

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**KRİPTOGRAFİK İŞLEM YAPAN ÜRÜNLERİN COMMON
CRITERIA (CC:2022 RELEASE-1) STANDARDI AÇISINDAN
İNCELENMESİ**

MERT GÜLER

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: DOÇ. DR. SERDAR SÜER ERDEM

TEMMUZ 2025

**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**KRİPTOGRAFİK İŞLEM YAPAN ÜRÜNLERİN
COMMON CRITERIA (CC:2022 RELEASE-1)
STANDARDI AÇISINDAN İNCELENMESİ**

MERT GÜLER

**YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

DANIŞMAN: DOÇ. DR. SERDAR SÜER ERDEM

TEMMUZ 2025

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

**ANALYSIS OF CRYPTOGRAPHIC PROCESSING
PRODUCTS IN ACCORDANCE WITH THE COMMON
CRITERIA (CC:2022 RELEASE-1) STANDARD**

MERT GÜLER

**A THESIS OF MASTER OF SCIENCE
DEPARTMENT OF ELECTRONICS ENGINEERING**

ADVISOR: ASSOC. PROF. DR. SERDAR SÜER ERDEM

JULY 2025

YÜKSEK LİSANS JÜRİ ONAY FORMU

GTÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulunun 23/06/2025 tarih ve 2025/33 sayılı kararıyla oluşturulan jüri tarafından 11/07/2025 tarihinde tez savunma sınavı yapılan Mert GÜLER'in tez çalışması Elektronik Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Doç. Dr. Serdar Süer ERDEM

ÜYE

: Dr. Öğr. Üyesi İhsan ÇİÇEK

ÜYE

: Prof. Dr. Bahattin TÜRETKEN

ONAY

Gebze Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Yönetim Kurulunun
...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

Bugün 2 yaşına giren, göz aydınlığım Fetih Erdem'e.

20.06.2025

ÖZET

Günümüzde bilgi güvenliği, bireylerin mahremiyetini korumanın ötesinde, kurumların operasyonel bütünlüğü ve devletlerin ulusal güvenliği için hayati bir öneme sahiptir. Bilgi sistemlerinin güvenilirliğini sağlamak ve güvenlik işlevlerinin doğruluğunu nesnel kriterlerle değerlendirebilmek amacıyla uluslararası kabul görmüş standartlara gereksinim duyulmaktadır. Bu bağlamda geliştirilen Common Criteria, bilgi teknolojisi ürün ve sistemlerinin güvenlik özelliklerini tanımlamak, doğrulamak ve belgelendirmek için kullanılan kapsamlı bir değerlendirme standardıdır. 1999 yılında uluslararası standart olarak kabul edilen Common Criteria, zaman içinde geçirdiği güncellemelerle günümüzde de bilgi güvenliği ürünlerinin değerlendirilmesinde etkin şekilde kullanılmaktadır. 2022 yılında yayımlanan Common Criteria Release-1 sürümü ise değerlendirme metodolojilerinde önemli yapısal değişiklikler ve modern güvenlik ihtiyaçlarına uyum sağlayan yeni yaklaşımlar içermektedir.

Bu tez çalışmasında, Common Criteria standardının CC:2022 Release-1 versiyonunun genel yapısı detaylı olarak incelenmiş ve kriptografik işlem gerçekleştiren SoftHSM ürünü, algoritma doğrulama testleri kapsamında ATE aşamasında değerlendirilmiştir. SoftHSM üzerinde, PKCS#11 arayüzü kullanılarak Java programlama diliyle AES algoritmasının CTR modunda şifreleme ve şifre çözme işlemleri gerçekleştirilmiş, elde edilen sonuçlar analiz edilmiştir. Doğrulama sürecinin güvenilirliğini artırmak amacıyla FIPS onaylı Bouncy Castle kriptografi kütüphanesi referans olarak kullanılmıştır.

Tez kapsamında yalnızca uygulama boyutu değil, aynı zamanda Common Criteria değerlendirme sürecinin yapısı da ele alınmış; Security Target dokümanının hazırlanmasından başlayarak, güvenlik hedeflerinin belirlenmesi, kullanıcı kılavuzu dokümanlarının incelenmesi, yaşam döngüsü süreçlerinin değerlendirilmesi, fonksiyonel testlerin gerçekleştirilmesi ve zafiyet analizlerinin yapılması gibi temel aşamaların varlığına ve işleyişine kapsamlı biçimde değinilmiştir.

Elde edilen bulgular, yazılımsal HSM ürünlerinin değerlendirme süreçlerine uygulanabilirlik açısından önemli bir örnek teşkil etmekte ve kriptografik ürünlerin doğrulama, güvenilirlik ile sertifikasyon süreçlerine ilişkin farkındalığı artırmaktadır. Bu çalışma, bilgi güvenliği alanında standartlara dayalı ürün geliştirme pratiğine önemli katkılar sağlamaktadır.

Anahtar Kelimeler: Bilgi Güvenliği, Common Criteria, CC:2022, HSM, Algoritma Doğrulama, Kriptografik Ürün Değerlendirmesi

ABSTRACT

In contemporary society, information security is paramount for safeguarding individual privacy, maintaining the operational integrity of organizations, and ensuring national security. This necessitates the establishment of internationally recognized standards for the reliability of information systems and the objective evaluation of security functions. In this context, the Common Criteria serves as a comprehensive evaluation standard for defining, validating, and certifying the security features of information technology products and systems. Recognized as an international standard in 1999, the Common Criteria has undergone periodic updates and continues to be an essential tool in the evaluation of information security products. The Common Criteria Release-1 version, published in 2022, introduces significant structural changes in evaluation methodologies and new approaches to address contemporary security requirements.

This thesis provides an in-depth examination of the overall structure of the Common Criteria standard CC:2022 Release-1 version, with a focus on the evaluation of the SoftHSM product during the ATE phase as part of algorithm validation tests. Encryption and decryption operations on SoftHSM were conducted using the AES algorithm in CTR mode via the PKCS#11 interface with the Java programming language, and the resulting data was analyzed. To enhance the reliability of the validation process, the FIPS-approved Bouncy Castle cryptography library was employed as a reference.

The thesis not only addresses the implementation aspect but also scrutinizes the structure of the Common Criteria evaluation process, encompassing stages such as the preparation of the Security Target document, the definition of security objectives, the review of user guide documentation, the evaluation of lifecycle processes, the conduct of functional tests, and the performance of vulnerability analyses. These stages are comprehensively examined in terms of their existence and operation.

The findings provide a significant example of the applicability of evaluation processes for software-based HSM products and enhance awareness of validation, reliability, and certification processes related to cryptographic products. This study makes substantial contributions to standard-based product development practices in the field of information security.

Keywords: Information Security, Common Criteria, CC:2022, HSM, Algorithm Verification, Cryptographic Product Evaluation

TEŞEKKÜR

Bu yüksek lisans süreci, yalnızca bir akademik yolculuk değil, aynı zamanda hayatımın en özel hikayelerinden birinin parçası oldu. Bu hikayenin içinde benimle birlikte yol yürüyen ve en zor anlarda bile gözlerimin içine bakarak bana başladığımız yeri hatırlatan sevgili eşim Kübra'ya; desteği, sabrı ve ilhamı için minnettarım.

Erdemli duruşuyla gönülleri fethetsin duası ederek, büyümesini heyecanla izlediğim Fetih Erdem'e, kucak dolusu sevgiyle teşekkür ederim.

Eğitim hayatım boyunca beni destekleyen, yanımda olduklarını her daim hissettiren anneme, babama ve kardeşime sonsuz teşekkürlerimi sunarım.

Kapısından her adım attığımda heyecanlandığım, bir parçası olmaktan onur duyduğum Gebze Teknik Üniversitesi'ne ve tüm değerli hocalarıma içtenlikle teşekkür ederim.

Bu çalışmanın, yoluma ışık tutan, kalbime dokunan tüm güzel insanlara sessiz bir teşekkür olarak ulaşmasını arzu ederim.

İÇİNDEKİLER

	Sayfa
ÖZET	vi
ABSTRACT	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
SİMGELER ve KISALTMALAR DİZİNİ	xi
ŞEKİLLER DİZİNİ	xiii
TABLolar DİZİNİ	xiv
1. GİRİŞ	1
2. TEMEL KAVRAMLAR	3
2.1. Kriptografi ve Bilgi Güvenliği Bağlamı	3
2.1.1. Kriptografik İşlem Yapan Ürünler	4
2.1.2. Donanımsal Güvenlik Modülü (Hardware Security Module – HSM)	4
2.2. Common Criteria Standardına Genel Bakış	5
2.2.1. Tarihsel Arka Plan ve Gelişim Süreci	6
2.2.2. Common Criteria 2022 Sürümünün Yapısı	8
2.2.3. Kriptografik Ürünler Açısından CC:2022 Sürümünün Önemi	10
3. COMMON CRITERIA DEĞERLENDİRME SÜRECİ	12
3.1. Değerlendirme Sürecindeki Paydaşlar ve Roller	12
3.2. EAL (Evaluation Assurance Level) Seviyeleri	13
3.3. Değerlendirme Sürecinin Akışı	15
3.3.1. ASE – Güvenlik Hedefi Değerlendirmesi (Security Target Evaluation)	16
3.3.2. ADV – Geliştirme Kanıtlarının Değerlendirilmesi (Development Evaluation)	18
3.3.3. AGD – Kılavuz Dokümantasyonun Değerlendirilmesi (Guidance Documentation Evaluation)	19
3.3.4. ALC – Yaşam Döngüsü Desteği (Life-Cycle Support Evaluation)	20
3.3.5. ATE – Test Etkinliği Değerlendirmesi (Tests for the Evaluated Product)	22
3.3.6. AVA – Zafiyet Değerlendirmesi (Vulnerability Assessment)	23
4. KRİPTOGRAFİK ÜRÜNLERDE ALGORİTMA DOĞRULAMA SÜREÇLERİ	28
4.1. Kriptografik Algoritma Doğrulamanın Önemi	28
4.2. Uluslararası Algoritma Doğrulama Sistemleri	29
4.2.1. NIST – CAVP ve ACVP	29
4.2.2. FIPS 140-3 Standardı	30
4.3. Uygulamalı Algoritma Doğrulama Testi: SoftHSM Örneği	30
4.3.1. SoftHSM Yapılandırması	31
4.3.2. Java ile PKCS#11 Üzerinden Algoritma Doğrulama Süreci	32
4.3.3. AES-CTR Modunda Algoritma Doğrulama Adımları	34
4.3.4. Bellek ve Zaman Performans Analizi	45

4.3.5. Doğrulama Yönteminin Özgünlüğü ve Yaklaşım Farkı	47
4.3.6. Tezin Katkısı ve Özgünlük Değeri	48
5. SONUÇ VE ÖNERİLER	49
5.1. Sonuç	49
5.2. Öneriler	49
KAYNAKLAR	51
ÖZGEÇMİŞ	53
TEZ METNİNDEN TÜRETİLEN YAYINLAR/SUNUMLAR	54

SİMGELER ve KISALTMALAR DİZİNİ

ACVP	: Automated Cryptographic Validation Protocol
ADV	: Development
AES	: Advanced Encryption Standard
AGD	: Guidance Documents
ALC	: Life Cycle Support
API	: Application Programming Interface
ARC	: Architecture
ASE	: Security Target Evaluation
ATE	: Tests
AVA	: Vulnerability Assessment
BİLGEM	: Bilişim ve Bilgi Güvenliği İleri Teknolojiler Araştırma Merkezi
CAVP	: Cryptographic Algorithm Validation Program
CC	: Common Criteria
CCL	: Conformance Claim Logical
CCRA	: Common Criteria Recognition Arrangement
CMC	: Configuration Management Capabilities
CMS	: Configuration Management Scope
CEM	: Common Evaluation Methodology
COV	: Coverage
CTR	: Counter Mode
CTCPEC	: Canadian Trusted Computer Product Evaluation Criteria
DEL	: Delivery
DPT	: Depth
DVS	: Development Security
ECC	: Elliptic Curve Cryptography
ECD	: Extended Component Definition
ECDSA	: Elliptic Curve Digital Signature Algorithm
ETR	: Evaluation Technical Report
FIPS	: Federal Information Processing Standard
FSP	: Functional Specification
FUN	: Functionality
HSM	: Hardware Security Module
IEC	: International Electrotechnical Commission
IMP	: Implementation
IND	: Independent Testing
INT	: Introduction
ISO	: International Organization for Standardization
IV	: Initialization Vector
LCD	: Life Cycle Definition
NIST	: National Institute of Standards and Technology
OBJ	: Security Objectives
OPE	: Operational Environment
PKCS	: Public Key Cryptography Standards
PKI	: Public Key Infrastructure
PQC	: Post Quantum Cryptography
PRE	: Preparative Procedures
REQ	: Requirements

RNG	: Random Number Generator
RSA	: Rivest-Shamir-Adleman
SAR	: Security Assurance Requirements
SFR	: Security Functional Requirements
SHA	: Secure Hash Algorithm
SPD	: Security Problem Definition
ST	: Security Target
TAT	: Test Assurance Tasks
TDS	: TOE Description Specification
TOE	: Target of Evaluation
TSE	: Türk Standartları Enstitüsü
TSF	: TOE Security Functions
TSFI	: TOE Security Function Interface
TSS	: TOE Summary Specification
TÜBİTAK	: Türkiye Bilimsel ve Teknolojik Araştırma Kurumu
VAN	: Vulnerability Analysis

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 3.1: Değerlendirme seviyeleri [8].	14
Şekil 3.2: Zafiyet derecelendirilmesi ve direnç seviyesi [9].	27
Şekil 4.1: Initialization işlemi.	31
Şekil 4.2: Slot görüntüleme işlemi.	31
Şekil 4.3: Oturum açma işlemi.	32
Şekil 4.4: AES anahtarı oluşturma işlemi.	32
Şekil 4.5: AES ile şifreleme işlemi.	32
Şekil 4.6: AES ile şifre çözme.	32
Şekil 4.7: SoftHSM ile PKCS#11 oturum başlatma örneği.	33
Şekil 4.8: AES-128 encryption test vektörleri.	35
Şekil 4.9: AES-192 encryption test vektörleri.	35
Şekil 4.10: AES-256 encryption test vektörleri.	36
Şekil 4.11: AES-128 decryption test vektörleri.	36
Şekil 4.12: AES-192 decryption test vektörleri.	37
Şekil 4.13: AES-256 decryption test vektörleri.	37
Şekil 4.14: AES_CTR_Enc() metodu.	38
Şekil 4.15: AES-192 encryption test vektörleri sonucu.	40
Şekil 4.16: AES_CTR_Dec() metodu.	41
Şekil 4.17: AES-128 decryption test vektörleri sonucu.	43
Şekil 4.18: BC_AES_CTR_Enc() metodu.	43

TABLÖLAR DİZİNİ

	<u>Sayfa</u>
Tablo 3.1: Atak potansiyeli hesaplama kriterleri [9].	25
Tablo 4.1: Bellek ve zaman performansı.	46

1. GİRİŞ

Bilgi ve iletişim teknolojilerinin hızla geliştiği günümüzde, dijital ortamda gerçekleşen veri iletimi ve depolama süreçlerinin güvenliğini sağlamak, bireyler, kurumlar ve devletler açısından kritik bir öncelik haline gelmiştir. Siber tehditlerin çeşitliliği ve karmaşıklığı artarken, bu tehditlere karşı geliştirilen savunma mekanizmalarının da sürekli olarak güncellenmesi ve sağlam temellere dayanması gerekmektedir. Bu bağlamda, güvenliğin temel taşlarından biri olan kriptografi, bilgi güvenliğinin sağlanmasında merkezi bir rol oynamaktadır.

