ilk elemeyi geçen ve özet fonksiyonu içeren sistemlerin temel özellikleri

Kriptografik Sistem	Permütasyon Genişliği	Blok Uzunluğu	Kapasite	XOF	Tur
ACE	320	64	256	x	16
ASCON	320	64	256	✓	12
Gimli	384	128	256	✓	24
XOODYAK	384	130	254	✓	12
Coral	256	32	224	x	28
PHOTON-Beetle	256	32	224	x	12
Sneikha	512	256	256	x	8
SIV-TEM-Photon	256	32	224	x	20
CLX	288	32	256	x	2560
DryGASCON	b	r	c	✓	nr
KNOT	256	32-128	224	x	68
HERON	b	r	c	x	32
Subterranean	257	32-33	224	✓	1
TRIAD	256	32-128	224	x	1024
SYCON	320	64	256	x	14
SPARKLE	384	128	256	x	11
SKINNY	384	128	256	x	29
SIV-Rijndael256	256	32-128	224-128	x	14
Shamashash	320	128	192	x	9
SATURNIN	x	256	x	x	16
ORANGE	256	128	128	x	12
GAGE	232	8	224-	x	32

Seçilen kripto-sistemler arasında özet fonksiyonu yapısında yalnızca SATURNIN sistemi Merkle-Damgard yapısını kullanmıştır. Diğer sistemler Sponge yapısını veya Sponge yapısının modlarını tercih etmiştir. Sponge yapısının Duplex ve Monkey-Duplex modlarının yanında sıkça kullanılan başka bir yapı genişletilmiş Sponge yapısıdır. Genişletilmiş Sponge yapısında sıkma aşamasındaki blok boyutu emilme aşamasındaki blok boyutundan uzundur. Bunun amacı özet değerinin oluşturulmasını hızlandırmak ve adım sayısını azaltmaktır. Fakat bu hızlandırmaya karşılık, emilme aşamasında durum matrisi üzerindeki kapasite bitlerinin bir kısmı sıkma aşamasında kullanıldığından özet fonksiyonunun genel ataklara karşı direnci zayıflamaktadır. Bu sistemlere bakılarak, permütasyon

tabanlı sistemlerin hafif-siklet uygulamalarda daha verimli çalıştığı düşünülebilir. Permütasyon tabanlı sistemler genelde tek permütasyon temelinde şifreleme işlemlerini gerçekleştirdiğinden, hafıza konusunda oldukça iyidir. Permütasyon boyutları da hafif-siklet kriptografinin uygulama alanları göz önünde bulundurulduğunda olabildiğince kısa tutulmaya çalışılmıştır.

Sistemler dikkatli bir şekilde incelendiğinde, blok uzunluğu kısmının kısa tutulduğu görülmüştür. Bu durumun iki sebebi olduğu düşünülmektedir. Birinci sebep, Sponge yapısında güvenlik, kapasite bitleri ile sağlandığından kapasite kısmı uzun tutulmaya çalışılmıştır. İkinci sebep ise hafif-siklet kriptografik sistemler sensör veya akıllı kartlar gibi küçük donanımlar için kullanıldığından, çıktı boyutu donanımı yormayacak şekilde belirlenmiştir. DryGASCON ve HERON sistemlerinin özet fonksiyonu için permütasyon uzunluğu, blok uzunluğu, kapasite ve tur sayısı değerleri Tablo 5.1'de verilmemiştir. Çünkü, bu sistemlerin parametreleri cebirsel olarak değişmektedir ve birbirleri ile bağlantılıdır. Ayrıca, ASCON, Gimli, XOODYAK, DryGASCON ve Subterranean sistemleri özet fonksiyonlarının yanında genişletilmiş çıktı fonksiyonu (XOF) da sunmuştur.

3. bölümde anlatılan sistemlerin kriptanalizleri incelendiğinde, birçoğu Sponge yapısına dayandırıldığından dolayı Sponge güvenliğinden sıkça yararlanıldığı gözlenmiştir. Bu sistemlerde özet fonksiyonlarına yapılan genel ataklara karşı güvenlik düzeyleri birbirleriyle çok yakın seviyededir. İncelenen sistemlerden görülmüştür ki, özet fonksiyonlarına yapılan genel atakların önüne geçebilmek için öncelikle kullanılan kriptografik yapının (rastgele permütasyon gibi) güvenlik düzeyi arttırılmıştır. Güvenlik düzeyinin arttırılması, hem parametrelerin boyutları ile hem de tur işlemleri ile sağlanmaktadır.

