
GAZİ ÜNİVERSİTESİ
Gazi
1926

FPGA ÜZERİNDE İLERİ ŞİFRELEME YÖNTEMİ TASARLANMASI

Cemil Baki KIYAK

DOKTORA TEZİ
İLERİ TEKNOLOJİLER ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AĞUSTOS 2025

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Cemil Baki KIYAK

18/08/2025

FPGA ÜZERİNDE İLERİ ŞİFRELEME YÖNTEMİ TASARLANMASI

(Doktora Tezi)

Cemil Baki KIYAK

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Ağustos 2025

ÖZET

Bu çalışma, gömülü sistemleri kod hırsızlığına karşı korumak amacıyla, kritik bir literatür boşluğunu dolduran, enerji-farkındalıklı (energy-aware) hibrit bir güvenlik çerçevesi (HGÇ) sunmaktadır. Yaklaşım; bit akışı şifrelemesi, mesaj-başı dinamik anahtar üretimi ve Dinamik Fonksiyon Değişimi (DFX) tabanlı bellek yanıltmasını bütünleştirerek, ROM kazıma ve yan kanal analizleri gibi fiziksel müdahalelere ve iletişim dinlemesi gibi mantıksal saldırılara karşı katmanlı bir donanım-yazılım karşı önlemi sunmaktadır. Xilinx Zynq-7020 SoC üzerinde prototiplenen yöntem, Ascon-XOF tabanlı bir Anahtar Türetme Fonksiyonu (KDF) ile her mesaj için taze bir Kimlik Doğrulamalı Şifreleme (AEAD) anahtarı üretir. Eş zamanlı olarak, gizli anahtar malzemesini ve seçili kod sayfalarını barındıran BRAM (Blok RAM) bloklarını DFX ile sürekli yer değiştirerek 'hareketli hedef' savunması uygular. Bu strateji fiziksel müdahaleleri zorlaştırırken, kriptografik gizlilik ve bütünlük ise hafif AEAD katmanı tarafından sağlanır. Yapılan donanım değerlendirmesi, önerilen HGÇ çerçevesinin yalnızca ASCON-128a ile değil, aynı zamanda ACORN-128, TinyJAMBU-128 ve JAMBU-PRESENT-128 gibi farklı hafif AEAD algoritmalarıyla da başarılı bir şekilde bütünleşebildiğini göstermiştir. Her bir algoritma için HGÇ'nin getirdiği ek gecikme ve enerji maliyeti nicel olarak ölçülmüş, böylece çerçevenin genel uygulanabilirliği ve ortogonalliği deneysel olarak kanıtlanmıştır. Çevrimiçi anahtar değişiminden kaçınarak ve uzun ömürlü sırların fiziksel konumunu sürekli değiştirerek, bu çerçeve FPGA tabanlı sistemlerde kod hırsızlığına karşı uygulanabilir ve katmanlı bir savunma sunmaktadır.

Bilim Kodu	:	93446
Anahtar Kelimeler	:	Kod hırsızlığı koruması, AEAD, Ascon-128a, hafif kriptografi, FPGA, Zynq-7000, Dinamik Fonksiyon Değişimi (DFX), hareketli hedef savunması, Yan Kanal Saldırısı (SCA) direnci, IoT, MITM
Sayfa Adedi	:	98
Danışman	:	Prof. Dr. Hasan Şakir BİLGE

DESIGNING ADVANCED ENCRYPTION METHOD ON FPGA

(Ph. D. Thesis)

Cemil Baki KIYAK

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

August 2025

ABSTRACT

This work presents an energy-aware hybrid security framework (HSF) to protect embedded systems against code theft, filling a critical literature gap. The framework integrates bitstream encryption, per-message dynamic key generation, and memory obfuscation based on Dynamic Function eXchange (DFX) to provide a layered hardware-software countermeasure against physical tampering, such as ROM scraping and side-channel analysis, as well as logical attacks like communication eavesdropping. Prototyped on a Xilinx Zynq-7020 SoC, the method employs an Ascon-XOF-based Key Derivation Function (KDF) to generate a fresh Authenticated Encryption with Associated Data (AEAD) key for each message. Concurrently, it implements a moving-target defense by continuously relocating the BRAM (Block RAM) blocks that host secret keying material and selected code pages using DFX. While this strategy thwarts physical attacks, cryptographic confidentiality and integrity are ensured by a lightweight AEAD layer. The hardware evaluation demonstrates that the proposed HSF framework can be successfully integrated not only with ASCON-128a but also with other diverse lightweight AEAD algorithms, such as ACORN-128, TinyJAMBU-128, and JAMBU-PRESENT-128. The overhead in latency and energy introduced by HSF was quantitatively measured for each algorithm, thereby experimentally proving the framework's general applicability and orthogonality. By avoiding online key exchange and continuously relocating the physical position of long-term secrets, the framework offers a practical and layered defense against code theft vectors in FPGA-based systems.

Science Code	:	93446
Key Words	:	Code theft protection, AEAD, Ascon-128a, lightweight cryptography, FPGA, Zynq-7000, Dynamic Function eXchange (DFX), moving-target defense, Side-Channel Attack (SCA) resistance, IoT, MITM
Page Number	:	98
Supervisor	:	Prof. Dr. Hasan Şakir BİLGE

TEŞEKKÜR

Bu tezin ortaya çıkmasını sağlayan ve doktora öğrenimim süresince desteklerini esirgemeyen tüm değerli hocalarıma, aileme ve dostlarıma en içten şükranlarımı sunarım. Bu sürecin mimarı, değerli danışman hocam Prof. Dr. Hasan Şakir BİLGE'ye sonsuz teşekkürlerimi sunarım. Kendisinin engin tecrübesi, yol gösterici eleştirileri ve sarsılmaz desteği, bu çalışmanın her aşamasında bana ışık tutmuştur. Akademik vizyonu ve sabırlı rehberliği, yalnızca bu tezin tamamlanmasında değil, bir araştırmacı olarak gelişimimde de en temel yapı taşı olmuştur. Tez izleme sürecimde jürimde yer alarak, kıymetli görüşleriyle çalışmamın teknik derinlik kazanmasına olanak tanıyan hocalarıma da teşekkürü bir borç bilirim. Yapıcı eleştirileri ve değerli yönlendirmeleri, araştırmamın doğru bir zeminde ilerlemesinde kilit rol oynamıştır. Hayatım boyunca bana koşulsuz sevgi ve destek sunan, başarılarımla her zaman gurur duyan canım annem Prof. Dr. Tülin KIYAK ve babam Prof. Dr. Suat KIYAK'a minnettarım. Bilgiye ve öğrenmeye olan tutkumu onlardan miras aldım; bu akademik yolculukta attığım her adımda onların ilhamı ve duaları en büyük gücüm oldu. Bu yoğun süreçteki en büyük dayanağım, hayat arkadaşım, sevgili eşime; varlığıyla hayatıma anlam katan çocuğuma ve tüm aileme sonsuz teşekkür ederim. Onların sabrı, anlayışı ve sevgisi olmasaydı bu yolculuk tamamlanamazdı. Beni her daim cesaretlendiren ve bu çalışmanın arkasındaki kahramanlar olan canım aileme minnettarım. Bu tez, en çok onların fedakârlığının ve desteğinin bir eseridir ve onlara armağan edilmiştir. Tez sürecinde fikir alışverişinde bulunduğumuz, desteğini ve dostluğunu her zaman hissettiğim değerli arkadaşım Dr. Gizem Büke ÖZTÜRK'e içtenlikle teşekkür ederim. Ayrıca, bu yolda bana eşlik eden, çalışmalarıma ilham veren ve motivasyonumu artıran tüm öğrencilerime ve yol arkadaşlarıma da teşekkürlerimi sunarım. Son olarak, bu çalışmanın herhangi bir aşamasında emeği geçen, ismini burada sayamadığım tüm dostlarıma ve meslektaşlarıma da en içten teşekkürlerimi sunarım.

Teşekkürlerimle.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xii

SİMGELER VE KISALTMALAR xiii

1. GİRİŞ 1

2. MEVCUT DURUM ÇALIŞMALARI 9

2.1. Bitstream Saldırıları ve Kod Koruma 9

2.2. Dinamik Yeniden Yapılandırma ve Hafif Kriptografi 10

3. KAVRAMSAL ÇERÇEVE 15

3.1. IoT ve Gömülü Sistemlerde Güvenlik Kavramları 15

3.2. Temel Kriptografik Yöntemler ve Protokoller 17

3.2.1. Asimetrik kriptografik yöntemler 18

3.2.2. Simetrik şifreleme ve hash yöntemleri 20

3.2.3. IoT'de kullanılan hafif kriptografi yaklaşımları 21

3.3. Donanımsal Güvenlik 22

3.4. PUF (Physically Unclonable Function) Tabanlı Sistemler 26

3.5. Saldırı Kavramları 29

3.5.1. Sözlük (Dictionary) ve brute force saldırıları 30

3.5.2. Yan kanal saldırıları (güç, EM, zamanlama analizleri) 30

3.5.3. Donanımsal manipülasyon ve truva atları (HTH) 31

Sayfa

3.5.4. Tersine mühendislik ve IP korsanlığı 31

3.5.5. Sonuç 32

4. MATERYALLER VE YÖNTEMLER 33

4.1. Hibrit Güvenlik Çerçevesi (HGÇ): Tasarım ve AEAD Arabirimi 33

4.1.1. Alıcı tarafı kabul denetimleri 37

4.1.2. İşletim prosedürleri 37

4.2. Deney Düzeneği 42

4.3. Donanım ve Yazılım Tasarımı 44

4.3.1. Platform ve üst düzey tasarım 45

4.3.2. Bellek ve konfigürasyon mimarisi 51

4.3.3. Güvenlik, çalışma zamanı ve uygulama ayrıntıları 56

4.3.4. BRAM'e gömülü kod sayfaları ve çalışma zamanı yükleme 58

4.3.5. Çoklu RM yerleşimi ve DFX zamanlama politikaları 60

4.3.6. Yer değiştirmenin güvenlik muhasebesi 61

5. BULGULAR 63

5.1. Ölçüm Düzeneği ve Disiplin 63

5.2. Üretici Tarafı Şifreleme (ZYNQ-7020) 64

5.3. Çözücü Tarafı Şifre Çözme (Arduino DUE) 67

5.4. Enerji Atfı ve DFX Ölçekleme 69

5.5. Güvenlik Analizi ve Saldırı Deneyleri 72

5.6. Bulguların Özeti 79

6. TARTIŞMALAR 81

7. SONUÇ VE ÖNERİLER 87

Sayfa

KAYNAKLAR 89

ÖZGEÇMİŞ 97

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. Literatürdeki temsilî FPGA güvenlik çalışmaları, sınırlamaları ve tespit edilen boşluklar 12

Çizelge 3.1. Donanım tabanlı güvenlik yöntemleri ve olası zafiyetleri..... 25

Çizelge 3.2. PUF tabanlı yaklaşımların avantaj ve dezavantajları 28

Çizelge 4.1. Nonce ve AAD alanlarının yapısı ve boyutları 34

Çizelge 4.2. Alıcı denetimleri 37

Çizelge 4.3. Alan semantiği ve güvenlik gerekçesi 40

Çizelge 4.4. Kısmi yeniden yapılandırma için konfigürasyon erişim portlarının karşılaştırması (PCAP/ICAP/MCAP)..... 53

Çizelge 4.5. Zynq-7020 üzerindeki bellek seçeneklerinin özellikleri ve kullanım gerekçesi 54

Çizelge 4.6. Güvenlik odaklı bellek seçimi (kazıma ve veri yolu izlemeye karşı)..... 56

Çizelge 4.7. Fonksiyon yükleyici parametreleri 59

Çizelge 5.1. Zynq-7020 şifreleme performansı: Referans AEAD'lerin HGÇ'li ve HGÇ'siz sürümlerinin karşılaştırması..... 65

Çizelge 5.2. Arduino DUE deşifreleme performansı: Referans AEAD'lerin HGÇ'li ve HGÇ'siz sürümlerinin karşılaştırması..... 68

Çizelge 5.3. Zynq üzerinde plato pencereleri (şönt yöntemi): Süre, akım, güç ve enerji 71

Çizelge 5.4. Kod koruma düzeneklerinin enerji/gecikme karşılaştırması (yalnız şönt ölçümü) 71

Çizelge 5.5. Veri sınırları (64-bit bloklar) 75

Çizelge 5.6. Şifre türleri ve AEAD durumu 76

Çizelge 5.7. Saldırı deneyi 1: Arduino Due üzerinde kaba-kuvvet doğrulama (DFX kapalı)..... 77

Çizelge 5.8. Saldırı deneyi 2: Arduino Due üzerinde sızıntı izolasyonu (DFX kapalı) 77

Çizelge **Sayfa**

Çizelge 5.9. Önerilen düzenek için saldırı dayanım özeti 78

Çizelge 6.1. Değerlendirilen platformlarda başa-başa karşılaştırma 81

Çizelge 6.2. Zynq-7020 (1×16 B): yalnız kripto gecikme/enerji ve ölçülen kompromi kapsamı 83

Çizelge 6.3. Farklı güvenlik yaklaşımlarının karşılaştırmalı analizi 84

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 1.1. Sabit konumlu bellek kullanımında QP tabanlı yan kanal saldırısı sırasında RAM'deki bitlerin okunması (Faraj ve Gebotys, 2021) (a. Kazıma, b. Bellek, c. Bitler) 2

Şekil 4.1. HGÇ–ASCON algoritma akış diyagramı. KDF (ASCON-XOF), Nonce/AAD, ASCON-128a 38

Şekil 4.2. Zynq-7000 SoC üst düzey (top level) blok diyagramı: PS–PL arabirimleri ve PCAP/ICAP konfigürasyon yolları gösterilmektedir (Kerling, 2019) 45

Şekil 4.3. Zynq-7020 üzerinde DFX kullanarak BRAM içeriklerini değiş-tokuş eden önerilen donanım mimarisi 46

Şekil 4.4. Dört yeniden yapılandırılabilir bölge ile yapılan tasarım 46

Şekil 4.5. Zynq-7020'nin altı saat bölgesine (clock region) ayrılmış PL topolojisi; sol üst bölgede PS yerleşimi bulunur. DFX bölge (RM) tanımları bu saat bölgesi sınırlarına göre yapılır (Kerling, 2019) 48

Şekil 4.6. Zynq-7020 PL üzerinde pblock sınırları ve iki RM (Pblock_Anahtar_Bolgesi / Pblock_B-Degisim_Bolgesi) için floorplan ekran görüntüsü; her RM içinde AXI BRAM Controller + BRAM yerleşimi..... 49

Şekil 4.7. Dört RM için floorplan ekran görüntüsü ve pblock sınırları; her RM içinde AXI BRAM Controller + BRAM yerleşimi 50

Şekil 4.8. Zynq-7020'nin PS/PL blok diyagramı ve sütun topolojisini gösteren şema (referans amaçlı genel görünüm) (Kerling, 2019)..... 51

Şekil 4.9. Yeniden yapılandırılabilir bir modülün (RM) iç görünümü 53

Şekil 4.10. BRAM üzerinde şifreli fonksiyon çalıştırma akış diyagramı 59

Şekil 4.11. DFX akışıyla her iletişim slot'unda rastgele bir yeniden yapılandırılabilir modülün (RM1–RM4) aktif edilmesi 61

Şekil 5.1. Rs = 18,408 Ω üzerinden Digilent Analog Discovery 3 ile alınan şönt yakalaması; UART, AEAD ve DFX platoları işaretlenmiştir 64

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklamalar
‖	Birleştirme (concatenation) işlemi.
α	Mesaj başına yapılan DFX takas (swap) sayısı.
ASCON-128a(K, N, A, M)	K anahtarı, N nonce'u, A ilişkili verisi ve M düz metni ile yapılan ASCON-128a kimlik doğrulamalı şifreleme işlemi.
ASCON-XOF128(...)	Girdilerden anahtar türetmek için kullanılan ASCON Genişletilebilir Çıktı Fonksiyonu.
C	Şifreli metin (Ciphertext).
C(n, k)	n'in k'lı kombinasyonu.
D	Anahtar sayısı (örneğin aygıt veya oturum sayısı).
dev_id	Aygıt tanımlayıcısı.
dom	Alan ayrımı (domain separation) için kullanılan protokol dizgesi.
E	Enerji (genellikle µWh cinsinden).
epoch, epoch_ctr	Yeniden başlatma sonrası tekrarı önleyen dönem (epoch) sayacı.
feat	Uygulamaya özgü özellikleri taşıyan alan.
H_loc	Konum belirsizliği (entropi).
I(t)	Zamana bağlı anlık akım.
idx	POOL dizisi içindeki dilim (slice) indeksi.
K	Her bir BRAM içindeki adreslenebilir konum sayısı.
K_i	Belirli bir i durumu için türetilmiş oturum anahtarı.

Simgeler	Açıklamalar
K_TR	DFX rotasyonu için kullanılan taşıma anahtarı.
L	Nonce uzunluğu (bit cinsinden).
M	1. Düz metin (Message/Plaintext). 2. Yeniden yapılandırılabilir BRAM/modül sayısı (Güvenlik Analizi bölümünde). 3. Anahtar başına mesaj sayısı (Güvenlik Analizi bölümünde).
msg, msg_ctr	Tek bir epoch içindeki mesaj sayacı.
N_swap	Mesaj başına yapılan takas (swap) sayısı.
nonce	Sürekli artan tek kullanımlık sayı.
P(n, k)	n'in k'lı permütasyonu.
P(t)	Zamana bağlı anlık güç.
P_coll	Çakışma olasılığı (Collision probability).
Pr[forge]	Sahtecilik olasılığı (Forgery probability).
pack128(...)	Girdileri 128-bitlik bir yapıya paketleyen fonksiyon.
POOL[idx]	Gizli malzemeyi içeren şifreli havuz ve onun idx indisli elemanı.
q	Saldırganın yaptığı toplam sorgu/deneme sayısı.
Q	Toplam mesaj sayısı (D·M).
R	1. Kullanılabilir DFX yuva (slot) sayısı (Güvenlik Analizi bölümünde). 2. Etkin veri yolu verimi (throughput) (Bant genişliği notunda).
rot_ctr	DFX rotasyon sayacı.
R_s	Şönt direncinin değeri.
s	Kök sır üzerinde dilim (slice) oluşturmak için kullanılan başlangıç ofseti.
S_ROOT	Aygıta özgü, uzun ömürlü 256-bitlik kök sır.

Simgeler	Açıklamalar
t	Kimlik doğrulama etiketinin (tag) uzunluğu (bit cinsinden).
tag	Kimlik doğrulama etiketi.
ts	İsteğe bağlı zaman damgası (timestamp) alanı.
Δt	Geçen süre, zaman penceresi.
ver	Protokol sürüm alanı.
V_load(t)	Zamana bağlı anlık yük gerilimi.
V_shunt(t)	Zamana bağlı anlık şönt gerilimi.

Kısaltmalar	Açıklamalar
A5/1	GSM için Akış Şifreleme Algoritması
AAD	İlişkili Veri (Associated Data)
ACORN	Hafif AEAD Algoritması
AEAD	Kimlik Doğrulamalı Şifreleme (Authenticated Encryption with Associated Data)
AES	Gelişmiş Şifreleme Standardı (Advanced Encryption Standard)
ARM	Advanced RISC Machine
ASCON	NIST Hafif AEAD Algoritmaları Kazananı
ASIC	Uygulamaya Özel Tümdevre (Application-Specific Integrated Circuit)
AXI	Gelişmiş Genişletilebilir Arayüz (Advanced eXtensible Interface)
BBRAM	Batarya Destekli RAM (Battery-Backed RAM)
BF	Kaba Kuvvet (Brute-Force)
BRAM	Blok RAM (Block RAM)
CBC	Şifre Bloğu Zincirleme (Cipher Block Chaining)

Kısaltmalar	Açıklamalar
CLB	Yapılandırılabilir Mantık Bloğu (Configurable Logic Block)
CPA	Korelasyon Güç Analizi (Correlation Power Analysis)
CRP	Challenge-Response Çifti (Challenge-Response Pair)
CSPRNG	Kriptografik Olarak Güvenli Sözde Rastgele Sayı Üreteci
DDR	Çift Veri Hızlı Bellek
DES	Veri Şifreleme Standardı (Data Encryption Standard)
Device DNA	Xilinx FPGA Tanımlayıcısı
DFT	Tasarım için Test Edilebilirlik (Design for Testability)
DFX	Dinamik Fonksiyon Değişimi (Dynamic Function eXchange)
DH	Diffie-Hellman
DL	Derin Öğrenme (Deep Learning)
DMA	Doğrudan Bellek Erişimi (Direct Memory Access)
DPA	Diferansiyel Güç Analizi (Differential Power Analysis)
DPR	Dinamik Kısmi Yeniden Yapılandırma (Dynamic Partial Reconfiguration)
DSP	Sayısal Sinyal İşlemcisi (Digital Signal Processor)
ECB	Elektronik Kod Defteri (Electronic Codebook)
ECC	Eliptik Eğri Kriptografisi (Elliptic Curve Cryptography)
EEG	Elektroensefalografi (Electroencephalography)
eFUSE	Tek Seferlik Programlanabilir Bellek (Elektronik Sigorta)
EM	Elektromanyetik (Electromagnetic)

Kısaltmalar	Açıklamalar
ETA	Tahmini Varış Zamanı (Estimated Time of Arrival)
FIFO	İlk Giren İlk Çıkar (First-In, First-Out)
FPGA	Alanda Programlanabilir Kapı Dizileri (Field-Programmable Gate Array)
FSBL	İlk Aşama Önyükleyici (First Stage Bootloader)
FW	Aygıt Yazılımı (Firmware)
G/Ç	Giriş/Çıkış (Input/Output)
GSM	Mobil İletişim için Küresel Sistem (Global System for Mobile Communications)
HGÇ	Hibrit Güvenlik Çerçevesi
HKDF	HMAC Tabanlı Anahtar Türetme Fonksiyonu
HSM	Donanım Güvenlik Modülü (Hardware Security Module)
HTH	Donanım Truva Atı (Hardware Trojan Horse)
ICAP	Dahili Yapılandırma Erişim Portu (Internal Configuration Access Port)
IND-CPA	Seçilmiş Düz Metin Saldırılarına Karşı Ayırt Edilemezlik
INT-CTXT	Şifreli Metin Bütünlüğü (Integrity of Ciphertext)
IP	Fikrî Mülkiyet (Intellectual Property)
IoT	Nesnelerin İnterneti (Internet of Things)
JAMBU	SIMON Tabanlı AEAD Algoritması
KDF	Anahtar Türetme Fonksiyonu (Key Derivation Function)
KiB	Kibibyte
LFSR	Doğrusal Geri Beslemeli Kaydırmalı Yazmaç
LUT	Arama Tablosu (Look-Up Table)

Kısaltmalar	Açıklamalar
MCAP	Management Configuration Access Port
MD5	Message-Digest Algorithm 5
MITM	Ortadaki Adam (Man-in-the-Middle)
NIST	Ulusal Standartlar ve Teknoloji Enstitüsü
OCM	Çip Üstü Bellek (On-Chip Memory)
OTP	Tek Seferlik Programlanabilir (One-Time Programmable)
PCAP	İşlemci Yapılandırma Erişim Portu (Processor Configuration Access Port)
PL	Programlanabilir Mantık (Programmable Logic)
PMU	Performans İzleme Birimi (Performance Monitoring Unit)
PRESENT	Hafif Blok Şifresi
PRF	Sözde Rastgele Fonksiyon (Pseudorandom Function)
PRNG	Sözde Rastgele Sayı Üreteci (Pseudorandom Number Generator)
PS	İşlem Sistemi (Processing System)
PSK	Önceden Paylaşılan Anahtarlar (Pre-Shared Key)
PUF	Fiziksel Olarak Klonlanamaz Fonksiyon (Physical Unclonable Function)
PWM	Darbe Genişlik Modülasyonu (Pulse-Width Modulation)
QP	Hareketsiz Fotonik (Quiescent Photonics)
RDPR	Uzaktan Dinamik Kısmi Yeniden Yapılandırma
RM	Yeniden Yapılandırılabilir Modül (Reconfigurable Module)
RO	Ring Osilatör (Ring Oscillator) veya Rastgele Kahin (Random Oracle)

Kısaltmalar	Açıklamalar
ROM	Salt Okunur Bellek (Read-Only Memory)
RP pin	Yeniden Yapılandırılabilir Bölüm Pini (Reconfigurable Partition Pin)
RSA	Rivest-Shamir-Adleman
SAT	Boolean Sağlanabilirlik Problemi (Boolean Satisfiability Problem)
SCA	Yan Kanal Saldırıları (Side-Channel Analysis)
SHA	Güvenli Özet Algoritması (Secure Hash Algorithm)
SoC	Çip Üzeri Sistem (System-on-Chip)
SRAM	Statik Rastgele Erişimli Bellek (Static Random-Access Memory)
TinyJAMBU	Çok Hafif AEAD Algoritması
TMTO	Zaman-Bellek Ödünleşimi (Time-Memory Trade-Off)
TRNG	Gerçek Rastgele Sayı Üreteci (True Random Number Generator)
UART	Evrensel Asenkron Alıcı/Verici (Universal Asynchronous Receiver/Transmitter)
URAM	UltraRAM
XOF	Genişletilebilir Çıktı Fonksiyonu
XOR	Özel VEYA (Exclusive OR)

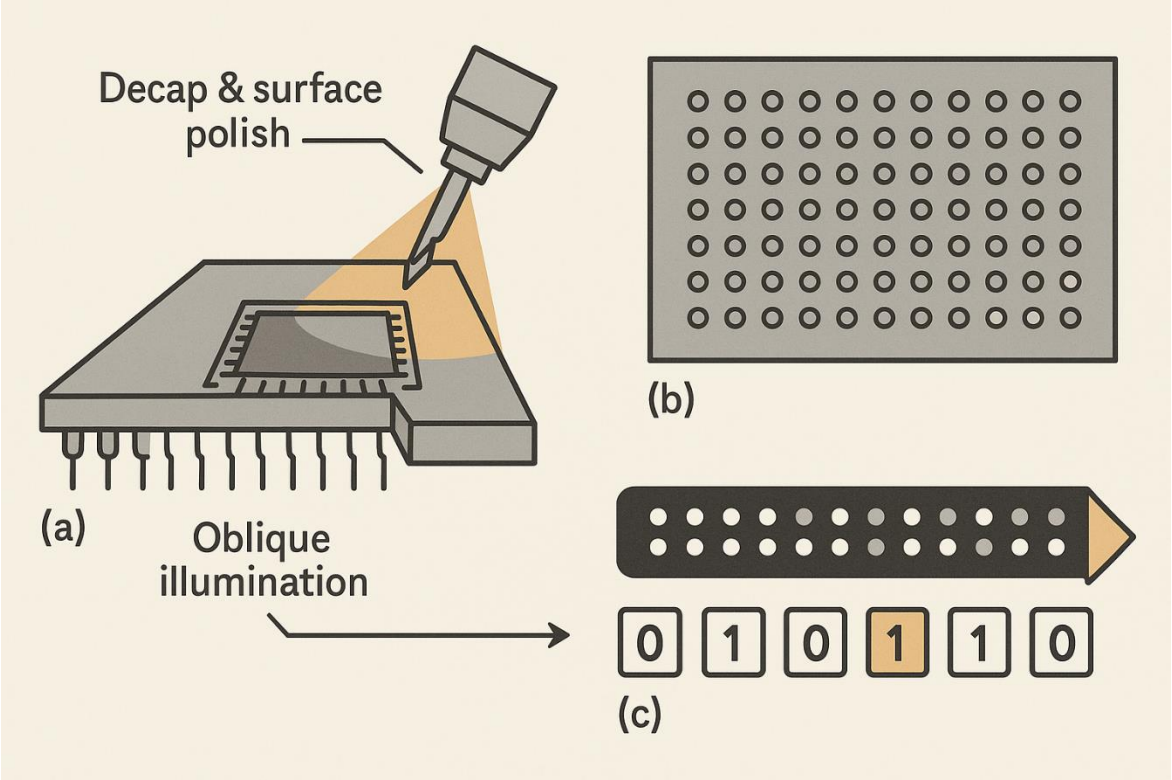
1. GİRİŞ

Gömülü sistem tümdevrelerine dayanan ticari ürünlerde, kod hırsızlığı, kalıcı olmayan belleğin doğrudan okunması (diğer adıyla "ROM kazıma"), yan kanal saldırıları ve iletişim dinlemesi ciddi bir sorun oluşturur (Proulx vd., 2023; Wanderley vd., 2011; Faraj ve Gebotys, 2021; Azriel vd., 2021). Bu tehditler, gömülü kod güvenliğini ve fikrî mülkiyeti tehlikeye atar. ROM kazıma, fiziksel erişim yoluyla ROM'dan doğrudan çıkarılan kodun çalınmasıdır. Bu, sıradan tümdevrelerde mümkün olmakla birlikte, özellikle SoC mimarisindeki MicroBlaze tasarımları içinde yer alan FPGA'lerde daha da zordur.

MicroBlaze IP'nin varsayılan gizlenmiş (obfuscated) yapısı ve Xilinx'teki bitstream şifreleme seçeneği, ROM'dan kod çıkarımını günümüz teknolojisiyle neredeyse 'imkânsız' olarak anılacak kadar zorlaştırır (Guin vd., 2014; Abideen, vd., 2024; Karam, vd., 2016; Stolz, vd., 2021). FPGA'nin Programmable Logic (PL) bölümüne gömülü bir soft-core işlemci olan MicroBlaze, kodunu uçucu konfigürasyon belleğinde tutar. Bu bellek doğası gereği doğrusal değildir ve FPGA entegresinin içinde yer aldığı (on-chip) için, harici bellek yongalarına (DDR RAM gibi) kıyasla fiziksel prob (physical probing) saldırılarına karşı doğal bir koruma seviyesi sunar. MicroBlaze için derlenen C kodu bu donanım konfigürasyonuna dönüştürülerek FPGA dokusunun ayrılmaz bir parçası olur. Bu nedenle ROM kazıma gibi fiziksel tekniklerle çıkarmak son derece zordur. Bu yapı, tehdit modelinde tanımlanan sorunlara karşı, gizlenmiş algoritmaların amaçladığı tersine mühendisliğin önlenmesi hedefini yerine getirir (Proulx vd., 2023; Wanderley vd., 2011). Bu çalışmanın Zynq-7020 kurulumunda, güvenlik açısından kritik C kodu parçaları yeniden yapılandırılabilir bölmeler içindeki AXI'ye eşlenmiş BRAM'lere yerleştirilmiş, DFX (PCAP) ile periyodik olarak yer değiştirilip temizlenmiştir; böylece sırların ve seçili kod sayfalarının fiziksel konumu da yalnızca MicroBlaze/konfigürasyon belleği gizlemesine güvenmek yerine hareketli hâle getirilmiştir (Şekil 1.1).