Kriptografi yalnızca teorik algoritmaların uygulanmasından ibaret değildir; aynı zamanda bu algoritmaların güvenli, tutarlı ve doğrulanabilir biçimde çalışmasını sağlayan yazılım ve donanım bileşenlerinin güvenliği de büyük önem taşımaktadır. Elektronik imza üreticilerinden akıllı kartlara, güvenli anahtar yönetim sistemlerinden donanımsal krypto modüllerine kadar pek çok farklı ürün grubu, hem bireysel hem kurumsal kullanımda aktif rol oynamaktadır. Bu ürünlerin güvenlik düzeylerinin nesnel kriterlere göre değerlendirilebilmesi, güvenin tesis edilmesi açısından vazgeçilmezdir.

Bu ihtiyaca cevap olarak geliştirilen Common Criteria (CC) standardı, bilgi teknolojisi ürünlerinin güvenlik işlevlerini ve bu işlevlerin güvenilirliğini değerlendirmek amacıyla uluslararası düzeyde kabul görmüş bir çerçeve sunmaktadır. 1999 yılında uluslararası standart olarak kabul edilen Common Criteria, ilk sürümünden sonra yıllar içerisinde aldığı revizyonlarla günümüzde hala bilgi güvenliği ürünlerinin değerlendirilmesi için başvurulacak bir standarttır. Standardın 2022 yılında yayımlanan son sürümünde önceki sürümlere kıyasla daha modüler, esnek ve değerlendirme süreçlerini daha sistematik hale getiren bir yaklaşım benimsenmektedir.

Bu tez çalışmasında, kriptografik işlem yapan ürünlerin Common Criteria standardı kapsamında nasıl değerlendirilebileceği incelenmektedir. Çalışmanın odağında, Common Criteria standardının istekleri anlatılacak olup 2022 versiyonunun getirdiği yenilikler üzerinde durulacaktır. Değerlendirme süreci içinde önemli yer tutan fonksiyonel testlerden bahsedilecek ve algoritma doğrulama yöntemleri, uluslararası ve ulusal düzeydeki uygulamalarla birlikte ele alınacaktır.

Tezin inceleme kısmında örnek bir ürün grubu olarak Hardware Security Module (HSM) seçilmiş olup, değerlendirme süreci bu ürün tipi üzerinden detaylandırılacaktır. HSM ürünleri, yüksek güvenliikli anahtar yönetimi ve kriptografik işlem gerçekleştirme kabiliyeti ile hem teknik karmaşıklık hem de güvenlik düzeyi açısından bu çalışma için uygun bir örnek teşkil etmektedir.

Bu kapsamda, önce kriptografi ve Common Criteria standardına dair temel kavramlar açıklanacak, ardından HSM ürünü üzerinden CC:2022 değerlendirme süreci adım adım incelenecek ve algoritma doğrulama faaliyetleri bağlamında değerlendirme tamamlanacaktır.

2. TEMEL KAVRAMLAR

Bu bölümde, bilgi güvenliği ve kriptografinin temel ilkeleri ile kriptografik ürünlerin güvenlik değerlendirmesinde kritik öneme sahip Common Criteria standardına genel bir bakış sunulacaktır. İlk olarak, kriptografi ve bilgi güvenliği bağlamındaki temel kavramlar ele alınacak; ardından, Common Criteria standardının yapısı, gelişimi ve kriptografik ürünler açısından taşıdığı önem detaylandırılacaktır.

2.1. Kriptografi ve Bilgi Güvenliği Bağlamı

Bilgi güvenliği, dijital sistemlerdeki verilerin gizliliğinin, bütünlüğünün ve erişilebilirliğinin korunmasını hedefleyen, teknik ve yönetsel önlemleri içeren bütünsel bir yaklaşımdır. ISO/IEC 27000 bilgi güvenliği standardına göre bu ifade, “bilginin gizliliğinin, bütünlüğünün ve kullanılabilirliğinin sağlanması ve sürdürülmesi” olarak tanımlanır [1].

Bu hedeflere ulaşmada en temel tekniklerden biri kriptografidir. Kriptografi, Claude Shannon’ın 1949 tarihli “Communication Theory of Secrecy Systems” adlı çalışmasından bu yana, hem kuramsal hem de pratik güvenliğin temelini oluşturur [2]. Modern kriptografi yalnızca şifreleme ile sınırlı olmayıp, veri bütünlüğü, kimlik doğrulama ve inkar edilememe gibi işlevleri de kapsar.

Kriptografinin sağladığı başlıca güvenlik ilkeleri şunlardır [3]:

- Gizlilik (Confidentiality): Bilginin yetkisiz kişilerce erişilememesini sağlar.
- Bütünlük (Integrity): Verinin izinsiz veya fark ettirilmeden değiştirilmemesini güvence altına alır.
- Kimlik Doğrulama (Authentication): Tarafların gerçekten iddia ettikleri kişi veya sistem olduğunu doğrular.
- İnkâr Edilemezlik (Non-repudiation): Gönderenin yaptığı işlemi sonradan reddetmesini engeller.

Bu güvenlik servisleri, dijital imza sistemlerinden elektronik oylamaya, finansal işlemlerden kamu veri altyapılarına kadar pek çok kritik uygulamanın temelini oluşturmaktadır.

2.1.1. Kriptografik İşlem Yapan Ürünler

Kriptografik işlem yapan ürünler, temelinde bir veya daha fazla kriptografik algoritmayı uygulayan ve güvenlik servislerini sağlayan yazılım, donanım ya da entegre sistem bileşenleridir.

Bu ürünler çok çeşitli biçimlerde olabilir:

- Yazılım tabanlı çözümler: Örneğin, anahtar yönetim yazılımları, dijital imza üretim yazılımları, güvenli e-posta sistemleri.
- Donanım tabanlı çözümler: Güvenli mikrodenetleyiciler, güvenlik token'ları, HSM cihazları.
- Karma çözümler: Örneğin, yazılımı içinde barındıran güvenli donanımlar.

Common Criteria gibi sertifikasyon sistemleri, bu ürünlerin güvenliğini değerlendirmek amacıyla uluslararası ortak bir dil ve metodoloji sunar. Kriptografik ürünlerin güvenlik değerlendirmesi, yalnızca algoritma seviyesinde değil, aynı zamanda ürünün genel mimarisi, saldırı yüzeyi, güvenlik politikaları ve kullanıcı etkileşimleri de dikkate alınarak yapılır [4].

2.1.2. Donanımsal Güvenlik Modülü (Hardware Security Module – HSM)

Donanımsal Güvenlik Modülü, kriptografik anahtarların güvenli şekilde üretilmesi, saklanması, işlenmesi ve yönetilmesi amacıyla tasarlanmış özel donanım cihazlarıdır. HSM'ler, yüksek güvenlik gereksinimleri olan uygulamalarda (örneğin dijital sertifika hizmetleri, finansal işlemler, elektronik imza üretimi, kimlik doğrulama altyapıları gibi alanlarda) yaygın biçimde kullanılmaktadır. Bu cihazlar, hassas kriptografik işlemlerin fiziksel ve mantıksal olarak güvenli bir ortamda gerçekleştirilmesini sağlar.

NIST'e göre bir HSM, “anahtarların dış ortama çıkmadan kriptografik işlemlerde kullanılmasını sağlayan, fiziksel olarak manipülasyona dirençli güvenli donanım platformudur” [5]. Bu cihazlar, FIPS 140-3 standardı kapsamında değerlendirilen kriptografik modüllerin bir alt kümesi olarak kabul edilir.

HSM'lerin temel işlevleri şunlardır:

- Anahtar üretimi ve saklama: Rastgele sayı üretimi kullanılarak oluşturulan kriptografik anahtarlar, HSM içinde güvenli bellekte saklanır.

- Kriptografik işlem gerçekleştirme: Şifreleme, şifre çözme, imzalama, doğrulama gibi işlemler anahtar dışarıya çıkmadan HSM içinde yapılır.
- Anahtar erişim kontrolü: Kimlik doğrulama, erişim politikaları ve denetim mekanizmalarıyla, yalnızca yetkili işlemlerin gerçekleştirilmesine izin verilir.
- Fiziksel güvenlik: Cihazlar, donanımsal müdahalelere karşı koruma sağlayan dedektörlerle ve silme mekanizmalarıyla donatılır.

HSM cihazları; bankacılık, açık anahtar altyapısı (PKI), ödeme sistemleri, dijital kimlik sistemleri gibi alanlarda yüksek güvenlik seviyesi (High Assurance Level) gerektiren uygulamalarda kullanılır. Bu nedenle, bu ürünlerin sadece işlevsel değil, aynı zamanda güvenlik yönünden de değerlendirilmesi gerekir.

Common Criteria değerlendirme süreçleri, HSM gibi karmaşık sistemler için ürünün güvenlik hedeflerinin açık bir biçimde tanımlanmasını, uygulanabilir güvenlik işlevlerinin belirtilmesini ve bu işlevlerin doğruluğunun nesnel testlerle gösterilmesini öngörür [6].

Özellikle CC:2022 Release-1 ile gelen modüler değerlendirme yapısı ve değerlendirme faaliyetlerine ilişkin yeni çerçeve dokümanı olan “Part 4: Framework for the Specification of Evaluation Methods and Activities”, HSM gibi donanım tabanlı ürünler için daha sistematik ve açık bir yaklaşım sunmaktadır. Bu tez çalışmasında HSM ürünü, kriptografik işlem yapan ürünlerin değerlendirilmesinde örnek bir senaryo olarak ele alınacaktır.

2.2. Common Criteria Standardına Genel Bakış

Bilgi teknolojisi sistemlerinin güvenlik değerlendirmelerinde en önemli gereksinimlerden biri, güvenlik düzeylerinin nesnel, tutarlı, tekrarlanabilir ve uluslararası ölçekte kabul görmüş standartlar doğrultusunda belirlenmesidir. Bu ihtiyaç, farklı ürün ve sistemlerin güvenlik özelliklerinin karşılaştırılabilir ve güvenilir bir şekilde değerlendirilmesini mümkün kılar. Bu amaca hizmet etmek üzere geliştirilen Common Criteria, bilgi teknolojisi ürünlerinin güvenlik gereksinimlerini tanımlamak, değerlendirmek ve belgelemek için uluslararası geçerliliğe sahip bir standart olarak kabul edilmektedir. Resmi adıyla Common Criteria for Information Technology Security Evaluation olan bu standart, ISO/IEC 15408 serisi kapsamında yayımlanmıştır. Hem ürün geliştiricilerine hem değerlendiricilere hem de kullanıcı ve

müşterilere ortak bir dil ve referans noktası sağlayarak, güvenlik ürünlerinin objektif ve karşılaştırılabilir şekilde değerlendirilmesine olanak tanır.

2.2.1. Tarihsel Arka Plan ve Gelişim Süreci

Common Criteria standardı, 1990'lı yılların ortasında Amerika Birleşik Devletleri'nin TCSEC (Orange Book), Kanada'nın CTCPEC ve Avrupa'nın ITSEC gibi farklı güvenlik değerlendirme yaklaşımlarını birleştirme çabasının bir sonucu olarak ortaya çıkmıştır. Bu üç temel kaynağın harmanlanmasıyla ilk olarak 1999 yılında ISO/IEC 15408 standardı kabul edilmiş, ardından çeşitli revizyonlarla yeni sürümleri yayımlanmıştır.

Teknolojik gelişmelerin hız kazanmasıyla birlikte; mobil cihazlar, gömülü sistemler, bulut çözümleri ve kriptografik donanımlar gibi alanlarda güvenlik gereksinimleri de çeşitlenmiş ve artmıştır. Bu gelişmelere uyum sağlamak amacıyla Common Criteria standardı, önceki sürümlerindeki eksiklikleri gidererek yeni yaklaşımlar içeren güncellenmiş sürümlerle geliştirilmeye devam etmektedir. Son olarak Kasım 2022'de yayımlanan CC:2022 Release-1 sürümü, 30 Haziran 2024 itibarıyla zorunlu hale getirilmiş ve önceki sürümlerin kullanımını geçersiz kılmıştır.

Common Criteria standardı, yalnızca teknik bir değerlendirme çerçevesi olmakla kalmaz; aynı zamanda uluslararası iş birliği çerçevesinde sürdürülen bir sertifikasyon ve tanıma mekanizmasıdır. Bu mekanizma, Common Criteria Recognition Arrangement (CCRA) çatısı altında işler ve ülkeler arasında karşılıklı tanıma esasına dayanır. CCRA'nın temel amacı, bir ülkede yapılan değerlendirmenin diğer ülkelere de kabul edilmesini sağlayarak, güvenlik değerlendirmelerinde çifte sertifikasyon ihtiyacını ortadan kaldırmaktır.

CCRA'ya taraf olan ülkeler iki kategoriye ayrılır:

- Sertifika veren ülkeler (Certificate Authorizing Members): Bu ülkeler, değerlendirme laboratuvarlarını akredite edebilir, değerlendirme süreçlerini denetleyebilir ve ürünlere resmi sertifika verebilir. Şu anda sertifika verme yetkisine sahip ülkeler şunlardır [7]:
 - Almanya
 - Amerika
 - Avustralya

- Fransa
 - Güney Kore
 - Hindistan
 - Hollanda
 - İspanya
 - İsveç
 - İtalya
 - Japonya
 - Kanada
 - Katar
 - Malezya
 - Norveç
 - Polonya
 - Singapur
 - Türkiye
- Sertifikaları tanıyan ülkeler (Certificate Consuming Members): Bu ülkeler, kendi ülkelerinde değerlendirme yapmasalar da CCRA kapsamında verilen sertifikaları tanır. Bu ülkeler ise şunlardır:
 - Avusturya
 - Belçika
 - Çek Cumhuriyeti
 - Danimarka
 - Endonezya
 - Etiyopya
 - Finlandiya
 - İngiltere
 - İsrail
 - Macaristan
 - Pakistan
 - Slovakya
 - Ukrayna
 - Ürdün
 - Yeni Zelanda
 - Yunanistan

Bu yapı sayesinde, örneğin Almanya’da Common Criteria sertifikası almış bir ürün, Türkiye’deki kamu kurumlarında veya Japonya’daki güvenlik standartlarına tabi bir sektörde de geçerli kabul edilmektedir. Bu durum, özellikle kriptografik donanım ürünleri gibi uluslararası pazara sunulan yüksek güvenli ürünler için ticari ve teknik açıdan ciddi bir avantaj sağlar.

Ayrıca CCRA, teknolojideki değişimlere uyum sağlamak amacıyla düzenli olarak güncellenmekte ve yeni versiyonlara (örneğin CC:2022) geçişte üye ülkelerin katkısıyla yön belirlemektedir.

2.2.2. Common Criteria 2022 Sürümünün Yapısı

CC:2022 sürümü, önceki sürümlerde alışılmış olan Part 1–2–3 yapısından farklı olarak, beş ana doküman ile bunlara bağlı değerlendirme faaliyetlerini tanımlayan CEM (Evaluation Methods and Activities) dokümanı etrafında yapılandırılmıştır. Her bir bölüm, değerlendirme sürecinin farklı bir yönünü kapsamakta ve önceki sürümlerdeki içerikler yeniden yapılandırılmış biçimde sunulmaktadır:

- Part 1 – Introduction and General Model:
 - Standardın kapsamını, temel kavramlarını ve değerlendirme sürecinin genel yapısını tanımlar. Bu bölüm, Common Criteria’nın mantıksal temelini ortaya koyar ve terminolojik netlik sağlar.
- Part 2 – Security Functional Components:
 - Ürünlerin yerine getirmesi beklenen güvenlik işlevlerinin sınıflandırıldığı kısımdır. CC:2022 sürümünde bu işlevler modüler bileşenler halinde yeniden organize edilmiştir. Örneğin kriptografik işlevlere, random number generation (RNG) ailesi dahil edilerek değerlendirme altındaki ürünlerin işlevleri geniş kapsamda ele alınır hale getirilmiştir.
- Part 3 – Security Assurance Components:
 - Bu bölüm, ürünün güvenliğine dair ne tür doğrulama ve değerlendirme faaliyetlerinin yapılması gerektiğini tanımlayan güvence bileşenlerini (SAR – Security Assurance Requirements) içerir. SAR’lar, geliştirilen ürünün güvenliğini kanıtlamak için geliştirici tarafından sağlanması gereken belgeler, analizler ve süreçleri kapsar. Aynı zamanda, değerlendirme sürecinde hangi yönlerin kontrol edileceğini belirler.