Tablo 5.2.2 NIST hafif-siklet kriptografi projesi kapsamında ilk elemeyi geçen ve özet fonksiyonu içeren sistemlerin doğrusal olmayan işlemi

ACE	Simeck-Box	**DryGASCON**	S-Box
ASCON	S-Box	**KNOT**	S-Box
Gimli	SP-Box	**HERON**	S-Box
XOODYAK	Keccak χ	**Subterranean**	Keccak χ
Coral	S-Box	**TRIAD**	S-Box
PHOTON-Beetle	S-Box	**SYCON**	S-Box
Sneikha	S-Box	**SPARKLE**	ARX-Box
SIV-TEM-Photon	S-Box	**SKINNY**	S-Box
CLX	NFSR	**SIV-Rijndael256**	S-Box
Shamashash	S-Box	**SATURNIN**	S-Box
ORANGE	S-Box	**GAGE**	S-Box

Tur içinde üzerinde durulan en önemli işlem doğrusal olmayan adımdır. Doğrusal olmayan adımlar, genelde bit veya baytların S-Box yardımı ile değiştirilmesi ile oluşmaktadır. Kullanılan S-Box yapılarının boyutları sistemin güvenlik isteğine ve S-Box yapısının tur sayısına göre değişmektedir. Doğrusal olmayan adımlarda S-Box yapısının yanında, S-Box ile aynı göreve hizmet eden Simeck-Box, SP-Box, ARX-Box, NLFSR ve Keccak sisteminin χ adımı kullanılmıştır. İncelenen sitemlerin neredeyse hepsinin cebirsel derecesi 2 olarak

belirtilmiştir. Bu cebirsel derece doğrusal olmayan adımlar ile sağlanmaktadır ve tur sayısı arttıkça cebirsel derece de artmaktadır. Bunun yanında doğrusal olmayan adımlarda kullanılan işlemlerin tersinin cebirsel derecesi, işlemlerin cebirsel derecesinden daha yüksek olacak şekilde ayarlanmıştır. Yüksek dereceli polinomları çözmek zor olacağından sistemlerin kriptanalizinde doğrusal olmayan adımlar çok önemli rol oynamaktadır.

5.3 Kullanım Alanları

Güvenliği tanımlanan bir özet fonksiyonu farklı sistemlerin içinde kullanılabilmektedir. Özet fonksiyonlarının kullanım alanlarından biri dijital imzalamadır. Özet fonksiyonları dijital imza algoritmalarının tamamında kullanılmaktadır. Özet fonksiyonlarının çok hızlı çalışması ve farklı girdiler için farklı özet değerleri üretmesi dijital imza için özet fonksiyonlarını vazgeçilmez kılmaktadır. Bunun yanında yalnızca özet fonksiyonu kullanılarak oluşturulmuş imza şemaları da mevcuttur. Bu imza şemalarına özet tabanlı imza şemaları denir. Özet tabanlı imza şemaları kuantum güvenliğin yanında yüksek performansa sahiptir. Yalnızca özet fonksiyonları ile oluşturulduğundan kullanılan özet fonksiyonu imzalama performansını etkilemektedir (Karatay et al., 2019).

Özet fonksiyonları parola saklama, büyük boyutlu dosyaların bütünlüğünün kontrolü gibi alanlarda sıkça kullanılırken son dönemlerde popülerleşen blokzincir teknolojisinde de kullanılmaktadır. Kripto-paralar ile ortaya çıkan blokzincir, günümüzde birçok alana entegre edilebilen ve sürekli gelişen bir teknolojidir. Blokzincir, doğası gereği güvenliğini sağlasa da özet fonksiyonlarının blokzincir güvenliğine büyük katkısı vardır. Blokzincirde şimdilik SHA-2 özet fonksiyonu sıkça tercih edilirken hızla gelişen bu teknoloji için yeni özet fonksiyonlarına ihtiyaç duyulabilir (Emec et al., 2019).

KAYNAKLAR DİZİNİ

Aagaard, M., AlTawy, R., Gong, G., Mandal, K. and Rohit, R., 2019, ACE: an authenticated encryption and hash algorithm, Submission to NIST-LWC.

Alkim, E., Bindel, N., Buchmann, J., Dagdelen, Ö., Eaton, E., Gutoski, G., Kramer, J. and Pawlega, F., 2017, Revisiting TESLA in the quantum random oracle model, International Workshop on Post-Quantum Cryptography.