Yan kanal saldırıları, güç tüketimi veya elektromanyetik yayılımlar gibi fiziksel yan verileri analiz ederek hassas verileri elde etmeyi amaçlar. Bitstream şifrelemesi doğrudan bit akışının okunmasını engellese de, şifre çözme donanımının kendisi bu tür saldırılara karşı zafiyet gösterebilir ve ek karşı önlemler gerektirir. İletişim dinlemesi ise pinler arasındaki veri akışını inceleyerek kod işlevini anlamaya çalışır; bu da özellikle ortadaki adam (MITM) yöntemleriyle dolaylı kod çıkarımını mümkün kılabilir. Bu çalışmada varsayılan tehdit, bir

saldırganın tümdevre içinde yer alan ticari (IP sınıfı) yazılımı (i) ROM kazıma ile, (ii) yan kanal analiziyle anahtar elde ederek veya (iii) iletişim örüntülerini analiz edip kod davranışını çıkararak ele geçirebilmesidir. Kaynak kısıtlı sistemlerde bu tehditlere karşı, düşük enerji ve sınırlı gecikmeye öncelik vermek için hafif AEAD ilkeleri (ör. NIST tarafından standartlaştırılan ASCON-128 ailesi) ve mesaj-başı anahtar kullanım politikası benimsenmiştir (McGrew, 2008; Dobraunig, vd., 2021; Turan vd., 2023).



Şekil 1.1. Sabit konumlu bellek kullanımında QP tabanlı yan kanal saldırısı sırasında RAM'deki bitlerin okunması (Faraj ve Gebotys, 2021) (a. Kazıma, b. Bellek, c. Bitler)

Literatürdeki kapsamlı ve ileri çalışmalar genellikle yalnızca FPGA'lere odaklanır; ancak iletişim hattı dinlemesi yoluyla kod işlevselliğinin açığa çıkarılmasını doğrudan hedefleyen—ve bunu etkin biçimde önleyen—özel çözümler sınırlıdır (Silva, vd., 2023). İlgili çalışmalar örneğin şu beş başlık altında toplanabilir: (i) Bitstream koruması: Guin (2014) sahte tümdevre tehdidini ele almış ve bitstream şifrelemeyi önermiştir; (ii) Gizleme (obfuscation): Engels (2022) logic locking güvenliğini değerlendirmiş, ancak iletişim güvenliğini vurgulamamıştır; (iii) IoT güvenliği: Silva (2023) hafif şifreleme algoritmalarını incelemiş, fakat kod işlevselliği korumasını ele almamıştır; (iv) Algorithm hopping: Soliman (2019), Dinamik Kısmi Yeniden Yapılandırma (DPR; güncel karşılığı DFX) ile birlikte

algoritma değiştirme yaklaşımının güvenliği iyileştirdiğini göstermiştir; (v) Fiziksel katman anahtar üretimi: MobileKey gibi güncel sistemler mobil aygıtlarda kablosuz kanal karşılıklılığından simetrik anahtarlar türetir; bu çalışmalar, burada hedeflenen yongaiçi kod hırsızlığını veya iletişim-semantik sızmasını önlemekten ziyade, uçlar arasında anahtar başlatmaya odaklanır (Song vd., 2022; Xiao vd., 2024; Xia vd., 2024). Dolayısıyla, iletişim dinlemesiyle kod işlevselliğinin açığa çıkmasını önleyen özgül karşı önlemler seyrektir. Önerilen yöntem, bu boşluğu doldurmayı amaçlar. Çalışma, (i) cihaz kimliği ve bir zaman damgası/nonce ile bağlanan önceden paylaşılmış bir kök sır ve küçük bir şifreli havuzdan, Ascon-XOF tabanlı bir KDF ile mesaj-başı AEAD anahtarları türeten; (ii) şifreli havuzu ve seçili kod sayfalarını DFX ile BRAM içinde hareketli hedef olarak yer değiştiren, düşük enerjili bir yöntem önerir (Bertoni vd., 2007, Bertoni vd., 2011; National Institute of Standards and Technology [NIST], 2015). Değerlendirme Zynq-7020 donanımı (ve Arduino sınıfı bir çözücü) üzerinde byte başına çevrim, uçtan uca gecikme ve (12 V/9,6 V) gerilimler altında enerji ölçümleriyle yapılmıştır. Yöntemin bilinen AEAD standartlarına karşı dayanımını göstermek için saldırı benzetimleri de, ayrıca gerçekleştirilmiştir.

Sonuç olarak, mevcut çözümlerin çoğunun bitstream şifrelemeye veya genel kriptografik algoritmaların enerji/süre optimizasyonuna odaklandığı; buna karşın, iletişimden yakalanan verilerle kod davranışının çıkarımını engelleyecek mekanizmaların literatürde yetersiz olduğu görülmüştür. Ayrıca akademik FPGA güvenliği önerilerinin önemli bir kısmı kuramsal kalmış ve ticari gereksinimleri karşılayamamıştır (Vipin ve Fahmy, 2018; Stolz vd., 2021). Bu akademi-sanayi uyumsuzluğu, bu çalışmada önerilen yöntemin—enerji verimliliği ve pratik devreye alınabilirlik gibi ticari gereksinimlerin yanında IP hırsızlığını da hedef alarak—bu boşluğu doldurabileceği beklentisini doğurmuştur. Literatürde, iletişim dinlemesi veya ROM kazıma yoluyla kod işlevselliğinin çıkarılmasını doğrudan hedefleyen belirli bir tehdit modeli ya da karşı önlem tespit edilememiştir. Güncel çalışmalar genellikle enerji ve zaman optimizasyonunu önceledığinden bu özgül tehdidi göz ardı etmektedir. Bu doğrultuda, uygulamaya dönük uygunluk, güvenlik ve enerji verimliliği; kısıtlı aygıtlarda yaygın hafif AEAD temel şifreleme algoritmalarına (Ascon-128a, ACORN, TinyJAMBU, JAMBU) karşı aynı ölçüm koşullarında değerlendirilmiştir (Dobraunig vd., 2021; Wu, 2016; Wu ve Huang, 2019, 2016; Bogdanov vd., 2007).

İletişim dinlemesiyle kod işlevi çıkarımı ve ROM kazımayı birlikte hedefleyen doğrudan karşılaştırılabilir bir çalışma bulunmadığından, önerilen hibrit güvenlik çerçevesi, kısıtlı

aygıtlarda yaygın hafif AEAD temel şifreleme algoritmalarına (ör. Ascon-128a, ACORN, TinyJAMBU) karşı değerlendirildi. Karşılaştırma, aynı test koşullarında byte başına çevrim, uçtan uca gecikme ve 12 V ile 9,6 V enerji metriklerine odaklandı. HGÇ katmanı, seçilen AEAD'den bağımsız olarak mesaj-başı dinamik anahtarlama ve DFX tabanlı bellek yer değiştirme katkısı sağlar. Geçmiş PSK tabanlı akış şifreleri (ör. A5/1) yalnızca kimlik doğrulamalı şifrelemeyi motive etmek ve statik anahtarlı LFSR tasarımlarının bu tehdit modeli için neden uygun olmadığını vurgulamak amacıyla referans verilmiştir (Biryukov vd., 2001).

Bu çalışmada önerilen hibrit güvenlik çerçevesi, yukarıdaki tehditlere karşı bitstream şifreleme, dinamik anahtar üretimi ve Dynamic Function eXchange (DFX) ile bellek gizlemeyi birleştirir (Soliman vd., 2019; Vipin ve Fahmy, 2018; Abideen vd., 2024; Raghunath ve Aravind, 2023). Xilinx Vivado'daki bitstream şifreleme seçeneği ROM üzerindeki yan kanal saldırılarını etkisini azaltmayı ve bit akışının tersine mühendisliğini ve yetkisiz kopyalanmasını hedeflerken, SoC Block Design içindeki MicroBlaze soft işlemcisinin "Sanal ROM'u" zaten gizli olduğundan fiziksel kazımaya doğal direnç gösterir. Bu çalışmanın Zynq-7020 uygulamasında sabit ROM konumlarına bel bağlamaktan da kaçınılır. Normalde ROM'da bulunacak güvenlik kritik kod parçaları ve tablolar, PL içinde DFX ile yer değiştirilebilir iki BRAM bölgesine yerleştirilip; PCAP üzerinden periyodik olarak yer değiştirilip temizlenir. Böylece fiziksel konumları da sürekli değişir. İletişim dinlemesi tehdidine karşı, çalışmada cihaz kimliği ve isteğe bağlı bir zaman damgası/nonce ile genişletilmiş önceden paylaşılmış bir kök sır ve bir şifreli havuzdan Ascon-XOF KDF ile mesaj-başı anahtar türetimi önerilir. Biçimsel olarak A5/1 gibi önceden paylaşılan anahtarlara dayanan şemalara benzese de—ki A5/1 bildirilen zafiyetlerine rağmen GSM ağlarında hâlen kullanımdadır—önerilen yöntem daha yüksek güvenlik sağlar (Biryukov vd., 2001).

A5/1 benzeri bir yapı (akış şifresi) burada seçilmemiştir. Çünkü bilinen kriptoanalitik zayıflıkları, 64-bit anahtar uzunluğu ve gizlenmemiş yapısı nedeniyle tek bir kök anahtarın RAM'den doğrudan elde edilme olasılığı yüksektir (Biryukov vd., 2001; Faraj ve Gebotys, 2021). Buna karşılık önerilen yöntem, (havuz + kök sır + cihaz kimliği + [isteğe bağlı zaman damgası/nonce]) gibi çoklu girdiden faydalanarak her mesaj için kriptografik olarak taze ve bağımsız bir 128-bit AEAD anahtarı türetir. Bu yaklaşım, kök anahtarın entropisini her bir mesaja özgü anahtarlar üretmek için kullanarak, tek bir anahtarın sürekli kullanımından

doğacak riskleri ortadan kaldırır. Böylece, olası bir mesaj-başı anahtar sızıntısının etkisi yalnızca o mesaja izole edilir ve sistem genelinde bir zafiyete yol açması engellenir. Bu bağlamda yöntem, literatürdeki bir boşluğu dolduran özgün bir tehdit modeli ve özel bir çözüm sunar.

Katkı donanım-yazılım hibriti olduğundan (çalışma zamanında iletişim semantiğini gizler ve DFX ile PL-BRAM'de hem sırları hem de seçili kod segmentlerini yer değiştirir), uygun karşılaştırma için temel algoritmalar, gömülü hedeflerde yaygın hafif AEAD temel şifreleme algoritmalarıdır. Asimetrik anahtar değişim protokolleri değildir. Buna göre, Hibrit Güvenlik Çerçevesi (HGÇ) standart AEAD'lerin (varsayılan Ascon-128a; alternatif olarak ACORN, JAMBU, TinyJAMBU) üzerine mesaj-başı anahtarlama katmanı olarak değerlendirilmiş; zaman/enerji aynı kurulumda raporlanmıştır. Eski PSK akış şifrelerine (ör. A5/1) yalnızca dinamik mesaj-başı anahtar seçiminin gerekçelendirilmesi (tek kök anahtarlı şemalara karşı) için atıf yapılmıştır. Bu çerçeve, yeniliği (dinamik anahtar + sırlar ve kod için DFX kullanarak taşınan BRAM) yalıtır ve değerlendirmeyi tehdit modeliyle hizalar. Önerilen yöntem mesaj-başı 128-bit AEAD anahtarları (Ascon ailesi) kullanır ve BRAM içeriğini DFX kullanarak yer değiştirir. DFX kriptografik anahtar entropisini artırmasa da, saldırganın fiziksel arama uzayını genişletir.

Önerilen hibrit yöntemin önemli alt bileşenlerinden biri DFX'tir. RAM bloklarını ve adreslerini sürekli yer değiştirerek secret pool'un (sır havuzu) ve seçili kod sayfalarının fiziksel konumunu gizler. Bu süreçte, gizli verileri içeren BRAM bloklarının FPGA üzerindeki fiziksel konumları sürekli olarak değiştirilir. Bu 'hareketli hedef' savunması, saldırganın hassas verinin bulunduğu fiziksel koordinatları tespit etmesini zorlaştırır. Ayrıca, pasif hale gelen bellek bloklarının içeriği, kalıntı veri bırakmamak için güvenli bir şekilde silinir (zeroized) ve boş alanlara anlamsız veri yerleştirilir. Böylece bir saldırganın gerçek veriyi yemlerden ayırması zorlaşır. Bu nedenle yöntem yalnızca entropiyi artırmakla kalmaz, iletişim örüntülerinin kod davranışıyla ilişkilendirilmesini de zorlaştırarak kod hırsızlığını engellemeyi amaçlar (Stolz vd., 2021). Pratikte, sır malzemeyle birlikte BRAM'de tutulan kod sayfalarının yer değiştirilmesi, kararlı sızıntı şablonlarını daha da bozar ve fiziksel kazıma/probla müdahaleyi zorlaştırır.

Yöntem tasarımında, önceden paylaşılan anahtarların (PSK) enerji verimliliği avantajları dikkate alınmıştır. Bu, kaynak kısıtlı gömülü sistemler ve benzeri IoT aygıtları için kritiktir.

Genellikle anahtar değişimine dayanan asimetrik güvenlik protokolleri (ör. Diffie–Hellman), yüksek hesaplama maliyeti nedeniyle enerji açısından kritik gömülü sistemlere uygun değildir (Abu Al-Haija vd., 2017; Silva vd., 2023).

Yöntem, gerçek zamanlı anahtar değişimi gerektirmeden dinamik anahtarlar üreterek yukarıda tanımlanan soruna alternatif bir çözüm sunar. Bu süreçte enerji tüketimi ve gecikmenin azaltılması da gözetilmiştir. Tasarım, her iletişim oturumunda şifreli havuzdan rastgele bir öğe seçilmesi ve bunun kök anahtar ile cihaz kimliğiyle birleştirilmesinden hareketle Ascon-XOF KDF ile mesaj-başı AEAD anahtarı türetilmesine dayanır.

Bu çalışmada varsayılan tehdit, saldırganın yongadışı veri yollarını (DDR/QSPI/AXI) probla izlemesi, yonga içi dizileri fiziksel olarak okumaya çalışması ve G/Ç'yi pasif gözlemleyerek program davranışını çıkarmasıdır. Buna karşılık, AXI üzerinden yalnızca şifreli secret pool taşınır ve sırları tutan BRAM örnekleri, fiziksel konumun mantıksal içerikten ayrıştırılması için DFX ile periyodik olarak yok edilip yeniden oluşturulur. Secret pool ile birlikte BRAM'de tutulan kod sayfalarının da yer değiştirilmesi, fiziksel konum ile mantıksal içeriğin eşzamanlılığını daha da bozar ve kazıma/prob şablonlarını başarısızlığa uğratır.

Önerilen yöntem, enerji ve süre açısından maliyeti düşük kalırken, tehdit kapsamında tanımlanan donanımsal saldırılara direnç gösterecek şekilde bilinçli olarak karmaşıklaştırılmıştır. Böylece donanım düzeyinde gizlemeye katkı sağlar (Abideen vd., 2024; Azriel vd., 2021; Karam vd., 2016). Önerilen yöntemin güvenlik felsefesi, iki temel katmana dayanmaktadır. Kriptografik katman, Kerckhoffs ilkesine tam uyumla, güvenliği tamamen açık ve standartlaşmış algoritmaların (ASCON-128a) ve gizli anahtarların yönetimine dayandırır. Buna ek olarak, fiziksel katmanda ise derinlemesine savunma (defense-in-depth) prensibi benimsenmiştir. Bu bağlamda DFX ile bellek konumlarının sürekli değiştirilmesi gibi gizleme (obfuscation) teknikleri, kriptografik güvenliğin yerine geçmeyi değil, ROM kazıma veya yan kanal analizi gibi fiziksel ve donanım tabanlı saldırıları gerçekleştirmek için gereken saldırgan iş yükünü (attacker's work factor) artırmayı hedefler. Böylece, sistemin güvenliği yalnızca kriptografik anahtarların gizliliğine bağlı kalırken, fiziksel müdahalelere karşı yüksek ölçüde birden fazla ek caydırıcılık katmanı oluşturulmuştur. Böylece bu yaklaşım, ayrıca, Shannon'ın (Shannon, 1949) "yeterli zaman verildiğinde her sistem kırılabilir" düsturunu gözeterek, gizlenmiş yapının tersine

mühendisliğini zorlaştırıp saldırganın zaman ve kaynak harcamasını artırmasıyla olası bir saldırının teorik olarak mümkün olsa bile pratik olarak gerçekleştirilemez kılınması hedefine hizmet eder. Bu bağlamda DFX, RAM adreslerini sürekli değiştirip anlamsız veriler ekleyerek istatistiksel analizlere karşı koruma sağlar (Stolz vd., 2021). Dolayısıyla hafif olmakla birlikte amaç-odaklı bu hibrit yapı—birden çok açıdan değerlendirilmiş ve güvenlikten ödün vermeden—kod davranışının dolaylı çıkarımını engelleyecek ölçüde yeterli karmaşıklık sunar; zira bir anahtar kırılsa bile yalnızca tek bir mesaj ele geçirilebilir.

İsteğe bağlı bir hazırlık (provisioning) evresi, bitstream şifreleme altında bir kök sır ve parametrelerin önceden kurulmasına imkân tanır. Böylece uçlar, Ascon-XOF ile TRNG-tohumlu seed/salt'tan aynı şifreli havuzu statik anahtar aktarmadan deterministik olarak türetebilir (Krawczyk ve Eronen, 2010; Chen., 2009; Barker vd., 2011; Wu ve Huang., 2019; Turan vd., 2023). Zaman damgası denemeli bir varyant kullanılabilir; ancak bu seçenek, değerlendirilen tehdit modelinden bağımsızdır ve gelecekteki çalışmalara bırakılmıştır. Bu hazırlık seçeneği, kısıtlı aygıtlarda çevrimiçi anahtar değişiminden kaçınarak düşük enerji ve gecikmeyi korur. Önemli olarak DFX, kazıma/probla müdahaleye karşı fiziksel arama uzayını büyütür; kriptografik kabuk entropisi 128 bit olarak kalır.

Bu konu tehdit modelinin dışında olsa da çalışma, "anahtar gömme" yaklaşımından ve literatürde belirtilen diğer şifreleme boşluklarından doğan zayıflıkları ele alacak gelecekteki işlere bir altyapı sunmayı amaçlayan isteğe bağlı bir mekanizma da önerir. Buna göre, hazırlık evresi tartışılmış ancak ana tehdit modeline yönelik çözümün resmî kapsamı dışında bırakılmıştır. Yine de çalışma, bir ön çözüm yolu sunarak bu ayrı konuyu gelecek araştırmalar için işaretlemektedir. Bu faz için kuramsal veya deneysel karşılaştırmalar ele alınmamıştır.

Sonuç olarak, bu çalışma FPGA tabanlı sistemlerde kod hırsızlığına karşı enerji-etkin ve pratik bir çözüm sunarak mevcut literatürde önemli bir boşluğu doldurur (Vipin ve Fahmy, 2018; Silva vd., 2023; Stolz vd., 2021). Buna göre: (i) iletişim hattı dinlemesi ve ROM kazıma yoluyla kod işlevselliğinin çıkarımına özgü bir tehdit modeli tanımlanmış ve buna karşı hibrit bir güvenlik çözümü sunulmuştur; (ii) A5/1 gibi örnekler üzerinden, önceden paylaşılan anahtarların IoT ortamlarında enerji verimliliği ve pratik uygulanabilirliği gerekçelendirilmiştir (Biryukov vd., 2001); (iii) Kerckhoffs ilkesi ışığında, algoritma bilinse bile DFX tabanlı gizleme tekniklerinin saldırı karmaşıklığını ve süresini artırarak gerçek

ortam güvenliğine katkısı ayrıntılı biçimde tartışılmıştır (Abideen vd., 2024; Stolz vd., 2021; Vipin ve Fahmy, 2018) ve (iv) önerilen yöntemin, aynı donanım üzerinde temsilî hafif AEAD temel şifreleme algoritmalarıyla karşılaştırıldığında getirdiği ek enerji ve gecikme maliyetleri de ölçülerek, bu maliyetlerin pratik bir çözüm için kabul edilebilir seviyede olduğu gösterilmiştir (Kerckhoffs, 1883).

Not: Bu çalışma, iki somut varyantı gerçekleştirmiştir: (i) doğrulama hedefi olan 2-BRAM (2 RM) tek-atım politika ve (ii) DFX davranışını nicemlemek için kullanılan 4-BRAM (4 RM) stres konfigürasyonu. Aksi belirtilmedikçe DFX'e ilişkin süre/enerji ölçümleri 4-BRAM üzerinde alınmış, 2-BRAM değerleri 4-BRAM tek-takas penceresinin doğrusal ölçeklenmesiyle verilmiştir. Erken sürümlerde yalnızca secret-pool BRAM'de taşınırken, bu sürümde seçili kod sayfaları da BRAM üzerinde DFX ile yer değiştirilmiştir. Tekrar üretilebilirliği ve gelecekteki araştırmaları kolaylaştırmak amacıyla, önerilen yöntem ile bilinen AEAD temel şifreleme algoritmalarının açık kaynak karşılaştırmasına, ilgili bağlantıda erişim verilmiştir (URL1).

2. MEVCUT DURUM ÇALIŞMALARI

Alan-programlanabilir kapı dizileri (FPGA'ler), tüketici elektroniğinden kritik altyapılara kadar uzanan gömülü bilgi-işlem sistemlerinde vazgeçilmez hâle gelmiştir. Yeniden yapılandırılabilir doğaları hızlı geliştirme ve devreye almayı mümkün kılarken; fikrî mülkiyet hırsızlığı, yan kanal analizi, hata enjeksiyonu, uzaktan Truva atı yerleştirme ve protokol düzeyi saldırılar gibi geniş bir tehdit yelpazesine de tasarımları açık bırakır. Geleneksel karşı önlemler bitstream şifreleme veya statik mantık kilitlemeye odaklanır; ancak son on yılda, gizliliğin tek başına yeterli olmadığını gösteren saldırılar çoğalmıştır. Aynı zamanda Nesnelerin İnterneti (IoT) düğümleri gibi kaynak kısıtlı aygıtlar, araştırmacıları hafif kriptografi ve enerji-farkındalıklı güvenlik modülleri geliştirmeye yöneltmiştir. Bu bölüm, enerji verimliliği ve pratik devreye alma vurgusuyla, kod koruma ve dinamik kısmi yeniden yapılandırma üzerine güncel çalışmaları kapsamlı biçimde özetler.

2.1. Bitstream Saldırıları ve Kod Koruma

Tedarikçi tarafından sağlanan AES tabanlı bitstream şifrelemeye rağmen, konfigürasyon verisinin gizliliğinin ihlâl edilebileceği yapılan çalışmalarda gözlemlenmiştir. 2011'den başlayarak Moradi ve diğerleri, Xilinx Virtex ve Altera Stratix FPGA'lerde yerleşik bitstream çözme biriminin yan kanaldan anahtar sızdırdığını ve AES algoritmasını kırmadan tam anahtar kurtarmayı mümkün kıldığını göstermiştir (Moradi, vd., 2011; Moradi, vd., 2013). Daha sonraki çalışmalar bu saldırıyı daha yeni cihazlara genellemiş, hatta şifreli bitstream'lerden anahtar çıkarmak için hata-enjeksiyon tekniklerini ortaya koymuştur (Swierczynski, vd., 2016). Bu bulgular, standart bitstream şifrelemenin tek başına sınırlı koruma sağladığı farkındalığını doğurmuştur. Anju Johnson ve diğerlerinden gelen uyarı niteliğindeki çalışma, Ethernet üzerinden uzaktan dinamik kısmi yeniden yapılandırmayı (DPR) kullanarak çalışan bir FPGA tasarımına donanım Truva atı yerleştirmiş; bu yolla AES anahtarlarının sızdırılması ve gömülü işlemcilerin kurcalanması mümkün olmuştur (Johnson vd., 2017). Çalışma, korunmayan dinamik yeniden yapılandırmanın bir saldırı vektörüne dönüşebileceğini vurgulamıştır.

Bu tehditleri bertaraf etmek için araştırmacılar, bitstream'leri belirli cihazlara bağlamayı ve devre düzeyinde mantığı gizlemeyi önermiştir. Maes ve diğerleri, çözme anahtarının kalıcı

bellekte saklanmak yerine silikon süreç varyasyonlarından türetildiği fiziksel olarak klonlanamaz fonksiyon (PUF) temelli anahtar üretimini tanıtmıştır (Maes, 2013). Van Herrewege ve diğerleri, bu PUF tabanlı şemaların Xilinx 7-serisi FPGA'lerde güvenilirliğini artırmıştır (Van Herrewege, 2015). 'Obfuscation' da ikinci bir savunma hattıdır. Karam ve diğerleri, doğruluk tablolarını FPGA'nin "karanlık silikon" bölgelerine haritalayarak, yalnızca çalışma zamanında doğru anahtar uygulandığında fonksiyonel kalan anahtar-bağımlı bir bitstream önermiştir (Karam, vd., 2016). Engels ve Shafique mantık kilitlemeyi analiz etmiş ve pek çok şemanın SAT ya da yapısal analizlere yenik düştüğünü göstermiştir (Engels, vd., 2022; Shafique, vd., 2024). Bu da daha sağlam mekanizmaların gerekliliğini ortaya koymuştur. Yakın zamanda Stolz ve diğerleri donanım-yazılım obfuscation'ını bütünleştirmiş; LifeLine çerçevesinde kriptografik algoritmayı FPGA dokusu ile gömülü işlemci arasında bölmüş ve çalışma zamanında DPR ile birleştirerek statik analizi boşa çıkarmıştır (Stolz, vd., 2021).

Buna paralel olarak akademide otomatik Truva atı ekleme ve bitstream kurcalama deneyleri yapılmıştır. Ender ve diğerleri, Xilinx 7-serisi bitstream şifrelemesini "Starbleed" adıyla ilk kez tamamen kırmış; belgelendirilmemiş konfigürasyon komutlarını kötüye kullanarak hem anahtarın hem verinin geri alınabildiğini göstermiştir (Ender, vd., 2020). Kataria ve diğerleri, Cisco Trust Anchor modülünün bitstream'inde birkaç biti çevirmenin bütünlük denetimini devre dışı bıraktığını ve keyfî değişikliklere izin verdiğini göstermiştir (Kataria, vd., 2019). Artık keyfî bitstream'lere donanım Truva atı ekleyebilen otomatik araç zincirleri mevcuttur (Ender vd., 2019). Bu da obfuscation ve çalışma-zamanı denetimlerinin gerekliliğini daha da somutlaştırır.

2.2. Dinamik Yeniden Yapılandırma ve Hafif Kriptografi

Kısmi yeniden yapılandırma, çalışma zamanında donanım konfigürasyonunu değiştirerek güvenliği iyileştirmek için kullanılmıştır. Soliman ve diğerleri, DPR kullanarak beş kimlik doğrulamalı şifre (AEAD) arasında rastgele geçiş yapan "algorithm hopping" güvenlik modülünü tanıtmıştır (Soliman vd., 2019). Tek bir şifre çekirdeğini çoklayarak beşini birden enstantiye etmek yerine kullanmaları, statik güç tüketimini yaklaşık %80 azaltmış ve tekil, yekpare bir gerçekleştirmeye göre LUT kullanımını %58'e düşürmüştür. Samir ve diğerleri buna tamamlayıcı bir yaklaşımla, eldeki pil gücüne göre farklı karmaşıklıkta şifreleri

yüklemiş; enerji-uyarlamalı HSM'leri yaklaşık %60 LUT tasarrufu sağlamış, dinamik gücü 10 mW ile sınırlandırırken yine de kimlik doğrulamalı şifreleme sunmuştur (Samir vd., 2019). Wei ve diğerleri DPR'yi PUF'lara uygulamış; modelleme saldırılarına karşı entropiyi ve dayanımı artırmak için üç farklı halka-osilatör PUF yerleşimi arasında çalışma sırasında geçiş yapmıştır (Wei, vd., 2020). Sunkavilli ve diğerleri, küçük devre parçalarını periyodik yeniden yapılandırarak donanım Truva atlarını zorlaştıran DPReDO adlı dinamik obfuscation şemasını önermiş; %3'ün altında alan ek yüküyle Truva atı başarı oranında %80'e varan düşüş bildirmiştir (Sunkavilli, vd., 2022).