- Part 4 – Framework for the Specification of Evaluation Methods and Activities:
 - 2022 sürümünün en dikkat çekici yeniliklerinden biridir. Değerlendirme yöntemlerinin ve faaliyetlerinin nasıl belirleneceğini sistematik bir şekilde açıklayan bu bölüm, özelleştirilebilir değerlendirme profilleri (Evaluation Activities) tanımlanmasına imkan tanır. Böylece yalnızca belge incelemesi değil, aktif test ve analiz faaliyetlerinin de yöntemsel olarak tanımlanması mümkün olur.
- Part 5 – Pre-defined Packages of Security Requirements:
 - Ürün bileşenlerinin daha önce değerlendirilmiş başka bileşenlerle nasıl birleştirileceğini ve yeniden kullanılabilir güvenlik değerlendirmelerinin nasıl yapılabileceğini ele alır. Özellikle modüler ürün mimarileri ve yazılım/donanım ayrımı olan sistemlerde önemli avantajlar sağlar.
- CEM – Evaluation Methods and Activities:
 - CC:2022 mimarisinin temel bileşenlerinden biri olan CEM dokümanı, Part 1–5’te tanımlanan gereksinimlerin nasıl değerlendirileceğini metodolojik olarak ortaya koyar. Değerlendirme faaliyetlerinin sistematik, tekrarlanabilir ve şeffaf şekilde yürütülmesini amaçlayan bu doküman; değerlendiriciler için gerekli test, analiz ve doğrulama yöntemlerini tanımlar. Aynı zamanda, değerlendirme profillerinin (Evaluation Activities) hazırlanmasına ve her ürün türüne uygun özelleştirilmiş değerlendirme yaklaşımlarının geliştirilmesine olanak sağlar. Bu yönüyle CEM, yalnızca teorik güvenlik gereksinimlerinin değil, uygulamalı güvenlik doğrulamasının da temel referansıdır.

Yeni sürüm, teknolojiye özel değerlendirme profillerinin (Protection Profiles) kullanımını teşvik ederken, aynı zamanda bunlara eşlik edecek değerlendirme faaliyetlerinin de standart bir biçimde yazılmasını öngörmektedir. Bu yaklaşım şu avantajları sağlar:

- Kriptografik ürünler, akıllı kartlar, mobil güvenlik çözümleri, ağ cihazları gibi farklı kategoriler için özel şablonlar geliştirilebilir.
- Aynı türdeki ürünler için daha adil, ölçülebilir ve benzer derinlikte değerlendirme yapılabilir.
- Ülke veya sektör bazlı farklı düzenleyici ihtiyaçlara uyum sağlanabilir.

Bu bağlamda, örneğin HSM cihazları için tanımlanmış bir Protection Profile ve buna bağlı Evaluation Activities seti, hem değerlendirme sürecini kısaltır hem de değerlendirme kalitesini artırır.

2.2.3. Kriptografik Ürünler Açısından CC:2022 Sürümünün Önemi

Kriptografik işlem yapan ürünlerde, güvenlik yalnızca algoritmanın doğruluğu ile değil; anahtar yönetimi, fiziksel koruma, algoritma seçimleri, uygulama arayüzlerinin güvenliği ve rastgele sayı üretimi gibi birçok parametreyle doğrudan ilişkilidir. Bu karmaşık yapının nesnel şekilde değerlendirilebilmesi için CC:2022'nin sunduğu modülerlik ve özelleştirilmiş değerlendirme aktiviteleri kritik bir avantaj sağlar. Örneğin bir HSM ürününün, hem algoritma düzeyinde (örn. AES, RSA, ECC) hem de donanım güvenliği düzeyinde (örn. fiziksel müdahale koruması, erişim denetimi) değerlendirilmesi gerekir. CC:2022, bu tür çok katmanlı analizlerin yönetilebilir olmasına olanak tanır. Kriptografik işlem yapan ürünler hem yazılım hem donanım unsurlarını içermeleri nedeniyle değerlendirmesi zor sistemlerdir. Bu tür ürünlerde dikkat edilmesi gereken bazı özgün konular şunlardır:

- Anahtar yönetimi ve anahtar izolasyonu
- Fiziksel güvenlik ve müdahale tespiti
- Algoritma doğruluğu ve uygulama güvenliği
- Rastgelelik ve entropi kaynakları
- Performans-güvenlik dengesi

CC:2022 ile bu tür ürünlerin değerlendirilmesinde kullanılan yöntemler artık daha açık, sistematik ve belgelenebilir hale gelmiştir. Ayrıca “Part 5 – Pre-defined Packages of Security Requirements” ile, daha önce değerlendirilmiş kriptografik modüllerin yeniden kullanımı mümkün hale gelmekte, bu da karmaşık sistemlerde modüler değerlendirme olanağı sunmaktadır. Örneğin, HSM içerisinde daha önce Common Criteria sertifikasına sahip bir kriptografik kütüphane varsa, bu modülün yeniden kullanımı ve modüler değerlendirme, CC:2022 Part 5 ile resmiyet kazanmıştır. Bu, değerlendirme sürecini hem hızlandırmakta hem de maliyeti azaltmaktadır.

Sonuç olarak, CC:2022 Release-1 sürümü, hem değerlendirme sürecinin kalitesini artırmış hem de ürün geliştiricilerine daha öngörülebilir ve yönetilebilir bir güvenlik değerlendirme süreci sunmuştur. Kriptografik işlem yapan ürünler gibi yüksek

güvenlikli sistemler için bu deęişiklikler, hem pratik hem de stratejik düzeyde ciddi kazanımlar sağlamaktadır.

3. COMMON CRITERIA DEĞERLENDİRME SÜRECİ

Common Criteria, yalnızca bir güvenlik çerçevesi değil; aynı zamanda değerlendirme süreci boyunca izlenecek belgelendirme, test etme ve analiz etme yöntemlerini tanımlayan bir metodolojidir. Bu bölümde, Common Criteria'nın temel bileşenleri olan Güvence Seviyeleri, Security Target dokümanı ve değerlendirme adımları detaylı olarak incelenecektir.

3.1. Değerlendirme Sürecindeki Paydaşlar ve Roller

Common Criteria standardı kapsamında üç temel paydaş grup yer almaktadır:

- Geliştirici (Developer):
 - Güvenlik işlevlerini uygulayan ve belgelerini hazırlayan ürün geliştiricisi tarafıdır. Geliştirici aynı zamanda Common Criteria sertifika müşterisi olarak da adlandırılabilir.
- Değerlendirici (Evaluator):
 - Ürünün güvenlik hedeflerine uygunluğunu ve belgelenmiş işlevlerin doğruluğunu kontrol edip ürüne bağımsız testler uygulayan akredite laboratuvar tarafıdır. Türkiye’de, Common Criteria standardı kapsamında değerlendirme faaliyetleri yürüten çeşitli değerlendirme laboratuvarları bulunmaktadır. Bu laboratuvarlar arasında, Türkiye’nin ilk Common Criteria değerlendirme laboratuvarı olma özelliğine sahip olan ve faaliyetlerini TÜBİTAK BİLGEM çatısı altında sürdüren OKTEM Değerlendirme Laboratuvarı öne çıkmaktadır.
- Sertifikasyon Kurumu (Certification Body):
 - Değerlendirme sonuçlarını denetleyerek ürüne resmi sertifikayı veren otoritedir. Türkiye’de Common Criteria sertifikası veren makam Türk Standartları Enstitüsü’dür.

3.2. EAL (Evaluation Assurance Level) Seviyeleri

Evaluation Assurance Level, Common Criteria kapsamında bir ürünün güvenlik değerlendirmesinin derinliği ve titizliği konusunda sistematik bir referans sunar. EAL, bir ürünün güvenlik iddialarının ne ölçüde incelendiğini ve doğrulandığını ifade eden, artan düzeyde güvenlik güvence faaliyetlerini temsil eden yedi ayrı seviyeden (EAL1'den EAL7'ye) oluşur. Her bir EAL seviyesi, ürünün güvenlik hedeflerine yönelik tasarım, uygulama ve test faaliyetlerine ilişkin güvence düzeyini belirler. Alt seviyelerde daha az detaylı incelemeler yapılırken, üst seviyelerde hem geliştirici hem de değerlendirici tarafından daha kapsamlı belgeler sunulmakta ve daha ayrıntılı test ve analizler gerçekleştirilmektedir.

EAL seviyeleri aşağıda özetlenmiştir:

- EAL1 – Ürünün temel güvenlik işlevlerinin belgelendiği ve bağımsız testlerle doğrulandığı en düşük güvence seviyesidir. Hızlı değerlendirme ihtiyacı duyulan ticari ürünler için uygundur.
- EAL2 – Geliştiriciden daha fazla dokümantasyon istenir. Ürünün yapısal tasarımı ile işlevleri arasında tutarlılık araştırılır. Mevcut bir ürünün sınırlı değişikliklerle değerlendirilmesi hedeflenir.
- EAL3 – Geliştirici tarafından sağlanan belgeler daha sistematiktir ve değerlendirme faaliyetleri daha geniştir. Tehditlere karşı daha yüksek düzeyde güvence sağlar. Ürün tasarım sürecinin kontrollü olduğu varsayılır. Saha ziyareti sürecini zorunlu kılar.
- EAL4 – Yaygın olarak kullanılan güvence seviyesidir. Geliştirici, ürünün güvenlik özelliklerini yöntemsel biçimde tanımlar. Kod incelemeleri, daha kapsamlı testler ve yönetsel süreçlerin denetimini içerir.
- EAL5 – Ürün, daha yüksek risklerin söz konusu olduğu ortamlarda kullanılmak üzere geliştirilmişse tercih edilir. Tasarımın sistematik ve açık bir şekilde tanımlanmasını ve saldırı potansiyeline karşı daha derinlemesine analiz gerektirir.
- EAL6 – Sistem tasarımının ve uygulamasının detaylı incelenmesini, kod yapısına ilişkin ek analizleri ve sıkı denetimleri içerir. Yüksek güvenlik gereksinimlerine sahip, karmaşık sistemlerin değerlendirilmesine olanak sağlar.

- EAL7 – En yüksek güvence seviyesidir. Kritik altyapılar veya yüksek güvenlik gerektiren askeri sistemler gibi alanlar için öngörülür. Değerlendirme oldukça zaman alıcı ve maliyetlidir.

Security Target dokümanında belirtilen EAL seviyesine göre, Common Criteria Part 5 kapsamında ilgili Güvence Gereksinimleri ve bunların kapsamı aşağıdaki şekilde gösterilmektedir. Bu şekil, değerlendirmenin hangi güvence bileşenlerini içermesi gerektiğini ve bu bileşenlerin ne kadar detaylı ele alınacağını belirleyerek, değerlendirme sürecinin sistematik ve tutarlı bir şekilde yürütülmesini sağlar.

Assurance class	Assurance family	Assurance components by evaluation assurance level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Development	ADV_ARC		1	1	1	1	1	1
	ADV_FSP	1	2	3	4	5	5	6
	ADV_IMP				1	1	2	2
	ADV_INT					2	3	3
	ADV_SPM						1	1
	ADV_TDS		1	2	3	4	5	6
Guidance documents	AGD_OPE	1	1	1	1	1	1	1
	AGD_PRE	1	1	1	1	1	1	1
Life-cycle support	ALC_CMC	1	2	3	4	4	5	5
	ALC_CMS	1	2	3	4	5	5	5
	ALC_DEL		1	1	1	1	1	1
	ALC_DVS			1	1	1	2	2
	ALC_LCD			1	1	1	1	2
	ALC_TAT				1	2	3	3
ST evaluation	ASE_CCL	1	1	1	1	1	1	1
	ASE_ECD	1	1	1	1	1	1	1
	ASE_INT	1	1	1	1	1	1	1
	ASE_OBJ	1	2	2	2	2	2	2
	ASE_REQ	1	2	2	2	2	2	2
	ASE_SPD		1	1	1	1	1	1
	ASE_TSS	1	1	1	1	1	1	1
Tests	ATE_COV		1	2	2	2	3	3
	ATE_DPT			1	1	3	3	4
	ATE_FUN		1	1	1	1	2	2
	ATE_IND	1	2	2	2	2	2	3
Vulnerability assessment	AVA_VAN	1	2	2	3	4	5	5

Şekil 3.1: Değerlendirme seviyeleri [8].

3.3. Değerlendirme Sürecinin Akışı

- Değerlendirme sürecinin başlatılabilmesi için, ürün geliştiricisi tarafından sürece temel teşkil edecek dokümantasyonun hazırlanması gerekmektedir. Bu dokümanların başında, değerlendirilecek ürünün güvenlik hedeflerini ve bu hedeflere ulaşmak üzere uygulanan güvenlik işlevlerini tanımlayan Security Target dokümanı gelmektedir. Security Target dokümanı ile birlikte, ürünün hangi varlıkları korumayı amaçladığı, bu varlıklara yönelik tehditlerin neler olduğu ve bu tehditlere karşı hangi güvenlik işlevlerinin geliştirildiği açık bir şekilde ortaya konur. Bu sayede ürünün güvenlik açısından nasıl bir çerçevede değerlendirileceği netleşmiş olur. Security Target dokümanının tamamlanmasının ardından, değerlendirme sürecinin ilerleyebilmesi için ürünün tasarımına ilişkin belgeler ile kullanıcı ve yönetici kılavuzları gibi tamamlayıcı dokümanların da hazırlanması gerekmektedir.
- Geliştirici tarafından hazırlanan dokümanlar ve değerlendirme altındaki ürün, değerlendirici kuruluş olan Common Criteria laboratuvarına gönderilir. Değerlendirici kuruluş, geliştirici tarafından sunulan bu dokümanları, seçilen EAL seviyesine uygun olarak Common Criteria standardı çerçevesinde sistematik bir biçimde inceler. İnceleme sürecinde, dokümantasyonda yer alan açıklamaların standarda uygunluğu, güvenlik hedefleri ile ürünün yapısal ve işlevsel uyumu denetlenir. Ayrıca, ürünün beyan edilen güvenlik işlevleri, değerlendirici tarafından bağımsız olarak gerçekleştirilen testler aracılığıyla doğrulanır.
- Eğer değerlendirme sırasında eksiklikler, tutarsızlıklar veya zayıflıklar tespit edilirse, değerlendirici laboratuvar bunları geliştiriciye resmi olarak bildirir. Geliştirici, belgeleri ve/veya ürünü revize ederek düzeltmeleri sağlar. Bu döngü, ürün ve/veya belgelerin birbiriyle tutarlı ve eksiksiz olacak son haline kadar tekrar edebilir.
- Bulguların bildirilmesi akabinde üründe ve/veya belgelerde son hale gelindikten sonra laboratuvar, elde edilen bulgu ve sonuçları içeren Evaluation Technical Report adlı kapsamlı raporu hazırlayarak yetkili sertifikasyon otoritesine (Certification Body) iletir.
- Otorite, bulguları denetler ve gerekmesi durumunda ek açıklamalar ya da düzeltmeler talep edebilir. Tüm kriterlerin sağlandığına kanaat getirirse ürüne

resmi Common Criteria sertifikasını verir. Sertifika, Common Criteria Portal üzerinden kamuya açık şekilde yayınlanır.

Değerlendirme sürecinde takip edilen güvence bileşenleri, EAL4 seviye güvence paketi temel alınarak aşağıdaki başlıklar altında incelenecektir.