Alkim, E., Ducas, L., Pöppelmann, T., and Schwabe, P., 2016, Post-quantum key exchange-a new hope, USENIX Security Symposium.

Bao, Z., Chakraborti, A., Datta, N., Guo, J., Nandi, M., Peyrin, T. and Yasuda, K., 2019, PHOTON-Beetle authenticated encryption and hash family, Submission to NIST-LWC.

Bao, Z., Guo, J., Iwate, T. and Song, L., 2019, SIV-Rijndael256 authenticated encryption and hash family, Submission to NIST-LWC.

Bao, Z., Guo, J., Iwata, T. and Song, L., 2019, SIV-TEM-PHOTON authenticated encryption and hash family, Submission to NIST-LWC.

Banik, S., Isobe, T., Meier, W. and Todo, Y., 2019, TRIAD v1 a lightweight AEAD and hash function based on stream cipher, Submission to NIST-LWC.

Bellare, M. and Kohno, T., 2004, Hash function balance and its impact on birthday attacks. In *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 401-418p.

Beierle, C., Biryukov, A., Santos, L. C.d., Großschadl, J., Perrin, L., Udovenko, A., Velichkov, V. and Wang, Q., 2019, SCHWAEMM and ESCH: lightweight authonticated encryption and hashing using the SPARKLE permutation family, Submission to NIST-LWC.

Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P. and Sim, A. M., 2019, SKINNY-AEAD and SKINNY-Hash, Submission to NIST-LWC.

Bernstein, D. J., Kölbl, S., Lucks, S., Massolino, P. M. C., Mendel, F., Nawaz, K. and Viguier, B., 2017, Gimli: a cross-platform permutation, *In International Conference on Cryptographic Hardware and Embedded Systems*, Springer, Cham. 299-320p.

Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2009, Keccak sponge function family main document, *Submission to NIST (Round 2)*, 3(30).

KAYNAKLAR DİZİNİ (devam)

Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2011, Cryptographic sponge functions.

Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2012, Permutation-based encryption, authentication and authenticated encryption. *Directions in Authenticated Ciphers*, 159-170p.

Biham, E. and Shamir, A., 1991, Differential cryptanalysis of DES-like cryptosystems, Journal of Cryptology, Vol 4, No 1, 3-72p.

Canetti, R., Goldreich, O. and Halevi, S., 2004, The random oracle methodology, revisited, *Journal of the ACM (JACM)*, *51*(4), 557-594p.

Canteaut, A., Duval, S., Leurent, G., Naya-Plasencia, M., Perrin, L., Pornin, T. and Schrottenloher, A., 2019, SATURNIN: a suite of lightweight symmetric algorithms for post-quantum security, Submission to NIST-LWC.

Chakraborty, B. and Nandi, M., 2019, ORANGE, Submission to NIST-LWC.

Claesen, L. J. M., Daemen, J., Genoe, M. and Peeters, G., 1993, Subterranean: A 600 mbit/sec cryptographic VLSI chip, Proceedings 1993 International Conference on Computer Design: VLSI in Computers & Processors, ICCD '93, Cambridge, MA, USA, IEEE Computer Society, 640-613 p.

Cox, D., Little, J. and O'Shea, D., 1996, Ideals, varieties, and algorithms an introduction to algebraic geometry and commutative algebra (second ed.), Springer Verlag, Berlin.

Curtin, M., and Dolske, J., 1998, A brute force search of des keyspace.; login.

Daemen, J., Hoffer, S., Peeters, M., Van Assche, G. and Van Keer, R., 2019, Xoodyak, a lightweight cryptographic scheme, Submission to NIST-LWC.

Daemen, J., Hoffert, S., Van Assche, G. and Van Keer, R., 2018, Xoodoo cookbook. *IACR Cryptology ePrint Archive*, 767.

Daemen, J., Massolino, P. M. C.and Rotella, Y., 2019, The subterranean 2.0 cipher suite, Submission to NIST-LWC.

Daemen, J. and Rijmen, V., 2001, Reijndael: The Advanced Encryption Standard. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, *26*(3), 137-139p.

Dobraunig, C., Eichlseder, M., Mendel, F. and Schläffer, M., 2016, Ascon v1. 2, *Submission to the CAESAR Competition*.

KAYNAKLAR DİZİNİ (devam)

Emeç M., Karatay M., Dalkılıç G., Alkım E., 2019, Consensus approaches of high-value crypto currencies and application in SHA-3", International Conference on Artificial Intelligence and Applied Mathematics in Engineering, ICAIAME2019, Antalya, Turkey.