Yeniden yapılandırmanın ötesinde, FPGA'ler için hafif kriptografi üzerine geniş bir çalışma gövdesi vardır. Sasdrich ve diğerleri, çalışma sırasında anlık olarak LUT içeriklerini değiştiren ve yan kanal saldırılarına direnç için "hareketli hedef" yaklaşımını kullanan PRESENT blok şifresini gerçekleştirmiştir (Sasdrich vd., 2015). Faraj ve Gebotys, SRAM'den sırları çıkarmak için fotonik yayılımları kullanmış ve kriptografik FPGA'lerde fotonik kalkanlamanın gereğini vurgulamıştır (Faraj ve Gebotys, 2021). Bommana ve diğerleri, DPR'yi derin öğrenme tabanlı yan kanal karşı önlemleriyle birleştirerek uyarlamalı koruma sağlamıştır (Bommana, vd., 2025). Son yıllarda Lorentz-kaos şifreleyiciler (Janakiraman vd., 2025), EEG güdümlü sahte-rastgele üreteçler (Kharidu ve Sudha, 2025), kuantum-kaos hibritleri (Shafique vd., 2024), çok-boyutlu kaos haritaları (Sivaranjani ve Amirtharajan, 2025) ve dinamik S-kutuları (Maazouz vd., 2022) gibi kaotik ve biyometrik şemalar FPGA'lerde incelenmiştir. Ciylan ve diğerleri, Virtex-7 üzerinde bir sistolik-dizi kaos konvolüsyon şifreleyici sunmuş; %55 LUT kullanımı ve 280 mW güç tüketimi ölçmüştür (Ciylan vd., 2023). Bu çeşitlilik, kaynak kısıtlı FPGA'lere uygun, hafif ama güvenli kriptografi arayışını ortaya koyar. Tamlık adına, fiziksel katman anahtar üretim sistemleri (ör. MobileKey ve güncel derlemeler) uçlar arasında anahtar başlatmaya odaklanır ve yonga-içi kod hırsızlığı ile iletişim semantiği sızmasını önleme hedefimizden bağımsızdır (Song vd., 2022; Xiao vd., 2024; Xia vd., 2024).

Yukarıdaki ilerlemelere karşın mevcut çalışmaların çoğu, tümleşik bir sistem bağlamında ya enerji verimliliğine ya da bitstream gizliliğine değinir; ikisini birden nadiren ele alır. A5/1 gibi eski akış şifreleri, 64-bit ön-paylaşımlı anahtarlar ve doğrusal geri beslemeli kaydırma yazmaçları (LFSR) kullandıkları için modern tehdit modellerinde zaman-bellek değiş tokuşu ve korelasyon saldırılarıyla güvensiz kabul edilmektedir (Ekdahl ve Johansson, 2003; Biryukov vd., 2001; Güneysu vd., 2008). Önerilen Hibrit Güvenlik Çerçevesi ise, kimlik

doğrulamalı şifrelemeyi (hafif AEAD) mesaj-başı anahtar türetimiyle birleştirir: Ascon-XOF KDF, şifreli bir havuzdan (ön-paylaşımlı kök sır, cihaz kimliği ve isteğe bağlı zaman damgası/nonce ile bağlı) her mesaj için anahtar üretir (Dobraunig, vd., 2021). Buna ek olarak DFX ile bellek obfuscation'ı kullanılarak derinlemesine savunma sağlanır.

Program işlevselliğinin dinlenen G/Ç üzerinden çıkarımıyla birlikte yonga-içi içeriğin fiziksel okunmasına aynı anda ve gömülü enerji/gecikme kısıtları altında direnç gösteren bir yaklaşıma literatürde açıkça rastlanmamıştır. Bu boşluk, hafif bir AEAD'i mesaj-başı anahtarlarla eşlemeyi ve güvenlik-kritik BRAM içeriğinin DFX ile çalışma zamanında yer değiştirilmesini eşleştirmeye motivasyon sağlar. Bu, ek anahtar entropisi iddiası gerektirmeden fiziksel arama uzayını büyütür. Aşağıdaki çizelgeler, önerilen yaklaşımın enerji tüketimini temsilî önceki çalışmalarla karşılaştırır, literatürdeki boşluğu özetler ve temel ödünleşimleri vurgular. Çizelge 2.1, son yıllardaki temsilî çalışmaları ve HGÇ yaklaşımını motive eden başlıca sınırlamaları özetler.

Çizelge 2.1. Literatürdeki temsilî FPGA güvenlik çalışmaları, sınırlamaları ve tespit edilen boşluklar

Yazar–Yıl	Yöntem	Platform	Uygulama	Sınırlamalar / Boşluklar
Biryukov vd., 2001	Gerçek zamanlı TMTO saldırısı (A5/1)	PC	GSM kriptoanalizi	Büyük ön hesaplama gerektirir; saldırı odaklıdır.
Ekdahl ve Johansson, 2003	Korelasyon saldırısı (A5/1)	PC	GSM kriptoanalizi	Uzun keystream gerektirir; IP koruması değildir.
Güneysu vd., 2008	COPACOBANA donanım saldırısı	FPGA kümesi	A5/1/DES kırma	Adanmış donanım; sadece saldırı, savunma değil.
Moradi vd., 2011	Yan kanal saldırıları	Virtex/Stratix	Üretici bitstream şifrelemesi	Gösterim düzeyi; karşı önlem sunmaz.
Maes, 2013	PUF-tabanlı anahtar bağlama	Virtex-6	Cihaz anahtar bağlama	Şifre çözümünden sonra zayıf (post-decryption).
Chakraborty vd., 2013	Bitstream'e Trojan ekleme	Virtex-5	Kavramsal gösterim	Tümleşik savunma yok.
Van Herrewege, 2015	PUF anahtar bağlama	Artix-7	Kopya-önleme	Yan kanala karşı hassas.
Sasdrich vd., 2015	Moving-target PRESENT	FPGA	SCA azaltma	Şifreye özgü; genel IP koruması değil.
Swierczynski vd., 2016	Şifreli bitstream'lere hata enjeksiyonu	Spartan-6	Anahtar sızıntısı	Fiziksel erişim gerek; aktif saldırı.
Karam vd., 2016	"Karanlık silikon" tabanlı obfuscation	Cyclone IV	IP koruması	Statik obfuscation; dinamik güncelleme yok.
Johnson vd., 2017	Uzaktan DPR saldırısı	Spartan-6	Trojan yerleştirme	Saldırı odaklı; savunma yok.

Çizelge 2.1. (devam) Literatürdeki temsilî FPGA güvenlik çalışmaları, sınırlamaları ve tespit edilen boşluklar

Adetomi vd., 2017	Şifreli kısmi bitstream yeniden konumlama (ATAL)	Virtex-5	Modül yer değiştirme	Uzak güncelleme/SCA koruması yok.
Kataria vd., 2019	Bitstream manipülasyonu	Virtex	Kimlik doğrulama atlatma	Dahili savunma yok; bit çevirme ile kırılabilir.
Ender vd., 2019	Otomatik Trojan ekleme	Artix-7	Trojan yerleştirme	Araç var; statik obfuscation kırılabilir.
Engels vd., 2022	Mantık kilitleme eleştirisi	—	Güvenlik incelemesi	SAT saldırılarına açık; DPR kullanılmıyor.
Soliman vd., 2019	Algorithm hopping (DPR ile)	Zynq-7020	IoT güvenliği	Bitstream koruması/doğrulaması yok.
Samir vd., 2019	Enerji-uyarlamalı HSM (DPR)	Zynq-7000	IoT güvenliği	Bitstream koruması yok; geçiş gecikmesi riski.
Wei vd., 2020	Dinamik CRO-PUF	Kintex-7	Yetkilendirme	Fonksiyonel mantık koruması yok.
Ender vd., 2020	Starbleed saldırısı	7-Serisi Xilinx	Bitstream kurtarma	Protokol açığı; yamalanmamış cihazlar savunmasız.
Moraitis ve Dubrova, 2020	LUT değiştirme saldırısı (SNOW-3G)	Spartan-6	Anahtar çıkarımı	LUT manipülasyonu; ek obfuscation gerekir.
Albartus vd., 2020	DANA netlist analiz çerçevesi	—	Tersine mühendislik	Analiz aracı; koruma sağlamaz.
Stolz vd., 2021	LifeLine HW/SW birlikte obfuscation	Zynq-7020	IP koruması	Karmaşıklık; şifreleme yok.
Faraj ve Gebotys, 2021	Fotonik yan kanal	SRAM-tabanlı FPGA	SRAM okuma	Saldırı çalışması; fotonik kalkan gerekir.
Nabeel vd., 2021	Hafif hash modülü	—	IoT güvenliği	FPGA kaynak döküm yok; bitstream koruması yok.
Sunkavilli vd., 2022	DPReDO obfuscation	UltraScale+	Trojan savunması	Prototip; güvenli DPR gerekir.
Maazouz vd., 2022	4D kaos tabanlı dinamik S-box	FPGA	Genel şifreleme	≈%35 LUT; kimlik doğrulama katmanı yok.
Yao vd., 2022	Yeniden yapılandırılabilir PUF mimarisi	Kintex-7	Kimlik doğrulama	Cihaz kimlik doğrulaması; kod koruması değil.
Ciylan vd., 2023	Sistolik dizi kaos konvolüsyonu	Virtex-7	Görüntü şifreleme	%55 LUT, ~280 mW; bütünlük/obfuscation yok.
Proulx vd., 2023	DPR ile FW/IP koruma	FPGA SoC	IP koruması	Çift çekirdek+RP; kripto/SCA koruması eksik.
Shafique vd., 2024	Kuantum–kaos hibrit	Simülasyon	Tıbbi görüntüleme	Karmaşık; donanım yok; simülasyon düzeyi.

Çizelge 2.1. (devam) Literatürdeki temsilî FPGA güvenlik çalışmaları, sınırlamaları ve tespit edilen boşluklar

Bommana vd., 2025	DL-tabanlı SCA savunması	FPGA	SCA azaltma	Enerji ek yükü belirsiz.
Janakiraman vd., 2025	Lorentz-kaos şifreleyici	Pynq-Z1	Görüntü şifreleme	Yüksek kaynak kullanımı; bütünlük yok.
Kharidu ve Sudha, 2025	EEG-tabanlı PRNG şifreleyici	Özel FPGA	Biyometrik IoT	EEG girdisi gerekir; kimlik doğrulama/bütünlük yok.
Sivaranjani ve Amirtharajan, 2025	2D-MTMHM kaos haritası	MATLAB	Görüntü şifreleme	Sadece yazılım; FPGA yok; yalnızca gizlilik.
Kıyak, 2025 (Bu çalışma)	Hibrit Güvenlik Çerçevesi	Zynq-7020	Kod hırsızlığına karşı koruma	Sözlük saklama gerekir; DFX karmaşıklığı.

3. KAVRAMSAL ÇERÇEVE

FPGA, ASIC ve SoC mimarilerinin esnekliği, gömülü sistemlerde yüksek performans ve özelleştirme avantajları sunarken, özellikle IoT cihazlarının yaygın ve çok katmanlı saldırı riskleriyle karşılaşması, güvenlik kavramlarının daha derin bir altyapıyla ele alınmasını gerektirmektedir. Bu bölümde, literatürde sıkça yer alan temel güvenlik kavramları ve kavramlarla bağlantılı teknolojiler detaylı biçimde incelenmektedir. İlk alt başlıkta (3.1) IoT ve gömülü sistemlerde güvenlik ele alınacak; IoT kavramı, sınırlı işlem ve bellek kaynakları, yaygın saldırı tipleri ve gömülü yapıya özgü temel kavramlar üzerinde durulacaktır. Ardından (3.2, 3.3, 3.4, 3.5 vb. alt kısımlarda) temel kriptografik protokoller, donanım güvenlik prensipleri, PUF yapıları, saldırı yöntemleri gibi konular birbirleriyle ilişkili biçimde irdelenecektir.

Literatürde, IoT cihazlarının tipik güvenlik gereksinimleri ile sınırlı kaynaklar arasındaki çelişkiyi gidermek amacıyla çeşitli yöntemler önerildiği, FPGA'nin programlanabilir yapısının da gömülü sistem güvenliğinde sıklıkla değerlendirildiği görülmektedir (Johnson vd., 2017; Samir vd., 2019; Karam vd., 2016). Aşağıdaki kısımlarda, bu yöntemlerin kavramsal temellerine zemin oluşturacak tanımlamalar ve ilkeler sunulmaktadır.

3.1. IoT ve Gömülü Sistemlerde Güvenlik Kavramları

IoT (Nesnelerin İnterneti), farklı ölçekte cihazların internet üzerinden bağlantı kurabildiği ve sensör, aktüatör gibi bileşenlerle veri alışverişinde bulunduğu bir ağ ekosistemi şeklinde tanımlanır. Bu yapı, akıllı şehirlerden ev otomasyonuna, sağlık hizmetlerinden endüstriyel kontrol sistemlerine uzanan geniş kullanım alanlarına olanak tanır. Bununla birlikte, IoT uygulamalarında yer alan gömülü sistemler, çoğunlukla kısıtlı işlemci gücü, bellek kapasitesi ve enerji kaynağıyla çalışır (Stallings, 2016). Böyle bir kısıtlama, yüksek güvenlik katmanlarının uygulanmasını zorlaştırmaktadır.

Sınırlı kaynaklar ve yaygın saldırı tipleri

Literatürde, sınırlı kaynakları sebebiyle asimetrik kriptografinin (örneğin Diffie-Hellman veya RSA) genellikle uzun hesaplama süreleri ve yüksek güç tüketimine sebep olduğuna işaret edilmektedir (Diffie ve Hellman, 1976; Rivest vd., 1978). Aynı şekilde, ECC gibi

küçük anahtar boyutları sunabilen yöntemlerde dahi matematiksel karmaşıklığın IoT'de ek işlem yükü oluşturduğu belirtilir (Miller, 1985; Soliman, 2019). Dolayısıyla IoT ortamının kaynak kısıtları, araştırmacıları hafif kriptografik çözümlere yöneltmiştir. Geçmişte bazı uygulamalarda, hesaplama maliyeti düşük olduğu için MD5 veya SHA-1 gibi hash fonksiyonları kullanılmış olsa da, bu algoritmaların günümüzde bilinen ciddi güvenlik zafiyetleri (özellikle çarpışma atakları) bulunmaktadır. Bu nedenle, modern ve güvenli IoT tasarımlarında bu fonksiyonların kullanımı kesinlikle önerilmemektedir. Bunun yerine, NIST'in hafif kriptografi standardizasyon süreciyle belirlenen ASCON gibi veya SHA-3 ailesinin daha hafif varyantları gibi güncel ve güvenliği kanıtlanmış algoritmalar tercih edilmelidir. (Johnson vd., 2017). Buna ilaveten, brute force (kaba kuvvet) gibi saldırılara karşı da her mesajda farklı anahtar kullanımı büyük önem taşır.

Başka bir boyut da, donanım truva atı (HTH) veya yan kanal saldırıları gibi spesifik donanım odaklı tehditlerdir. FPGA gibi yeniden programlanabilir devrelerde, saldırgan uzaktan dinamik kısmi yeniden yapılandırma (RDPR) imkanını kötüye kullanarak sisteme zararlı modül ekleyebilir veya saat sinyalini manipüle edebilir (Karam vd., 2016). Bu noktalarda bit akışının (bitstream) şifrelenmesi, FPGA içi test edilebilirlik mekanizmalarının (DFT) devreye alınması veya bellek konumlarının periyodik olarak değiştirilmesi gibi yöntemler, IoT'nin düşük işlem gücü kısıtlarını da gözeterek ek koruma katmanları sunabilmektedir.

Gömülü sistemlerin güvenliğine dair temel kavramlar

Bir gömülü sistem, sensörlerden veya ağdan gelen veriyi alan, önceden tanımlanmış işlevleri yerine getiren ve belirli bir çıkış (kontrol sinyali veya iletişim cevabı) üreten elektronik yapıdır. Bu sistemlerin çoğunda güvenlik iki ana noktadan incelenir: (1) verinin gizliliği ve bütünlüğü, (2) sistemin yan kanal saldırıları ve fiziksel manipülasyonlara karşı dayanıklılığı. Geleneksel yazılım tabanlı önlemler (örneğin, yalnızca AES ile şifreleme) gömülü sistem düzeyinde yetmeyebilmektedir. Çünkü FPGA veya ASIC tabanlı tasarımda ek lojik bloklar, saat yönetimi veya bit akış erişimi gibi fiziksel katmanlara saldırı mümkündür. Donanım-yazılım birlikte gizleme (co-obfuscation) metodolojileri, bu sebeple gömülü sistemlerin korumasında sıkça atıf yapılır (Samir vd., 2019). İlaveten, IoT ekosisteminde çok farklı üreticilerin devre tasarımları bir araya geldiğinden, tedarik zinciri güvenliği ve tasarımın test edilebilirliği (DFT) de kavramsal olarak önemlidir (Johnson vd., 2017).

Bu bağlamda, FPGA gibi yeniden programlanabilir donanımların gömülü sistem yapısına entegrasyonu, hem esneklik hem de saldırılara açık kapılar anlamına gelebilir. Örneğin, bir Zynq SoC platformunda ARM çekirdekleri ve FPGA bölümü (PL) aynı yonga üzerindedir. Bu, işlemciden kaynaklanan saat sinyalleri daha güvenli kabul edilirken, saldırgan FPGA bölümüne yönelik bir RDPR atağıyla saat manipülasyonu veya bellek erişimini kötüye kullanabilir. Ayrıca bellek bloklarının (BRAM) sabit konumda bırakılması, sözlük ataklarında kritik kelime listesinin kolayca yakalanmasına sebebiyet verebilir. Dolayısıyla, IoT ve gömülü sistem güvenliği noktasında FPGA tabanlı tasarımlarda "dinamik bellek konum değiştirme", "timestamp ve nonce tabanlı anahtar yenileme" ve "bit akışı şifreleme" gibi stratejilerin gündeme gelmesi bu kısıtlarla doğrudan ilişkilidir (Karam vd., 2016; Samir vd., 2019).

Özetle, IoT ve gömülü sistemler güvenlik gereksinimlerinde hem yazılım hem de donanım düzeyinde eşgüdümlü önlemler gerektirir. Sınırlı kaynaklara rağmen, asimetrik kriptografinin ağır maliyeti veya sabit anahtarlı simetrik çözümlerin sözlük ataklarına açıklığı, literatürde birçok çalışmayı dinamik yenileme ve konum değiştirme gibi yöntemlere yöneltmiştir (Johnson vd., 2017). Aşağıdaki bölümlerde, bu yaklaşımı kavramsal olarak destekleyen kriptografik prensipler, donanım katmanında kullanılan teknolojiler ve saldırı türleri daha detaylı biçimde açıklanacak, önerilen yöntemin dayandığı temel kavramlar teknik detaylarıyla yer alacaktır.

3.2. Temel Kriptografik Yöntemler ve Protokoller

Gömülü sistemlerin ve IoT cihazlarının güvenliğinde, hem asimetrik hem de simetrik kriptografik yöntemlerin geniş bir yelpazede kullanıldığı görülmektedir. Bunun nedeni, IoT uygulamalarının farklı güvenlik ihtiyaçları (örneğin, uçtan uca gizlilik, kimlik doğrulama, veri bütünlüğü) yanında, ciddi kısıtlamalara (düşük işlem gücü, az bellek, sınırlı enerji) sahip olmasıdır. Literatürde, Diffie-Hellman, RSA ve ECC gibi asimetrik protokollerin güçlü güvenlik ilkelerine dayanmakla birlikte gömülü uygulamalarda yüksek işlem maliyeti veya uzun hesaplama sürelerine sebep olduğu çok sayıda deneyle doğrulanmıştır (Diffie ve Hellman, 1976; Rivest vd., 1978; Miller, 1985). AES gibi simetrik yöntemler ise hem yazılım hem de donanım tabanlı olarak düşük gecikme ve görece düşük enerji tüketimi sunabilmekte, fakat anahtarın yönetimi ve saklanması sorunu nedeniyle ek savunma katmanları gerektirmektedir (Stallings, 2016). Aşağıda, yaygın kriptografik protokol ve

algoritmaların IoT ve gömülü sistem bağlamında kullanımı, avantajları ile dezavantajları çerçevesinde geniş biçimde açıklanmıştır.

3.2.1. Asimetrik kriptografik yöntemler

Açık ve gizli anahtar çiftlerine dayanan asimetrik yöntemler, güvenli anahtar değişimi ve dijital imza gibi temel kriptografik işlevleri sağlamakta olup, bu başlık altında alandaki en temel protokoller incelenmektedir.

Diffie-Hellman (DH) protokolü

Diffie-Hellman, asimetrik kriptografinin ilk kavramsal temellerinden birini oluşturarak, iki tarafın güvenli olmayan bir kanal üzerinden ortak bir gizli anahtar paylaşmasına olanak tanır (Diffie ve Hellman, 1976). Protokol, büyük asal sayıların modüler aritmetiğine dayanır; taraflar ortak bir taban ve asal sayı üzerinde hesap yaparak bir anahtar üretir. Literatürde IoT cihazları üzerinde yapılan denemelerde, büyük asal sayılar kullanıldığında yüksek işlem süresi ve yüksek enerji tüketimi gözlemlenmiştir. Bir çalışma, 256-bit güvenlik seviyesini korumak istediğinde dahi, küçük bir gömülü işlemcinin DH hesaplamalarını birkaç saniyede tamamlayabildiğini; ancak bu sırada yoğun enerji çektiğini rapor etmiştir (Abu Al-Haija vd., 2017). Yan kanal saldırılarına yönelik bazı araştırmalar, DH hesaplamaları sırasındaki ara değerlerin saldırgan tarafından izlenebilmesi hâlinde, anahtarın kısmen tahmin edilebileceğini ileri sürmektedir (Johnson vd., 2017).

DH protokolünün temel avantajı, iletişim öncesi herhangi bir paylaşılan sır gerektirmemesidir. Yani iki taraf, açık bir kanaldan tablo değerleri veya asal sayılar iletebilmektedir. Fakat IoT koşullarında büyük asal sayılarla işlem yapmak; sabit, düşük güçlü sensör üniteleri için pratikte uzun hesaplama sürelerine ve pil ömrünün kısalmasına neden olmaktadır (Samir vd., 2019). Bu dezavantaj, bazı çalışmalarda DH'yi yalnızca ilk bağlantı esnasında kullanıp, sonrasında simetrik anahtarlara geçme stratejisini teşvik etmiştir.

RSA

RSA, Rivest, Shamir ve Adleman tarafından geliştirilen ve asal çarpanlara ayırmanın zorluğuna dayanan bir asimetrik protokoldür (Rivest vd., 1978). Dijital imza, açık anahtarlı

şifreleme gibi alanlarda yaygın kullanıma sahiptir. RSA'nın IoT bağlamındaki en büyük sıkıntısı, büyük asal sayı çiftlerinin üretilmesi ve işlenmesi sırasında ortaya çıkan yüksek işlem maliyeti ve uzun anahtar oluşturma süresidir. Bir literatür çalışmasında, 2048 bit uzunluğunda anahtar kullanılan bir RSA uygulamasının, küçük bir gömülü işlemcide saniyeler mertebesinde hesaplama gerektirdiği ve bu sırada pil tüketiminin zirve yaptığı gözlemlenmiştir. Ayrıca RSA ile yapılan şifre çözme ve imzalama işlemlerinin de görece uzun sürmesi, IoT cihazlarında gerçek zamanlı uygulamaları zorlaştırır (Karam vd., 2016).

Yine de RSA'nın avantajı, standartlaşmış ve yaygın kullanım alanına sahip olmasıdır. Büyük sunucu veya masaüstü platformlarında donanımsal hızlandırmayla kolayca entegrasyon mümkündür. Ancak gömülü sensör dizinlerinde, işlem ve enerji yükü sebebiyle daha hafif alternatiflerin aranmasına yol açmıştır (Samir vd., 2019). Yan kanal saldırıları ve saat manipülasyon senaryolarında da, RSA anahtarına ait ara değerler elde edilirse güvenlik büyük ölçüde riske girebilir (Johnson vd., 2017).

ECC (Eliptik Eğri Kriptografisi)

Eliptik eğri kriptografisi, asal çarpanlara ayırma yerine eliptik eğriler üzerindeki ayrık logaritma problemini esas alarak küçük anahtar boyutları ile yüksek güvenlik sağlamayı hedefler (Miller, 1985). Böylece RSA'ya kıyasla aynı güvenlik seviyesinde daha kısa anahtarlar kullanarak işlem verimliliği artar. Yine de IoT ortamında ECC tabanlı hesapların, özellikle montgomery çarpımı gibi matematiksel işlemlerde kayda değer işlem gücü gerektirdiği bildirilmiştir. Bazı araştırmalar, ECC'yi 160-224 bit anahtarlarla uygulayarak hafif bir form kazandırmaya çalışmış, ancak enerji tüketiminin yine de düşük güçlü cihazlar için önemli sayılabilecek düzeyde olduğu belirtilmiştir (Soliman vd., 2019).

ECC'nin avantajı, daha kısa anahtar boyutuyla asymetrik kriptografinin sağladığı güvenliği korurken, RSA'ya göre hesaplama bakımından görece avantajlı olmasıdır (Samir vd., 2019). Öte yandan, FPGA üzerinde ECC hızlandırması yapılan bir çalışmada, gelişmiş matris çarpımları ve modüler hesaplamalar için özel donanım çekirdekleri kullanmış, fakat ek donanım alanı ve ek güç harcaması gibi sonuçlara da dikkat çekilmiştir. Ayrıca makine öğrenimiyle ECC anahtarlarını çözme çabalarının varlığı, yine algoritma üzerinde ek maskeleme veya rastgele parazit ekleme yöntemleri uygulanmasını gerektirebilir (Karam vd., 2016).

3.2.2. Simetrik şifreleme ve hash yöntemleri

Tek bir gizli anahtarın kullanıldığı simetrik şifreleme ve veri bütünlüğü için geliştirilen hash yöntemleri, asimetrik yaklaşımlara göre daha yüksek performans sundukları için kaynak kısıtlı sistemlerde sıkça tercih edilmektedir.

AES (Advanced Encryption Standard)

AES, hem donanım hem de yazılım tabanlı olarak çok farklı platformlarda uygulanabilen, blok şifreleme esaslı bir simetrik algoritmadır. 128, 192 ve 256 bit gibi farklı anahtar boyutlarıyla kullanılabilen AES, hızlı ve güvenli bir yapı sunar (Stallings, 2016). IoT ortamında AES, düşük bit sürümleri veya turları azaltılmış versiyonlar şeklinde uygulanarak enerji tüketimi düşürülmüştür. Bir çalışmada, FPGA üzerinde AES'in 128 bitlik modda donanımsal çekirdek olarak entegre edilmesiyle, yazılım uygulamasına kıyasla çok daha yüksek veri hızı elde edildiği rapor edilmiştir. Fakat sabit bir anahtar kullanan AES uygulamalarında, sözlük ataklarına karşı koruma kısmen zayıf kalmaktadır; saldırgan anahtarı ele geçirdiğinde sistemin tamamı uzun süre kırılgan kalır (Johnson vd., 2017).

AES avantaj bakımından etkin bellek ve işlem kullanımı, gömülü platformlarda yazılım optimizasyonuyla kolay implementasyon ve yüksek dayanıklılık sunar. Dezavantaj olarak, sabit ya da uzun süre yenilenmeyen anahtarın gömülü hafızada saklanması, dictionary atak ve brute force girişimleri için açıklık doğurur. Ayrıca yan kanal saldırılarıyla güç dalgalanmasının incelenmesi veya tablolama teknikleri (S-Box'ların analizi gibi) saldırganlarca kullanılabilir.

MD5 (Message-Digest Algorithm 5) ve hafif hash uygulamaları

MD5, veri bütünlüğünü doğrulamak amacıyla 128 bitlik bir özet değer (hash) üretir; IoT uygulamalarında da hafif bir özet fonksiyonu olarak benimsenmiştir. Ne var ki MD5, çakışma ataklarına karşı kırılganlığı sebebiyle "kriptografik açıdan güvensiz" kategorisine doğru kaymıştır (URL2). Buna karşın, düşük işlem maliyeti ve basit implementasyon imkanı, bazı gömülü uygulamalarda hız avantajına sebep olmaktadır (Karam vd., 2016). Dolayısıyla MD5 veya benzeri hash fonksiyonları IoT tarafında yazılım tabanlı şekilde

kullanılabiliyorsa da, veri bütünlüğü ve kimlik doğrulama için tek başına yeterli görülmez; ek protokoller veya ek maskeleme mekanizmaları gerekir (Samir vd., 2019).

Literatürde, SHA-1 ve SHA-256 gibi daha güçlü hash fonksiyonları da sıklıkla analiz edilmiş, IoT kısıtları altında bit uzunluğu ve ek turların hesaplama yükü üzerine vurgu yapılmıştır. Dolayısıyla, hash fonksiyonlarının IoT'deki rolü genellikle kimlik doğrulama veya veri bütünlüğü ile sınırlıdır; anahtar oluşturma sürecine katılmaları hâlinde fazladan bellek tüketimi ve işlem süresi ortaya çıkar (Soliman vd., 2019). Kimi araştırmalarda, hafif hash algoritmaları (örn. PHOTON, SPONGENT vb.) incelenmiş (Poschmann, 2011; Bogdanov vd., 2007), ancak standartlaşma eksikliği ve güvenlik analizlerinin tam olgunlaşmaması konuları tartışılmıştır.

3.2.3. IOT'de kullanılan hafif kriptografi yaklaşımları

Asimetrik protokollerin IoT cihazlarında uzun hesaplama süresi ve yüksek enerji maliyetine sebep olduğu gerçeği, literatürü "hafif kriptografi" (lightweight cryptography) araştırmalarına yöneltmiştir (Stallings, 2016). Bazı çalışmalarda AES'in turları azaltılmış veya blok boyutu 64 bite düşürülmüş sürümleri değerlendirilmiş, böylece veri kapasiteleri düşük olan IoT cihazlarında optimum denge aranmıştır (Karam vd., 2016). Buna ek olarak, CAESAR gibi yarışmalar sonucunda seçilen kimlik doğrulamalı şifreleme algoritmaları (örneğin AEGIS, ASCON) IoT ortamında test edilmiş; ACORN, JAMBU, MORUS, CLOC, Pi-Cipher gibi yöntemlerin görece düşük donanım ihtiyacıyla yüksek hız sunabildiği belirtilmiştir (Silva vd., 2023).

Kimi zaman bu hafif şifreleme algoritmaları, FPGA üzerinde dinamik kısmi yeniden yapılandırma eşliğinde yüklenip yalnızca ihtiyaç duyulduğunda etkinleştirilerek ek enerji tasarrufu elde edildiği rapor edilmiştir (Karam vd., 2016). Literatürde, ICAP kullanımının her şifre değiştirme sürecinde ek enerji maliyeti getirdiğine yönelik uyarılar vardır, ancak buna karşın sabit bir anahtar veya algoritma kullanan sistemin saldırılara açık kapı bırakması da ciddi bir risk olarak tanımlanır. Bu nedenle "algoritma sıçrama" (algorithm hopping) mekanizmasıyla dinamik konfigürasyonun birlikte kullanıldığı çeşitli prototipler geliştirilmiş ve saldırganın belirli bir algoritmaya göre saldırı stratejisi oluşturmasını güçleştirdiği gösterilmiştir (Samir vd., 2019).