3.3.1. ASE – Güvenlik Hedefi Değerlendirmesi (Security Target Evaluation)

ASE değerlendirme sınıfı, değerlendirme sürecindeki ürünün güvenlik hedeflerini tanımlayan ve ürünün güvenlik gereksinimlerine dair beyanını içeren Security Target dokümanının analiz edildiği kritik bir aşamadır. Bu değerlendirme sınıfı, Security Target dokümanının hem yapısal hem de içeriksel bütünlüğünü sorgular.

Security Target, ürünün değerlendirileceği temel belgedir ve aşağıdaki başlıklardan oluşur:

- ASE_INT: TOE Tanımı ve Kapsamı (TOE Definition – TOE Description)
 - Bu bölümde, değerlendirmeye konu olan ürünün (TOE – Target of Evaluation) temel tanımı yapılır ve ürünün sunduğu güvenlik işlevlerine genel hatlarıyla yer verilir. TOE'nin hangi ortamda çalıştığı, sistem mimarisi içindeki konumu ve sınırları açıkça belirtilir. Ayrıca TOE'nin teslimat sürecine ilişkin bilgiler (örneğin, ürünün kullanıcıya ne şekilde ulaştırıldığı, hangi bileşenlerle birlikte teslim edildiği) de bu başlık altında sunulur. Bu kapsamda, değerlendirmenin hangi bileşenleri kapsadığı ve TOE'nin fiziksel ya da yazılımsal yapısının sınırları net şekilde ortaya konulur.
- ASE_CCL: Uygunluk Beyanı (Conformance Claim)
 - Conformance Claim bölümü, hazırlanan Security Target dokümanının Common Criteria standardının hangi versiyonuna ve hangi kapsamına uygun olarak oluşturulduğunu beyan ettiği kısımdır. Bu beyan, değerlendirme sürecinin temel referans çerçevesini oluşturur.
- ASE_SPD: Güvenlik Problemi Tanımı (Security Problem Definition)
 - Security Target dokümanında yer alan bu bölüm, değerlendirilen ürünün güvenlik açısından maruz kalabileceği tehdit ortamını ve korunması gereken varlıkları tanımlar. Ürüne ve/veya çalışma ortamına ilişkin olası tehditler, organizasyonel güvenlik politikaları ve çevresel

varsayımlar aracılığıyla kapsamlı bir güvenlik problemi modellemesi ortaya konulur.

- ASE_OBJ: Güvenlik Hedefleri (Security Objectives)
 - ASE_SPD bölümünde tanımlanan güvenlik problemlerine etkin şekilde karşılık verecek güvenlik hedeflerinin belirlendiği bölümdür. Bu bölümün amacı, ürün özelinde tanımlanmış tehditler, organizasyonel güvenlik politikaları ve çevresel varsayımlar doğrultusunda uygun ve tutarlı güvenlik hedeflerinin ortaya konulmasını sağlamaktır.
- ASE_ECD: Genişletilmiş Bileşen Tanımı (Extended Components Definition)
 - Ürün geliştirme sürecinde, Common Criteria standardında mevcut olmayan özel güvenlik ihtiyaçlarının karşılanabilmesi amacıyla geliştirici tarafından tanımlanan özel gereksinimler bu başlık altında yer alır. ASE_ECD bölümü, bu genişletilmiş bileşenlerin açık ve doğru biçimde tanımlandığı, standardın biçimsel kurallarına uygun olarak ifade edildiği ve değerlendirme sürecine entegre edilebilirliğinin sağlandığı kısımdır. Bu bölüm sayesinde, TOE'ye özgü özel gereksinimlerin değerlendirme sürecinde standarda uygun biçimde yer alması mümkün hale gelir.
- ASE_REQ: Güvenlik Gereksinimleri (Security Requirements)
 - ASE_REQ bölümü, ASE_OBJ bölümünde tanımlanan güvenlik hedeflerini gerçekleştirmek üzere ürüne uygulanacak olan güvenlik gereksinimlerinin sistematik biçimde sunulduğu bölümdür. Bu bölümde, güvenlik hedefleriyle uyumlu şekilde seçilen gereksinimlerin Common Criteria Part 2 dokümanındaki tanımlar çerçevesinde doğru, tutarlı ve eksiksiz olarak sunulması beklenir.
- ASE_TSS: TOE Özeti Spesifikasyonu (TOE Summary Specification)
 - ASE_TSS bölümü, Security Target dokümanında belirtilen güvenlik işlevsel gereksinimlerinin TOE tarafından nasıl yerine getirildiğini açıklayan bölümdür. Bu kapsamda, ürünün gerçekleştirdiği güvenlik işlevleri teknik detaylarıyla sunulur ve her bir SFR için ürünün hangi mekanizmaları kullanarak bu gereksinimi sağladığı net biçimde ortaya konulur. TOE tarafından sunulan bu güvenlik işlevlerinin, Security Target dokümanındaki hedeflerle ve gereksinimlerle tutarlılığı, değerlendirmenin temel dayanak noktalarından biridir. Bu nedenle

ASE_TSS bölümü, ürünün güvenlik özelliklerinin teknik bir özetini içeren ve değerlendirme laboratuvarına doğrulama imkanı sağlayan kritik bir bölümdür. Açıklamaların doğruluğu, tutarlılığı ve yeterliliği, değerlendirme sürecinin başarısı açısından büyük önem taşır.

3.3.2. ADV – Geliştirme Kanıtlarının Değerlendirilmesi (Development Evaluation)

ADV değerlendirme sınıfı, değerlendirme sürecinin teknik derinliğini temsil eden ve TOE'nin tasarımına, mimarisine, arayüzlerine ve kaynak koduna kadar uzanan geliştirme kanıtlarının analiz edildiği kritik bir aşamadır. Bu değerlendirme, geliştirici tarafından sağlanan teknik belgelerin Common Criteria standardına uygun olarak hazırlanıp hazırlanmadığını ve ürünün güvenlik işlevlerinin güvenilir şekilde hayata geçirilip geçirilmediğini ortaya koymayı amaçlar.

ADV sınıfı, ürünün iç yapısının değerlendirme laboratuvarı tarafından sistematik olarak anlaşılmasını ve güvenlik gereksinimlerinin ürün içinde doğru biçimde karşılandığının kanıtlanmasını sağlar. Bu kapsamda aşağıdaki alt başlıklar değerlendirilir:

- ADV_FSP: Fonksiyonel Spesifikasyon (Functional Specification)
 - TOE tarafından sunulan dış arayüzlerin ve bu arayüzler üzerinden gerçekleştirilen güvenlik işlevlerinin tanımlandığı bölümdür. Bu kapsamda, ürünün işlevsel davranışı dış gözlemle anlaşılabilir hale getirilir. Değerlendirici, bu arayüzleri kullanarak ürünü test edebilmekte ve dıştan görülebilen davranışların Security Target dokümanında belirtilen güvenlik işlevleriyle tutarlı olup olmadığını doğrulayabilmektedir.
- ADV_TDS: Detaylı Tasarım (TOE Design Specification)
 - TOE'nin iç yapısının modüller düzeyinde detaylandırıldığı bölümdür. Her modülün işlevi, diğer modüllerle ilişkisi ve güvenlik işlevlerine katkısı açıkça tanımlanır. Bu yapı sayesinde, ürünün hangi iç yapılar aracılığıyla hangi güvenlik gereksinimlerini yerine getirdiği izlenebilir hale gelir. Bu bölüm, hem tasarımın anlaşılabilirliğini hem de güvenlik analizlerinin izlenebilirliğini sağlayan temel unsurlardandır
- ADV_ARC: Güvenlik Mimarisi Tanımı (Security Architecture)

- TOE'nin güvenlik işlevlerini gerçekleştiren bileşenlerinin genel mimari yapısının tanımlandığı bölümdür. Bu yapı içerisinde güvenlik işlevlerinin birbirleriyle olan ilişkileri, bilgi akış yolları ve güvenlik sınırları açıklanır. Güvenlik açısından kritik bölümler ile diğer yapılar arasındaki ayrımların netliği, güvenli tasarım ilkeleri doğrultusunda değerlendirilir.
- ADV_IMP: Uygulama Temsili (Implementation Representation)
 - ADV_IMP bölümü, geliştiricinin ürünü hangi kaynak kod ya da donanımsal yapı ile gerçekleştirdiğini gösteren temsili içermektedir. Değerlendirme seviyesine bağlı olarak kaynak kod ve donanım tasarımı gibi somut çıktılar bu bölümde yer alabilir. Bu yapı sayesinde değerlendirici, ürünün tasarlandığı biçimde gerçekten uygulanıp uygulanmadığını inceleyebilir. Kod düzeyinde güvenlik açıkları ya da uygunsuzluklar bu aşamada tespit edilebilir.

3.3.3. AGD – Kılavuz Dokümantasyonun Değerlendirilmesi (Guidance Documentation Evaluation)

AGD değerlendirme sınıfı, TOE'nin güvenli bir şekilde kurulması, yapılandırılması ve kullanılmasına olanak sağlayan rehber dokümanlarının incelendiği bölümdür. Bu aşama, ürünün son kullanıcılar ve yöneticiler tarafından hatasız ve güvenlik hedeflerine uygun şekilde kullanılabilmesini teminat altına almayı amaçlar. Common Criteria standardı, yalnızca ürünün teknik yapısına değil, aynı zamanda ürünle birlikte sunulan dokümantasyonun yeterliliğine de büyük önem vermektedir.

Geliştirici tarafından sağlanan rehber belgeler, kullanıcıların ürünü güvenlik gereksinimlerine uygun biçimde kullanabilmesini sağlamalı ve herhangi bir yanlış yapılandırma ya da kullanım hatasının güvenlik açıklıklarına yol açmasının önüne geçmelidir. AGD sınıfı aşağıdaki iki ana başlıktan oluşur:

- AGD_PRE: Hazırlık Kılavuzu (Preparative Procedures Guidance)
 - AGD_PRE bölümü, TOE'nin kurulumu ve ilk yapılandırma süreci için geliştirici tarafından sunulan rehber dokümanların değerlendirildiği kısımdır. Bu belgeler, ürünün güvenli bir şekilde kurulumunu gerçekleştirmek ve değerlendirme ortamında doğru yapılandırmayı sağlamak üzere hazırlanmıştır. Kurulum işlemi sırasında yapılması

gereken ön ayarlamalar, gerekli bileşenlerin listesi, ortam gereksinimleri ve güvenlik açısından dikkat edilmesi gereken adımlar (örn. varsayılan parolaların değiştirilmesi, güvenli bağlantıların kurulması vb.) bu belgelerde yer almalıdır. AGD_PRE'nin amacı, değerlendiricinin ürünü doğru biçimde yapılandırarak değerlendirme testlerini gerçekleştirmesini sağlamanın yanı sıra, ürünün gerçek dünyadaki dağıtım sürecinde güvenli şekilde kurulabileceğini de kanıtlamaktır.

- AGD_OPE: Operasyonel Kullanım Kılavuzu (Operational User Guidance)
 - AGD_OPE bölümü, TOE'nin son kullanıcıları için hazırlanmış kullanım kılavuzlarının değerlendirildiği kısımdır. Bu belgelerde ürünün nasıl başlatılacağı, kullanılacağı ve normal operasyon sırasında hangi adımların izleneceği ayrıntılı şekilde açıklanmalıdır. Ayrıca kullanıcıların erişebileceği fonksiyonlar, sınırlar, yetkilendirme süreçleri ve güvenli kullanım senaryoları açık biçimde sunulmalıdır. Bu bölümdeki temel amaç, ürünün kullanım aşamasında güvenlik hedeflerinin tehlikeye atılmasına neden olabilecek hatalı işlemleri engelleyecek net ve doğru rehberliğin sağlandığının doğrulanmasıdır.

3.3.4. ALC – Yaşam Döngüsü Desteği (Life-Cycle Support Evaluation)

ALC değerlendirme sınıfı, değerlendirme konusu ürünün geliştirilme, yapılandırma, teslimat ve yaşam döngüsü boyunca güvenli bir şekilde yönetilip yönetilmediğini inceleyen bölümdür. Bu değerlendirme, geliştirici organizasyonun güvenlik açısından ürünün yaşam döngüsünü ne derece kontrollü ve güvenilir yönettiğini ortaya koymayı hedefler. Ayrıca, kaynak kod düzeyinde yapılan değişikliklerin kontrolü, geliştirme ortamının güvenliği ve ürünün kullanıcıya teslim sürecindeki bütünlük de bu başlık altında değerlendirilir.

Aşağıda ALC sınıfına ait alt başlıklar detaylı bir biçimde verilmiştir:

- ALC_CMC: Yapılandırma Yönetimi Yetenekleri (Configuration Management Capabilities)
 - Bu bölümde, TOE'nin tüm bileşenlerinin nasıl tanımlandığı, sürüm kontrolünün nasıl sağlandığı ve yapılan değişikliklerin nasıl izlendiği

değerlendirilir. Amaç, ürünün her aşamasının tutarlı ve tekrarlanabilir şekilde yönetildiğini gösterecek yapılandırma yönetimi mekanizmalarının mevcut olduğunu kanıtlamaktır.

- ALC_CMS: Yapılandırma Yönetimi Belgeleri (Configuration Management Scope)
 - ALC_CMS bölümü, geliştirici tarafından yapılandırma yönetimi kapsamında hazırlanan belgelerin içeriğini ve yeterliliğini inceler. Bu belgelerde TOE'nin hangi parçalarının yapılandırma yönetimi altında olduğu, hangi dosyaların/öğelerin izlendiği ve ürünün nasıl kontrol edildiği gibi bilgiler yer alır.
- ALC_DEL: Teslimat Süreci (Delivery)
 - Bu başlık altında, TOE'nin geliştiriciden kullanıcıya güvenli bir şekilde nasıl teslim edildiği değerlendirilir. Burada teslimat sırasında ürünün yetkisiz değişikliklere maruz kalmadığı, bütünlüğünün korunduğu ve alıcı tarafından doğrulanabilir bir sistemin sunulup sunulmadığı incelenir.
- ALC_DVS: Geliştirme Ortamı Güvenliği (Development Security)
 - ALC_DVS, TOE'nin geliştirildiği fiziksel ve dijital ortamların güvenliğini değerlendirir. Geliştirici firmanın, ürün üzerinde çalışan personelin yetkilendirilmesi, kaynak kodlara erişimin kontrolü ve üretim sürecinde bilgi güvenliğinin sağlanmasına yönelik uygulamaları bu başlıkta ele alınır. Geliştirme yapılan ortamın, değerlendirme laboratuvarı ve sertifikalandırma otoritesi tarafından denetlendiği saha ziyareti süreci bu adımda iletir.
- ALC_LCD: Yaşam Döngüsü Tanımı (Life-Cycle Definition)
 - Bu bölümde, TOE'nin tüm yaşam döngüsü boyunca (tasarım, geliştirme, test, dağıtım, bakım ve sonlandırma) nasıl yönetileceği açıklanır. Amaç, ürünün tüm aşamalarında planlı ve güvenli bir şekilde geliştirildiğini ve sürdürüldüğünü gösterecek yeterlilikte bir yaşam döngüsü tanımının sağlanmasıdır.
- ALC_TAT: Teknik Destek Yeterliliği (Tools and Techniques)
 - ALC_TAT değerlendirmesi, geliştirici tarafından kullanılan yazılım geliştirme araçları, derleyiciler, analiz araçları gibi teknik araçların

yeterliliğini ve belgelenme düzeyini inceler. Aynı zamanda, kaynak kod seviyesinde yapılan işlemlerin izlenebilirliği ve değerlendirme laboratuvarı tarafından doğrulanabilirliği açısından da önem taşır.

3.3.5. ATE – Test Etkinliği Değerlendirmesi (Tests for the Evaluated Product)

ATE değerlendirme sınıfı, değerlendirme altındaki ürünün güvenlik işlevlerini ne ölçüde doğru ve eksiksiz yerine getirdiğini ortaya koymak amacıyla test faaliyetlerinin yürütüldüğü bölümdür. Bu sınıf kapsamında geliştirici tarafından gerçekleştirilen testlerin uygunluğu değerlendirilir, test sonuçlarının doğruluğu analiz edilir ve ayrıca değerlendirme laboratuvarı bağımsız testler gerçekleştirerek ürünün güvenlik işlevlerini sınar. Amaç, TOE'nin güvenlik işlevlerinin pratikte beklenen şekilde çalıştığını ispatlamaktır.