FIPS, P., 1999, Data Encryption Standard (DES). *National Institute of Standards and Technology*, *46-3*, *25*(10), 1-22.

Gligoroski, D., Mihajloska, H. and Otte, D., 2019, GAGE and InGAGE, Submission to NIST-LWC.

Goldreich, O., 2009, *Foundations of cryptography: volume 2, basic applications*, Cambridge university press, 383-389p.

Guo, J., Peyrin, T. and Poschmann, A., 2011, The photon family of lightweight hash functions. In: P. Rogaway (Ed.), CRYPTO 2011, LNCS, vol. 6841, Springer, 222-239 p.

Güneysu, T., Lyubashevsky, V., and Pöppelmann, T., 2012, Practical lattice-basel cryptography: a signature scheme for embedded systems, CHES, LNCS 7428, 530-547p.

Karatay M., Demiroz D., Alkım E., Gürsoy A., 2019, Comparing some hash functions in cryptographic signature, International Marmara Sciences Congress Spring, IMASCON'2019, Kocaeli, Turkey.

Katz, J. and Lindell, Y., 2014, Introduction to modern cryptography., Chapman and Hall/CRC.

Kelsey, J., Chang, S. J. and Perlner, R., 2016, *SHA-3 derived functions: cSHAKE, KMAC, TupleHash, and ParallelHash* (No. NIST Special Publication (SP) 800-185 (Draft)), National Institute of Standards and Technology.

KNOT: algorithm specifications and supporting document, 2019, Submission to NIST-LWC.

Koblitz, N. and Menezes, A., 2013, Another look at security definitions, *Advances in Mathematics of Communications*, *7*(1), 1-38p.

Küçük, O., 2012, Design and analysis of cryptographic hash functions, *Leuven: Katholieke Universiteit Leuven*.

Matsui, M., 1994, Linear cryptanalysis method for DES cipher, Advances in CryptologyEUROCRYPT' 93, Springer-Verlag, 386-397p.

KAYNAKLAR DİZİNİ (devam)

Merkle, R., 1990, One way hash functions and DES, In G. Brassard, editor, Advances in Cryptology - CRYPTO ’89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA; Proceedings, volume 435 of Lecture Notes in Computer Science, Springer, 428– 446p.

Montes, M. and Penazzi, D., 2019, Yarara and coral v1, Submission to NIST-LWC.

Naito, Y. and Ohta, K., 2014, Improved indifferentiable security analysis of PHOTON, In Abdalla, M., Prisco, R.D., eds.: Security and Cryptography for Networks - 9th International Conference, 201.

Nuriyeva F., Karatay M., 2017, On the analysis of an encryption sheme, [ITHEA ISS] Scientific and Practical Conference "Theoretical and Applied Aspects of Program Systems Development" ,TAAPSD'2017, Kiev, Ukraine.

Özkaynak, F., 2015, Kriptolojik rasgele sayı üreteçleri, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 8(2):37-45s.

Penazzi, D. and Montes, M., 2019, Shamash and shamashhash version 1, Submission to NIST-LWC.

Riou, S., 2019, DryGASCON lightweight cryptography standardization process round 1 submission, Submission to NIST-LWC.

Saarinen, M-J. O., 2019, Sneiken and sneikha, Submission to NIST-LWC.

Sako, K., 2011, Semantic security, *Encyclopedia of Cryptography and Security*, 1176-1177 p.

Sarkar, S., Mandal, K. and Saha, D., 2019, SYCON v1.0 submission to lightweight cryptographic standarts, Submission to NIST-LWC.

Shannon, C. E., 1949, Communication theory of secrecy systems, *Bell system technical journal*, *28*(4), 656-715p.

Shor, P.W., 1994, Algorithms for quantum computation: discrete logarithms and factoring, In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 124–134 p.

Shor, P.W., 1997, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput., 26(5), 1484–1509p.

Stallings, W., 2011, Cryptography and network security: principles and practice, Upper Saddle River: Pearson.

KAYNAKLAR DİZİNİ (devam)

Standard, S. H., 2002, Federal information processing standards publication 180-2.

Wu, H., and Huang, T., 2019, CLX: a family of lightweight authenticated encryption algorithms, Submission to NIST-LWC.

Ye, D., Shi, D., Ma, Y. and Wang, P., 2019, HERN & HERON: lightweight aead and hash constructions based thin sponge (v1), Submission to NIST-LWC.