Özetle, IoT ve gömülü sistemlerde temel kriptografik protokoller, geleneksel asimetrik yöntemlerden hafif simetrik çözümlere kadar geniş bir yelpazeye yayılır. Diffie-Hellman, RSA, ECC gibi yöntemler güçlü güvenlik altyapısı sağlasa da gömülü cihazların güç ve işlem kısıtları sebebiyle pratik olmayan durumlar doğurabilir. AES gibi simetrik yöntemler hız ve düşük enerji avantajı sunsa da anahtar yönetimi kritik bir zayıf noktadır. MD5 gibi hash fonksiyonları, basitliğine rağmen kriptografik açıdan eskimiş kabul edilir; bunun yerine daha gelişmiş hash'ler veya hafif kripto standartları kullanılmaya başlanmıştır (Stallings, 2016). Sonuç olarak literatür, IoT'de optimal yaklaşım olarak hafif ve dinamik çözümlere yönelme eğilimindedir. Bir yandan anahtarın her mesajda değişmesi, dictionary atağına karşı konum değiştirme gibi önlemler devreye alınırken, diğer yandan ICAP tüketimi ve FPGA üzerinde PUF veya ek donanım modüllerinin karmaşıklığı da kontrol edilmeye çalışılır (Johnson vd., 2017; Samir vd., 2019). Buradan hareketle, sonraki alt başlıklarda donanımsal güvenlik kavramları, FPGA/SoC mimarilerinin saldırı yüzeyleri ve saldırı türleri (dictionary, brute force, yan kanal) anlatılarak, esas uygulamalara teori düzeyinde temel oluşturulacaktır.

3.3. Donanımsal Güvenlik

Yazılım düzeyindeki savunma önlemleri, gömülü sistemlerin ve IoT cihazlarının güvenliğinde önemli bir yer tutsa da, donanım katmanındaki açıklar dikkate alınmadığında çok daha karmaşık ve tehlikeli saldırı vektörlerine maruz kalınabilmektedir. Literatürde, özellikle FPGA, SoC ve ASIC benzeri programlanabilir veya sabit tasarıma sahip entegre devrelerde, bit akışının (bitstream) manipüle edilmesi, donanım truva atı (HTH) eklenmesi ve yan kanal analizleri gibi saldırıların ciddi tehdit oluşturduğu vurgulanmıştır (Johnson vd., 2017; Karam vd., 2016; Samir vd., 2019). Bu alt başlık, donanımsal güvenlik kavramını, FPGA tabanlı sistemlerde sık karşılaşılan saldırı türlerini ve bu saldırıların önüne geçmek için kullanılan teknikleri, kavramsal temelleriyle birlikte açıklamayı amaçlamaktadır.

FPGA, ASIC ve SoC kavramları (Zynq 7020, MicroBlaze gibi yapılar)

Donanım güvenliği, çoğu zaman tasarımın hedef platformuna göre değişen önlemler ve optimizasyonlar gerektirir. FPGA, içerisinde programlanabilir mantık hücreleri ve ara bağlantı ağları barındıran, kullanıcı tanımlı işlevlerin donanım seviyesinde hızla gerçeklenebildiği esnek devrelerdir. Bu yapı, bit akışı (bitstream) adı verilen bir konfigürasyon dosyası yüklenerek işlevsellik kazanır. ASIC (Application-Specific

Integrated Circuit) ise tasarımı üretim sırasında sabitlenmiş, yeniden programlanamayan yongaları ifade eder. FPGA'nin aksine ASIC, üretim sonrası değiştirilemediğinden donanım truva atı yerleştirme riskini tedarik zinciri aşamasına odaklar. SoC yapıları, örneğin Zynq 7020 ailesi, FPGA mantığıyla birlikte işlemci çekirdeklerini (ARM tabanlı) aynı çipte barındırır; bu da yazılım ve donanımın iç içe geçtiği, güvenlik gereksinimi açısından çok katmanlı bir platform sunar. MicroBlaze gibi soft-core işlemciler de FPGA üzerinde çalışan yazılım süreçlerine elverişli bir işlemci mimarisi oluşturarak, gömülü sistemlerin yüksek esneklikle tasarlanmasına yardımcı olur.

Bu entegre platformlarda, hem işlemci tarafında (PS - Processing System) hem de FPGA tarafında (PL - Programmable Logic) pek çok güvenlik bileşeni devreye alınır. Örneğin, Device DNA adı verilen benzersiz kimlik numarası, FPGA çipinin her birini donanımsal olarak ayırt eden bir özellik olarak kullanılabilir. eFUSE veya BBRAM gibi bellek birimleri, FPGA içindeki anahtarları saklamak için tasarlanmıştır ve enerji kesilse bile veriyi koruyabilir ya da tek yazımla kilitlenebilir. Bu imkânlar, bit akışı şifrelemesi veya donanım tabanlı kimlik doğrulaması gibi yüksek güvenlikli uygulamalarda sıkça kullanılır.

Hardware Trojan (HTH), yan kanal saldırıları ve bitstream manipülasyonu

Donanım truva atı (HTH), tasarım aşamasında ya da tedarik zinciri boyunca entegre devreye gizlice eklenebilen zararlı bileşenlerdir. FPGA gibi yeniden programlanabilir sistemlerde, HTH riski bazen "Remote Dynamic Partial Reconfiguration (RDPR)" imkânını ele geçirmiş bir saldırganın FPGA'ye kötü amaçlı bit akışları yüklemesiyle gerçekleşir (Johnson vd., 2017). Bu sayede saldırgan, FPGA içinde ek lojik devreler barındırarak kriptografik modülleri sabote edebilir, gerçek rastgele sayı üreteci (TRNG) çıktısını öngörülebilir hâle getirebilir veya saat sinyali üzerinde oynama yaparak sistem bütünlüğünü sarsabilir.

Yan kanal saldırıları da donanım düzeyinde sıklıkla incelenir: güç tüketimi, elektromanyetik (EM) yayılım veya zamanlama üzerinden gizli anahtarların sızdırılabildiği çalışmalar mevcuttur (Karam vd., 2016). FPGA üzerinde yürütülen kriptografik işlemler, bu saldırılara hedef olabilir; özellikle saat sinyali manipülasyonuyla gecikme değerleri ölçülüp ara değerlerin tahmini yapılabilir. Bazı araştırmacılar, "co-obfuscation" yaklaşımıyla donanım ve yazılım bloklarını birlikte gizleyerek saldırı yüzeyini daraltmayı ve bit akışını tersine mühendislik yöntemlerinden korumayı başarmıştır (Samir vd., 2019).

Bitstream manipülasyonu, FPGA güvenliğinde bir başka kritik başlıktır. FPGA konfigürasyon verisi olan bit akışı, saldırgan eline geçtiğinde tersine mühendislik veya zararlı modül ekleme riskine neden olur. Buna karşı geliştirilen bit akışı şifrelemesi ve obfuscation yöntemleri, FPGA'nin yalnızca kendi dahili anahtarları (örn. eFUSE, BBRAM) ile çözülebilen bit akışlarını kabul etmesini ve dışarıdan gelen yetkisiz modüllerin işe yaramamasını amaçlar. Dahası, "on-chip AES decryption circuit" gibi mekanizmalarla bit akışı FPGA içinde çözülerek, hem saat sinyali üzerinde saldırı hem de dışarıdan gizli bilgiyi elde etme ihtimali minimal düzeye indirilmeye çalışılır. Bu yaklaşım, IP korumasında da sıklıkla gündeme gelir.

DFX (Dynamic Function eXchange) mantığı ve temel prensipleri

Dinamik Fonksiyon Değişimi (DFX), FPGA üzerinde çalışma anında belirli fonksiyonların yerini veya içeriğini değiştirebilme özelliğidir. Bu yaklaşım, partial reconfiguration (kısmi yeniden yapılandırma) temelinde çalışarak ICAP (Internal Configuration Access Port) üzerinden FPGA'nin hedef bölgesine yeni bit akışları yüklenmesini sağlar. Böylece farklı algoritmalar veya modüller, sistem çalışırken peş peşe devreye girebilir. Bazı çalışmalar, DFX yöntemini "algoritma sıçrama (algorithm hopping)" ile birleştirmiştir. Bu sayede saldırganın sabit bir şifreleme çekirdeğine yönelik analiz yapması veya tersine mühendislik uygulaması zorlaşır (Samir vd., 2019). Ancak ICAP kullanımının ek enerji maliyeti ve tasarım karmaşıklığı getirdiğine de dikkat çekilir (Karam vd., 2016).

Araştırmacılar, DFX ile Bellek (BRAM) konum değiştirme sürecini de yöneterek sabit hedefler sunmaktan kaçınır. Bu, sözlük saldırılarını veya brute force sırasında saldırganın aynı adresi uzun süre gözlemlemesinn önüne geçer. Literatürde, FPGA konfigürasyon verilerinin dinamik taşınmasıyla hem IP koruması hem de saldırı yüzeyinin parçalanması açısından olumlu sonuçlar elde edildiği rapor edilmiştir. Öte yandan, her yeniden yapılandırma adımında bit akışının şifreli kalması ve saat sinyallerinin saldırılara karşı korunması şarttır. Aksi takdirde, saldırgan "initial configuration with intermediate readback" gibi yöntemlerle içerideki veriyi yakalayabilir veya zafiyete sebep verebilir (Johnson vd., 2017).

Tasarım için test edilebilirlik (Design for Testability - DFT) ve cihaz kimlik doğrulaması

Donanım tasarımında, DFT ilkeleri sayesinde, FPGA içerisindeki kısmi yeniden yapılandırma bölgelerinde test ve izleme devreleri eklenerek zararlı bit akışları veya HTH eklentilerinin anormal davranışları fark edilebilmektedir (Samir vd., 2019). Sensör benzeri bir görev gören test devreleri, güç tüketiminde olağandışı artış, beklenmeyen saat kaymaları veya ek lojik blok hareketlerini izleyebilmektedir. Böylece, RDPR yoluyla sisteme sızdırılan bir modülün varlığı erkenden anlaşılabilir.

Cihaz kimliği doğrulaması için ise Device DNA gibi FPGA'ye özgü benzersiz kimlik numaraları, bit akışının sadece ilgili çipte çalışmasını sağlayacak bir şifreleme mekanizmasında kullanılabilir. Bazı sistemler, eFUSE veya BBRAM destekli bellek alanlarında saklanan anahtarla bit akışını şifreler ve sadece hedef FPGA bu bit akışını çözümleyebildiği için üretimden sahaya kadar IP hırsızlığı veya truva atı ekleme riski ciddi şekilde azaltılır. Buna ek olarak, PUF (Physical Unclonable Function) tabanlı yaklaşımlar da donanımsal kimlik doğrulama amacıyla sıklıkla kullanılır, fakat çevresel hassasiyet ve makine öğrenimi saldırı riski göze çarpar (Karam vd., 2016).

Aşağıdaki çizelge, donanım katmanında çeşitli güvenlik önlemlerinin hangi saldırı senaryolarına karşı koruma sağladığını ve hangi noktalarda ek risklerin söz konusu olduğunu Çizelge 3.1'de özetler.

Çizelge 3.1. Donanım tabanlı güvenlik yöntemleri ve olası zafiyetleri

Yöntem	Korumak İstediği Saldırılar	Olası Zafiyet ve Dezavantajlar
Bit Akışı Şifrelemesi	- Bit akışı manipülasyonu - Tersine mühendislik	- Yan kanal analizine açık (şifre çözme sırasında güç ölçümü) - Anahtar yönetimi zorluğu (eFUSE, BBRAM)
On-Chip AES Decryption	- FPGA dışında bit akışını çözmeye çalışan saldırı	- Donanım ek kaynak tüketimi - AES devresi yan kanal saldırılarına hedef olabilir
Dinamik Fonksiyon Değişimi (DFX)	- Sabit topolojiye dayalı HTH - Yan kanal modellemesi	- ICAP kullanımı sırasında ek enerji harcaması - Yeniden yapılandırma sürecinin güvenli yönetimi gereklidir

Çizelge 3.1. (devam) Donanım tabanlı güvenlik yöntemleri ve olası zafiyetleri

Bellek (BRAM) Konum Değiştirme	- Dictionary veya brute force - Yan kanal (güç/EM) sabit noktası	- Her taşıma işlemi konfigürasyon süreci ister-Sistemde senkronizasyon zorlukları - Enerji tüketimi artabilir
PUF Tabanlı Kimlik (PUF + CRP)	- Cihaz klonlama - Geleneksel saklı anahtar saldırıları	- Çevresel faktör hassasiyeti (sıcaklık, voltaj)-CRP modellemesiyle makine öğrenimi saldırısı - FPGA'de entegrasyon karmaşıklığı
Yan Kanal Önlemleri (Dummy Load vb.)	- Güç analizi- EM sızıntısı	- Ek lojik ve enerji tüketimi - Tüm saldırı vektörlerini kapatmakta yetersiz kalabilir
Donanım Truva Atı Tespiti (DFT)	- Gizli eklenmiş lojik blok-Beklenmeyen güç tüketimi	- Ek devre karmaşıklığı- Her saldırıyı tespit garantisi yok - Yazılım/donanım ortak analizi gerektirir

Çizelge 3.1'den anlaşılacağı üzere, FPGA tabanlı donanım güvenlik yöntemleri çeşitli saldırı kategorilerine odaklanmakta, ancak her çözümün yan kanal, enerji tüketimi ve ek donanım karmaşıklığı gibi dezavantajları bulunmaktadır (Samir vd., 2019; Johnson vd., 2017). Bu yöntemlerin bir arada uygulanması, saldırı yüzeyini önemli ölçüde daraltsa da, tasarım sürecini oldukça karmaşık hale getirmektedir.

Sonuç olarak, donanım güvenliği, gömülü sistemlerin IoT ekosisteminde karşı karşıya kaldığı karmaşık saldırı vektörlerine bütüncül bir bakış açısı gerektirir. FPGA'nin dinamik yapılandırma kabiliyeti, bit akışı şifrelemesi, bellek konum değiştirme, PUF tabanlı kimlik, tasarımın test edilebilirliği (DFT) gibi yöntemler, hem birbirini tamamlayıcı hem de ek karmaşıklık sağlayacak unsurlar olarak değerlendirilmelidir (Karam vd., 2016; Samir vd., 2019). Böylece, yan kanal incelemesinden hardware trojan yerleştirmeye, sözlük saldırısından saat sinyali manipülasyonuna kadar, geniş bir saldırı tipine karşı dirençli sistemler tasarlanabilmektedir. Bu kavramsal temel, ileri bölümlerde FPGA üzerinde dinamik anahtar üretme ve bellek konumu değiştirme gibi stratejilerin nasıl şekillendiğinin daha net anlaşılmasını sağlayacaktır.

3.4. PUF (Physically Unclonable Function) Tabanlı Sistemler

Donanım temelli güvenlik araştırmaları, gizli anahtarları veya kimlik doğrulama mekanizmalarını yazılım yerine fiziki özellikler aracılığıyla sağlamanın avantajlarını

sıklıkla vurgular. Bu bağlamda, PUF (Physically Unclonable Function) yapıları, üretim sürecindeki rastgele varyasyonlardan yararlanarak cihazı donanım düzeyinde benzersiz kılan bir yöntem olarak öne çıkar (Al-Meer ve Al-Kuwari, 2023). Her FPGA veya ASIC, işlem sırasında ortaya çıkan doğal gecikme, bellek davranışı veya başka fiziksel ögeler sayesinde, yazılım kopyalamayı veya donanım sahteciliğini çok daha zorlu hâle getiren özgün bir "parmak izi" üretir. Bu yaklaşım, gömülü sistemlerin IoT bağlamında karşılaştığı saldırılara düşük maliyetli ve uzun vadeli bir çözüm sunma hedefiyle literatürde çeşitli uygulamalara sahiptir.

PUF yapılarına dair temel ilkeler

Bir PUF, belirli bir "challenge" girişi alır ve o cihaza özgü fiziksel özelliklerden kaynaklanan "response" değeri üretir. Belirli sayıdaki challenge-response çifti (CRP), kimlik doğrulamada veya anahtar üretiminde kullanılabilir. Örneğin ring osilatör (RO) PUF tasarımında, her bir osilatörün üretim aşamasından gelen mikroskobik farklılıkları, farklı frekanslar oluşturmakta ve bu frekanslar ölçülerek PUF yanıtı elde edilmektedir (Samir vd., 2019). Arbiter PUF modelleri ise eşzamanlı yollardan geçen sinyallerin hangisinin önce ulaştığını karşılaştırarak bir bit üretir. Memory-based PUF yapılarında ise SRAM hücreleri gibi bellek elemanlarının ilk çalışma ânında aldığı rastgele başlangıç hâli değerlendirilir. Tüm bu mekanizmaların ortak noktası, saldırganın aynı fiziksel yapıyı kopyalayamamasına ve cihaza ait CRP'leri statik bir anahtar gibi tahmin edememesine dayalı olmasıdır (Karam vd., 2016).

Çevresel faktörlere hassasiyet ve CRP yönetimi

PUF tabanlı sistemler, her ne kadar kopyalanamazlığı arttırsa da sıcaklık, gerilim dalgalanması, yaşlanma etkileri gibi dış faktörlerden dolayı yanıt kararlılığını kaybetme riski taşır (Al-Meer ve Al-Kuwari, 2023). Hata düzeltme kodları veya ek lojik devreler, CRP değerlerini stabilize etmek üzere kullanılabilir, ancak bu yöntemler FPGA veya ASIC'te ek alan ve ek enerji tüketimi anlamına gelir. Literatürde, özellikle IoT cihazlarında PUF'un CRP sayısı çok fazla sunulursa makine öğrenimiyle model oluşturulabileceği ve PUF'un taklit edilebileceği uyarısı yapılmıştır. Bu nedenle CRP çıktılarının sınırlandırılması veya dinamik olarak değiştirilmesi tavsiye edilir (Samir vd., 2019).

Çevresel faktörlere hassasiyet ve CRP yönetimi

Bazı çalışmalar, FPGA içindeki kısmi yeniden yapılandırma (DPR) olanaklarından yararlanarak "dinamik PUF" tasarlama yoluna gitmiş, ring osilatörlerin yol ve sayısını çalışma anında değiştirip saldırganın sabit bir fiziksel modeli ele geçirmesini zorlaştırmıştır (Karam vd., 2016). Başka bir yaklaşımsa, FPGA'de gömülü bellek bloklarının (BRAM) konumunu değiştirerek PUF verilerini sabit bir lokasyona dayalı inceleme girişimlerini geçersiz kılar. Yine de bu yöntemlerin çevresel hassasiyeti ve yeniden yapılandırma sırasında ICAP kullanımından kaynaklanan enerji yükü literatürde tartışma konusu olmayı sürdürmüştür (Samir vd., 2019).

Avantaj ve dezavantajların kapsamlı değerlendirmesi

Aşağıdaki çizelgede, PUF tabanlı sistemlerin IoT ortamında getirdiği avantajlar ve karşı karşıya kaldığı dezavantajlar özetlenmiştir. Aynı çizelge, farklı araştırmalarda tespit edilen zayıf noktaları da içerir.

Çizelge 3.2. PUF tabanlı yaklaşımların avantaj ve dezavantajları

Özellik	Avantajlar	Dezavantajlar
Fiziksel Olarak Kopyalanamayan Kimlik	- Cihaz üretim sürecindeki rastgele varyasyonları kullanarak benzersiz kimlik üretir. - Yazılım kopyalama veya klonlamaya kıyasla daha üst düzey koruma sağlar.	- Üretim partileri arasındaki varyasyon farkları denetim gerektirir. - Bazı PUF türleri (örn. bellek tabanlı) sıcaklık, gerilim dalgalanmaları karşısında kararlılık sorunu yaşayabilir.
CRP (Challenge-Response Pair) Mekanizması	- Tekrarlanmayan kimlik doğrulaması ve anahtar üretimi yapılabilir - Büyük CRP seti, saldırganın tam modelleme yapmasını zorlaştırır	- CRP sayısı çok fazla olursa makine öğrenimi tabanlı modelleme riskini artırır (Samir vd., 2019) - Yanlış ölçümler, hata düzeltme katmanı gerektirir (ek lojik/enerji)
FPGA Üzerinde Dinamik PUF	- Kısmi yeniden yapılandırma ile ring osilatör yolları veya hafıza blokları çalışma sırasında değiştirilebilir - Donanım truva atı ve yan kanal saldırılarına karşı ek direnç	- Her yeniden yapılandırma adımında ICAP kullanımının ek enerji maliyeti (Karam vd., 2016) - FPGA'de tasarım karmaşıklığı, test edilebilirlik mekanizmalarının hassas ayar gerektirmesi

Çizelge 3.2. (devam) PUF tabanlı yaklaşımların avantaj ve dezavantajları

Donanım Kimlik Doğrulamasında Kullanımı	- Gizli anahtar depolamaya gerek kalmadan cihaza özgü kimlik sunar - Yazılım korsanlığına ve bellek üzerinden anahtar sızdırmaya karşı ek koruma	- PUF yanıtlarındaki belirsizlik, çevresel faktörler ve üretim toleranslarından etkilenebilir - CRP yanıtlarının yönetimi ve saklanması için ek protokoller gerekebilir
Çevresel Koşullara Duyarlılık (Termal, Gerilim vb.)	- Bazı tasarımlar yüksek adaptasyonla daha da rastgele yanıtlar üretebilir - Bazı araştırmalar, çevresel değişimin doğal rastgeleliği artırdığını ileri sürer	- PUF yanıtlarının güvenilirliği düşebilir (Al-Meer ve Al-Kuwari, 2023) - Voltaj saldırıları veya termal saldırılarla sahte CRP değeri oluşturulabilir

Çizelge 3.2'deki veriler, FPGA tabanlı PUF uygulamalarında dikkat edilmesi gereken önemli hususları ortaya koymaktadır (Johnson vd., 2017). Çevresel etkenler, CRP yönetimi, kısmi yeniden yapılandırma senaryoları ve makine öğrenimi saldırıları gibi konularda mevcut literatür, henüz tam anlamıyla olgunlaşmış çözümler geliştirme aşamasındadır. Bununla birlikte, kimlik doğrulama ve anahtar üretimi açısından yazılım tabanlı yaklaşımlara göre donanım seviyesinde önemli avantajlar sunduğu belirtilmektedir.

Donanım truva atı (HTH) veya yan kanal saldırılarına karşı, PUF'un dinamik reconfigurable yapılara entegrasyonu da araştırılmaktadır. Birden fazla ring osilatör yoluna sahip FPGA tasarımının, çalışma sırasında bu yolları rastgele şekilde seçerek saldırganın sabit bir modeli ele geçirmesini engellediği gözlemlenmiştir (Karam vd., 2016). Diğer yandan, CRP verilerinin sızması hâlinde makine öğrenimi analiziyle PUF'u modellemenin mümkün olduğu da rapor edilmiştir (Samir vd., 2019). Bu nedenle her PUF uygulamasında CRP yönetimi ve güvenli veri saklama büyük önem taşımaktadır.

3.5. Saldırı Kavramları

Gömülü sistem ve IoT cihazlarının geniş saldırı yüzeyi, yalnızca kriptografik protokollerin değil, donanım platformunun güvenliğini de bütüncül şekilde ele almayı zorunlu kılar. Sözlük (dictionary) atağı, brute force, yan kanal (güç/EM/zamanlama analizi), donanım truva atı (HTH) ve tersine mühendislik gibi saldırı yöntemleri literatürde sıkça ele alınmaktadır (Johnson vd., 2017; Samir vd., 2019). Bu bölümde, özellikle IoT'nin kısıtlı

kaynakları ve FPGA tabanlı tasarımlarla ilgili saldırı vektörleri ve bu vektörlere karşı kullanılan savunma konseptleri incelenir.

3.5.1. Sözlük (Dictionary) ve brute force saldırıları

Sözlük atağı, genellikle kısa bir kelime veya parola listesine dayalı sistemlerde kullanılır. Saldırgan, hedef sistemin anahtarını veya parola tabanlı bileşenini, elindeki listede sıralayarak dener. Özellikle gömülü sistemlerde sabit bir kelime listesi veya uzun süre değişmeyen bir anahtar saklanıyorsa, saldırının başarı şansı yükselir (Karam vd., 2016). Brute force ise her olası anahtarı sırayla deneyerek çözüm bulmayı hedefleyen daha genel bir yaklaşım olup, eğer sistem kısa anahtar boyutu veya statik anahtar kullanıyorsa pratik bir zaman diliminde kırılabilme riski içerir (Johnson vd., 2017).

Yazılım tabanlı dinamik anahtar yenileme ve bellek konum değiştirme yöntemleriyle, sözlük atağının etkinliği büyük ölçüde düşürülebilir. Araştırmalar, her mesajda farklı anahtar kullanılmasının saldırganın “bir kez kır, sonsuza dek kullan” senaryosunu işe yaramaz hâle getirdiğini göstermiştir (Samir vd., 2019; Soliman vd., 2019). Bellek alanında tuttuğu kelime listesinin sabit konumda olmaması (dinamik yer değiştirme) da saldırganın aynı adresi hedefleyerek listeyi ele geçirme çabasını anlamsız kılar (Stolz vd., 2021). Bu yöntemler, ek konfigürasyon adımı gerektirse bile IoT ortamı için çok daha uzun soluklu koruma getirdiğini deneysel bulgularla doğrulamaktadır.

3.5.2. Yan kanal saldırıları (güç, EM, zamanlama analizleri)

Yan kanal saldırıları, FPGA veya ASIC üzerinde çalışan şifreleme algoritmalarının güç tüketimi, elektromanyetik sızıntı veya zamanlama verilerini inceleyerek anahtarları ya da gizli parametreleri sızdırmak üzere tasarlanır (Karam vd., 2016). Bir çalışmada, saldırgan FPGA devresinde belirli mantık kapılarının ne zaman etkin olduğunu güç dalgalanmalarını veya EM yayılımını ölçerek analiz etmiş ve kriptografik çekirdeğin ara değerlerini tahmin etmiştir. Başka bir senaryoda, saat sinyalinin manipülasyonu yoluyla (örneğin, saat frekansını anlık olarak yükseltmek) algoritmadaki kritik döngülerin dışa vurduğu zamanlama profilini inceleyerek anahtar çıkarma girişimleri kaydedilmiştir (Samir vd., 2019).

Bu tür saldırılara karşı, sistem tasarımında dummy load, rastgele maskeleme, ara değerlerin saklandığı konumun sürekli değiştirilmesi gibi yöntemler önerilmektedir. Bazı çalışmalar, FPGA'de kısmi yeniden yapılandırma kullanarak kriptografik modülleri periyodik olarak farklı konumlara taşımanın, saldırganın yan kanal ölçümlerini tutarlı bir modele dönüştürmesini zorlaştırdığını göstermiştir (Johnson vd., 2017). Aynı zamanda, dahili saat kaynaklarının (örneğin Zynq işlemcisinden gelen saat) dış müdahaleye daha kapalı olması da sistemin zamanlama saldırılarına karşı direnç kazandırmaktadır.

3.5.3. Donanımsal manipülasyon ve truva atları (HTH)

Donanım truva atı saldırıları, entegre devre içinde gizli bir lojik devre yerleştirerek sistemin normal işleyişini bozmayı veya veri sızdırmayı amaçlar. FPGA gibi esnek platformlarda, saldırgan uzaktan malicious bit akışları yükleyerek HTH ekleyebilir (Karam vd., 2016). Bu senaryoda, remote dynamic partial reconfiguration aracılığıyla sisteme yetkisiz modül eklenmesi ve kritik modüllerin clock veya veri yoluna müdahale edilmesi mümkündür (Johnson vd., 2017).

Farklı araştırmalarda, bit akışının şifrelenmesi ve FPGA'de yalnızca cihazın dahili anahtarlarıyla (eFUSE veya BBRAM) çözülebilmesi, bu tür HTH ekleme saldırılarını büyük ölçüde önlemektedir. Ayrıca, tasarımın test edilebilirliği (DFT) yöntemleriyle çalışma sırasında anormal güç çekişi veya beklenmedik saat sinyali manipülasyonu tespit edilebilmektedir (Samir vd., 2019). HTH saldırısı, donanımdaki trojan devrenin belirli bir koşulla etkinleşmesi şeklinde de kurgulanabildiğinden, sürekli izleme ve lojik bütünlük kontrolü gibi tasarım içi güvenlik katmanları önem kazanır.

3.5.4. Tersine mühendislik ve IP korsanlığı

Tersine mühendislik, saldırganın bit akışını veya ASIC tasarımını çözerek devrenin işlevini anlaması, IP (fikri mülkiyet) bloklarını kopyalaması veya zafiyetleri istismar etmesi riskini ifade eder. FPGA bit akışı şifreleme veya obfuscation teknikleri, bu riski azaltmayı hedefler (Samir vd., 2019). Ayrıca Bellek (BRAM) konum değiştirme, saldırganın sabit bir modüle dair "geçerli" netlist veya kapı düzeyi harita çıkarmasını anlamsız kılar. Ancak, saldırgan netlisti elde etse dahi, anahtar yönetimi ve zaman damgası gibi dinamik parametreler olmadan sistemi klonlaması veya çalıştırması mümkün olmayabilir (Karam vd., 2016).

3.5.5. Sonuç

Bu bölümde açıklanan saldırı kavramları, IoT ve gömülü sistemlerin düşük işlem gücü ve sınırlı enerji ortamında maruz kalabileceği geniş yelpazeyi yansıtmaktadır. Sözlük (dictionary) atağı, brute force, yan kanal analizleri, donanım truva atı ve tersine mühendislik riskleri, sadece yazılım katmanında değil, donanım tasarımının özünde de önlemler almayı zorunlu kılar (Johnson vd., 2017). FPGA tabanlı tasarımlarda kısmi yeniden yapılandırma (DFX) ve bit akışı şifrelemesi gibi yöntemler, saldırganın uzaktan kötü amaçlı modül ekleme veya sabit bellek konumlarını hedef alması senaryolarını boşa çıkarmaya odaklanır. Aynı zamanda, her mesajda anahtar yenileme veya bellek içeriği konum değiştirme gibi stratejiler, sözlük ve brute force ataklarını pratikte işe yaramaz hâle getirebilir. Bu kavramsal altyapı, tez içerisinde ele alınacak dinamik bellek yönetimi ve yazılım tabanlı anahtar üretimi tekniklerine zemin oluşturmaktadır.