ATE sınıfı aşağıdaki alt başlıklardan oluşur:

- ATE_COV: Test Kapsamı (Coverage)
 - Bu bölümde geliştirici tarafından hazırlanan testlerin, Security Functional Requirements ile ne ölçüde örtüştüğü değerlendirilir. Amaç, her bir güvenlik işlevi için en az bir testin tanımlandığının ve bu testlerin işlevlerin tümünü kapsadığının gösterilmesidir. Test kapsamı, değerlendiricinin temel doğrulama alanlarından biridir.
- ATE_DPT: Test Derinliği (Depth)
 - ATE_DPT bölümünde, geliştirici tarafından hazırlanan testlerin TOE içindeki bileşenlerin yapısı, sınırları ve etkileşimleri açısından ne kadar ayrıntılı ve derinlemesine gerçekleştirildiği değerlendirilir. Burada yalnızca işlevsel testlerin yapılması değil, aynı zamanda maksimum uzunlukta parola, hatalı giriş, boş bırakma gibi sınır durumlarının ve hata senaryolarının da test edilip edilmediği analiz edilir.
- ATE_FUN: Fonksiyonel Testler (Functional Testing)
 - Bu alt başlık altında geliştirici tarafından gerçekleştirilen testler detaylı şekilde ele alınır. Geliştirici, ürünün güvenlik işlevlerini yerine getirdiğini gösteren test senaryolarını, yöntemlerini ve sonuçlarını doküman eder. Değerlendirici bu testleri inceleyerek geçerliliklerini kontrol eder.

- ATE_IND: Bağımsız Testler (Independent Testing)
 - ATE_IND değerlendirmesi, bağımsız laboratuvarın (değerlendiricinin) geliştirici tarafından sunulan testleri tekrar etmesi ya da kendi hazırladığı testleri uygulaması yoluyla gerçekleştirilir. Amaç, geliştiricinin sunduğu testlerin doğruluğunu sınamak ve gerektiğinde ilave testlerle güvenlik işlevlerinin gerçekten uygulandığını kanıtlamaktır. Bağımsız testler, değerlendirme sürecinde doğrulayıcı ve tamamlayıcı bir rol üstlenir.

3.3.6. AVA – Zafiyet Değerlendirmesi (Vulnerability Assessment)

AVA değerlendirme sınıfı, değerlendirilen ürünün bilinen ve potansiyel güvenlik açıklarına karşı ne derece dirençli olduğunu analiz etmeyi ve bu zafiyetlerin kötüye kullanılabilirliğini değerlendirmeyi amaçlayan sınıftır. Bu sınıf, hem geliştirici tarafından yapılan açıklık analizlerini hem de değerlendirici laboratuvar tarafından yürütülen bağımsız zafiyet değerlendirmesini kapsar. Değerlendirme kapsamında, ürünün teknik belgeleri, uygulama yapısı, arayüzleri ve dışa açık bileşenleri incelenerek; mevcut tehdit ortamında geçerli saldırı teknikleriyle TOE'nin güvenlik mekanizmaları test edilir. AVA sınıfı, ürünün tanımlı güvenlik hedeflerine ve seçilen EAL seviyesinin gerektirdiği “attack potential” seviyesine uygun bir güvenlik sağlamasını temin etmeyi amaçlar.

- AVA_VAN: Zafiyet Analizi (Vulnerability Analysis)
 - AVA_VAN, AVA sınıfı altında tek alt başlıktır ve zafiyet analizine dair faaliyetlerin tamamını kapsar. Değerlendirici laboratuvarın gerçekleştirdiği bağımsız test ve analiz süreçleri bu başlık altında yürütülür. Bu süreçte kamuya açık saldırı bilgileri dikkate alınarak TOE'ye yönelik potansiyel açıklar araştırılır. TOE'nin dış arayüzleri ve kullanım senaryoları analiz edilerek, bu arayüzler üzerinden yapılabilecek kötüye kullanımlar değerlendirilir. Tüm bu analizler, seçilen EAL seviyesinin gerektirdiği saldırı potansiyeli (attack potential) göz önünde bulundurularak belli seviyelerde gerçekleştirilir. Bu seviyeler, analizlerin kapsamını ve derinliğini belirler. Aşağıda, AVA_VAN alt sınıfının seviyelerine ilişkin açıklamalar yer almaktadır:

- AVA_VAN.1 – Giriş Düzeyi Zafiyet Analizi (Vulnerability Survey):

Temel belgelerin taranması ve bilinen açıklar için literatürün incelenmesi ile sınırlı, yüzeysel bir analizdir. Yeni veya gelişmiş analiz yöntemleri içermez.

- AVA_VAN.2 – Basit Yöntemsel Zafiyet Analizi (Vulnerability Analysis):

Daha derinlemesine analizlerin yapıldığı, ancak hala sınırlı saldırı potansiyeli ve karmaşıklığında testlerin uygulandığı seviyedir.

- AVA_VAN.3 – Yoğun Yöntemsel Zafiyet Analizi (Intensive Methodical Vulnerability Analysis):

Gelişmiş teknikler ve detaylı test senaryolarının kullanıldığı, orta-yüksek seviyede saldırı potansiyeline sahip analizler içerir. Bu seviyede, TOE'nin dış arayüzleri üzerinden gelebilecek saldırılar sistematik olarak incelenir. Yazılım ve donanım bileşenlerinin güvenlik incelemesi yapılır; kod analizi ve tasarım gözden geçirmeleri ile potansiyel yazılım açıkları ve tasarım zayıflıkları değerlendirilir. Ayrıca, kullanıcı ve yönetici dokümantasyonu incelenerek ürünün doğru ve güvenli kullanımını sağlayacak yönergelerin yeterliliği kontrol edilir.

- AVA_VAN.4 – Gelişmiş Yöntemsel Zafiyet Analizi (Advanced Methodical Vulnerability Analysis):

Karmaşık saldırı teknikleri, özel araçlar ve yüksek beceri gerektiren yöntemlerle gerçekleştirilen ileri düzey analizler bu seviyede yapılır. Bu seviyede, yan kanal saldırılarına karşı direnç testleri gerçekleştirilir. Güç analizi (power analysis), zamanlama saldırıları (timing attacks) ve elektromanyetik sızıntı testleri gibi yan kanal tekniklerine karşı ürünün dayanıklılığı sınanır. Cihaza fiziksel müdahale (tamper testi) ve hata enjeksiyonu (fault injection) yöntemleri uygulanarak güvenlik zafiyetleri araştırılır. Aynı zamanda güvenlik politikalarının ve prosedürlerinin tutarlılığı, anahtar yönetimi,

erişim kontrolleri ve güncelleme mekanizmaları gibi kritik alanlarda doğrulanır.

- AVA_VAN.5 – Gelişmiş Seviye Saldırı Direnci (Advanced Attack Resistance):

Çok yüksek saldırı potansiyeline sahip, genellikle devlet destekli ve iyi finanse edilmiş saldırganlara karşı ürünün dayanıklılığının değerlendirildiği en üst seviyedir. Kritik güvenlik ürünleri için uygulanır. Bu kapsamda, önceki seviyelerde yapılan tüm testler ve analizler en üst düzeyde uygulanır ve karmaşık, özel tasarlanmış saldırı senaryolarıyla ürünün güvenliği sınanır.

Bu incelemeler sonucunda bulunan zafiyetler, risk düzeylerine göre sınıflandırılır ve değerlendirilir. Ürünün zafiyetlere karşı gösterdiği direnç seviyesi, sertifikasyon kararında belirleyici olur. Atak potansiyeli aşağıdaki tabloda bulunan kriterlere göre hesaplanır.

Tablo 3.1: Atak potansiyeli hesaplama kriterleri [9].

Factor	Value
Elapsed Time	
<= one day	0
<= one week	1
<= two weeks	2
<= one month	4
<= two months	7
<= three months	10
<= four months	13
<= five months	15
<= six months	17
> six months	19
Expertise	
Layman	0
Proficient	3 ^a

Tablo 3.1: Devam.

Expert	6
Multiple experts	8
Knowledge of TOE	
Public	0
Restricted	3
Sensitive	7
Critical	11
Window of Opportunity	
Unnecessary / unlimited access	0
Easy	1
Moderate	4
Difficult	10
None	**b
Equipment	
Standard	0
Specialised	4 ^c
Bespoke	7
Multiple bespoke	9
^a When several proficient persons are required to complete the attack path, the resulting level of expertise still remains “proficient” (which leads to a 3 rating).	
^b Indicates that the attack path is not exploitable due to other measures in the intended operational environment of the TOE.	
^c If clearly different test benches consisting of specialised equipment are required for distinct steps of an attack, this should be rated as bespoke.	

Her bir değerlendirme kriteri (bilgi seviyesi, uzmanlık, ekipman, zaman ve erişim türü) için verilen puanlar ayrı ayrı belirlendikten sonra bu değerler toplanarak toplam saldırı (atak) puanı elde edilir. Bu toplam puan, Common Criteria tarafından tanımlanmış eşik değerlere göre yorumlanır ve böylece saldırının gerektirdiği kaynak düzeyine karşılık gelen atak potansiyeli (saldırı seviyesi) belirlenir.

Values	Attack potential required to exploit scenario:	Meets assurance components:	Failure of components:
0-9	Basic	-	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4, AVA_VAN.5
10-13	Enhanced-Basic	AVA_VAN.1, AVA_VAN.2	AVA_VAN.3, AVA_VAN.4, AVA_VAN.5
14-19	Moderate	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3	AVA_VAN.4, AVA_VAN.5
20-24	High	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4	AVA_VAN.5
=>25	Beyond High	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4, AVA_VAN.5	-

Şekil 3.2: Zafiyet derecelendirilmesi ve direnç seviyesi [9].

4. KRİPTOGRAFİK ÜRÜNLERDE ALGORİTMA DOĞRULAMA SÜREÇLERİ

Kriptografik ürünlerin güvenliğini değerlendirirken yalnızca fiziksel güvenlik önlemleri ya da sistem tasarımı değil, aynı zamanda kullanılan kriptografik algoritmaların doğruluğu ve uygulanma biçimi de büyük önem taşır. Bir algoritmanın teorik olarak güvenli olması, onun bir donanım veya yazılım bileşeni içinde doğru uygulandığını garanti etmez. Bu nedenle kriptografik ürünlerde algoritma doğrulama süreçleri, hem bağımsız sertifikasyon sistemleri hem de Common Criteria gibi çerçeveler içerisinde kritik bir değerlendirme alanıdır.

4.1. Kriptografik Algoritma Doğrulamanın Önemi

Kriptografik algoritmalar, bilgi güvenliğinin temel yapı taşlarını oluşturur. Simetrik şifreleme (örneğin AES), asimetrik şifreleme (örneğin RSA, ECC) ve özet alma algoritmaları (örneğin SHA) gibi teknikler, bütünlük, gizlilik ve kimlik doğrulama gibi güvenlik servislerini sağlamak üzere kritik roller üstlenir. Ancak bu algoritmaların güvenliği, yalnızca teorik olarak doğru tanımlanmış olmalarına değil, aynı zamanda doğru bir şekilde uygulanmalarına da bağlıdır. Uygulamada yapılan en küçük bir hata, örneğin anahtar yönetiminde bir kusur, blok şifreleme modunun yanlış kullanımı veya IV (Initialization Vector) üretiminin hatalı yapılması gibi durumlar, sistemin bütün güvenlik yapısını tehlikeye sokabilir [10], [11]. Bu bağlamda, algoritmaların doğru şekilde implemente edilip edilmediğini bağımsız ve sistematik testlerle doğrulamak kritik öneme sahiptir. İşte bu amaçla algoritma doğrulama testleri devreye girer.

Common Criteria değerlendirme sürecinde, kriptografik algoritmaların doğruluğu genellikle ATE adımı kapsamında incelenir. Özellikle ATE_FUN (Functional Testing) kriteri altında, değerlendirilen ürünün algoritmik çıktılarının referans vektörlerle karşılaştırılarak doğruluğu test edilebilir.

Dünya genelinde, birçok kriptografik ürünün algoritmalarını doğrulamak için NIST Cryptographic Algorithm Validation Program (CAVP) gibi resmi test setleri kullanılmaktadır. Bu testler, algoritmanın belirli girdiler altında doğru çıktılar üretip

üretmediğini denetler ve sertifikasyon süreçlerinde temel referans noktası olarak kullanılır [12].

Ayrıca, bu testlerin önemi yalnızca bireysel ürün güvenliği ile sınırlı değildir. Geniş kapsamlı dijital altyapılarda kullanılan kriptografik bileşenlerin doğruluğu, bütün bir sistemin güvenlik bütünlüğü açısından da doğrudan belirleyicidir. Bu nedenle hem donanımsal hem yazılımsal HSM'ler gibi kriptografik modüller, değerlendirme süreçlerinde algoritma doğrulama adımı olmadan tam bir güvenlik değerlendirmesinden geçmiş sayılmaz.

Sonuç olarak, algoritma doğrulama testleri; güvenli ürün geliştirme ve sertifikasyona uygunluk açısından yalnızca tavsiye edilen bir uygulama değil, aynı zamanda güvenilirliğin vazgeçilmez bir gereğidir.

4.2. Uluslararası Algoritma Doğrulama Sistemleri

4.2.1. NIST – CAVP ve ACVP

Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), kriptografik algoritmaların doğrulamasına yönelik iki önemli test ve sertifikasyon sistemi işletmektedir:

- CAVP (Cryptographic Algorithm Validation Program):
 - NIST Kriptografik Algoritma Doğrulama Programı (CAVP), onaylanmış (yani FIPS onaylı ve NIST tarafından önerilen) kriptografik algoritmaların ve bunların bireysel bileşenlerinin doğrulama testini sağlar [12]. Geliştirici tarafından üretilen çıktıların NIST tarafından belirlenen test vektörleriyle karşılaştırılması temeline dayanır.
- ACVP (Automated Cryptographic Validation Protocol):
 - CAVP'nin otomasyon ve uzaktan erişim desteğiyle genişletilmiş halidir. ACVP üzerinden geliştiriciler kendi kriptografik modüllerini bir istemci yazılım aracılığıyla doğrudan NIST sunucularına karşı test edebilir [13].

Her iki sistemin sonucunda da başarılı bulunan algoritmalar, NIST tarafından onaylı hale gelir ve bu modüller, FIPS 140-3 gibi üst düzey güvenlik sertifikasyonları için uygun kabul edilmektedir.

4.2.2. FIPS 140-3 Standardı

FIPS 140-3 (Federal Information Processing Standard Publication 140-3), Amerika Birleşik Devletleri hükümeti tarafından geliştirilen ve kriptografik modüllerin güvenlik gereksinimlerini belirleyen bir standarttır. FIPS 140-2'nin yerini almak üzere hazırlanmış olup, 2020 yılında yürürlüğe girmiştir [14].

Bu standart, özellikle HSM, akıllı kartlar, şifreleme kütüphaneleri gibi kriptografik işlevler sağlayan ürünlerin güvenli bir şekilde geliştirilmesini ve doğrulanmasını sağlar.

FIPS 140-3, Common Criteria gibi genel amaçlı bir güvenlik değerlendirmesi yapmaz; yalnızca kriptografik modülün özelliklerine odaklanır. Ancak kriptografik işlem yapan pek çok ürün hem Common Criteria hem de FIPS değerlendirmesine girer.