TEŞEKKÜR

Tez çalışmam boyunca,

bilgi ve tecrübelerini benden esirgemeyen ve her türlü desteği göstererek yanımda olan danışman hocam sayın Prof. Dr. Urfat NURİYEV'e

bilgi birikiminden ve deneyimlerinden çok şey öğrendiğim, akademik hayatıma yön veren ikinci danışman hocam sayın Dr. Öğr. Üyesi Erdem ALKIM'a

desteğini her zaman hissettiğim aileme ve sevgili arkadaşım Atakan AYLANÇ'a,

2210-C Öncelikli Alanlara Yönelik Yüksek Lisans Burs Programı ile tez çalışmam boyunca maddi destek veren TÜBİTAK-BİDEB'e

sonsuz teşekkürlerimi sunarım.

ÖZGEÇMİŞ

2 Mayıs 1995 tarihinde Balıkesirin Edremit ilçesinde doğdu. Lise eğitimini Edremit Anadolu Lisesi'nde tamamladı. 2013 yılında Ege Üniversitesi Matematik Bölümünü kazandı. 2018 yılında Ege Üniversitesi Matematik Bölümü, Bilgisayar Bilimleri opsiyonundan mezun oldu. Yüksek lisans eğitimine Ege Üniversitesi Matematik Bölümü'nün Bilgisayar Bilimleri bilim dalında devam etti.

E-Posta : karataymlk9@gmail.com

SERTİFİKALAR

2017 TÜBİTAK Kriptoloji Yaz Okulu

2017 Computational Number Theory

2017 ISCTurkey Bilgi Güvenliği ve Kriptoloji

2017 Sakarya Üniversitesi - Yapay Zeka

2017 Sosyal Medya ve Toplum

2017 TAAPSD Ukraine

2018 HAVELSAN Pardus/Linux

2018 Gazi Üniversitesi CTF Siber Güvenlik

2018 Ideathon Endüstri 4.0 – İzmir Yüksek Teknoloji Enstitüsü

2018 ISCTurkey Bilgi Güvenliği ve Kriptoloji

2018 Cisco Networking Academy – Introduction to Cybersecurity

2018 TOBB ETÜ CTF Siber Güvenlik

2018 Deep Learning Türkiye - Mentor

2019 Cisco Networking Academy – Mobility Fundamentals

2019 Genç Beyinler Yeni Fikirler Tez Yarışması - Hakem

2019 BTK Siber Talimhane

2019 STM CTF Siber Güvenlik

2019 IMASCON Fen Bilimleri Kongresi (Yazar)

YABANCI DİL

İngilizce : Yazma – İyi Okuma – İyi Konuşma – Orta

YAYINLAR

1) Nuriyeva F., Karatay M., 2017, “On the Analysis of an Encryption Sheme”, [ITHEA ISS] Scientific and Practical Conference "Theoretical and Applied Aspects of Program Systems Development" ,TAAPSD'2017, Kiev, Ukraine.

2) Emeç M., Karatay M., Dalkılıç G., Alkım E., 2019, “Consensus Approaches of High-Value Crypto Currencies and Application in SHA-3”, International Conference on Artificial Intelligence and Applied Mathematics in Engineering, ICAIAME2019, Antalya, Turkey.

3) Karatay M., Demiroz D., Alkım E., Gürsoy A., 2019, “Comparing Some Hash Functions in Cryptographic Signature”, International Marmara Sciences Congress Spring, IMASCON’2019, Kocaeli, Turkey.

4) Aylanç A., Karatay M., Gursoy K. N., Gursoy A., 2019, “Application of Blockchain Technology on Taxation System”, International Marmara Sciences Congress Autumn, Kocaeli, Turkey.

5) Karatay M., Aylanç A., Gursoy A., Gursoy K. N., 2019, “Structures and Security of the Blockchain Technology”, International Marmara Sciences Congress Autumn, Kocaeli, Turkey.

6) Karatay M., Alkım E., Nuriyev U., 2019, “Efficient Random Number Generation for Lattice-Based Cryptography”, International Marmara Sciences Congress, IMASCON’2019, Kocaeli, Turkey.

YETKİNLİKLER

Kriptografi, Şifreleme Yazılımları, Güvenlik Testleri, Bilgi Güvenliği, Bilgisayar Bilimleri,

Algoritmalar, Microsoft Office, R Programlama Dili, MATLAB, C Programlama Dili, PHP,

JAVA, SQL, Magma, Ayrık Yapılar, Bilgisayar Cebri, C# Programlama Dili.