4. MATERYALLER VE YÖNTEMLER

Bu bölüm, iletinin her biri için kimlik doğrulamalı şifrelemeyi (AEAD) AXI üzerinden eşlenmiş BRAM üzerinde saklanan gizli havuzun ve seçili kod sayfalarının çalışma zamanında Dynamic Function eXchange (DFX) ile yerinin değiştirilmesiyle birleştiren hibrit güvenlik çerçevesinin (HGÇ) nasıl uygulandığını açıklar. Anlatım, şifreleme algoritmasının donanım tasarımına odaklanır. Ölçüm ve kıyaslama ayrıntıları Bulgular bölümünde raporlanmaktadır.

4.1. Hibrit Güvenlik Çerçevesi (HGÇ): Tasarım ve AEAD Arabirimi

256 bitlik bir kök sır (S_{ROOT}), bir aygıt tanımlayıcısı (dev_{id}, 32 bit, hatta "big-endian") ve 128 bitlik dilimlerden oluşan bir havuz (POOL[ ]) AXI üzerinden eşlenmiş BRAM'de tutulur. Havuz girdileri, 256 bitlik kökün üzerinde ardışık olmayan ofsetlerden alınan 128 bitlik pencerelerdir. idx→dilim eşlemesi BRAM'de tutulur ve seçili kod sayfalarıyla birlikte DFX tarafından düzenli aralıklarla taşınır; böylece hem sırların hem de kodun fiziksel konumu hareketli bir hedefe dönüşür. Kimlik doğrulamalı şifrelemede ASCON-128a kullanılır. Politika kontrollü anahtar türetimi ASCON-XOF128 ile yapılır. Nonce ve AAD açıktır ancak bütünlüğe bağlıdır; kabullenme yalnızca etikete dayanır.

Anahtarlama ve alan ayrımı (KDF Politikaları)

ASCON-XOF128, 256 bitlik S_{ROOT} üzerinde, kamuya açık (public) bağlamda anahtarlı bir PRF/RO olarak modellenir. İki kurulum politikası desteklenmektedir. Her ikisi de bir protokol dizgesi dom ile açık alan ayrımı uygular (Bertoni vd., 2007; Bertoni vd., 2011;NIST, 2015).

Politika A (dilim başına anahtar)

$$K_i = ASCON - XOF128(\text{“}KDF|ASCON|v1\text{”} \parallel S_ROOT \parallel dev_id \parallel POOL[idx]) \qquad (4.1)$$

Anahtarlar aygıt+dilim bazındadır. Anahtar başına nonce tekilliği zorunludur. Bir K_i'nin ele geçirilmesi, yeniden anahtarlamaya veya dilim döndürmeye kadar yalnızca aynı (dev_{id}, idx) ile gönderilen paketleri etkiler (Eş. 4.1).

Politika B (isteğe bağlı, ileti başına anahtar)

$$K_i = ASCON - XOF128(\text{“}KDF|ASCON|v1\text{”} \parallel S_ROOT \parallel dev_id \parallel POOL[idx] \parallel epoch \parallel msg) \quad (4.2)$$

Anahtarlar paket bazındadır; tek bir K_i'nin ele geçirilmesi, $(dev_id, idx, epoch, msg)$ tamamen bilinse bile diğer paketlere yardımcı olmaz (Eş. 4.2).

Her iki politika için de S_{ROOT} aygıttan asla çıkmaz ve dom dizgesi, çapraz protokoller arasında alan ayrımı sağlar. Politika B'de (epoch, msg) ayrıca paket başına alan tanımlar (Bertoni vd., 2007; Bertoni vd., 2011; Dobraunig, vd., 2021). Bu nedenle bir K_i'nin açığa çıkması, $(dev_{id}, idx, epoch, msg)$ bilinse bile herhangi bir K_j ($j \neq i$) için fayda sağlamaz.

Nonce ve AAD oluşturma

Kamuya açık nonce 16 bayttır ve $dev|epoch|msg$ alanlarını "big-endian" olarak paketler (Eş. 4.3):

$$nonce = pack128(dev_id[4], epoch_ctr[4], msg_ctr[8]) \quad (4.3)$$

Yeniden kullanım, $(epoch_{ctr}, msg_{ctr})$ üzerinde tekdüze bir politika ile engellenir (Dworkin, 2007; Dobraunig vd., 2021).

Çizelge 4.1. Nonce ve AAD alanlarının yapısı ve boyutları

Alan	Kaynak	Boyut (bit)	Açıklama ve Sınırlar
`dev_id`	Nonce/AAD	32	Aygıt kimliği.
`epoch`	Nonce	32	Yeniden başlatma sayacı. Güç kesintisine karşı kalıcı bellekte saklanmalı. 2^{32} yeniden başlatmaya kadar tekillik sağlar.
`msg`	Nonce/AAD	64	Mesaj sayacı. Epoch başına 2^{64} mesaja kadar tekillik sağlar. Sarmalanmadan (wrap around) önce `epoch` artırılmalıdır.
`ver`	AAD	8	Protokol sürümü.
`idx`	AAD	8	Havuz (POOL) indeksi. `
`feat`	AAD	8	Uygulamaya özel özellik bitleri.
`ts`	AAD (isteğe bağlı)	64	Zaman damgası.

Not: Nonce, dev_{id}, epoch ve msg alanlarının big-endian sıralamasıyla birleştirilmesiyle (concatenation) oluşturulur. AAD'nin toplam boyutu, ts alanı olmadan 15 bayt (120 bit), ts alanı ile 23 bayt (184 bit) olur.

Güvenlik çerçevesinin temelinde, standart ve anahtarsız bir Genişletilebilir Çıktı Fonksiyonu (XOF) olan ASCON-XOF'un, anahtarlı bir sözde rastgele fonksiyon (PRF) olarak kullanılması yatar. Bu modelde, uzun dönemli gizli sır S_{ROOT} anahtar olarak işlev görür ve dev_{id}, $POOL[idx]$ gibi kamusal girdileri işleyerek mesaj-başı anahtarları türetir. Alan ayrımı (domain separation), "$KDF|ASCON|v1$" gibi sabit bir dizge ile sağlanarak, bu KDF'in diğer protokollerdeki kriptografik fonksiyonlarla karışması önlenir. Bu çalışmada kullanılan "havuz" (POOL), S_{ROOT} üzerinde belirli ofsetlerden alınan 128-bitlik pencerelerden oluşur; bu yaklaşım, KDF girdisini çeşitlendirerek ek bir değişkenlik katmanı sunar.

İşletim kuralları:

(i) Yeniden başlatmada epoch'u artır; (ii) msg sayacını tekdüze ilerlet; (iii) sayaç sarmalanmadan çok önce yeniden anahtar (yeni idx ya da yeni bir S_{ROOT} türevi) uygula. Anahtar başına tekil nonce, AEAD güvenliği için zorunludur.

Gerekçe

dev alanı uçları bağlar ve yanlış aygıttan gelen paketlerin reddini sağlar; $epoch$ yeniden başlatma sonrası sayaç çakışmalarını önler; msg, tekdüzelik dayatır, $epoch$ başına tazelik sunar ve taşıma katmanı eşzamanlamasına yardım eder. Bu alanlar gizli değildir; açık taşınmaları bir zafiyet oluşturmaz, çünkü kabullenme yalnızca doğru anahtar altında üretilen kimlik doğrulama etiketinin doğrulanmasına bağlıdır. İlişik veriler (AAD) AEAD etiketiyle bütünlük korumasına sahiptir ve 15 baytlık $ver|dev|idx|msg|feat$ düzenini veya isteğe bağlı ts alanını ekleyen 23 baytlık uzatmayı kullanır. Burada ver protokol sürüm uyumluluğunu kodlar; dev uç eşleşmesini zorlar; idx hangi dilimin kullanıldığını gösterir; msg taşıma katmanı eşzamanlamasını destekler ve yeniden oynatmayı kontrol eder; $feat$ uygulamaya özgü politika bitlerini taşır. Mevcutsa isteğe bağlı ts alanı zaman damgası doğrulamasını etkinleştirir. AAD şifrelenmez; ancak AEAD etiketiyle kriptografik olarak bağlandığından, herhangi bir bit değişikliği doğrulamayı başarısız kılar ($INT - CTXT$) (McGrew, 2008). Politika B seçildiğinde ($epoch$, msg) de paket başına alan ayrımı için KDF'ye girer; Politika A'da girmez.

Dilim havuzu (POOL)

S_{ROOT} üzerindeki dilimler, 256 bitlik dizge üzerinde 128 bitlik kayan pencerelerdir; başlangıç ofsetleri s∈[1,129] kümesinden seçilir ve seçim sırası tekdüze değildir (ör. beşinci anahtar, s=100 iken [100..227] penceresini kullanabilir). AAD içindeki idx alanı, yalnızca hangi pencerenin kullanıldığını gösteren bir etikettir. Pencere içeriği gizli kaldığından idx'in açık olması bir zafiyet oluşturmaz. Aynı idx yeniden kullanılsa dahi, nonce tekilliği sayesinde AEAD güvenliği korunur. Artık risk, olası anahtar sızıntılarının tek bir idx üzerinde yoğunlaşmasıdır. Bu risk, rastgele idx seçimi ve DFX ile azaltılır. POOL içindeki dilimler, S_{ROOT} üzerinde ardışık olmayan ofsetlerden seçilen 128 bitlik pencerelerdir (ör. [1..128], [100..227], …). Politika, sabit bir sıralamadan kaçınır ki bir gözlemci, açık idx değerlerini kökün kararlı alt aralıklarına eşleyemesin.

Başlatma

Başlatmada S_{ROOT}, dev_{id} ve $POOL[\,]$ BRAM'e yüklenir; ($epoch_{ctr}$, msg_{ctr}) ayarlanır; sır havuzunu ve seçili kod sayfalarını barındıran BRAM bölgeleri için DFX ile yer değiştirme/silme etkinleştirilir. Yeniden başlatmada nonce yeniden kullanımını önlemek için $epoch_{ctr}$ artırılır. Açılışta DFX denetleyicisi, sırlar ve seçili kod için iki BRAM bölgesi ayırır ve periyodik yer değiştirmeyi etkinleştirir.

Kararlı durum

Her ileti için nonce (dev_{id}, $epoch_{ctr}$, msg_{ctr}) ile oluşturulur; K_i, Politika A veya Politika B uyarınca (S_{ROOT}, dev_id, $POOL[idx][\ \|\ epoch\ \|\ msg]$) kullanılarak türetilir; $(C, tag) = ASCON-128a_{Enc}(K_i, nonce, AAD, M)$ hesaplanır ve kayıt iletilir. Alıcı, "Alıcı-tarafı Kabul Denetimleri" bölümündeki kontrolleri uygular ve yalnızca etiket doğrulamadan başarıyla geçerse kabul eder. Ardışık iletilerde aynı idx'in kullanılması, nonce'lar tekil olduğu sürece güvenliği düşürmez; Politika A'da ele geçirilen bir anahtar, yeniden anahtarlamaya kadar aynı (dev_{id}, idx) ile gönderilen paketleri etkiler. Politika B'de etki tek bir paketle sınırlıdır.

4.1.1. Alıcı tarafı kabul denetimleri

Bir paket, [L1]–[L9] koşullarının tümü sağlanırsa kabul edilir; aksi takdirde reddedilir. Çizelge 4.2, bahsi geçen koşulları listeler.

Çizelge 4.2. Alıcı denetimleri

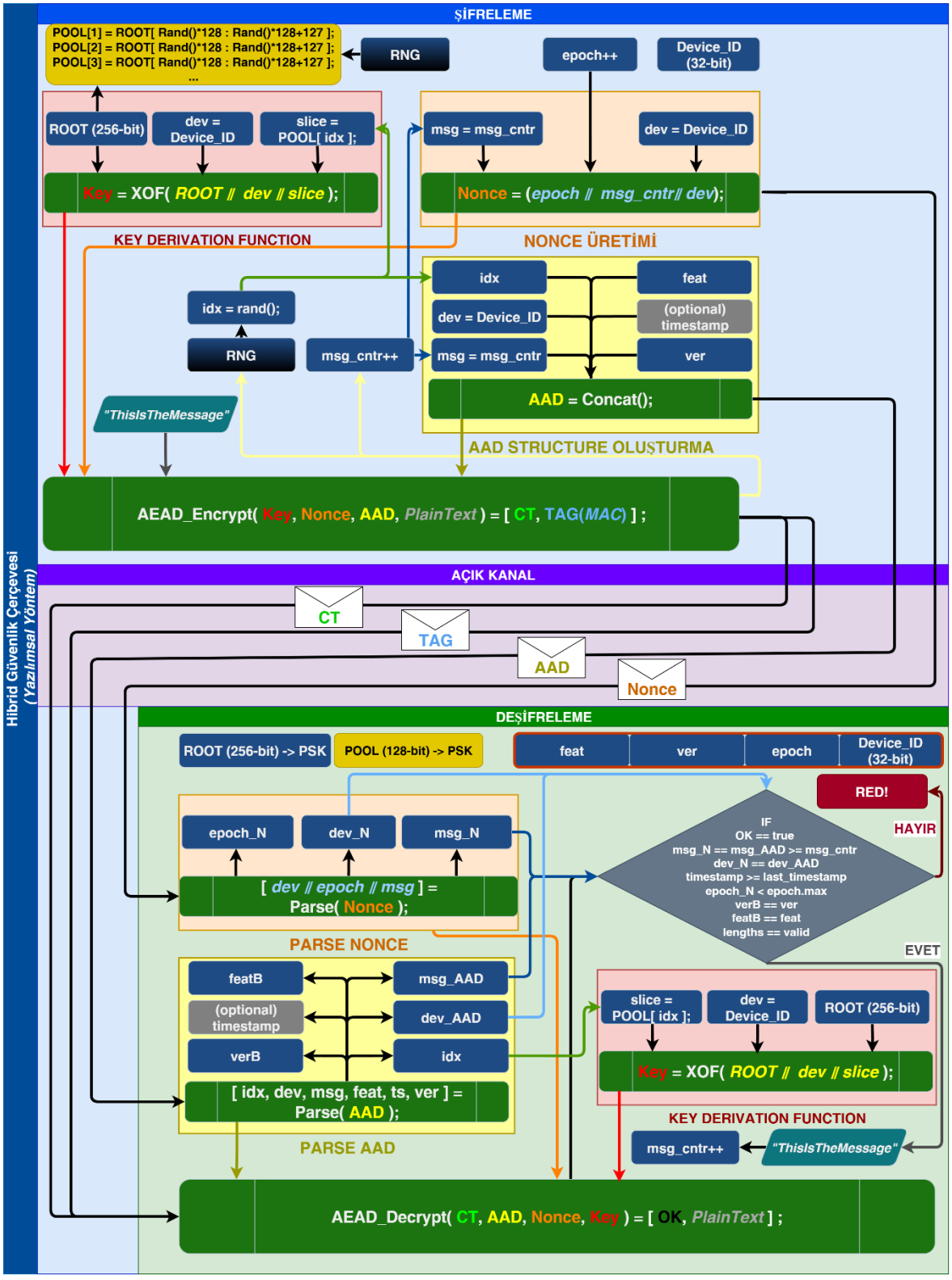
Adım	Koşul (başarısızlıkta reddet)
L1	|nonce| = 16, |tag| = 16, |AAD| ∈ {15, 23}
L2	idx < |POOL|; dev_AAD = dev_nonce
L3	msg_AAD = msg_nonce
L4	Politika bitleri: ver/feat ve AAD yerleşimi geçerli
L5	Yeniden oynatma karşıtı: (epoch, msg) taze (tekdüze ya da pencere)
L6	İsteğe bağlı zaman damgası: ts varsa, ts ≥ ts_last ya da pencere sınırı uygulanır
L7	K_i = ASCON-XOF128("KDF|ASCON|v1" ‖ S_ROOT ‖ dev_id ‖ POOL[idx])[0..127]
L8	(İsteğe bağlı) Paket başı alan ayrımı için KDF girdisine epoch ‖ msg ekle (Politika B)
L9	ASCON-128a_Dec(K_i, nonce, AAD, C, tag) başarılı

Güvenlik notu

dev/idx/msg/epoch tasarım gereği açıktır. Gizlilik K_i ile, bütünlük etiketle sağlanır (Krawczyk ve Eronen, 2010; Barker vd., 2011; Chen, 2009). idx veya dev sızması gizliliği zayıflatmaz. Çünkü S_{ROOT} ve $POOL[idx]$ bilinmez; $(epoch, msg)$ ise paket başına alan ayrımı sağlar. Politika A'da (dilim başına), türetilmiş bir anahtarın ele geçirilmesi, yeniden anahtarlama ya da dilim döndürme yapılana kadar aynı (dev_{id}, idx) ile gönderilen paketleri etkiler. Politika B'de (paket başına), $(epoch, msg)$'in KDF'ye dâhil edilmesi, etkiyi yalnızca ilgili pakete sınırlar. Her iki politika için de anahtar başına nonce tekilliği gerekir.

4.1.2. İşletim prosedürleri

Şifreli metinler ve etiketler ASCON-128a ile üretilir. Kabullenme yalnızca etiket doğrulamasınadır; açık metin işaretleyicileri kullanılmaz. Etiket sahteciliği olasılığı yaklaşık 2^{-128}'dir; uygulama düzeyi zaman yan kanallarını azaltmak için etiket karşılaştırması sabit zamanda gerçekleştirilir (Dobraunig vd., 2021). Hasar sınırlaması, "KDF Politikaları" alt bölümünde tanımlanan Politika A/B'ye göre yürütülür. Şekil 4.1, hem sır havuzunun hem de seçili kod sayfalarının DFX destekli BRAM yer değişimi sırasında yürütülen şifreleme algoritmasının KDF girdilerini, Nonce/AAD oluşturma sürecini ve alıcı kontrollerini özetler.



Şekil 4.1. HGÇ–ASCON algoritma akış diyagramı. KDF (ASCON-XOF), Nonce/AAD, ASCON-128a

DFX, kriptografik anahtar entropisini artırmaz; bunun yerine kazıma/probla okuma saldırılarına karşı saldırganın fiziksel arama uzayını büyütür ve sızıntı şablonlarını kararsızlaştırır. Bu etki, kriptografik anahtar uzayından bağımsızdır. Havuz boyutu ve dizin

seçimi parametrelerdir: Daha büyük havuz, aynı dilimin kazara yeniden kullanım olasılığını azaltır; idx AAD içinde açıktır, dilim içeriği ise gizli kalır. DFX yer değişimi ve BRAM silme, fiziksel arama uzayını büyütür; kriptografik anahtar entropisi 128 bitlik anahtar uzunluğu ile tanımlanır. Bir dinleyici yalnızca idx etiketini ve AAD/nonce başlıklarını görür. idx→dilim eşlemesi BRAM içinde gizlidir ve DFX ile periyodik olarak yer değiştirildiğinden, açık kanalda idx'in gözlenmesi dilim içeriğinin veya S_{ROOT}'un çıkarımını mümkün kılmaz.

Parametre rolleri

tag, şifreli metni kimlik doğrular ve bütünlüğünü sınar; AAD açık iletilir fakat etiketle kriptografik olarak bağlandığından fark edilmeden değiştirilemez; KDF (ASCON-XOF), S_{ROOT}, dev_{id} ve gizli $POOL[idx]$'ten 128 bitlik oturum anahtarı türetir. Böylece gizlilik (IND-CPA) ve bütünlük (INT-CTXT) sağlanır. Hasar sınırlaması, "KDF Politikaları" alt bölümündeki Politika A/B'ye göre yürütülür. Her paket için oturum anahtarı K_i, ASCON-XOF128 ile Politika A veya Politika B'ye göre türetilir *(McGrew, 2008).*

Şifreleme

Düz metin M ve başlık alanları *(idx, epoch_ctr, msg_ctr, ver, feat[, ts])* verildiğinde:

(i) $POOL[idx]$ ve dev_id, BRAM'den okunur;
(ii) K_i, yukarıdaki gibi ASCON-XOF128 ile türetilir;
(iii) nonce ve AAD oluşturulur;
(iv) $(C, tag) = ASCON-128a_{Enc}(K_i, nonce, AAD, M)$ hesaplanır;
(v) tek satırlık UART çerçevesi yayılır:

$$PKT;\ ver = 1;\ nonce_hex = \cdots;\ aad_hex = \cdots;\ ct_hex = \cdots;\ tag_hex = \cdots$$

Şifre çözme

(i) nonce/AAD/tag'in uzunluk ve yerleşimini sınar;
(ii) idx < |POOL| ve AAD'deki dev ile nonce'taki dev'in eşit olduğunu doğrular;
(iii) $(epoch_ctr, msg_ctr)$ üzerinde yeniden oynatma karşıtını uygular;

(iv) K_i'yi ASCON-XOF128 ile (S_{ROOT}, dev_id, $POOL[idx]$) kullanarak türetir;
(v) yalnızca $ASCON-128a_{Dec}(K_i, nonce, AAD, C, tag)$ başarı döndürürse M'yi teslim eder ve yeniden oynatma penceresini ilerletir. Başarısız doğrulamalar günlüğe alınır ve (epoch, msg) penceresi ilerletilmez; kabul edilen paketlerde pencere ilerletilir.

Alan anlamları ve güvenlik gerekçesi. Paket, sabit bir kamuya açık alan kümesini (Nonce, AAD) gösterir ve gizli öğeleri (key, ROOT, slice) saklar. Aşağıda, şekille tutarlı her simgenin kapsam/boyutu, kaynağı ve kesin güvenlik/doğrulama rolü Çizelge 4.3'te özetlenmiştir *(…_N "Nonce'tan", …_AAD "AAD'den" anlamındadır)*.

Çizelge 4.3. Alan semantiği ve güvenlik gerekçesi (Şekil 4.1. HGÇ–ASCON akış diyagramındaki sembollerin açıklamaları)

Simge	Kapsam / Boyut	Kaynak / Tanım	Rol ve Güvenlik Gerekçesi
S_{ROOT} (ROOT)	Gizli / 256 b	Aygıt-içi kök sır	Yalnız KDF girdisi; aygıttan çıkmaz. Kaba kuvvet ≈ 2^{256}; etkili sistem güvenliği 128 bit AEAD/etiket ile sınırlıdır.
$POOL[idx]$ (slice)	Gizli / 128 b	S_{ROOT} üzerinde ardışık olmayan ofsetlerden 128 b pencere	KDF girdisini çeşitlendirir. Pencereler üst üste binebilir; *idx* tekrarları (epoch, msg) KDF'ye dahilse paket tekilliğini azaltmaz.
idx	Açık / (politikaya göre) 8 b	RNG/CSPRNG veya politika sayacı	Havuzdaki dilimi seçer. Sızıntısı zararsızdır: S_{ROOT} bilinmeden dilim değeri çıkarılamaz.
dev, dev_N, dev_{AAD}	Açık / 32 b	Aygıt tanımlayıcısı	Paketleri aygıta bağlar; $dev_N = dev_{AAD}$ olmalıdır. Kimlik doğrulama etikete dayanır, *dev*'in gizliliğine değil.
$epoch$, $epoch_N$	Açık / 32 b	Tekdüze epoch sayacı	Yeniden oynatma kovası. Yeniden başlatmada artır veya son durumu kalıcı tut.
msg, msg_N, msg_{AAD}, msg_{cntr}	Açık / 64 b	Tekdüze ileti sayacı	Epoch başına tazelik; dev ve epoch ile birlikte Nonce'u oluşturur. $msg_N = msg_{AAD}$ olmalı; msg_cntr yerel durumdur.
ts (isteğe bağlı)	Açık / 64 b	Politika isterse zaman damgası	Günlükleme/pencereleme için opsiyoneldir; (epoch, msg) varken güvenlik için gerekli değildir.
ver, $verB$	Açık / 8 b	Protokol sürümü; verB alıcı politikası	Uzlaşma/uyumluluk. Uyumsuzluk ⇒ reddet.
$feat$, $featB$	Açık / 8 b (bit maskesi)	Özellik/politika bitleri; featB alıcı politikası	Yetenek kapılamayı sağlar (ör. uzun AAD). Gizlilik-kritik değildir; bütünlük etikettedir.
$Nonce$	Açık / 128 b	pack128(dev, epoch, msg)	AEAD Nonce'u. Anahtar başına tekillik zorunlu; tekdüze (epoch, msg) ile sağlanır.
AAD (15/23 B)	Açık	ver	dev
K_i	Gizli / 128 b	XOF(ROOT ‖ slice ‖ dev [‖ epoch ‖ msg])	Paket başı anahtar. Tek bir K_i'nin sızması, (epoch, msg) alan ayrımıyla diğer paketlere yardımcı olmaz.
CT, TAG	Gizli çıktı / 128 b etiket	AEAD (ASCON-128a)	IND-CPA + INT-CTXT sağlar. Sahtecilik olasılığı ≤ 2^{-128}.

Şifreleme algoritması $HGC_{Encrypt}(M, idx, epoch, msg, ver, feat[, ts])$

1. KDF: K←ASCON-XOF128("KDF|ASCON|v1"‖S_{ROOT}‖dev_{id}‖ POOL[idx])[0..127]

2. (Politika B – isteğe bağlı): KDF girdisine (epoch ‖ msg) ekle, sonra kısalt

3. Nonce oluştur: nonce ← pack128(dev_id, epoch, msg)

4. AAD oluştur: AAD ← ver | dev | idx | msg | feat [| ts]

5. Şifrele: (C, tag) ← ASCON-128a_Enc(K, nonce, AAD, M)

6. Döndür: "PKT;ver=1;nonce_hex=...;aad_hex=...;ct_hex=...;tag_hex=..."

Deşifreleme algoritması ($HGC_{Decrypt}(PKT)$)

1. Ayrıştır & uzunlukları kontrol et: `parse nonce, AAD, C, tag; require

2. Tutarlılık & anti-replay: `require idx <

3. KDF: K←ASCON-XOF128("KDF|ASCON|v1"‖S_{ROOT}‖dev_{id}‖ POOL[idx])[0..127]

4. Doğrula & çöz: (ok, M) ← ASCON-128a_Dec(K, nonce, AAD, C, tag)

5. Döndür: M if ok else Reject

İsteğe bağlı hazırlık (provisioning)

Bitstream koruması altında bir seed/salt kurulabilir; böylece her iki uç da kablo üzerinde statik anahtar göndermeden aynı $POOL[\,]$ dizisini belirleyici biçimde türetir. Hazırlık iletileri gönderildiğinde gizlilik ve bütünlük için ASCON-128a kullanılır; düzenek, değerlendirilen tehdit modelinin kapsamı dışında olup tekrarlanabilirliği desteklemek için burada özetlenmiştir (Krawczyk ve Eronen, 2010; Barker vd., 2011; Chen, 2009).

4.2. Deney Düzeneği

Tüm ölçümler donanım üzerinde yapılmıştır; birincil platform Xilinx Zynq-7020 *(Snickerdoodle Black)* olup Vivado/Vitis 2020.2 ile yapılandırılmıştır.

Yazılım geliştirme ortamı

Vivado 2020.2 (DFX/PCAP akışı, kısmi bitstream yönetimi) ve Vitis 2020.2 (derleme/koşum) kullanılmıştır. Zaman ölçümleri Zynq Processing System üzerinde ARM Global Timer ve PMU çevrim sayaçları ile alınmıştır; şifreleme/çözme akışı ASCON-128a ve ASCON-XOF128 KDF çağrılarını içeren C tabanlı uygulama olarak koşturulmuştur. Günlükler (UART satırları) toplanmıştır.

Donanım özeti (FPGA — Snickerdoodle Black / Zynq-7020)

• Çip: Xilinx Zynq-7000 SoC (XC7Z020)
• FPGA mantıksal kaynakları: ~85k logic-cell sınıfı
• DSP blokları: 220
• Dahili BRAM: ≈630 KB (140 × 36 Kb = 5 040 Kb ≈ 630 KB)
• İşlemci: Çift çekirdekli ARM Cortex-A9 (867 MHz)
• Kullanıcı I/O: 100+
• Bağlantı: Wi-Fi ve Bluetooth 4.0 (kart sürümüne bağlı)

Donanım testleri Snickerdoodle Black (Zynq-7020) üzerinde yürütülmüş; Dynamic Function eXchange *(DFX/PCAP)* ile AXI-haritalı BRAM bölgelerinde dinamik yer değiştirme *(swap/scrub)* uygulanmıştır. Xilinx Vivado 2020.2 ve Vitis 2020.2, bitstream şifreleme ve kısmi yeniden yapılandırma akışını desteklemiştir. Uygulama derleme/koşumları Vitis üzerinden gerçekleştirilmiştir. Arduino Due, alıcı/karşı-saldırı koşum takımı olarak Nonce/AAD alanlarını ayrıştırıp ASCON-XOF128 ile K_i türeterek etiket *(tag)* doğrulaması yapmış; ayrıca etiket bit-çevirme ve *(epoch, msg)* tekrar denemeleri ile anti-replay testi gerçekleştirilmiştir.