FIPS 140-3 standardı, teknik temelini ISO/IEC 19790 standardından almaktadır. Her iki standart da kriptografik modüllerin güvenliğini modül seviyesi üzerinden değerlendirir. FIPS 140-3, bu uluslararası standardı Amerika Birleşik Devletleri hükümeti sistemlerine adapte ederken, ISO/IEC 19790 ise daha evrensel ve ülke bağımsız bir yaklaşım sunar. Dolayısıyla, FIPS 140-3 uyumlu bir ürün aynı zamanda ISO/IEC 19790 kriterlerine büyük oranda uygunluk göstermektedir [15].

Türkiye, FIPS 140-3 standardını doğrudan uygulamamakla birlikte, kriptografik modül güvenliği değerlendirmelerinde ISO/IEC 19790 standardını esas alır. Ülkemizde bu alandaki çalışmalar, TÜBİTAK BİLGEM öncülüğünde yürütülmekte olup ISO/IEC 19790 temelinde çeşitli kriptografik modül testleri yapılmakta ve aynı zamanda Common Criteria değerlendirme süreçlerini desteklemektedir.

4.3. Uygulamalı Algoritma Doğrulama Testi: SoftHSM Örneği

Bu çalışma kapsamında, Common Criteria değerlendirme sürecinde ATE_FUN (Functional Testing) adımı bağlamında ele alınan algoritma doğrulama testleri, yazılımsal bir HSM ürünü üzerinde uygulanmıştır. Donanımsal bir HSM ürününe erişim kısıtları nedeniyle, açık kaynaklı olarak geliştirilen ve yaygın olarak test amaçlı kullanılan SoftHSM çözümü tercih edilmiştir. Bu yazılım, PKCS#11 standardına uygun biçimde çalışmakta olup, gerçek bir HSM'in temel fonksiyonlarını yazılım ortamında gerçekleştirmektedir.

SoftHSM, test ortamına kurulmuş ve Java programlama dili kullanılarak PKCS#11 API üzerinden erişilmiştir. Bu kapsamda aşağıdaki adımlar uygulanmıştır:

- SoftHSM ortamının kurulumu ve başlatılması (token ilklendirme, veri deposu yapılandırma)
- PKCS#11 standardına uygun olarak oturum açma, kullanıcı ve şifre yönetimi işlemlerinin gerçekleştirilmesi
- AES algoritmasının CTR (Counter Mode) modunda hem şifreleme hem de şifre çözme işlemlerinin uygulanması

Uygulama sırasında kullanılan test senaryoları, açık metin ve anahtar çiftleri ile şifreleme/şifre çözme işlemlerinin doğruluğunu test etmeye yöneliktir. Böylelikle HSM ürününün algoritmik işlevleri, kriptografik doğrulama bağlamında fonksiyonel olarak test edilmiştir.

Bu uygulama, ATE_FUN kriterine doğrudan karşılık gelmekte olup, değerlendirilen ürünün kriptografik doğruluğunu ortaya koymayı amaçlamaktadır. Aynı zamanda test süreci, algoritmanın doğru uygulandığını ve HSM ürününün güvenlik gereksinimlerine uygun çalıştığını göstermektedir. Bu yönüyle çalışma, yalnızca teorik bir değerlendirme değil, aynı zamanda gerçek test ortamında yapılan uygulamalı bir katkı sunmaktadır. Bu tez kapsamında SoftHSM ile gerçekleştirilen işlemler, CyberHashira'nın GitHub üzerinde yayımladığı PKCS#11 eğitim materyalleri ve örnek uygulamalarına dayanılarak yürütülmüştür [16].

4.3.1. SoftHSM Yapılandırması

- Token Yapılandırması

```
softhsm2-util --init-token --slot 0 --label "Token1"
```

Şekil 4.1: Initialization işlemi.

- Slot 0 üzerine "Token1" adında bir HSM token'ı oluşturuldu.
- Komut çalıştığında SO PIN ve User PIN girildi.

- Slot Bilgilerini Görüntüleme

```
softhsm2-util --show-slots
```

Şekil 4.2: Slot görüntüleme işlemi.

- Sistemde tanımlı token ve slot bilgilerini görüntülemek için kullanılır.
- Oturum Açma

```
pkcs11-tool --module /usr/lib/softhsm/libsofthsm2.so  
--login --pin 12345
```

Şekil 4.3: Oturum açma işlemi.

- Token'a login olurken kullanıcı PIN'i kullanılır.
- Daha önce oluşturulmuş nesneler listelenir.
- AES Anahtarı Oluşturma

```
pkcs11-tool --module /usr/lib/softhsm/libsofthsm2.so  
--login --pin 12345 \ --keygen --key-type aes:256  
--label "MyAESKey"
```

Şekil 4.4: AES anahtarı oluşturma işlemi.

- 256-bit AES anahtarı oluşturur.
- AES İle Şifreleme

```
pkcs11-tool --module /usr/lib/softhsm/libsofthsm2.so  
--login --pin 12345 \ --encrypt --key-label
```

Şekil 4.5: AES ile şifreleme işlemi.

- plaintext.txt dosyası şifrlenerek ciphertext.txt olarak kaydedilir.
- AES İle Şifre Çözme

```
pkcs11-tool --module /usr/lib/softhsm/libsofthsm2.so --  
login --pin 12345 \ --decrypt --key-label "MyAESKey" --  
input-file ciphertext.txt --output-file decrypted.txt
```

Şekil 4.6: AES ile şifre çözme.

- Şifrelenmiş dosya orjinal haline getirilir.

4.3.2. Java ile PKCS#11 Üzerinden Algoritma Doğrulama Süreci

PKCS#11, kriptografik donanım veya yazılım token'ları üzerinde anahtar yönetimi ve kriptografik işlemlerin gerçekleştirilmesine olanak tanıyan, yaygın olarak kullanılan bir API standardıdır. SoftHSM üzerinde token yapılandırması ve AES anahtar üretimi gibi işlemler başarıyla tamamlandıktan sonra, algoritma doğrulama aşamasına geçilmiştir. Bu süreçte, HSM yalnızca komut satırı araçları aracılığıyla değil, aynı

zamanda PKCS#11 arayüzü üzerinden Java programlama dili kullanılarak erişilmiştir. Java'nın PKCS#11 desteği sayesinde, AES algoritmasının CTR modunda şifreleme ve şifre çözme işlemleri gerçekleştirilmiş; elde edilen çıktılar, FIPS onaylı Bouncy Castle kütüphanesiyle karşılaştırılarak doğrulama yapılmıştır. Bu testlerin temel amacı, kullanılan algoritmaların beklenen şekilde çalışıp çalışmadığını doğrulamak ve elde edilen çıktıları güvenilir bir referansla karşılaştırarak doğruluk ve bütünlük açısından değerlendirmektir.

Aşağıda, HSM ile temel seviyede bağlantı kurmayı ve oturum yönetimini gerçekleştiren kod yapısı sunulmaktadır. Bu yapı, test öncesi gerekli olan HSM ilklendirme (initialize), oturum açma (session open) ve kullanıcı giriş işlemlerini gerçekleştirmektedir.

```
package hsmTest;
import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Rule;
import org.junit.rules.ExpectedException;
import
sun.security.pkcs11.wrapper.CK_C_INITIALIZE_ARGS;
import sun.security.pkcs11.wrapper.PKCS11;
import sun.security.pkcs11.wrapper.PKCS11Exception;
import java.io.File;
import java.io.IOException;
import static
sun.security.pkcs11.wrapper.PKCS11Constants.*;
public abstract class AltSeviyeTest {
    static protected long hSession;
    static protected PKCS11 p11;
    @Rule
    public ExpectedException thrown =
ExpectedException.none();

    @BeforeClass
    public static void init() throws IOException,
PKCS11Exception {

        CK_C_INITIALIZE_ARGS initArgs = new
CK_C_INITIALIZE_ARGS();
        p11 =
PKCS11.getInstance("D:\\SoftHSM2\\lib\\softhsm2-
x64.dll", "C_GetFunctionList", null, false);
        hSession =
p11.C_OpenSession(p11.C_GetSlotList(true)[0],
CKF_SERIAL_SESSION | CKF_RW_SESSION, null, null);
        char[] pin = {'1', '2', '3', '4', '5'};
        p11.C_Login(hSession, CKU_USER, pin);
    }
}
```

Şekil 4.7: SoftHSM ile PKCS#11 oturum başlatma örneği.

```

@AfterClass
    public static void destroy() throws PKCS11Exception
    {
        p11.C_Logout(hSession);
        p11.C_CloseSession(hSession);
    }
    public static File createFile(String fileName) throws
    IOException {
        File file = new File(fileName);
        if (file.exists()) {
            file.delete();
        }
        file.createNewFile();
        return file;
    }
}

```

Şekil 4.7: Devam.

Bu örnekte, “sun.security.pkcs11.wrapper.PKCS11” sınıfı kullanılarak SoftHSM modülüne erişim sağlanmaktadır. “C_GetSlotList”, HSM üzerindeki mevcut token slotlarını listelerken; “C_OpenSession” oturum açmakta, “C_Login” fonksiyonu ise kullanıcı doğrulamasını gerçekleştirmektedir. Kodun sonunda oturum kapatılmakta ve sistemle bağlantı sonlandırılmaktadır. Bu yapı, sonraki aşamalarda algoritma doğrulama işlemlerinin gerçekleştirilmesi için temel bir hazırlık niteliğindedir.

4.3.3. AES-CTR Modunda Algoritma Doğrulama Adımları

Oturum yönetimi başarıyla gerçekleştirildikten sonra, AES algoritmasının CTR modundaki doğruluğu, test vektörleri kullanılarak hem şifreleme hem de şifre çözme işlemleri için uygulanmıştır. Bu doğrulama, PKCS#11 standardı üzerinden doğrudan SoftHSM ile haberleşerek gerçekleştirilmiştir.

Algoritmanın işleyişini değerlendirmek amacıyla, AES-CTR modunda 128, 192 ve 256 bit anahtar uzunlukları için, her biri 100 adet olmak üzere toplamda 300 test vektörü kullanılmıştır. Bu vektörler, .req uzantılı giriş dosyalarından ayrıştırılmış ve işlem sonucunda elde edilen çıktı verileri .rsp uzantılı dosyalarda kayıt altına alınmıştır.

Kullanılan test vektörleri üçer örnek olarak aşağıda sunulmuştur. Her bir örnek, ilgili şifreleme ya da şifre çözme işleminde kullanılan parametrelerin formatını göstermektedir:

```

[Enc]

[128]

Count = 1
KEY = 9af1a2a1b6e69d408fab6aa6ece98915
NONCE = 62c5ca558f534bc3
COUNTER = 744f10cf18288e7d
PLAINTEXT = 70d934d91857b1084c5687a74eb0ba2c

Count = 2
KEY = 98cf48a0cf85be27b4bf6ad210c56818
NONCE = 90991c3a34e1b156
COUNTER = 9c88681f10d5fff6
PLAINTEXT = c3d7208c31ee1d9c7329af65ad36507b

Count = 3
KEY = ee3f6e89ed1328660504fb2952657f0c
NONCE = a8d24004247c8175
COUNTER = 34e2d0b20e64bfc3
PLAINTEXT = 3831334d240218539de7c939ba02c1ab

```

Şekil 4.8: AES-128 encryption test vektörleri.

```

[Enc]

[192]

Count = 1
KEY = 94af7923cafbcb4802eb7385a08d9ba04262c5493a0d0db7c
NONCE = 57637fce2b89adf9
COUNTER = 09538fb23f85f2e2
PLAINTEXT = e75ff5e92d9cd1ff2b0f7e8646eb9c5c

Count = 2
KEY = dc3d3e513f46effab958d27b4840bab2e70628aa0ce225e1
NONCE = 463704dddbba580
COUNTER = 10f918bbf39a063d
PLAINTEXT = 71ca34d7fb3ad98ebbd81d22019dba6c

Count = 3
KEY = 6fcf1f3c99662d3ab4c480ddb45283cbf326e62c7ef2caa9
NONCE = 8466f80daf73ee6e
COUNTER = 147f91fe3105607f
PLAINTEXT = b64b1facee77b912ed7d7295f2ed29dd

```

Şekil 4.9: AES-192 encryption test vektörleri.

```
[Enc]

[256]

Count = 1
KEY =
0b2c62afe8f01ae3aba3ea8bfd928252fa6d9d2c6b4f07bedd7bb1ebb5d04cd9
NONCE = 651ef67f1cb4d7cd
COUNTER = fcb5dfa4830a0359
PLAINTEXT = 790ef2897f82d2f4145101d73f4692e6

Count = 2
KEY =
3f65237144545d9d150d22781d840793a2069c802d2a5bea69ad04b2bf0f072e
NONCE = a81cc88d13f03ab8
COUNTER = 128034e809cdb1f6
PLAINTEXT = deebe2fae1292193135f129c3f10d53b

Count = 3
KEY =
0b6377e21b0ea2a9f69881d806afc8bbe1a6a54f60986f40e16d0318239129a3
NONCE = 40fd1eccf4e041f8
COUNTER = 5b1e51850166034e
PLAINTEXT = 0195ebb4fc85c8ec33aaa00f65b0001f
```

Şekil 4.10: AES-256 encryption test vektörleri.

```
[Dec]

[128]

Count = 1
KEY = 5f55ecc859e792f4b911404be24ba8bc
NONCE = d5b6ccd271bbc0a5
COUNTER = 60aa0e64724db7ff
CIPHERTEXT = 2fd786d6767f15c72304ef53cceab946

Count = 2
KEY = 3908cca5d3adbb2a79fadedfbc46fc9e4
NONCE = 25fe2fdf6ee0b327
COUNTER = 13d4c80750f7ea44
CIPHERTEXT = c5be51f1087d6d281509ac642dae7f01

Count = 3
KEY = 6b10278e1039f066b4a414771a867a6a
NONCE = eebf2b8ebe8988f8
COUNTER = a41b199f71e4b9de
CIPHERTEXT = 714a803a29a9e178736d59ac52f9bfb1
```

Şekil 4.11: AES-128 decryption test vektörleri.

```

[Dec]

[192]

Count = 1
KEY = f9ce396eef8ed77ee16af71651f9c77a49d5be0fb480daa4
NONCE = d59273e9f926bc1b
COUNTER = b93a94e35795f5c4
CIPHERTEXT = 1b24d5e4954797181043c69482058ffa

Count = 2
KEY = 85931b366196fb6ad847fa2fa1d58bbb96b4992c9fc66c8b
NONCE = 11cfaad3849a73a9
COUNTER = 1a47ef027bf1f3a1
CIPHERTEXT = 4ddfdb46df224c8f8b930d18e25e1518

Count = 3
KEY = 1732f3608117a16e17ef2acb29f7e52c97b963dfbfe155e2
NONCE = b00ef871b54e037a
COUNTER = e25c77720eda0fc4
CIPHERTEXT = 361c9557ae6443f5bb3c7a41e6cccd6f

```

Şekil 4.12: AES-192 decryption test vektörleri.

```

[Dec]

[256]

Count = 1
KEY =
2d9e984c664a9f6a71ba29b102ceb9f422f02dc32bc36fa77be8028bf0c8edab
NONCE = f4adc41599a75673
COUNTER = 2125f7b148b4cf61
CIPHERTEXT = 2cbe2b223dbffc9f9aff61c39ba9bfd1

Count = 2
KEY =
61dc4cbae4be352b81479d0ea2b579b815608b85edb93d7d4098a6b08af7cafe
NONCE = b831eaae97f7efd7
COUNTER = 58d48b954aee8827
CIPHERTEXT = 6b19a3659561866d4923a300d9d31246

Count = 3
KEY =
0c80a20a4642c9bad85c99fdb050e20280ade22b4db2749ada23c9a81cc7057
NONCE = fd3afc4dc92fb622
COUNTER = 126408d01c4a2800
CIPHERTEXT = 61cbf9a4c768f9a824bc7c8e6a943093

```

Şekil 4.13: AES-256 decryption test vektörleri.