Donanım özeti (Arduino Due)

• Mikrodenetleyici: Atmel SAM3X8E (ARM Cortex-M3)
• Saat frekansı: 84 MHz
• Flash: 512 KB
• RAM: 96 KB (64 KB + 32 KB)
• Dijital I/O: 54 (12 PWM)
• Analog giriş: 12
• Bağlantı: USB ve seri haberleşme

Değerlendirilen şifreler, ASCON-128a ve hafif AEAD kümesini *(ACORN-128, TinyJAMBU-128, JAMBU–PRESENT-128)* kapsar (Dobraunig vd., 2021; Wu, 2016; Wu ve Huang, 2019; Wu ve Huang, 2016; Bogdanov vd., 2007). Tüm bu AEAD algoritmaları, hem standart ("sade") yapılandırmalarında hem de önerilen Hibrit Güvenlik Çerçevesi (HGÇ) katmanı eklenmiş olarak ayrı ayrı değerlendirilmiştir. Bu sayede HGÇ'nin getirdiği ek maliyet her bir algoritma için doğrudan ölçülmüştür. Gizli veriler ve seçili kod sayfaları, kısmi yeniden yapılandırılabilir bölmeler içinde AXI ile eşlenmiş iki BRAM bölgesinde tutularak; Dynamic Function eXchange (DFX/PCAP) testler sırasında bu bölgeleri periyodik olarak yer değiştirip, temizlemiştir (Xilinx, 2018; Vipin ve Fahmy, 2018). Çalışma süresi ölçümleri, Zynq Processing System üzerinde ARM Global Timer ve PMU çevrim sayaçları ullanılarak alınmıştır (ARM Ltd., 2010). Sonuç olarak, takip eden UART çıktısı üretilmiştir:
$[time]\ us_total|us_kdf|us_aead|us_glue,$
$[pmu]\ cyc_total|cyc_kdf|cyc_aead, PKT; ver = 1; \dots;$
bu satırlar günlük (log) olarak saklanmıştır. Burada us_{glue}, KDF/AEAD dışındaki kriptografik olmayan ek yükü (paket çerçeveleme/ayrıştırma, UART G/Ç, tampon kopyaları) kapsar. PMU sayaçları *(cyc_*)* ise buna karşılık gelecek biçimde raporlanmıştır. Giriş hattındaki anlık akım, hassas bir şönt üzerinden ve Digilent Analog Discovery 3'ün osiloskop kipinde ölçüldü; paket başına enerji, yazılım penceresi AEAD çağrısına hizalanarak

$$P(t) = V_{in} \cdot I(t) \tag{4.4}$$

eşitliğine entegre edilmesiyle elde edildi (Eş. 4.4). Uygun olduğunda, ana güç hattında PicoTest M3511A 6½ haneli multimetre ile kararlı yük altında akım ölçülerek ortalama akım

doğrulaması yapıldı. Paket başına enerji, osiloskop integrasyonundan türetildi. Bir Arduino Due, alıcı/saldırı kancası olarak kullanıldı: Nonce ve AAD alanlarını *(dev, idx, epoch, msg)* ayrıştırdı; aşağıdaki şekilde anahtarı türetti; $ascon_{aead128a} - decrypt$ fonksiyonunu çağırdı ve etiketleri sabit sürede karşılaştırdı. Negatif testler, etiket bitlerinin çevrilmesini ve eski *(epoch, msg)* ikililerinin yeniden oynatılmasını içererek yeniden-oynatma (anti-replay) denetimini doğruladı (Eş. 4.5).

$$K_{\mathrm{i}} = ASCON - XOF128(\text{“}KDF\text{”} \parallel S_ROOT \parallel dev \parallel POOL[idx])[0..127] \qquad (4.5)$$

Doğruluk

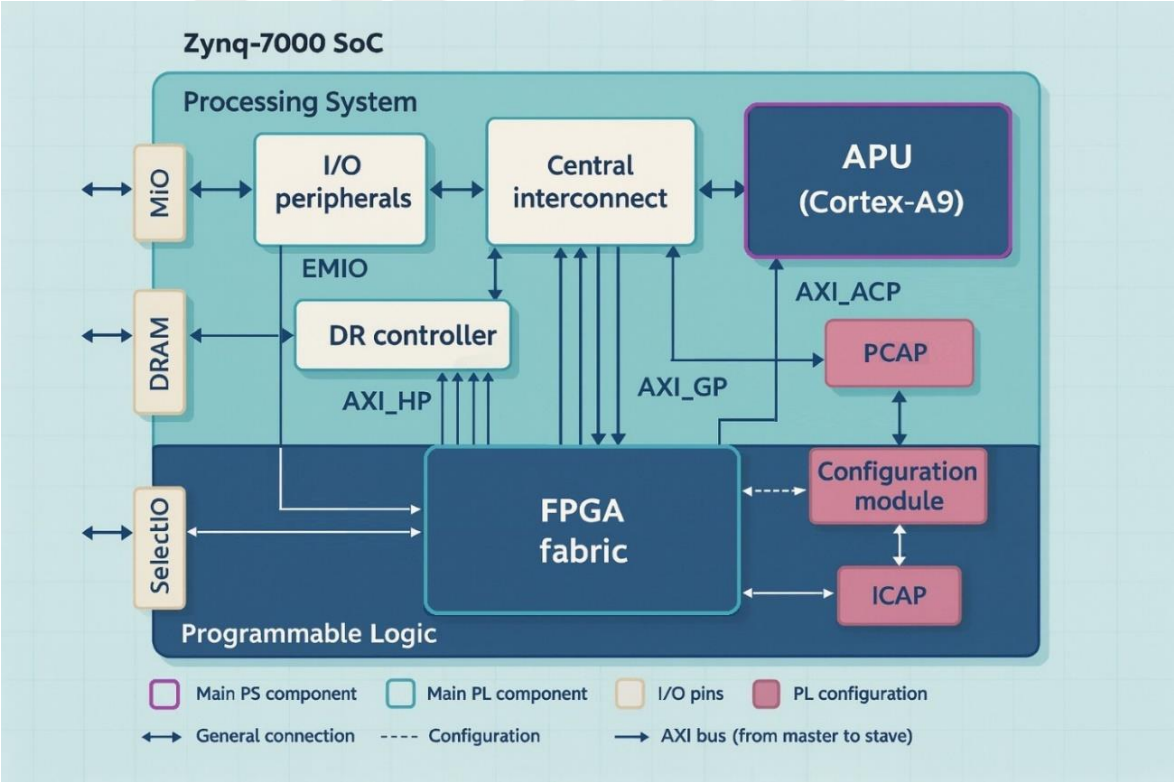
ASCON, ACORN, TinyJAMBU ve JAMBU için resmi test vektörlerine karşı uygulamalar çapraz kontrol edildi. Şifreleme ve doğrulama çıktıları bit düzeyinde birebir uyuştu. Bu çalışmada hem 2 BRAM (işlevsel doğrulama hedefi) hem 4 BRAM (stres/ölçüm) düzenekleri gerçekleştirilmiştir. DFX'e ilişkin ölçülen pencereler 4 BRAM üzerinde alınmış, 2 BRAM değerleri aynı koşullarda doğrusal ölçeklenerek verilmiştir. Dört veya sekiz bölge kullanmak da doğrudan uygulanabilir olup kazıma/probla taramaya karşı fiziksel arama uzayını daha da genişletir. Kriptografik entropiyi değiştirmeden olası saldırılara karşı rastgeleliği artırır. Ayrıca, çözme ve saldırı analizi için Arduino Due kullanıldı. Enerji tüketimi analizleri donanım ortamında yapıldı. FPGA ve Arduino Due üzerindeki ölçümler, algoritmanın gerçek dünya senaryolarındaki enerji verimliliğini değerlendirmiştir. Tüm bileşenler, algoritmanın etkinliğini ve geleneksel yöntemlerle olan performans farklarını ölçecek şekilde dikkatle yapılandırıldı. Her araç, belirli bir performans metriğine veya saldırı türüne odaklanarak kapsamlı bir analiz sağladı. Böylece algoritmanın donanım ortamındaki performansı ile güvenliği ayrıntılı biçimde incelendi. Deney düzeneği ayrıca ticari gömülü sistemlerdeki pratik uygulama zorluklarını dikkate alarak, önceki çalışmalarda vurgulanan akademi–sanayi uyumsuzluğunu ele almıştır (Vipin ve Fahmy, 2018; Stolz vd., 2021).

4.3. Donanım ve Yazılım Tasarımı

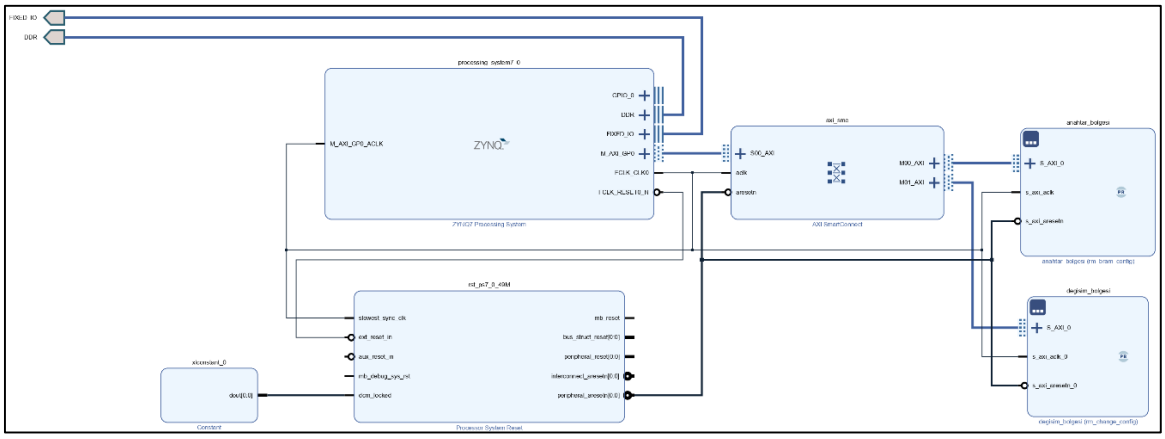
Platformun bütünsel görünmesi ve DFX mekanizmasının anlaşılabilirliği için, Zynq-7000 SoC'de PS–PL veri yolları ile konfigürasyon yolları (PCAP/ICAP) Şekil 4.2'de özetlenmiştir.

4.3.1. Platform ve üst düzey tasarım

Güvenlik hedefi iki yönlüdür: (i) kodu ve uzun ömürlü sırları kolayca kazınabilir ve dışarıdan gözlenebilir belleklerin dışında tutmak; (ii) hassas verinin fiziksel konumunu Dynamic Function eXchange (DFX) ile "hareketli hedef" hâline getirmek. Prototip, Zynq-7020 üzerinde PCAP yoluyla gerçekleştirilmiştir. Yöntem, gerekirse MicroBlaze'e de taşınabilir *(Sanal çekirdek-[softcore] işlemcinin PL içindeki BRAM talimat belleği DFX ile yer değiştirilebilir)*. Sert çekirdek *(hardcore)* ARM (PS), denetimi güvenli tarafta tutarken, PL'deki iki yeniden-yapılandırılabilir BRAM, şifreli durumu saklar (Çizelge 4.6, Şekil 4.3 ve Şekil 4.4.). DFX süre/enerji pencereleri, aksi not edilmedikçe 4 RM konfigürasyonunda ölçülmüş değerlerdir. 2 RM için raporlanan sayılar, aynı koşullardaki tek-takas penceresinin yaklaşık yarısı olacak şekilde ölçeklenmiş karşılıklardır.

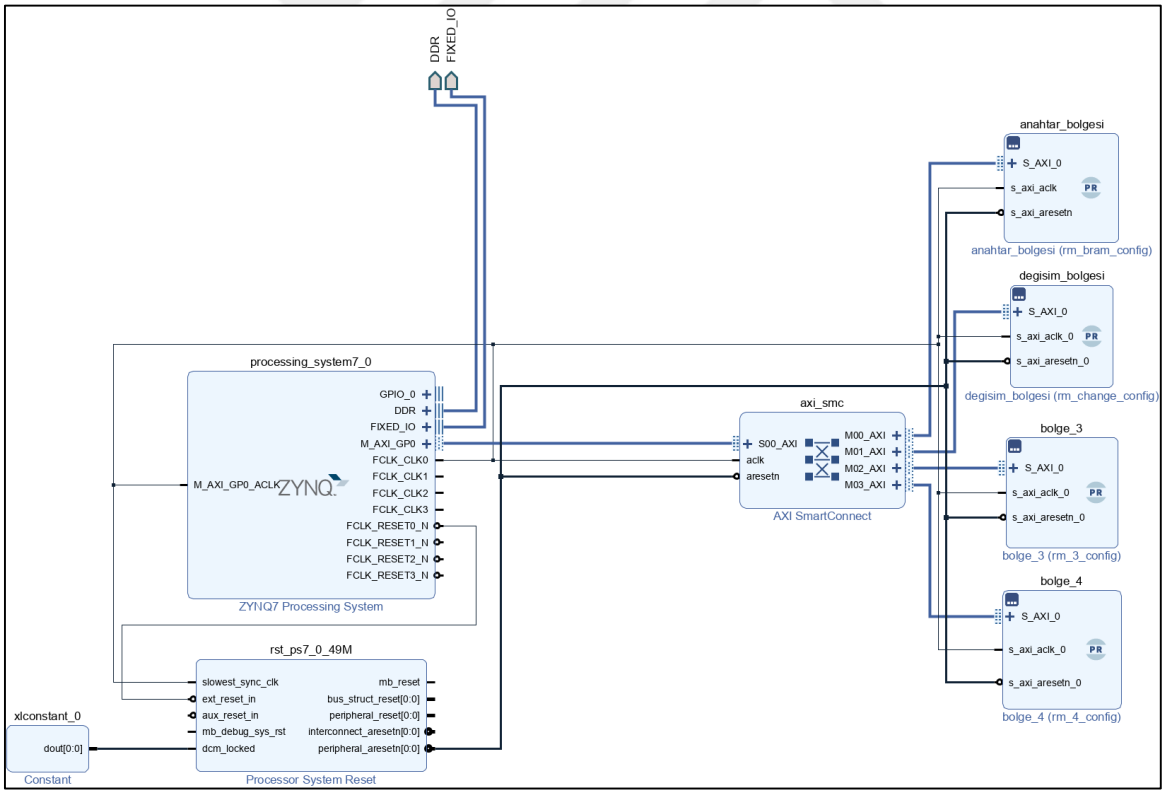


Şekil 4.2. Zynq-7000 SoC üst düzey (top level) blok diyagramı: PS–PL arabirimleri ve PCAP/ICAP konfigürasyon yolları gösterilmektedir (Kerling, 2019)



Şekil 4.3. Zynq-7020 üzerinde DFX kullanarak BRAM içeriklerini değiş-tokuş eden önerilen donanım mimarisi

Not: İki yeniden yapılandırılabilir bölge mevcuttur. Kaynaklar, AXI üzerinden SmartConnect ile ARM işlemci sistemine bağlanır. DFX BRAM içeriklerini değiştirmek için kullanılır.



Şekil 4.4. Dört yeniden yapılandırılabilir bölge ile yapılan tasarım (adres haritası 0x4000_0000; 0x4100_0000; 0x4200_0000; 0x4300_0000)

Önerilen çerçeve, Xilinx Zynq-7020 üzerinde Vivado/Vitis 2020.2 ile uygulanmıştır (AMD, 2023; AMD 2025). Mevcut tasarım MicroBlaze kullanmaz; tüm yazılım Zynq Processing System (çift çekirdekli Cortex-A9) üzerinde çalışır. PL'deki sırlar ve seçilmiş BRAM-

yerleşimli kod sayfaları DFX ile yer değiştirir. Programlanabilir mantıkta (PL) *Pblock* kısıtları ile çoklu kısmi yeniden yapılandırılabilir bölge (RM) tanımlanır (Şekil 4.6, Şekil 4.7 ve Şekil 4.8). Statik tasarım Zynq PS, bir AXI SmartConnect ve *Pblock* ile sınırlandırılmış birden fazla kısmi yeniden yapılandırılabilir modülden (RM) oluşur. Her bir RM bir AXI BRAM Controller ve bir Block Memory Generator örneği içerir (Şekil 4.9). Zynq-7020'nin programlanabilir mantığı, saat bölgesi (clock region) ızgarası ve sütun tabanlı kaynak dizilimi (CLB/DSP/BRAM sütunları) üzerine kurulur. Kısmi yeniden yapılandırma (DFX) süreçlerinde güvenli sınırlar elde etmek için, her yeniden yapılandırılabilir modülün (RM) bir Pblock aracılığıyla saat bölgesi kenarlarına ve BRAM sütunlarına karo hizalı (tile-aligned) olarak konumlandırılması önerilir. Bu yaklaşım, kısmi bit akışı oluşturulurken yönlendirmenin (routing) belirlenen sınırları aşmasını ("sınır kesimi") engelleyerek Vivado DFX tasarım kurallarına uyumu güvence altına alır. Statik alan; Zynq PS, AXI SmartConnect ve yeniden yapılandırma sırasında değişmeyecek çevresel IP'leri barındırır. RM'lerin her biri ise kendi AXI BRAM Controller + BRAM ikilisini içerir; BRAM'i verinin fiziksel yerini değiştirerek taşıyan asıl obje budur.

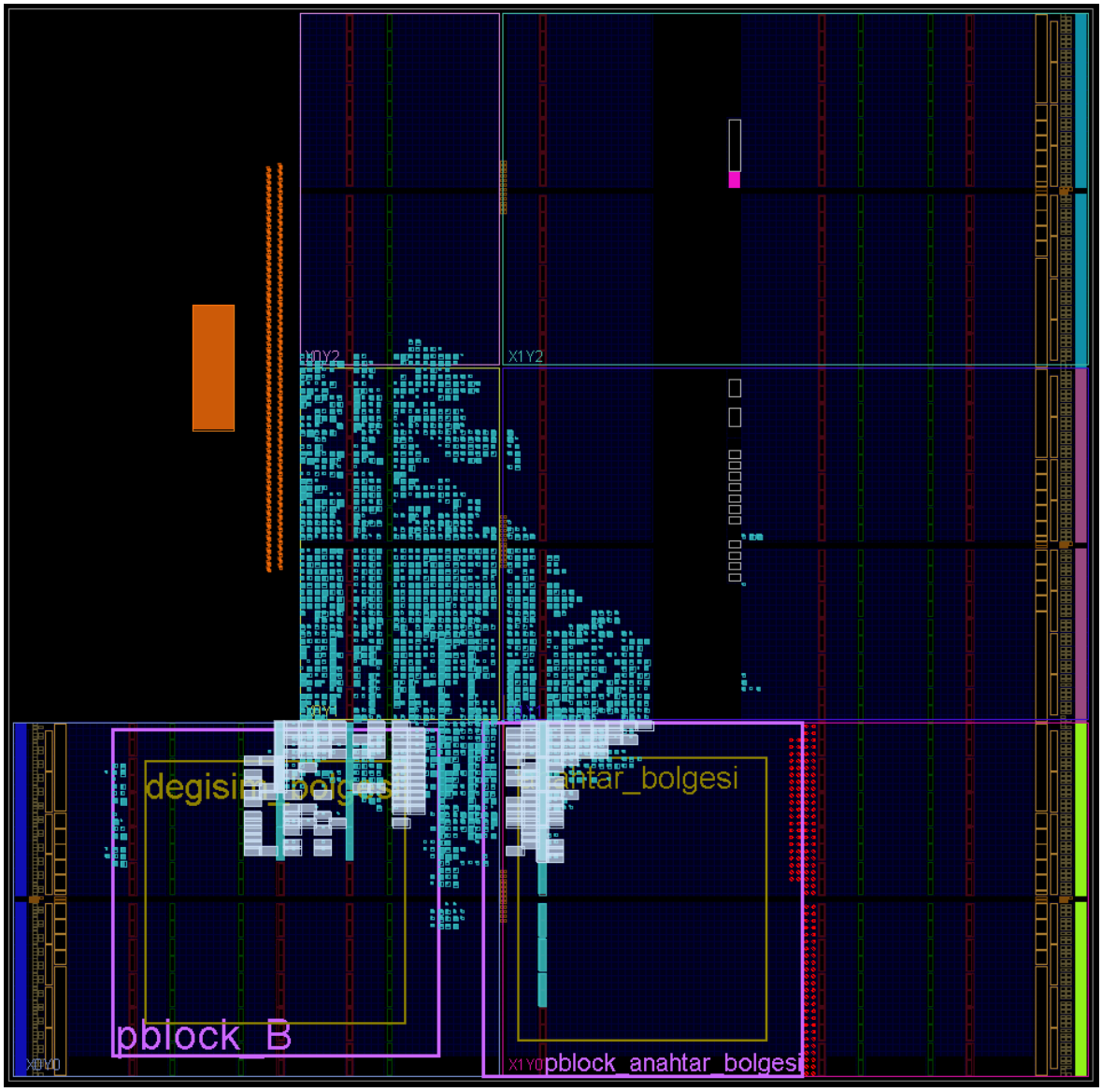
Bu çalışmada iki RM ayrı Pblock'lar ile tanımlanmıştır. Ekran görüntüsünde "pblock_anahtar_bolgesi / anahtar_bolgesi" (anahtar/kelime listesi deposu) ve "pblock_B / degisim_bolgesi" (sıradaki yedek bölge) olarak işaretlenmiştir. Her iki Pblock da en az bir tam BRAM sütununu ve ona komşu yönlendirme kaynağını kapsayacak şekilde boyutlandırılmış, I/O kolonları, saat tamponları ve PS sınırları gibi sabit makro bölgeler bilinçli biçimde hata vermeyecek şekilde gerekli önlemler alınmıştır *(SNAPPING_MODE = ON)*. RM sınırlarında, Vivado'nun reconfigurable partition pin (RP pin) mekanizması ile AXI sinyalleri statik alana taşınır; böylece kısmi yeniden yapılandırma sırasında statik yönlendirme donuk kalır. Bu yerleşim, DFX turunda aktif BRAM'in "anahtar bölgesinden" "değişim bölgesine" taşınmasına izin verir; bir sonraki turda rolleri tekrar değiştirerek fiziksel bellek hücrelerinin konumunu sürekli değişken kılar (moving-target).

DFX bölgelerinin saat bölgesi sınırlarına uygun yerleştirilmesi gerektiğinden, Zynq-7020'nin PL saat bölgelerine ayrılmış topolojisi Şekil 4.5'te verilmiştir; PS sol üstte yer alır.

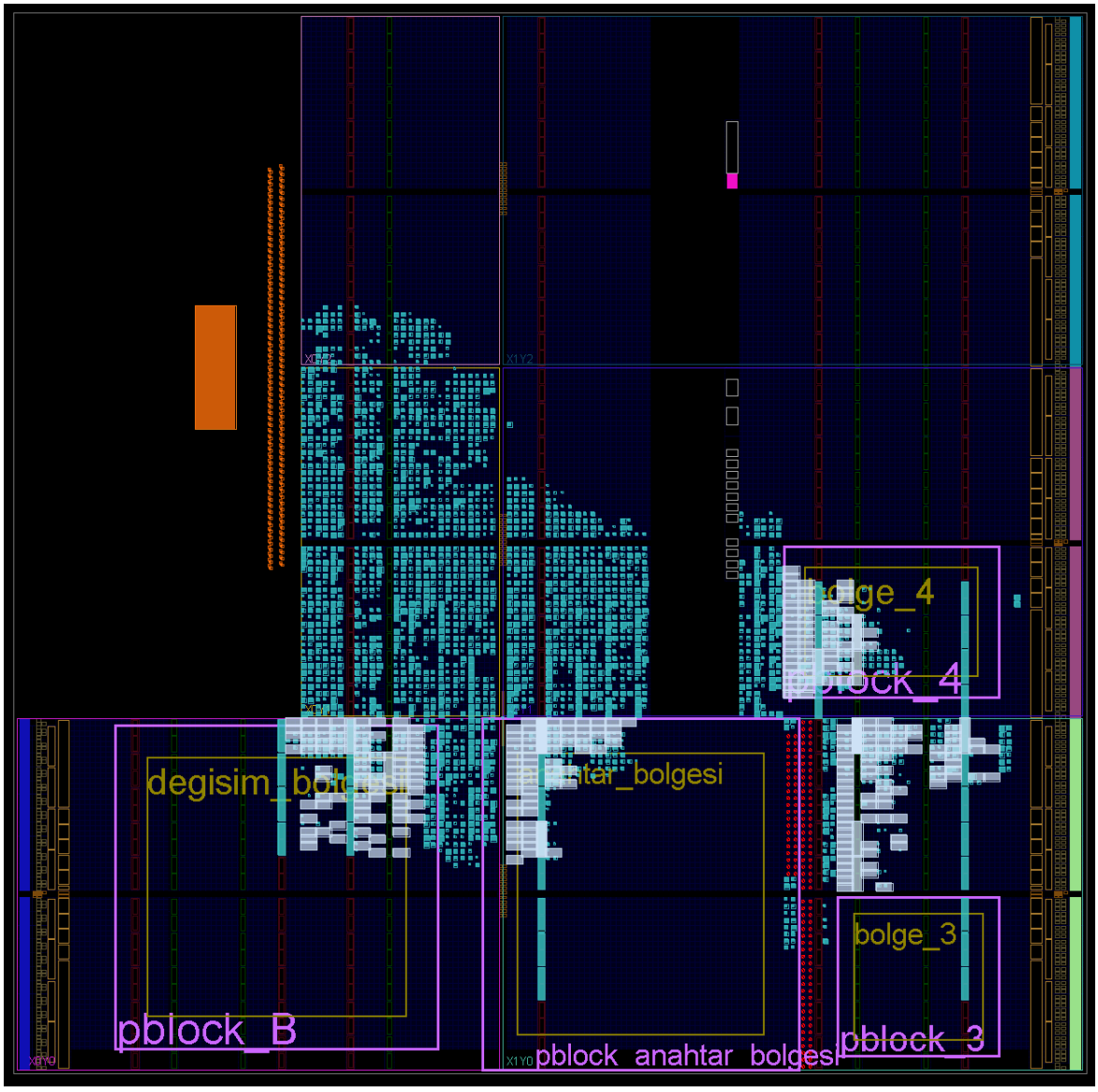


Şekil 4.5. Zynq-7020'nin altı saat bölgesine (clock region) ayrılmış PL topolojisi; sol üst bölgede PS yerleşimi bulunur. DFX bölge (RM) tanımları bu saat bölgesi sınırlarına göre yapılır (Kerling, 2019)

Zynq-7020'nin altı saat bölgesine ayrılmış PL topolojisi, BRAM sütunlarının bağımsız bloklar halinde gruplanması sayesinde RM'lerin tek saat bölgesi yüksekliğinde tutulmasına olanak tanır. Bu da kısmi bit-akışlarının küçük kalmasını ve yeniden yapılandırma gecikmesinin düşük olmasını sağlar. Toplam etki; (i) anahtar/kelime listesinin çip-üstünde kalması, (ii) içeriğin yalnızca şifreli olarak taşınması, (iii) fiziksel hücrelerin DFX ile sürekli yer değiştirmesi ve (iv) AXI erişiminin statik-dinamik sınırda kontrollü şekilde sonlandırılmasıdır. Bu mimari, veri yolu kazıma ve yan kanal korelasyonlarını hem taşıma katmanında hem de fiziksel yerleşimde zorlaştırır (AMD, 2023; AMD, 2025).

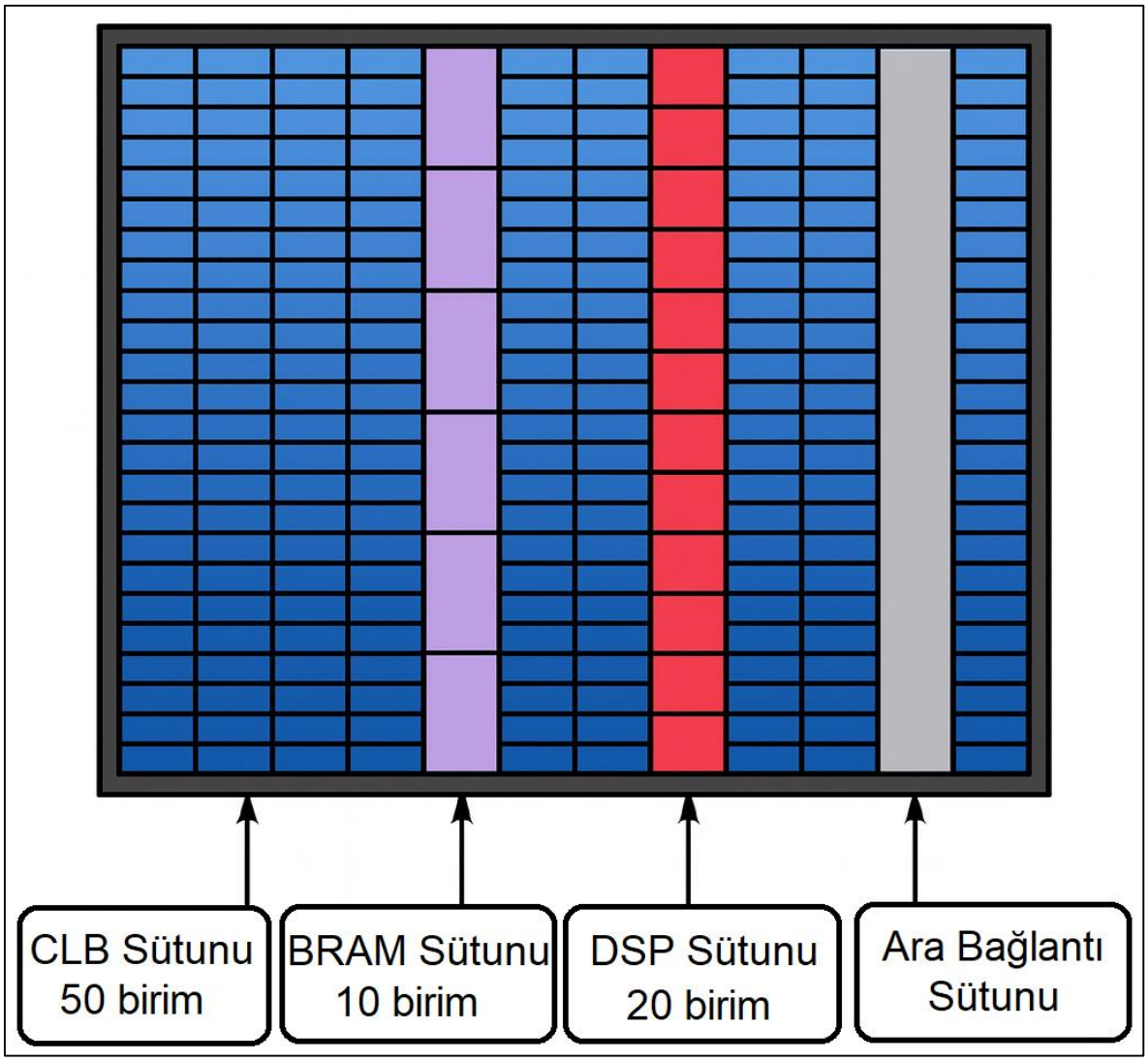


Şekil 4.6. Zynq-7020 PL üzerinde pblock sınırları ve iki RM (Pblock_Anahtar_Bolgesi / Pblock_B-Degisim_Bolgesi) için floorplan ekran görüntüsü; her RM içinde AXI BRAM Controller + BRAM yerleşimi



Şekil 4.7. Dört RM için floorplan ekran görüntüsü ve pblock sınırları; her RM içinde AXI BRAM Controller + BRAM yerleşimi (Bölüm 5'te raporlanan DFX süre/enerji ölçümleri bu yerleşim üzerindedir)

Statik kısım; Zynq PS, bir AXI SmartConnect ve Pblock'larla sınırlandırılmış iki kısmen yeniden yapılandırılabilir modülden oluşur. Her RM içinde bir AXI BRAM Controller ve bir Block Memory Generator örneği bulunmaktadır (Şekil 4.9). Bu adres haritalaması, Şekil 4.6'da gösterilen Pblock sınırlarıyla hizalıdır; böylece kısmi bit-akışları BRAM sütunlarını tam kapsayacak biçimde üretilir ve statik yönlendirme değişmeden kalır (Şekil 4.7) (AMD, 2025). Aynı DFX + taşıma anahtarı + yeniden paketleme politikaları, gerekirse MicroBlaze/ICAP varyantına da değiştirilmeden uygulanabilir; bu çalışmada gösterim, PS/PCAP üzerinde enerji/verim odaklı olarak tercih edilmiştir.



Şekil 4.8. Zynq-7020'nin PS/PL blok diyagramı ve sütun topolojisini gösteren şema (referans amaçlı genel görünüm) (Kerling, 2019)

4.3.2. Bellek ve konfigürasyon mimarisi

Her RM, çalışma zamanında DFX ile bağımsız biçimde yeniden yapılandırılabilir. Normal işletimde BRAM'lerden biri etkin kelime listesini ve oturum anahtarlarını tutarken, ikinci BRAM çevrim dışı hazırlanır. PCAP, kısmi bitstream'i yükleyerek iki BRAM'in rolünü değiştirir. Sürekli BRAM yer değiştirme, anahtar konumunu gizler ve yan kanal/saplama (probing) saldırılarını zorlaştırır (Çizelge 4.6). Şekil 4.9, her RM'nin iç yapısını göstermektedir: bir AXI BRAM Controller, bir Block Memory Generator örneğine bağlıdır. Birinci BRAM 0x4000_0000 – 0x4000_7FFF adreslerini, ikinci BRAM ise 0x4200_0000 – 0x4200_7FFF adreslerini kaplar, böylece her biri 32 KiB olmak üzere iki adet çip-üstü bellek elde edilir *(4 BRAM varyantında adresler 0x4000_0000; 0x4100_0000; 0x4200_0000; 0x4300_0000 olarak haritalanır).* Bu bellekler DFX ile sırayla etkinleştirilir, devreden

çıkarılır ve yeniden oluşturulur; anahtar ve kelime liste saklama bu şekilde güvence altına alınır. SmartConnect ve adresleme klasik AXI kurallarını izler; yenilik, yalnız içerik güncellemek yerine fiziksel hücrelerin konumunu sürekli değiştirmededir (Şekil 4.3, Şekil 4.4 ve Şekil 4.9).

Rotasyon başına yeniden paketleme ve bağımsızlık

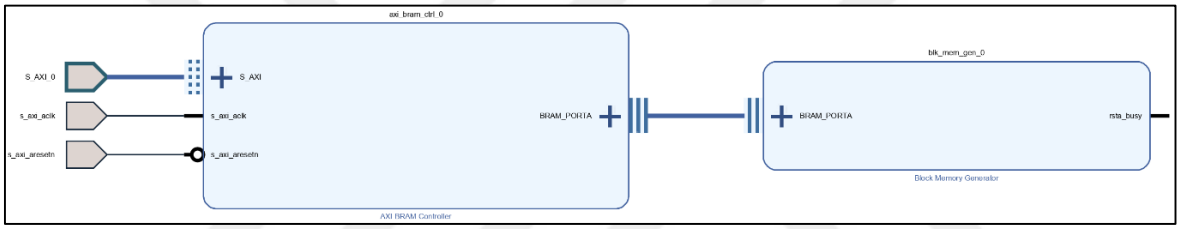
Her DFX turundan önce PS, taze bir taşıma anahtarı (K_{tr}^{r}) türetir ve aktif havuzdaki girdileri hedef RM'ye yazmadan önce yeniden paketler. Her bir giriş, kendi giriş kimliği (idx) ve rotasyon sayacı (rot_{ctr}) gibi ayırt ediciler AAD/nonce olarak kullanılarak ASCON-128a ile yeniden şifrelenir. Böylece aynı mantıksal içerik (ör. aynı sözlük dilimi veya kod sayfası) iki ardışık turda bile AXI hattında ve BRAM içinde istatistiksel olarak bağımsız şifreli metinlere karşılık gelir (Eş. 4.6). Tekrar eden desen sızıntıları bastırılır ve korelasyon tabanlı yan kanal riski azaltılır.

Kod sayfalarının taşınması

Seçilmiş BRAM-yerleşimli kod sayfalarının da aynı yeniden paketleme ve rotasyon disiplinine dahil edilmesi önerilir: açık metin kod yalnızca PS çekirdek yazmaçlarında anlık olarak bulunur; yeniden paketleme tamamlandıktan sonra yazmaçlar zeroize edilir ve DFX tetiklenerek kod sayfası hedef RM'ye taşınır. Mevcut deneylerde kod sayfaları açık metin taşınmış, şifreli/AEAD paketleme 4.3.4'te öneri olarak bırakılmıştır. Buna rağmen kodun açık hâli yalnızca PS yazmaçları ve kısa ömürlü RAM tamponunda anlık bulunur; rotasyon sonunda tamponlar zeroize edilir ve DFX ile sayfa hedef RM'ye aktarılır.

Zynq-7000 cihazlarda PS, PL'yi yazılım tarafından erişilen DEVCFG çevre birimi üzerinden PCAP ile programlar; PCAP tam/kısmi ve şifreli bitstream'leri destekler (AMD, 2023; AMD, 2025). PL içinde yüksek bant genişliği için ICAP ilkel bloğu da vardır; Zynq üzerinde ICAP kullanmak, PCAP sahipliğini ICAP'e devretmeyi ve ICAP'i özel mantık ya da bir DFX denetleyicisiyle sürmeyi gerektirir (AMD, 2025). Bu nedenle tüm DFX işlemleri için, güvenilir denetim düzlemini ARM PS içinde tutmak amacıyla PCAP tercih edilmiştir. Bu arayüzlerin temel farkları Çizelge 4.4'te özetlenmiştir. Zynq-7020'de mevcut bellek türlerinin kriptografik anahtar saklamaya uygunluğu Çizelge 4.5 (özellikler) ve Çizelge 4.6 (güvenlik odaklı seçim) altında karşılaştırılmıştır. Tasarım bilinçli olarak BRAM kullanır;

çünkü çip içindedir, PS/PL tarafından AXI üzerinden görülebilir ve DFX ile yer değiştirilebilir. Kalıcı eFUSE veya pille destekli BBRAM, Xilinx araçlarınca bitstream şifreleme için 256-bit AES anahtarını saklamakta kullanılır; ancak tek seferlik programlanabilir yapıları ve sınırlı kapasiteleri dinamik anahtar parametreleri (POOL gibi) için uygun değildir. Yonga dışı DDR ve QSPI bellekler yüksek kapasite sunar fakat veri yolu analizi ile prob'lanabilir; bu yüzden yalnızca gizli olmayan kod depolamak için ayrılır. Yonga içi OCM yalnızca PS tarafından erişilebilir ve DFX ile taşınamaz (Çizelge 4.5 ve Çizelge 4.6). Dört veya sekiz yeniden yapılandırılabilir BRAM bölgesi kullanmak, bölge sayısının artırarak kriptografik entropiyi değiştirmeden kazıma/problama için fiziksel arama uzayını genişletir. M BRAM ve her birinde K adres varsa, tek hücre konum belirsizliği $H_{loc} = \log^2(MK)$ olur (Xilinx, 2018; Vipin ve Fahmy, 2018).



Şekil 4.9. Yeniden yapılandırılabilir bir modülün (RM) iç görünümü (her RM, kodu ve gizli POOL'u tutan bir AXI BRAM Controller ve bir BRAM örneği içerir. DFX sırasında denetleyici ve bellek kaldırılır ve alternatif örnekle değiştirilir)

Çizelge 4.4. Kısmi yeniden yapılandırma için konfigürasyon erişim portlarının karşılaştırması (PCAP/ICAP/MCAP)

Arayüz	Cihaz Ailesi	Erişim Yolu	Açıklamalar
PCAP	Zynq-7000	PS (DEVCFG)	Tam/kısmi yeniden yapılandırma; şifreli bitstream desteği; bu tasarımda kullanılan konfigürasyon portu budur.
ICAP	Tüm seriler	PL logic	PL içinde yüksek bant genişliği; özel ICAP denetleyicisi gerektirir; bu tasarımda kullanılmaz.
MCAP	UltraScale(+)	PCIe uç noktası	Sadece UltraScale/Versal sınıfında mevcuttur; Zynq-7020 tarafından desteklenmez.

Xilinx cihazlarda üç temel konfigürasyon yolu bulunur. PCAP, Zynq-7000 ailesinde işlemci sistemi (PS) ile konfigürasyon motoru arasındaki bağlantıdır ve bu ailede tam konfigürasyon için tek arayüzdür. PCAP, şifreli ve kimlik doğrulamalı bitstream kabul eder; tam ve kısmi yeniden yapılandırmayı destekler ve DEVCFG çevre birimi üzerinden yazılımla sürülür.

Bununla birlikte, veri yolu AXI slave portu olduğundan aktarımı PS meşgul eder ve bant genişliği sınırlıdır. ICAP (Internal Configuration Access Port), programlanabilir mantık (PL) içinde yer alır ve özel mantık ile sürülerek yüksek bant genişlikli kısmi yeniden yapılandırma sağlar. ICAP, bitstream çözme (decryption) yapamaz; şifreli çerçeveler önce PCAP üzerinden geçirilmelidir. Ayrıca PCAP'ten ICAP'e geçiş, konfigürasyon motorunun güvenli biçimde durdurulmasını ve DEVCFG yazmaçlarındaki bazı denetim bitlerinin temizlenmesini gerektirir (AMD, 2025). Son olarak MCAP (Management Configuration Access Port), UltraScale/Versal cihazlarda PCIe üzerinden konfigürasyon veri yolunu dışa açar; Zynq-7000'de bulunmaz, bu nedenle mevcut uygulama için ilgili değildir. Bu tasarımda PCAP tercih edilmiştir; çünkü ARM PS ile doğal bütünleşme sunar ve Vivado DFX akışıyla uyumludur (AMD, 2023; AMD, 2025) (Çizelge 4.4).

Konu bütünlüğü adına, Çizelge 4.4 daha önceki bellek karşılaştırmasını genişleterek Zynq-7020'deki başlıca RAM/ROM kaynaklarını listeler. İşlemci sisteminde, her Cortex-A9 çekirdeği için ayrı 32 KiB komut ve veri önbelleği ile ortak 512 KiB L2 önbellek bulunur; bu önbellekler yazılımdan şeffaftır ve kriptografik anahtar saklamak için uygun değildir. 256 KiB OCM (On-Chip Memory), hızlı kod/veri için sıkı bağlı RAM olarak hizmet verir. Çift kanallı DDR3 (PS DDR), çok daha büyük uçucu depolama sunar; QSPI flash ise kalıcı birincil bitstream ve kullanıcı verisini tutar. Programlanabilir mantık tarafında, distributed RAM (LUT tabanlı) küçük FIFO/kayıtçılar için kullanılır; BRAM blokları 18–36 KiB, UltraRAM destekli cihazlarda 288 KiB bloklar sunar (XC7Z020'de URAM yoktur). Cihaz ayrıca, yerleşik bitstream çözme motorunun kullandığı AES anahtarını saklamak için 256-bit eFUSE dizisi ve pil destekli BBRAM içerir. Bu belleklerin erişim hakları ve uçuculuk özellikleri farklıdır; çip-içi, DFX ile yer değiştirilebilir ve secret-pool (sır havuzu) için yeterli kapasite sağlayan tek seçenek BRAM'dir (Çizelge 4.5 ve Çizelge 4.6).

Çizelge 4.5. Zynq-7020 üzerindeki bellek seçeneklerinin özellikleri ve kullanım gerekçesi

Bellek	Uçuculuk	Sahiplik	Erişim	Kapasite	Kullanım	Gerekçe
QSPI flash	Hayır	PS	PS / haricî	MB'ler	Boot bitstream / veri	Kalıcı; haricî erişilebilir; önyükleme imajları için kullanılır.
eFUSE / BBRAM	Hayır	PS	Boot kontrol	256 bit	Bitstream şifreleme anahtarı	OTP; liste için çok küçük; bitstream şifreleme için ayrılmıştır.

Çizelge 4.5. (devam) Zynq-7020 üzerindeki bellek seçeneklerinin özellikleri ve kullanım gerekçesi

OCM	Evet	PS	Yalnız PS	256 KiB	Yığın/kod	Hızlı çip-içi; sabit yer; FSBL sürecinde kullanılır.
BRAM (PL)	Evet	PL	PS/PL AXI	18–36 KiB/blok	Sır havuzu, kod penceresi	Çip-içi; DFX ile taşınabilir; AXI adresli; gizli veriler için seçilmiştir.
PS DDR3	Evet	PS	PS / haricî	MB'ler	Uygulama verisi	Haricî; veri yolu izlemeye açık; gizli olmayan yükler için uygundur.
UltraRAM	Evet	PL	Yalnız PL	≥288 KiB	Büyük tampon	XC7Z020'de mevcut değil (UltraScale özelliği).
Dağıtık RAM	Evet	PL	Yalnız PL	Birkaç KiB	Küçük tamponlar	LUT-RAM; çok küçük ve sabit.
L1/L2 önbellekler	Evet	PS	CPU	32–512 KiB	Önbellek	Donanımca yönetilir; doğrudan adreslenemez; anahtar saklamaya uygun değil.

Özetle, BRAM güvenlik/performans için yer değiştirilebilir, PCAP ise Zynq-7000 üzerinde şifreli DFX'i PS tarafından sürmeye olanak tanır (AMD, 2023; AMD, 2025). MCAP bu cihaz sınıfında yoktur; ICAP ise mevcut tehdit modelinde belirgin bir güvenlik kazancı olmadan PL mantık ve denetim karmaşıklığı ekler. Bu nedenle PCAP kullanımı, yeniden yapılandırmayı doğrudan PS'den kontrol ederek yazılımı basitleştirir; devcfg sürücüsü bitstream ayrıntılarını soyutlar ve çalışan sistemle eşzamanlılığı sağlar. Son olarak, eFUSE/BBRAM, PL bitstream'ini koruyan cihaza özgü AES anahtarı için ayrılmıştır; sık değişen sır havuzları (pool) için uygun depolar değildir. Her dönüşte yeniden anahtarlama (per-rotation re-keying), her taşınan pool'a *(veya kod penceresine)* taze şifreli metin kazandırır; tekrar eden desen sızıntılarını önleyerek yan kanal ve kazıma saldırılarını zorlaştırır. Seçilen yapılandırma, haricî arabirim eklemeden güvenliği en üst düzeye çıkarır (Çizelge 4.4 ve Çizelge 4.5).

4.3.3. Güvenlik, çalışma zamanı ve uygulama ayrıntıları

Saldırgan modeli şunları içerir: (i) paket veya kart üzerinde katman sökme/prob yerleştirme (DDR, QSPI, AXI üzerinde veri yolu dinleme); (ii) çip içi dizilerin invaziv okunması (Çizelge 4.6). Buna karşı şu politikalar izlenir: (a) uzun ömürlü anahtarlar haricî ortamlarda tutulmaz; (b) PL içindeki BRAM'de yalnızca şifreli pool slice'lar ve yardımcı durum bilgisi saklanır; (c) açık metin (cleartext) materyal birkaç çevrimden uzun olmayacak şekilde sadece PS çekirdek yazmaçlarında yaşar; (d) fiziksel BRAM örneği DFX yoluyla sürekli "sıfırlanıp yeniden kurulur" (Şekil 4.9); (e) Rotasyonlar arasında şifreli metin bağımsızlığı: K_{tr}^{r} ile yeniden paketlenen havuz/kod girdileri, *idx* ve rot_{ctr} temelli AAD/nonce sayesinde iki tur arasında istatistiksel olarak bağımsız kalır; AXI ve BRAM üzerinde tekrar eden desenler oluşmaz. Bu "hareketli hedef + şifreli taşıma" stratejisi, hem fiziksel kazıma hem de korelasyon/güç analizi saldırılarının eşiğini yükseltir.

Çalışma zamanında önce ASCON-XOF128 ile K_{tr}^{r} türetilir; ardından aktif RM'deki havuz/kod girdileri, giriş kimliği *(idx)* ve rotasyon sayacı (rot_{ctr}) bağlamı AAD/nonce olacak şekilde ASCON-128a altında hedef RM için yeniden paketlenir. Geçici açık metin tamponları zeroize edilir ve *PCAP/XDevCfg* ile kısmi bit-akışı yüklenerek hedef RM etkinleştirilir; önceki RM scrub edilerek pasifleştirilir ve roller çevrilir.

Çizelge 4.6. Güvenlik odaklı bellek seçimi (kazıma ve veri yolu izlemeye karşı)

Bellek	Çip-içi	DFX-taşınabilir	Notlar
PS DDR	Hayır	Hayır	Yüksek kapasite; veri yolu izlemeye açık; açık anahtar/pool tutulmaz.
QSPI	Hayır	Hayır	Önyükleme imajlarını tutar; açık sır saklanmaz.
OCM	Evet	Hayır	Çip-içi ve hızlı, ancak sabit ve yalnız PS erişimli; hareketli hedef savunmasına uygun değil.
eFUSE / BBRAM	Evet	Hayır	PL bitstream'ini çözen AES-256 anahtarı için ayrılmıştır; dinamik anahtarlar için kullanılmaz.
PL BRAM	Evet	Evet	Şifreli secret-pool ve gizli kod penceresi için seçilmiştir; DFX ile fiziksel olarak taşınır ve silinir.

Bu çalışmada MicroBlaze tabanlı bir varyant mümkün olmakla birlikte, güvenilir kontrol düzlemi ve DFX orkestrasyonu bilinçli olarak Zynq Processing System (PS) üzerinde tutulmuştur. Bu tercih, (i) PCAP ve onun olgun sürücü yığınıyla doğal entegrasyon, (ii) programlanabilir mantık (PL) kaynak yükünün azaltılması ve (iii) uzun ömürlü sırların yonga dışı belleklerden yalıtılması gerekçelerine dayanmaktadır (AMD, 2023; AMD, 2025). MicroBlaze seçeneğinde kısmi yeniden yapılandırma ICAP üzerinden sürülür ve komut/veri BRAM'leri DFX ile yer değiştirilebilir; ancak bu, ek PL mantığı ve kontrol karmaşıklığı doğurur. Zynq-7020'nin donanımsal Cortex-A9 PS'i, PCAP temelli kısmi yeniden yapılandırma için daha yalın bir yazılım yolu ve daha yüksek uygulama performansı sunar. Önerilen yöntemin "hareketli hedef" bellek yer değiştirme ve "yalnızca yazmaçta" açık metin ömrü ilkeleri korunduğunda, yan kanal ve fiziksel kazıma türü saldırılara karşı direnç PS tabanlı tasarımda da sağlanmaktadır (Vipin ve Fahmy, 2018; AMD, 2025).

Zynq-7000 / Vitis 2020.2 ortamında kısmi bitstream aktarımı için "XDevCfg" sürücüsü kullanılmaktadır: "XDcfg_CfgInitialize" ile ilklendirme yapılır, PCAP DMA aktarımı yapılandırılır, kısmi bitstream tamponu "XDcfg_Transfer" ile gönderilir, durum ve kesmeler "XDcfg_IsDmaBusy" ve "XDcfg_IntrGetStatus" ile izlenir/temizlenir.

UltraScale(+)/ZynqMP ailesinde benzer işlevler XilFPGA *(XFpga_PL_BitStream_Load)* üzerinden de yürütülebilse de, Zynq-7000 sınıfı için uygun yol XDevCfg/PCAP'tir (AMD, 2023; AMD, 2025). Önyükleme sırasında statik PL yapılandırması PCAP ile yüklenir; PL bitstream'inin şifresi, eFUSE veya BBRAM'de saklanan aygıta özgü AES anahtarıyla çözülür (AMD, 2023). Bitstream'i çözen bu cihaz-özgü AES anahtarı, çalışma zamanında paket başına ASCON-XOF128 ile türetilen anahtarlarla ilişkili değildir ve hiçbir koşulda pool/kod yeniden paketleme malzemesine karışmaz. Fakat ileri çalışmalar için bitstream'i her döngüde okuyup özgün yöntemle tekrar şifreleme işlemi, bitstream şifresinin kırılması ihtimaline karşı dinamik şifreleme ile ekstra dayanıklılık sağlar.

Çalışma zamanında paket başına anahtarlar ASCON-XOF128 ile *(S_{ROOT}, dev, POOL[idx], epoch, msg)* girdilerinden türetilir; iletimde ASCON-128a kimlik doğrulamalı şifreleme kullanılır (Dobraunig vd., 2021). AXI üzerinden yalnızca şifreli yük (ciphertext) ve etiket (tag) ile birlikte kamuya açık Nonce/AAD alanları taşınır; açık metin anahtar malzemesi PS yazmaçlarında anlık olarak işlenir ve kalıcı belleğe yazılmaz. Her mesajdan sonra PS, PCAP ile kısmi bitstream yükleyerek alternatif yeniden yapılandırılabilir bölgeyi (RM)

etkinleştirir; önceki bölge silinir (scrub) ve pasifleştirilir. Alıcı tarafta *(örneğin Arduino Due üzerinde)*, aynı *(idx, epoch, msg, dev)* alanlarıyla ASCON-XOF128 üzerinden paket anahtarı yeniden türetilir ve ASCON-128a doğrulama/çözme işlemi tamamlanır. Böylece her rotasyonda AXI'den yalnızca şifreli veri ve yeniden şifrelenmiş secret-pool girdileri geçer.

Kriptografik işlemler ve anahtar yönetimi PS üzerinde (Vitis/C) yürütülür. Sistem açılışında statik PL, PCAP ile güvenli biçimde yüklenir. Bitstream çözme anahtarı eFUSE/BBRAM'de tutulur. Bu anahtar, önerilen yöntemin dinamik (paket başına) anahtarlarıyla ilişkili değildir (AMD, 2023). İşlemci, SmartConnect ve AXI BRAM Controller bileşenlerini ilklendirir, RM1/RM2 için ilk kısmi bitstream'i PCAP ile yükler ve aktif BRAM bölgesine şifreli secret-pool ile yardımcı durum bilgilerini yerleştirir. Mesajlar ASCON-128a ile şifrelenir ve iletilir. İleri çalışmalar için verilen öneriyle, her DFX takası öncesinde ASCON-XOF128 ile taze bir taşıma anahtarı üretilir; aktif sırlar PS yazmaçlarında yeniden şifrelenerek hedef RM'nin BRAM'ine yazılır; ardından kısmi bitstream yüklenerek roller çevrilir ve yazmaçlar sıfırlanır. Sürekli BRAM yer değişimi, fiziksel problama ve korelasyon temelli yan kanal saldırılarını zorlaştırır.

$$K_{tr}^{r} = ASCON - XOF128(\text{“}TR\text{”} \parallel S_{ROOT} \parallel dev_{id} \parallel epoch_{r} \parallel RM_{id} \parallel rot_{ctr})[0..127] \quad (4.6)$$

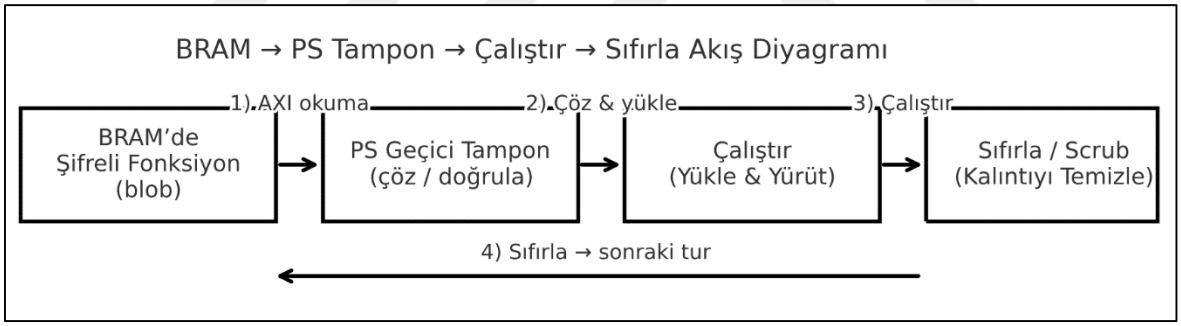
Gerekçe

K_{tr}^{r}'nin $(S_{ROOT},\ dev_{id})$ ile bağlanması, anahtarı cihaza özgü kılar. Monoton rot_{ctr} rotasyonlar arasında tekilliği sağlar. "TR" alan ayrımı etiketi, diğer XOF çağrılarıyla olası karışmaları önler. Doğru bir rot_{ctr} politikası ve nonce tekilliği sağlandığında ek alanlar *(ör. epoch, RM kimliği, içerik özeti)* girişi büyütse de anlamlı bir güvenlik kazanımı sağlamaz; bu nedenle yalın form tercih edilmiştir.

4.3.4. BRAM'e gömülü kod sayfaları ve çalışma zamanı yükleme

Önceki sürümlerde BRAM yalnızca secret-pool için kullanılırken, bu çalışmada seçili kod sayfaları da BRAM'de tutulup DFX ile yer değiştirilmiştir. Seçilmiş kod sayfalarının programlanabilir mantıkta (PL) yer alan BRAM üzerinde kimlik doğrulamalı şifreli paketler (ciphertext + tag) halinde tutulması ve her DFX turunda hedef yeniden yapılandırılabilir modüle (RM) taşınması planlanmıştır. Taşıma öncesi PS tarafında ASCON-XOF128 ile taze bir taşıma anahtarı (K_{tr}^{r}) türetilmesi önerilir. Her havuz/kod girdisi, giriş kimliği (ör. idx) ve

rotasyon sayacı (rot_{ctr}) gibi ayırt edici bağlamlar AAD/nonce olarak kullanılarak ASCON-128a ile hedef RM için yeniden paketlenmesi sayesinde, aynı mantıksal içerik ardışık iki turda dahi AXI hattında ve BRAM içinde istatistiksel olarak bağımsız şifreli metinlere karşılık gelir. Tekrar eden desenlerin görünürlüğü azalır ve korelasyon temelli yan kanal şablonlarının yeniden kullanımı zorlaşır. Böylece kodun açık metin hâli yalnızca PS çekirdek yazmaçlarında ve kısa ömürlü bir RAM tamponunda anlık olarak bulunur; doğrulama (tag) başarısız olduğunda yürütme yapılmaz. Yeniden paketleme tamamlandığında geçici açık metin zeroize edilir ve PCAP/XDevCfg yoluyla kısmi bit-akışı yüklenerek roller çevrilir; pasif RM scrub edilerek kalıntılar temizlenir. Bitstream'i çözen cihaz-özgü AES anahtarı yalnızca bitstream güvenliğine aittir; paket başına ASCON-XOF128 ile türetilen anahtarlarla ilişkili değildir ve havuz/kod yeniden paketlemeye karışmaz. Akışın adımları ve bellek/önbellek etkileşimi Şekil 4.10'da, güncel yükleyici parametreleri ve alanlar *(ör. AAD/nonce bağlamları)* Çizelge 4.7'de özetlenmiştir. Çalışmanın mevcut deney düzeneğinde kodun açık metin hali kullanılmıştır. Bu bahis, ileri çalışmalar için bir öneri olarak bırakılmıştır.



Şekil 4.10. BRAM üzerinde şifreli fonksiyon çalıştırma akış diyagramı

Çizelge 4.7. Fonksiyon yükleyici parametreleri

Tasarım	BRAM Ofsetleri (hex)	Blob uzunluğu (bayt, ör.)	Anahtarlandırma	DFX turu (r)	DFX Değişim Penceresi (ms)
2×BRAM	0x4000_0000– 0x4000_7FFF; 0x4200_0000– 0x4200_7FFF	8 192	– Deneysel: "Açık metin (AEAD yok)" – Öneri: "ASCON-128a (AAD/nonce: idx, rot_ctr)".	Round-robin veya rastgele slot	≈ 90 (Bulgular)
4×BRAM	0x4000_0000; 0x4100_0000; 0x4200_0000; 0x4300_0000 (her biri 32 KiB)	4 096~ 16 384	– Deneysel: "Açık metin (AEAD yok)" – Öneri: "ASCON-128a (AAD/nonce: idx, rot_ctr)".	Round-robin veya rastgele slot	≈ 180 (Bulgular)

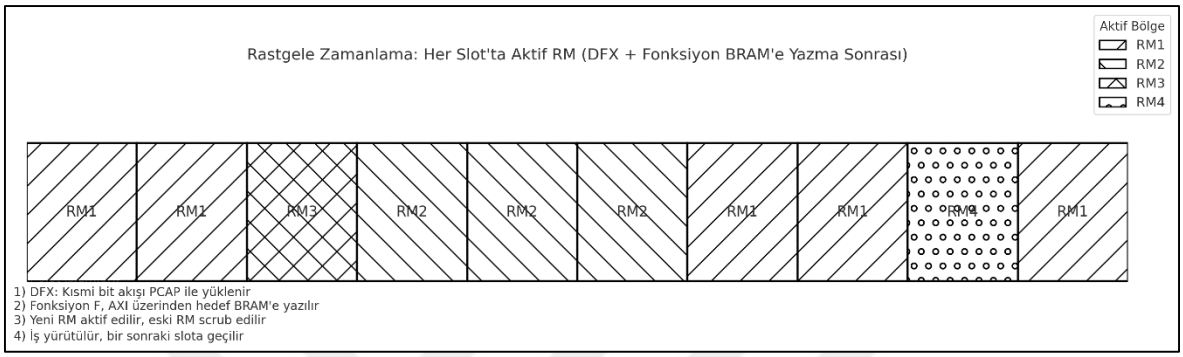
Not: n = 8 KiB, R = 128 MB/s kabulüyle yalnız aktarım süresi $t \approx 64\ \mu s$ olur. (AXI_HP teoriği 1.2 GB/s $\rightarrow$ $t \approx 6.8\ \mu s$.) Üzerine önbellek bakımı + fonksiyon çağrı/sıfırlama eklenerek ≈ 60 µs üst-sınır bütçesi elde edilir. (Kerling, 2019; AMD, 2025; Becher vd., 2016; Vipin ve Fahmy, 2014). Ancak bu değerler kurulum/handshake/PCAP kontrolcüsü ve sistem etkileşimlerini içermez. Bu çalışmada ölçülen per-swap DFX penceresi (PCAP + kontrol + eşzamanlama) ms ölçeğindedir (≈180 ms @ 4-RM) (Zeroize ≈45 ms/BRAM); bu yüzden mikro-saniyelik alt sınır zaman bütçesine katılmamıştır. Sayısal sonuçlar için Bulgular bölümüne bakınız.