Her test vektörü için; anahtar (KEY), başlatma sayacı (NONCE + COUNTER), açık metin (PLAINTEXT) veya şifreli metin (CIPHERTEXT) gibi değerler sırasıyla işlenmekte; “CK_AES_CTR_PARAMS” veri yapısı ve “CK_MECHANISM” nesnesi kullanılarak uygun algoritma parametreleri belirlenmektedir.

Aşağıda sunulan kod yapısı, AES-CTR modunda çalışan algoritmanın hem şifreleme (encryption) hem de şifre çözme (decryption) işlemlerine ait test senaryolarının, SoftHSM ortamında nasıl yürütüldüğünü göstermektedir.

Ayrıca, karşılaştırma amacıyla aynı test vektörleri kullanılarak Bouncy Castle kütüphanesiyle yazılım tabanlı doğrulama işlemleri de gerçekleştirilmiştir. Böylece uygulama çıktılarının eşleşip eşleşmediği gözlemlenmiş ve testlerin geçerliliği pekiştirilmiştir.

- AES_CTR_Enc() Metodu:
 - AESCTRMCTEnc192.req dosyasından alınan test vektörleri okunur.
 - Her bir vektördeki KEY, NONCE, COUNTER ve PLAINTEXT verileri ayrıştırılır.
 - NONCE ve COUNTER birleştirilerek Initialization Vector üretilir.
 - CK_AES_CTR_PARAMS yapısı ile Initialization Vector, PKCS#11 için uygun forma dönüştürülür.
 - Anahtar nesnesi oluşturularak “C_EncryptInit” ve “C_Encrypt” fonksiyonlarıyla şifreleme yapılır.
 - Sonuçlar “.rsp” uzantılı çıktıya yazılır.

```
@Test
public void AES_CTR_Enc() throws IOException, PKCS11Exception
{ //*****ENCRYPTION*****
    String line;
    ArrayList<String> Count
        = new ArrayList<>(100); //count
    ArrayList<String> KEY
        = new ArrayList<>(100);
    ArrayList<String> NONCE
        = new ArrayList<>(100);
    ArrayList<String> COUNTER
        = new ArrayList<>(100);
    ArrayList<String> PT
        = new ArrayList<>(100);

    File AESEncryption = createFile("AESCTRMCTEnc192.rsp");
    try (FileWriter fw = new FileWriter(AESEncryption)) {
        try(BufferedReader br = new BufferedReader(new
    FileReader("AESCTRMCTEnc192.req"))) {
    for (int i=0; i<603; i++) {
        line = br.readLine();

        if (line.contains("Count ="))
    {Count.add(line.substring(8));}
        else if (line.contains("KEY ="))
    {KEY.add(line.substring(6));}
```

Şekil 4.14: AES_CTR_Enc() metodu.

```

        else if (line.contains("NONCE ="))
        {NONCE.add(line.substring(8));}
        else if (line.contains("COUNTER ="))
        {COUNTER.add(line.substring(10));}
        else if (line.contains("PLAINTEXT ="))
        {PT.add(line.substring(12));}

    }

    for (int f = 0; f<10; f++){
        System.out.println("Count : " + Count.get(f));
        System.out.println("KEY : " + KEY.get(f));
        System.out.println("NONCE : " + NONCE.get(f));
        System.out.println("COUNTER : " +
COUNT.get(f));
        System.out.println("PLAINTEXT : " +
PT.get(f));

    }
    fw.write("[Enc]");
    fw.write("\n\n[192]");

    for (int j = 0; j < 100; j++){
        try {
            String counter_format =
NONCE.get(j).trim() + COUNTER.get(j).trim();
            byte[] counter_byte =
DatatypeConverter.parseHexBinary(counter_format);

            System.out.println(counter_byte.length);

            for (int k = 0; k < 16; k++){
                System.out.print(counter_byte[k]);
            }
            System.out.println();

            CK_ATTRIBUTE [] KeyTemp =
AESKey(KEY.get(j).trim());
            long Key =
p11.C_CreateObject(hSession,KeyTemp);

            byte[] ciphertext = new byte[4096];

            byte[] plaintext =
DatatypeConverter.parseHexBinary(PT.get(j).trim());

            CK_MECHANISM mech = new
CK_MECHANISM(CKM_AES_CTR);
            mech.pParameter = new
CK_AES_CTR_PARAMS(counter_byte);

            p11.C_EncryptInit(hSession,mech,Key);
            int outLength =
p11.C_Encrypt(hSession,0,plaintext,0,plaintext.length,0,cipher
text,0,ciphertext.length);
            ciphertext = Arrays.copyOf(ciphertext,
outLength);

```

Şekil 4.14: Devam.

```

System.out.println("*****");
                System.out.println("Count = " +
Count.get(j));

System.out.println(DatatypeConverter.printHexBinary(ciphertext
).toLowerCase());

System.out.println("*****");
                fw.write("\n\nCount = " + Count.get(j) +
"\nKEY = " + KEY.get(j) + "\nNONCE = " +
                NONCE.get(j) + "\nCOUNTER = " +
COUNTER.get(j) + "\nPLAINTEXT = " + PT.get(j) +
                "\nCIPHERTEXT = " +
DatatypeConverter.printHexBinary(ciphertext).toLowerCase());

                } catch (Exception e){
                System.out.println(e);
                }
            }
        }
    }
}

```

Şekil 4.14: Devam.

Bu işlemin sonucunda üretilen .rsp uzantılı çıktı dosyasından, ilk üç test vektörüne ait örnekler aşağıda sunulmuştur.

```

[Enc]

[192]

Count = 1
KEY = 94af7923cafbcb4802eb7385a08d9ba04262c5493a0d0db7c
NONCE = 57637fce2b89adf9
COUNTER = 09538fb23f85f2e2
PLAINTEXT = e75ff5e92d9cd1ff2b0f7e8646eb9c5c
CIPHERTEXT = 847ebf004d3fbfac1beda7548151a889

Count = 2
KEY = dc3d3e513f46effab958d27b4840bab2e70628aa0ce225e1
NONCE = 463704dddbba580
COUNTER = 10f918bbf39a063d
PLAINTEXT = 71ca34d7fb3ad98ebbd81d22019dba6c
CIPHERTEXT = 34a6c2f399095d7ed756a86da159a324

Count = 3
KEY = 6fcf1f3c99662d3ab4c480ddb45283cbf326e62c7ef2caa9
NONCE = 8466f80daf73ee6e
COUNTER = 147f91fe3105607f
PLAINTEXT = b64b1facee77b912ed7d7295f2ed29dd
CIPHERTEXT = b816790083432ddb5b589c59674fe92e

```

Şekil 4.15: AES-192 encryption test vektörleri sonucu.

- AES_CTR_Dec() Metodu:
 - Bu işlem AESCTRMCTDec128.req dosyasındaki CIPHERTEXT vektörleri üzerinde gerçekleştirilir.
 - Yukarıdakiyle benzer süreç izlenerek şifre çözme (decryption) işlemi yapılır.
 - Her vektörün çıktısı hem konsola yazdırılır hem de çıktı dosyasına kaydedilir.

```

@Test
public void AES_CTR_Dec() throws IOException,
PKCS11Exception {
//*****DECRYPTION*****
String line;
ArrayList<String> Count
    = new ArrayList<>(100); //count
ArrayList<String> KEY
    = new ArrayList<>(100);
ArrayList<String> NONCE
    = new ArrayList<>(100);
ArrayList<String> COUNTER
    = new ArrayList<>(100);
ArrayList<String> CT
    = new ArrayList<>(100);
File AESEncryption =
createFile("AESCTRMCTDec128.rsp"); try (FileWriter
fw = new FileWriter(AESEncryption)) {
try(BufferedReader br = new BufferedReader(new
FileReader("AESCTRMCTDec128.req"))) {

for (int i=0; i<603; i++) { //satır sayısı
    line = br.readLine();

    if (line.contains("Count ="))
{Count.add(line.substring(8));}
    else if (line.contains("KEY ="))
{KEY.add(line.substring(6));} else if
(line.contains("NONCE ="))
{NONCE.add(line.substring(8));} else if
(line.contains("COUNTER ="))
{COUNTER.add(line.substring(10));} else if
(line.contains("CIPHERTEXT ="))
{CT.add(line.substring(13));}

}

for (int f = 0; f<10; f++){
System.out.println("Count : " + Count.get(f));
System.out.println("KEY : " + KEY.get(f));
System.out.println("NONCE : " + NONCE.get(f));
System.out.println("COUNTER : " + COUNTER.get(f));
System.out.println("CIPHERTEXT : " + CT.get(f));

}
}

```

Şekil 4.16: AES_CTR_Dec() metodu.

```

fw.write("[Dec]");
fw.write("\n\n[128]");

for (int j = 0; j < 128; j++){
    try {

        String counter_format = NONCE.get(j).trim() +
        COUNTER.get(j).trim();
        byte[] counter_byte =
        DatatypeConverter.parseHexBinary(counter_format);

        CK_ATTRIBUTE [] KeyTemp =
        AESKey(KEY.get(j).trim());
        long Key = p11.C_CreateObject(hSession,KeyTemp);

        byte[] plaintext = new byte[4096];

        byte[] sourceData =
        DatatypeConverter.parseHexBinary(CT.get(j).trim());

        CK_MECHANISM mech = new CK_MECHANISM(CKM_AES_CTR);

        mech.pParameter = new CK_AES_CTR_PARAMS(counter_byte);

        p11.C_DecryptInit(hSession,mech,Key);
        int outLength =
        p11.C_Decrypt(hSession,0,sourceData,0,sourceData.length,0,
        plaintext,0,plaintext.length);
        plaintext = Arrays.copyOf(plaintext, outLength);

        System.out.println("*****");
        System.out.println("Count = " + Count.get(j));
        System.out.println(DatatypeConverter.printHexBinary(plaintext).toLowerCase());

        System.out.println("*****");
        fw.write("\n\nCount = " + Count.get(j) + "\nKEY
        = " + KEY.get(j) +
            "\nNONCE = " + NONCE.get(j) + "\nCOUNTER
        = " + COUNTER.get(j) + "\nCIPHERTEXT = " + CT.get(j) +
            "\nPLAINTEXT = " +
            DatatypeConverter.printHexBinary(plaintext).toLowerCase());
        } catch (Exception e){
            System.out.println(e);
        }
    }
}
}

```

Şekil 4.16: Devam.

Bu işlemin sonucunda üretilen .rsp uzantılı çıktı dosyasından, ilk üç test vektörüne ait örnekler aşağıda sunulmuştur.

```
[Dec]

[128]

Count = 1
KEY = 5f55ecc859e792f4b911404be24ba8bc
NONCE = d5b6ccd271bbc0a5
COUNTER = 60aa0e64724db7ff
CIPHERTEXT = 2fd786d6767f15c72304ef53cceab946
PLAINTEXT = c7e28ba5320b1d51d35c601bfb9157ab

Count = 2
KEY = 3908cca5d3adbb2a79fadeefbc46fc9e4
NONCE = 25fe2fdf6ee0b327
COUNTER = 13d4c80750f7ea44
CIPHERTEXT = c5be51f1087d6d281509ac642dae7f01
PLAINTEXT = 0c66e78c6129ec234b7170ba41fc8c53

Count = 3
KEY = 6b10278e1039f066b4a414771a867a6a
NONCE = eebf2b8ebe8988f8
COUNTER = a41b199f71e4b9de
CIPHERTEXT = 714a803a29a9e178736d59ac52f9bfb1
PLAINTEXT = dd776a91bd5b1a24a3e5f52f8bd9ff0e
```

Şekil 4.17: AES-128 decryption test vektörleri sonucu.

- BC_AES_CTR_Enc() Metodu:
 - Aynı test vektörleri Bouncy Castle kütüphanesi ile işlenir.
 - Amaç, HSM üzerinden elde edilen çıktıların bir Java kriptografi sağlayıcısı ile tutarlılığını kontrol etmektir.

```
@Test
public void BC_AES_CTR_Enc() throws IOException { // BOUNCY CASTLE
    İLE ENC//
    String line;
    ArrayList<String> Count
        = new ArrayList<>(100);
    ArrayList<String> KEY
        = new ArrayList<>(100);
    ArrayList<String> NONCE
        = new ArrayList<>(100);
    ArrayList<String> COUNTER
        = new ArrayList<>(100);
    ArrayList<String> PT
        = new ArrayList<>(100);
    File AESEncryption = createFile("AESCTRMCTEnc192_BC.rsp");
    try (FileWriter fw = new FileWriter(AESEncryption)) {
        try (BufferedReader br = new BufferedReader(new
            FileReader("AESCTRMCTEnc192.req"))) {
```

Şekil 4.18: BC_AES_CTR_Enc() metodu.

```

for (int i=0; i<603; i++) {
    line = br.readLine();

    if (line.contains("Count ="))
{Count.add(line.substring(8));}
    else if (line.contains("KEY ="))
{KEY.add(line.substring(6));}
    else if (line.contains("NONCE ="))
{NONCE.add(line.substring(8));}
    else if (line.contains("COUNTER ="))
{COUNTER.add(line.substring(10));}
    else if (line.contains("PLAINTEXT ="))
{PT.add(line.substring(12));}

}

for (int f = 0; f<10; f++){
    System.out.println("Count : " + Count.get(f));
    System.out.println("KEY : " + KEY.get(f));
    System.out.println("NONCE : " + NONCE.get(f));
    System.out.println("COUNTER : " + COUNTER.get(f));
    System.out.println("PLAINTEXT : " + PT.get(f));

}

fw.write("[Enc]");
fw.write("\n\n[192]");

for (int j = 0; j < 100; j++) {
    try {

        String counter_format = NONCE.get(j).trim() +
COUNTER.get(j).trim();
        byte[] counter_byte =
DatatypeConverter.parseHexBinary(counter_format);
        byte[] secret_key =
DatatypeConverter.parseHexBinary(KEY.get(j).trim());

        Key skeyspec = new SecretKeySpec(secret key,
"AES");

        Security.addProvider(new
BouncyCastleProvider());
        Cipher encryptionCipher =
Cipher.getInstance("AES/CTR/NoPadding", "BC");
        IvParameterSpec iv = new
IvParameterSpec(counter_byte);
        encryptionCipher.init(Cipher.ENCRYPT_MODE,
skeyspec, iv);

        byte[] msg =
DatatypeConverter.parseHexBinary(PT.get(j));
        encryptionCipher.doFinal(msg);

        byte[] encryptedMsg =
encryptionCipher.doFinal(msg);

        String ciphertext =
Hex.toHexString(encryptedMsg);

```

Şekil 4.18: Devam.

```

        System.out.println("CipherText = " +
ciphertext);

        fw.write("\n\nCount = " + Count.get(j) + "\nKEY
= " + KEY.get(j) +
        "\nNONCE = " + NONCE.get(j) + "\nCOUNTER
= " + COUNTER.get(j) +
        "\nPLAINTEXT = " + PT.get(j) +
"\nCIPHERTEXT = " + ciphertext.toLowerCase());

        } catch (Exception e){
            System.out.println(e);
        }
    }
}
}
}

```

Şekil 4.18: Devam.

Bu işlemin sonucunda üretilen .rsp uzantılı çıktı dosyası, Şekil 4.15’te sunulan sonuçlarla birebir örtüşmektedir.

4.3.4. Bellek ve Zaman Performans Analizi

AES algoritmasının CTR modundaki doğruluğu test edildikten sonra, aynı test senaryoları kullanılarak şifreleme ve şifre çözme işlemlerinin sistem kaynaklarına etkisi değerlendirilmiştir. Bu kapsamda her bir işlem için kullanılan bellek miktarındaki değişim (Δ Memory) ve işlem süresi (ms) ölçülerek analiz edilmiştir.

Bellek ölçümleri, Java sanal makinesi üzerinde çalışan programın heap belleği içindeki tüketimi temel alacak şekilde yapılmıştır. Ölçümlerden önce “System.gc()” komutu ile çöp toplayıcı manuel olarak çağrılmış, ardından “Runtime.getRuntime().totalMemory()” ve “freeMemory()” metotları yardımıyla şifreleme ve çözme işlemleri öncesi ve sonrası bellek kullanımı belirlenmiştir. Zaman ölçümleri ise “System.nanoTime()” metodu ile gerçekleştirilmiş ve milisaniye (ms) cinsine dönüştürülmüştür.

Her biri 100 test vektöründen oluşan AES-128, AES-192 ve AES-256 anahtar uzunlukları için yapılan şifreleme ve şifre çözme işlemlerine ait ortalama bellek ve zaman değerleri aşağıda sunulmuştur:

Tablo 4.1: Bellek ve zaman performansı

AES Modu	İşlem Türü	ΔMemory (KB)	Süre (ms)
AES-128	ENC	13.275	375
AES-192	ENC	13.275	338
AES-256	ENC	13.274	333
AES-128	DEC	13.219	458
AES-192	DEC	13.219	368
AES-256	DEC	13.218	352

Tablodaki değerler incelendiğinde, her üç anahtar uzunluğu için de bellek tüketiminin oldukça benzer olduğu görülmektedir. Δ Memory değerleri, işlem türünden (ENC/DEC) ve anahtar uzunluğundan bağımsız olarak yaklaşık 13.2 KB seviyesindedir. Bu durum, kullanılan PKCS#11 arayüzü ve SoftHSM ortamının, şifreleme algoritmasının içsel karmaşıklığından bağımsız olarak benzer bellek ayak izi ile çalıştığını göstermektedir.

Zaman performansı açısından değerlendirildiğinde ise, 128-bit anahtar uzunluğu ile yapılan şifreleme ve özellikle çözme işlemlerinin, diğer anahtar uzunluklarına kıyasla biraz daha uzun sürdüğü gözlemlenmiştir. Bu farkların küçük olması ve beklenen sıralamanın aksine bazı sapmalar göstermesi, işlem sürelerinin sadece anahtar uzunluğu değil aynı zamanda sistem kaynaklarının anlık durumları ve Java sanal makinesinin optimizasyonları gibi etkenlerden de etkilendiğini göstermektedir.

Ayrıca, işlem sırasında cihazın sıcaklığı izlenmiş ancak sıcaklık değerlerinde ölçülebilir bir artış (delta) tespit edilmemiştir. Bu durum, gerçekleştirilen AES-CTR şifreleme ve şifre çözme işlemlerinin nispeten düşük işlem yükü ve kısa süreli gerçekleşmesi nedeniyle cihazın termal dengesinin korunmasından kaynaklanmaktadır. Dolayısıyla, performans analizinde sıcaklık değişimi dikkate alınmamıştır.

Sonuç olarak, yapılan bellek ve zaman ölçümleri, algoritmanın kaynak tüketimi açısından oldukça tutarlı ve yönetilebilir seviyelerde çalıştığını göstermektedir. Özellikle sabit kalan bellek ayak izi, PKCS#11 üzerinden çalışan HSM benzeri ortamların deterministik ve kontrollü kaynak yönetimi sunduğunu ortaya koymaktadır.

4.3.5. Doğrulama Yönteminin Özgünlüğü ve Yaklaşım Farkı

Bu çalışmanın önemli katkılarından biri, algoritma doğrulama sürecinde kullanılan yaklaşımın klasik yöntemlerden farklı olmasıdır. Bilindiği üzere, FIPS 140-3 kapsamında algoritma doğrulama süreçleri CAVP aracılığıyla yürütülür; burada ürün, NIST tarafından yayımlanan test vektörlerine yanıt verir ve sonuçları doğrudan resmi bir makam tarafından doğrulanır. Benzer şekilde, ISO/IEC 19790 standardı çerçevesinde Türkiye'de TSE (Türk Standartları Enstitüsü) gibi ulusal otoritelerce algoritma testleri yürütülmekte ve onay mekanizması sağlanmaktadır.

Ancak bu tez kapsamında, doğrudan bir sertifikasyon otoritesine erişim olmaksızın, algoritma doğrulama testi için farklı ve pratik bir yaklaşım benimsenmiştir. FIPS 140-3 tarafından onaylanmış olan Bouncy Castle kütüphanesi referans olarak alınmıştır. Doğrulama süreci şu şekilde yapılandırılmıştır:

1. Test vektörleri, ilk olarak PKCS#11 standardı üzerinden haberleşilen SoftHSM ürününe uygulanmıştır.
2. Aynı test vektörleri, Bouncy Castle kütüphanesi üzerinde işlenmiş ve referans çıktılar elde edilmiştir.
3. Eğer SoftHSM'in verdiği çıktılar, Bouncy Castle'dan elde edilen sonuçlarla birebir uyuyorsa, bu durumda ürünün ilgili algoritmaları doğru ve güvenli şekilde implemente ettiği kabul edilmiştir. Bu eşleşme doğrultusunda ürünün algoritma işleyişinin FIPS onaylı referans çıktılarla tutarlılığı değerlendirilmiştir.

Bu yöntem, resmi CAVP süreci işletilmeksizin de standartlara uygun referans çıktılar elde edilmesini mümkün kılmakta; böylece, doğrulama süreci daha erişilebilir ve tekrarlanabilir hale gelmektedir. Bu yöntem ile geliştiriciler, ürünlerini değerlendirme makamına göndermeden önce kendi iç test süreçlerinde FIPS düzeyinde ön doğrulama yapabilme imkanına erişebilmektedirler. Ayrıca bu süreç, akademik düzeyde güvenilirlik açısından da anlamlı bir doğrulama imkanı sunmaktadır. Açık kaynaklı sistemlerde sınırlı kaynaklarla doğrulama testi gerçekleştirmek isteyen araştırmacılar için uygulanabilir ve tekrarlanabilir bir yaklaşımdır.

Bu yönüyle çalışma, kriptografik ürün geliştirme döngüsünde erken aşama testlere katkı sağlayan bir model önerisi sunmakta; aynı zamanda açık kaynak araçlarla

standartlara uyumlu şekilde gerçekleştirilebilecek doğrulama stratejileri üzerine değerli bir örnek teşkil etmektedir.

4.3.6. Tezin Katkısı ve Özgünlük Değeri

Bu tez, Common Criteria standardının 2022 sürümüne odaklanan ilk Türkçe akademik çalışmalardan biri olması bakımından literatürde önemli bir boşluğu doldurmaktadır. Gerek ulusal düzeyde gerekse akademik çevrelerde, yeni versiyonun yapısal ve kavramsal değişikliklerine dair kaynak eksikliği göz önünde bulundurulduğunda, bu çalışma, okuyucuların CC:2022'ye dair temel bir farkındalık geliştirmesine ve güncel standartlara aşinalık kazanmasına doğrudan katkı sunmaktadır.

Tez kapsamında sunulan yöntemsel yaklaşım; Common Criteria'ya ilgi duyan araştırmacılar, değerlendirme kuruluşları, ürün geliştiricileri ve yüksek lisans/doktora düzeyinde çalışma yürüten öğrenciler için somut bir örnek teşkil etmektedir. Özellikle algoritma doğrulama süreçlerinin ele alınış biçimi, güvenlik değerlendirme alanına kavramsal derinlik kazandıracak ve benzer çalışmalara metodolojik bir zemin sunacak şekilde yapılandırılmıştır.

Ayrıca, tezde tercih edilen açık kaynak çözümler, standartlara dayalı ürün doğrulama süreçlerinin yalnızca ticari HSM cihazlarıyla değil, daha erişilebilir ve deneysel ortamlarda da gerçekleştirilebileceğini ortaya koyarak, güvenlik değerlendirmelerine yönelik daha kapsayıcı ve ulaşılabilir bir yaklaşım sunmaktadır. Bu yönüyle çalışma, hem akademik araştırmalarda hem de uygulamalı geliştirme projelerinde referans olarak değerlendirilebilecek niteliktedir.

Sonuç olarak, bu tez; sadece bir ürünün algoritma doğrulama sürecini belgelemekle kalmayıp, aynı zamanda CC:2022'nin yerli literatürde tanıtılması ve anlaşılmasına yönelik öncü bir çaba olarak değerlendirilebilir. Bu yönüyle gelecekte benzer konularda yapılacak çalışmalara hem yöntemsel hem de içeriksel olarak zemin hazırlaması beklenmektedir.

5. SONUÇ VE ÖNERİLER

5.1. Sonuç

Bu çalışmada, Common Criteria standardı kapsamında bir HSM ürününün değerlendirilmesi kuramsal ve pratik yönleriyle ele alınmıştır. Açık kaynaklı ve yazılım tabanlı bir çözüm olan SoftHSM kullanılarak, algoritma doğrulama süreçlerinin gerçekleştirilebilirliği test edilmiştir. Bu kapsamda, PKCS#11 standardı üzerinden Java diliyle AES algoritmasının CTR modunda şifreleme ve şifre çözme işlemleri yapılmış, test vektörleri kullanılarak çıktıların doğruluğu değerlendirilmiştir. Algoritmaların hem SoftHSM hem de FIPS onaylı bir açık kaynak kütüphane olan Bouncy Castle üzerinden yürütülmesiyle karşılaştırmalı bir doğrulama sağlanmıştır.

Bu yaklaşım, doğrudan resmi bir sertifikasyon sağlamamakla birlikte, akademik düzeyde güvenilirlik açısından anlamlı bir doğrulama sunmaktadır. Aynı zamanda açık kaynaklı sistemlerde, sınırlı kaynaklarla doğrulama testi gerçekleştirmek isteyen araştırmacılar için uygulanabilir ve tekrarlanabilir bir yöntem önermektedir. SoftHSM'in kolay yapılandırılabilir olması, PKCS#11 standardıyla uyumlu çalışması ve açık kaynaklı bir ortamda doğrulama yapılabilmesi, özellikle kavramsal ispat ve eğitimsel amaçlarla yürütülen değerlendirmelerde ciddi bir esneklik sağlamaktadır.

Gerçekleştirilen doğrulama çalışması, CC:2022 standardının getirdiği daha yapılandırılmış değerlendirme yöntemleri ile de örtüşmektedir. CC:2022 sürümü, önceki versiyonlara göre daha açık ve modüler bir değerlendirme yaklaşımı sunmaktadır. Bu bağlamda, test vektörleriyle yapılan algoritma doğrulama faaliyetleri, CC:2022'nin teknik derinliği artırılmış yeni yaklaşımına uygun olarak nesnel ve tekrarlanabilir bir değerlendirme kanıtı sunmuştur.

5.2. Öneriler

- Açık kaynaklı SoftHSM gibi ürünlerin, geliştirici ortamlarında kullanılmak üzere belirli test çerçeveleri ile birlikte sunulması, ürünlerin güvenlik değerlendirmesine hazırlanma süreçlerini kolaylaştırabilir.

- Bu tezde uygulanan yöntem, sertifikasyon öncesi ürün içi algoritma doğrulama testleri için bir temel teşkil edebilir. Böylece geliştiriciler, ürünlerini değerlendirmeye sunmadan önce ön analizler gerçekleştirebilir ve eksiklikleri erken aşamada tespit edebilir.
- Geliştirilen test altyapısı AES algoritmasıyla sınırlı kalmayıp, RSA, ECDSA, SHA gibi farklı algoritmaların doğrulama testlerine de uyarlanabilir. Böylece daha geniş kapsamlı bir değerlendirme yapılabilir.
- Kuantum bilgisayar teknolojilerinin gelişimiyle birlikte klasik kriptografik algoritmaların güvenliği sorgulanır hale gelmiştir. Bu bağlamda, post-kuantum kriptografi (PQC) alanında geliştirilen algoritmaların da benzer doğrulama süreçlerine tabi tutulması gerekmektedir. Lattice tabanlı veya kod tabanlı algoritmalar gibi kuantum dirençli yapılar, yakın gelecekte güvenlik ürünlerinin temel bileşenleri arasında yer alacaktır. Bu nedenle, benzer çalışmaların CRYSTALS-Kyber, Dilithium gibi NIST PQC finalist algoritmalarına yönelik test vektörleri ile gerçekleştirilmesi, CC değerlendirme sürecinin kuantum sonrası döneme hazır hale gelmesi açısından kritik bir adımdır.

KAYNAKLAR

- [1] ISO, (2018), Information technology — Security techniques — Information security management systems — Overview and vocabulary, ISO/IEC 27000:2018, International Organization for Standardization.
- [2] Shannon C. E., (1949), “Communication Theory of Secrecy Systems”, Bell System Technical Journal, 28 (4), 656–715.
- [3] Stallings W., (2017), “Cryptography and Network Security: Principles and Practice”, 7th Edition, Pearson.
- [4] CCDB, (2022), Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CC:2022 Revision 1, Common Criteria Development Board.
- [5] Barker E., Barker W., (2019), “Recommendation for Key Management: Part 2 – Best Practices for Key Management Organizations”, Technical Report No: NIST SP 800-57 Part 2 Rev. 1, National Institute of Standards and Technology.
- [6] CCDB, (2022), Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the Specification of Evaluation Methods and Activities, CC:2022 Revision 1, Common Criteria Development Board.
- [7] CCRA, (2025), <https://www.commoncriteriaportal.org/ccra/members/index.cfm>, (Erişim Tarihi: 22/06/2025).
- [8] CCDB, (2022), Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined Packages of Security Requirements, CC:2022 Revision 1, Common Criteria Development Board.
- [9] CCDB, (2022), Common Methodology for Information Technology Security Evaluation, CEM:2022 Revision 1, Common Criteria Development Board.
- [10] Ferguson N., Schneier B., Kohno T., (2010), “Cryptography Engineering: Design Principles and Practical Applications”, Wiley.
- [11] Paar C., Pelzl J., (2010), “Understanding Cryptography: A Textbook for Students and Practitioners”, Springer.
- [12] NIST, (2025), <https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program>, (Erişim Tarihi: 22/06/2025).
- [13] NCCoE, (2025), <https://www.nccoe.nist.gov/automation-nist-cryptographic-module-validation-program>, (Erişim Tarihi: 22/06/2025).

- [14] Schaffer K., (2020), “FIPS 140-3 derived test requirements (DTR): CMVP validation authority updates to ISO/IEC 24759”, Technical Report No: NIST SP 800-140, National Institute of Standards and Technology.
- [15] NIST, (2019), Security requirements for cryptographic modules, FIPS 140-3, National Institute of Standards and Technology.
- [16] Hashira C., (2025), CyberHashira/PKCS-11-Tutorials, <https://github.com/CyberHashira/PKCS-11-Tutorials>, (Erişim Tarihi: 22/06/2025).

ÖZGEÇMİŞ

Mert GÜLER, lisans öğrenimini 2017 yılında Karabük Üniversitesi Elektrik-Elektronik Mühendisliği (İngilizce) bölümünde tamamlamış ve bu programdan 3.12 genel not ortalaması ile mezun olmuştur. Yüksek lisans eğitimine Gebze Teknik Üniversitesi Elektronik Mühendisliği Anabilim Dalı'nda devam etmiş ve 2025 yılında mezun olmuştur. Yüksek lisans öğrenimini 3.43 genel not ortalaması ile tamamlamıştır. 2019 yılından itibaren TÜBİTAK BİLGEM bünyesinde çalışmaktadır.

TEZ METNİNDEN TÜRETİLEN YAYINLAR/SUNUMLAR

Güler M., (2025), “Common Criteria Değerlendirme Sürecinin CC:2022 Versiyon Özelinde İncelenmesi”, Ayasofya Multidisipliner Lisansüstü Öğrenci Kongresi, 138, İstanbul, Türkiye, 20-21 Mayıs.

Güler M., (2025), “Bir HSM Ürününün Common Criteria (CC:2022) Kapsamında Değerlendirilmesi ve Algoritma Doğrulama Süreci”, 9. GTÜ Lisansüstü Araştırmalar Sempozyumu, 37, Kocaeli, Türkiye, 26-27 Mayıs.