Bu çalışmanın deney düzeneklerinde kod sayfaları açık metin olarak BRAM'de gezdirilmiş ve DFX ile taşınmıştır. AEAD ile şifreli taşıma tasarım seçeneği olarak sunulmuş, ayrıntılı zaman/enerji etkileri Bulgular bölümünde değerlendirilmek üzere bırakılmıştır. Şekil 4.10 akışı, Çizelge 4.7 ise deneysel (açık metin) ve öneri (ASCON-128a) ayrımını birlikte özetler.

4.3.5. Çoklu RM yerleşimi ve DFX zamanlama politikaları

Prototip hem iki RM'li doğrulama düzenini hem de dört RM'li stres düzenini gerçekledi. Dört RM'li yapı, yer değiştirmenin zaman/enerji davranışını gözlemlemek ve konum rastgeleliği altında güç/EM şablonlarının bozulmasını incelemek için kullanıldı. Doğrulama aşamasında ise iki RM'li tek atış (single-shot) politika tercih edildi. Şekil 4.11, her iletişim slot'unda DFX ile kısmi bit-akışının yüklenmesini, hedef BRAM'in etkinleşmesini ve bir önceki BRAM'in scrub edilmesini gösterir. Round-robin döngü, istenirse tek adımlık rastgele sapma ile karıştırılarak fiziksel yerleşimin öngörülebilirliği azaltılır. DFX penceresi bu çalışmada per-swap (tek kısmi bit-akışı yüklemesi) olarak tanımlıdır. Dört RM'li stres konfigürasyonunda, osiloskopla şönt üzerinden izlenen akım platosu "DFX anına" karşılık gelerek per-swap pencereyi ≈180 ms olarak göstermektedir. Aynı işletim koşullarında iki RM'li tek-atış politikada, kısmi bit-akışı yaklaşık yarıya indiğinden per-swap sürenin ≈90 ms düzeyinde olacağı ilk-mertebe bir öngörüdür. Profil ve enerji kırılımları Bulgular bölümünde raporlanacaktır. Önemli bir ayrım olarak, zeroize/temizleme maliyeti DFX penceresinden ayrıdır ve bellek yazımları ile tampon temizliğine bağlıdır. Laboratuvar ölçümlerine dayanarak yapılan öngörü ile hesaplama yapıldığında, BRAM başına ≈45 ms seviyesinde bir zeroize penceresi ortaya çıkmıştır. Ancak bu değer per-swap DFX süresinin bir parçası değildir *(örtüşme yoksa toplam süreye eklenir)*. Ayrıştırılmış zaman pencereleri, plateau segmentleri ve 2-RM/4-RM karşılaştırmaları metnin Bulgular kısmında

sunulmaktadır. Bu bölümde sayısal yineleme yapılmamış, mimari seçeneklerin ve zamanlama politikalarının bütünsel çerçevesi sunulmuştur. Bu ayrımla tutarlı olarak, Bölüm 5'te raporlanan DFX süre/enerji pencereleri 4 RM ölçümleridir; 2 RM sonuçları, aynı koşullarda tek-takasın yaklaşık yarısı olacak biçimde ölçeklenmiştir. 4-RM varyantının floorplanı Şekil 4.7'dedir.



Şekil 4.11. DFX akışıyla her iletişim slot'unda rastgele bir yeniden yapılandırılabilir modülün (RM1–RM4) aktif edilmesi

Not: Her slot'ta sıralı işlem adımları: (1) Kısmi bit akışının PCAP üzerinden yüklenmesi (DFX), (2) Fonksiyon F'nin AXI yoluyla hedef BRAM'e yazılması (fonksiyon kodu/verisi RAM'de tutulur ve dışarıdan okunması zorlaştırılır), (3) Yeni RM'nin aktif edilmesi ve bir önceki RM'nin scrub edilmesi (temizlenmesi), (4) İşin yürütülmesi ve bir sonraki slota geçiş.

4.3.6. Yer değiştirmenin güvenlik muhasebesi

DFX temelli yer değiştirme mekanizması, sistemin kriptografik anahtar uzayını genişletmez. Bunun yerine, fiziksel saldırılara karşı ek bir güvenlik katmanı oluşturarak saldırganın iş yükünü artırır. Bu katkı, bir saldırganın hassas veriye fiziksel olarak erişmek için çözmesi gereken konum belirsizliğinin (positional uncertainty) artırılmasıyla sağlanır. Bu belirsizlik, farklı seviyelerde incelenebilir. En temel modelde, bir saldırganın tek bir "gizli hücreyi" bulma zorluğu, olası konumların sayısıyla ölçülür.

$$H_{loc} = \log_2(M \cdot K) \tag{4.7}$$

Yukarıdaki ifadede (M), yeniden yapılandırılabilir BRAM bölgesi (RM) sayısını ve (K) ise her bir BRAM'deki adreslenebilir hücre sayısını temsil etmektedir (Eş. 4.7). Örneğin, bu çalışmada kullanılan 4 BRAM'li ($M = 4$) ve her biri 1024 ($K = 2^{10}$) adresten oluşan bir

konfigürasyonda, tek bir hücrenin tam yerini belirlemek için gereken ek bilgi miktarı $\log_2(4 * 1024) = 12$ bit'e karşılık gelir. Daha gerçekçi bir senaryoda, gizli veri k adet hücreye yayılır ve kalan $M * K - k$ hücre sahte/yem verilerle doldurulur. Bu durumda saldırganın, $M * K$ hücrelik toplam uzay içinden doğru k hücrenin hangileri olduğunu tahmin etmesi gerekir. Bu durum, belirsizliği kombinatoryal olarak artırır (Eş. 4.8).

$$H_{comb} = \log_2\big(C(M * K, k)\big) \tag{4.8}$$

Bu belirsizlik faktörlerinin hiçbiri, sistemin 128-bitlik kriptografik güvenliğine doğrudan eklenmez. Bunlar, fiziksel saldırı (probing, micro-scraping, side-channel analysis) senaryolarında saldırganın maliyetini ve karmaşıklığı artıran bağımsız terimler olarak değerlendirilmelidir. Dinamik yerleşim gibi ek faktörlerin de bu iş yüküne katkısı vardır. Bu bölümün amacı, DFX'in getirdiği bu çok katmanlı fiziksel güvenlik konseptini tanıtmaktır; yaklaşımların nicel analizi ise Bulgular ve Güvenlik Analizi bölümlerinde detaylandırılmıştır.

5. BULGULAR

Bu bölüm, üretici tarafında (Xilinx Zynq-7020) şifreleme ve çözücü/saldırı düzeneği tarafında (Arduino Due) şifre çözme için yapılan donanım ölçümlerini raporlar. Değerlendirilen küme, ASCON-128a, ACORN-128, TinyJAMBU-128 ve JAMBU–PRESENT-128 algoritmalarını içerir. Bu algoritmaların tümü hem standart (sade) halleriyle hem de önerilen Hibrit Güvenlik Çerçevesi (HGÇ) katmanı uygulanmış versiyonlarıyla karşılaştırmalı olarak değerlendirilmiştir. Zynq üzerinde gecikme ve çevrim sayıları ARM Global Timer ve PMU ile yakalanmış; akım/gerilim ölçümleri aşağıda ayrıntılandırıldığı üzere besleme hattında yapılmıştır. Çizelge 5.1 ve Çizelge 5.2'de görülen alt-aşama zamanlamaları ve çevrim dökümleri doğrudan PMU günlüklerinden ve birleştirilmiş çizelgelerden alınmıştır.

Not: Zynq alt-aşama çevrimleri ve süreleri (µs): HGÇ–ASCON toplam 12 µs (KDF 6,63 µs; AEAD 5,26 µs); ASCON-128a toplam 6 µs (AEAD 5,52 µs); ACORN-128 500 µs; TinyJAMBU-128 172 µs; JAMBU–PRESENT-128 366 µs.

5.1. Ölçüm Düzeneği ve Disiplin

Zynq kartının 12 V girişine, Rs = 18,408 Ω değerinde hassas bir seri şönt yerleştirilmiştir. Digilent Analog Discovery 3, şönt üzerindeki düşümü $V_{shunt(t)}$ olarak örneklemiştir. Anlık akım ve yük gerilimi şu şekilde hesaplanmıştır (Eş. 5.1, Eş. 5.2, Eş. 5.3).

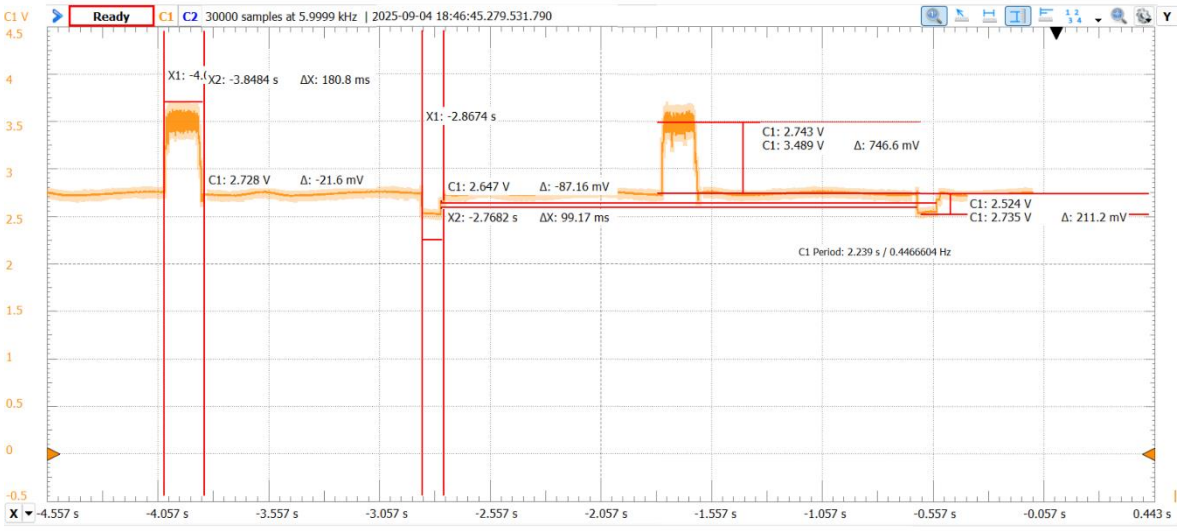
$$I(t) = \frac{V_{shunt(t)}}{R_S} \quad (5.1)$$

$$V_{load(t)} = 12.0 - V_{shunt(t)} \quad (5.2)$$

$$P(t) = I(t) \cdot V_{load(t)} \quad (5.3)$$

Kripto pencereleri, gözlenen $V_{load} \approx 8{,}5–9{,}5\,V$ aralığında çalışır; bu nedenle pencere enerjileri sabit bir değer yerine pencereye özgü V_{load} kullanılarak hesaplanır. İzlenebilirlik amacıyla 12 V hattının ortalama akımı PicoTest M3511A multimetre ile ayrıca ölçülmüş ($P_{12} = 12 \cdot I_{12}$) ve yalnızca çizelgelerde izlek amaçlı kullanılmıştır. Pencere bazlı enerjiler şönt yönteminden raporlanmıştır. Alıcı/çözücü platformu olan Arduino Due üzerinde yapılan enerji analizlerinde, deneyler sırasında kartın güç tüketiminin kararlı olduğu gözlemlenmiştir. Gerek boşta (idle) gerekse kriptografik işlemler sırasında yapılan hassas

akım ölçümlerinde, güç tüketiminde anlamlı bir dalgalanma tespit edilmemiştir. Bu deneysel bulguya dayanarak, Arduino platformunun güç tüketimi analizlerde $P = 21.6\,mW\ (12V\ x\ 1.8\,mA)$ olarak sabit alınmış ve enerji maliyetleri $E = P * \Delta t$ formülüyle, doğrudan işlem süresiyle orantılı olarak hesaplanmıştır. Dolayısıyla enerji süreyle doğrusal ölçeklenir: $E_{(\mu Wh)} = 6 \times 10^{-6} \cdot duration_{(\mu s)}$. Şönt yakalaması üç plato gösterir: (i) düşük akımlı bir UART gönderim penceresi, (ii) KDF+AEAD yürütmesini kapsayan idle{+}crypto penceresi ve (iii) kısmi yeniden yapılandırmaya bağlı kısa süreli aşırı-akım DFX penceresi. Pencere tablosundaki "Delta" satırları (Çizelge 5.3), belirtilen referansa göre imzalı farkları raporlar ($\Delta V_{load} = -\Delta V_{shunt}$) (Şekil 5.1). Aşağıdaki pencere tablosunda "AEAD delta (UART'a göre)" büyüklüğü, düşük akımlı UART platosu üzerinde kısa bir şifrelemenin artımsal enerjisini izole ederek kriptografik işi kaydetme/günlükleme yükünden ayrıştırır. DFX penceresi, yeniden konumlandırma adımına karşılık gelir. 4-RM tek-atım süre/enerji değerleri doğrudan ölçülmüş, 2-RM tek-atım değerleri aynı çalışma koşullarında doğrusal ölçekleme ile elde edilmiştir.



Şekil 5.1. Rs = 18,408 Ω üzerinden Digilent Analog Discovery 3 ile alınan şönt yakalaması; UART, AEAD ve DFX platoları işaretlenmiştir

5.2. Üretici Tarafı Şifreleme (ZYNQ-7020)

Çizelge 5.1, Zynq-7020 üzerinde üretici tarafı şifreleme sonuçlarını birleştirir ve değerlendirilen tüm AEAD algoritmalarının hem standart (sade) hem de HGÇ sargılı

versiyonları için PMU/GT kaynaklı çevrim sayımları ve süreleri ile alt-aşamaları *(KDF, init, AAD, msg, fin)* raporlar. Güç ve enerji, kripto platosundaki şönt-türevli yük çalışma noktasından elde edilmiştir; izlenebilirlik için 12 V PicoTest ölçüm toplulukları da gösterilir. Enerjiler $E_{\mu Wh} = P \cdot \frac{\Delta t_{\mu s}}{3600}$ bağıntısıyla hesaplanır. Bu çizelge, ana makine tarafı yapıştırıcı işlemleri (host-side glue) hariç tutularak, üreticinin mesaj-başı hesaplama maliyetini tekdüze bir ölçüm disiplininde niceler. Üretici çizelge, HGÇ altında bir korunan mesajın (KDF+AEAD) kripto çalışma noktasındaki maliyetini raporlar. Aşama ve alt-aşama süre/çevrimleri PMU günlükleri ve birleştirilmiş çizelgelerden türetilmiştir. Çizelge 5.1, bu karşılaştırmaların alt kırımlarıyla birlikte cycle, süre, güç, enerji karşılaştırmalarını detaylı biçimde göstermektedir.

Çizelge 5.1. Zynq-7020 şifreleme performansı: Referans AEAD'lerin HGÇ'li ve HGÇ'siz sürümlerinin karşılaştırması

Algoritma / Alt-aşamalar	Cycle	Süre (µs)	I@9.3V (A)	I@12V (A)	P@9.3V (W)	P@12V (W)	Enerji (µWh) @9.3 / @12
ASCON-128a	5,200	6.00	0.149	0.125	1.38	1.50	0.0023 / 0.0025
AEAD (total)	4,793	5.52	0.149	0.125	1.38	1.50	0.0021 / 0.0023
init	1,225	1.41	0.149	0.125	1.38	1.50	0.0005 / 0.0006
aad	1,343	1.55	0.149	0.125	1.38	1.50	0.0006 / 0.0006
msg	1,096	1.25	0.149	0.125	1.38	1.50	0.0005 / 0.0005
fin	1,129	1.26	0.149	0.125	1.38	1.50	0.0005 / 0.0005
ASCON-128a (HGÇ)	10,400	12.00	0.149	0.125	1.38	1.50	0.0046 / 0.0050
KDF	5,511	6.63	0.149	0.125	1.38	1.50	0.0025 / 0.0028
AEAD (total)	4,375	5.26	0.149	0.125	1.38	1.50	0.0020 / 0.0022
init	980	1.18	0.149	0.125	1.38	1.50	0.0005 / 0.0005
aad	1,183	1.42	0.149	0.125	1.38	1.50	0.0005 / 0.0006
msg	934	1.12	0.149	0.125	1.38	1.50	0.0004 / 0.0005
fin	978	1.18	0.149	0.125	1.38	1.50	0.0005 / 0.0005

Çizelge 5.1. (devam) Zynq-7020 şifreleme performansı: Referans AEAD'lerin HGÇ'li ve HGÇ'siz sürümlerinin karşılaştırması

ACORN-128	433,100	500.00	0.149	0.125	1.38	1.50	0.1912 / 0.2083
AEAD (total)	432,997	499.88	0.149	0.125	1.38	1.50	0.1911 / 0.2083
init	227,347	262.46	0.149	0.125	1.38	1.50	0.1004 / 0.1094
aad	50,615	58.43	0.149	0.125	1.38	1.50	0.0223 / 0.0243
msg	53,078	61.28	0.149	0.125	1.38	1.50	0.0234 / 0.0255
fin	101,916	117.66	0.149	0.125	1.38	1.50	0.0450 / 0.0490
ACORN-128 (HGÇ)	437,523	505.00	0.149	0.125	1.38	1.50	0.1936 / 0.2104
KDF	5,550	6.00	0.149	0.125	1.38	1.50	0.0023 / 0.0025
AEAD (total)	431,907	498.00	0.149	0.125	1.38	1.50	0.1908 / 0.2075
init	226,353	260.99	0.149	0.125	1.38	1.50	0.1002 / 0.1087
aad	50,627	58.37	0.149	0.125	1.38	1.50	0.0224 / 0.0243
msg	52,955	61.06	0.149	0.125	1.38	1.50	0.0234 / 0.0254
fin	101,944	117.54	0.149	0.125	1.38	1.50	0.0451 / 0.0490
TinyJAMBU-128	149,263	172.00	0.149	0.125	1.38	1.50	0.0658 / 0.0717
AEAD (total)	149,158	171.88	0.149	0.125	1.38	1.50	0.0657 / 0.0716
init	19,617	22.61	0.149	0.125	1.38	1.50	0.0086 / 0.0094
aad	29,344	33.81	0.149	0.125	1.38	1.50	0.0129 / 0.0141
msg	87,877	101.26	0.149	0.125	1.38	1.50	0.0387 / 0.0422
fin	12,244	14.11	0.149	0.125	1.38	1.50	0.0054 / 0.0059
TinyJAMBU-128 (HGÇ)	154,669	178.00	0.149	0.125	1.38	1.50	0.0682 / 0.0742
KDF	5,615	6.00	0.149	0.125	1.38	1.50	0.0023 / 0.0025
AEAD (total)	148,985	172.00	0.149	0.125	1.38	1.50	0.0660 / 0.0717
init	19,631	22.66	0.149	0.125	1.38	1.50	0.0087 / 0.0094
aad	29,345	33.88	0.149	0.125	1.38	1.50	0.0130 / 0.0141
msg	87,733	101.29	0.149	0.125	1.38	1.50	0.0388 / 0.0422
fin	12,249	14.14	0.149	0.125	1.38	1.50	0.0054 / 0.0059

Çizelge 5.1. (devam) Zynq-7020 şifreleme performansı: Referans AEAD'lerin HGÇ'li ve HGÇ'siz sürümlerinin karşılaştırması

JAMBU–PRSNT-128	317,142	366.00	0.149	0.125	1.38	1.50	0.1400 / 0.1525
AEAD (total)	316,896	365.72	0.149	0.125	1.38	1.50	0.1398 / 0.1524
init	53,568	61.82	0.149	0.125	1.38	1.50	0.0236 / 0.0258
aad	105,319	121.54	0.149	0.125	1.38	1.50	0.0465 / 0.0506
msg	105,407	121.60	0.149	0.125	1.38	1.50	0.0465 / 0.0507
fin	52,580	60.68	0.149	0.125	1.38	1.50	0.0232 / 0.0253
JAMBU–PRSNT-128 (HGÇ)	322,514	372.00	0.149	0.125	1.38	1.50	0.1427 / 0.1550
KDF	5,485	6.00	0.149	0.125	1.38	1.50	0.0023 / 0.0025
AEAD (total)	316,836	366.00	0.149	0.125	1.38	1.50	0.1401 / 0.1525
init	53,506	61.81	0.149	0.125	1.38	1.50	0.0236 / 0.0258
aad	105,322	121.66	0.149	0.125	1.38	1.50	0.0466 / 0.0507
msg	105,406	121.76	0.149	0.125	1.38	1.50	0.0466 / 0.0508
fin	52,580	60.74	0.149	0.125	1.38	1.50	0.0233 / 0.0253

Not: Kripto platoları şöntten türetilmiş $V_{load} \approx 9.265\,V$ ve $I \approx 0.149\,A$ noktasında ölçülmüştür; buna göre $P_{load} \approx 1.38\,W . 12\,V$ sütunu, PicoTest multimetre ile ölçülen ortalama akımı ($I \approx 0.125\,A \Rightarrow P_{12V} \approx 1.50\,W$) gösterir. Enerjiler $E_{\mu Wh} = P \cdot \frac{\Delta t_{\mu s}}{3600}$ bağıntısıyla hesaplanmıştır.

5.3. Çözücü Tarafı Şifre Çözme (Arduino DUE)

Çizelge 5.2, Arduino Due üzerinde aynı mesaj-başı çerçevelemede çözücü tarafı şifre çözmeyi izole eder. Sabit 12 V ve 1,8 mA (21,6 mW) altında çalışıldığından enerjiler süreyle doğrusal ölçeklenir. Çizelgede, değerlendirilen tüm algoritmaların (hem HGÇ'li hem de sade versiyonların) parse, KDF ve AEAD gibi alt aşama dökümleri de sunulmuştur ve birleştirilmiş değerlerle tutarlıdır. Bu değerler; girdilerin ayrıştırılması, mesaj-başı anahtarların türetilmesi ve doğrulama/çözme işlemlerinin mikrokontrolcüdeki maliyetini yakalar. HGÇ–ASCON toplamı ile parse/KDF/AEAD dökümü 21,6 mW'da hesaplanmış olup birleştirilmiş enerji çizelgesiyle eşleşir (örn. Due üzerinde HGÇ–ASCON için işlem başına 0,005772 µWh).

Çizelge 5.2. Arduino DUE deşifreleme performansı: referans AEAD'lerin HGÇ'li ve HGÇ'siz sürümlerinin karşılaştırması

Algoritma / Alt-aşama	Duration (µs)	I@12 V (mA)	P@12 V (mW)	Energy (µWh) @12
HGÇ–ASCON-128a	962	1.8	21.6	0.00577
parse	486	1.8	21.6	0.00292
KDF	254	1.8	21.6	0.00152
AEAD	222	1.8	21.6	0.00133
ASCON-128a	710	1.8	21.6	0.00426
parse	486	1.8	21.6	0.00292
AEAD	224	1.8	21.6	0.00134
HGÇ–ACORN-128	21 977	1.8	21.6	0.13186
parse	486	1.8	21.6	0.00292
KDF	254	1.8	21.6	0.00152
AEAD	21 237	1.8	21.6	0.12742
ACORN-128	21 663	1.8	21.6	0.12998
parse	486	1.8	21.6	0.00292
AEAD	21 177	1.8	21.6	0.12706
HGÇ–TinyJAMBU-128	5 576	1.8	21.6	0.03346
parse	486	1.8	21.6	0.00292
KDF	254	1.8	21.6	0.00152
AEAD	4 836	1.8	21.6	0.02902
TinyJAMBU-128	5 302	1.8	21.6	0.03181
parse	486	1.8	21.6	0.00292
AEAD	4 816	1.8	21.6	0.02890
HGÇ–JAMBU–PRESENT-128	17 646	1.8	21.6	0.10588
parse	486	1.8	21.6	0.00292
KDF	254	1.8	21.6	0.00152
AEAD	16 906	1.8	21.6	0.10144
JAMBU–PRESENT-128	17 299	1.8	21.6	0.10379
parse	486	1.8	21.6	0.00292
AEAD	16 813	1.8	21.6	0.10088

Not: Arduino Due ölçümleri sabit I = 1.8 mA (P = 21.6 mW) kabulüyle PicoTest multimetre ile ölçülen 12 V sabit gerilimine dayanmaktadır. Enerjiler $E(\mu Wh) = P \cdot \frac{\Delta t_{\mu s}}{3600}$ eşitliğiyle hesaplanır. HGÇ–ASCON toplamı ve parse/KDF/AEAD dökümü, birleştirilmiş enerji çizelgesiyle eşleşmektedir (örn. HGÇ–ASCON için işlem başına 0.00577 µWh). ASCON/ACORN/TinyJAMBU 128 bit etiket (AEAD) kullanır; JAMBU–PRESENT 64 bit etiket kullanır (PRESENT-ECB, AEAD olmayan bir karşılaştırma temelidir.).

5.4. Enerji Atfı ve DFX Ölçekleme

Bu alt bölüm, Çizelge 5.3'deki plato bölütlemesini Çizelge 5.4'teki algoritma özetleriyle birleştirerek taşıma (transport), kriptografi ve yeniden konumlandırma (relocation) enerji paylarını atfeder. Bu bölümdeki DFX pencereleri 4-BRAM (4 RM) düzenek üzerinde ölçülmüştür (Şekil 4.7); 2-BRAM değerleri tek-takas penceresinin ≈½'si kabul edilerek ölçeklenmiştir. "AEAD delta (UART'a göre)" büyüklüğü, idle{+}crypto penceresi içinde şifrelemenin artımsal işini izole ederken; DFX satırları tek-atım kısmi yeniden yapılandırmayı temsil eder. Ölçülen 4-RM pencere, mesaj başına tek takas başına referans değeri sağlar; aynı çalışma koşullarında 2-RM tek-atım süresi ve enerjisi yaklaşık yarıya ölçeklenir ve özet çizelgede her iki varyant da açıkça raporlanır. "AEAD delta (UART'a göre)" satırı, düşük akımlı UART platosu üzerinde kısa bir şifrelemenin artımsal enerjisini izole ederek kriptografik işi kaydetme/günlükleme yükünden ayrıştırır. DFX penceresi, kısmi yeniden yapılandırmaya bağlı kısa bir aşırı-akım sergiler; çizelgede hem toplam hem de idle'a göre delta katkıları listelenir. DFX süre/enerji değerleri 4 RM ölçümüne dayalıdır; 2 RM karşılıklar aynı koşullarda tek-takas penceresinin ≈½'si olarak verilmiştir. Kripto (KDF+AEAD) yolunun gecikmesi RM sayısından bağımsızdır. Şifreler arası karşılaştırmalarda üretici için idle{+}crypto çalışma noktası benimsenerek, takip eden eşitliklere göre belirlenir (Eş. 5.4, Eş. 5.5).

$$P_{enc} = I_{enc} \cdot V_{load,enc} \tag{5.4}$$

$$E_{alg} = P_{enc} \cdot \Delta t_{alg} \tag{5.5}$$

Metodoloji notu - bant genişliği üst-sınırları ve enerji entegrasyonu

Ölçülen pencere süreleri (örn. 4-RM için $t_{DFX} \approx 180.8\,ms$, 2-RM için $t_{DFX} \approx 90.4\,ms$) bant genişliği temelli üst-sınır kestirimlerle tutarlıdır. Küçük tamponlar için işlem süresi kaba olarak $t \approx \frac{n}{R}$, kısmi yeniden yapılandırma için $t_{PR} \approx \frac{n_{PR}}{R_{PCAP}}$ ile sınırlandırılabilir; burada n aktarım boyutu, R ilgili yolun etkin verimidir. Enerji, pencere boyunca şönt ölçümünden türetilen $E = \int V(t)I(t)dt$ ile hesaplanmıştır. Bu nedenle çizelgeler pencere-özel $V_{load}(t)$ değerini kullanır. Bu üst-sınırlar PCAP/denetim eklerini kapsamadığından, nihai metrikler yalnız ölçümle raporlanmıştır (Çizelge 5.3 ve 5.4). Eşitlikler yalnızca sezgisel tavan olarak sunulur. HGÇ–ASCON karşılaştırma çizelgesinde iki kez raporlanır:

(i) yalnız kripto (KDF+AEAD, yeniden konumlandırma yok) ve (ii) kripto+DFX (yeniden konumlandırma maliyeti dahil).

Yer değişim politikası ölçeklendirme (4-RM ölçüm ve 2-RM doğrulama)

Zorlu deneyler, yeniden konumlandırmayı sınamak üzere dört BRAM destekli yeniden yapılandırılabilir modülü (RM) çalıştırmıştır. Doğrulama tasarımı ise iki RM kullanır. Çizelge 5.3'Teki DFX maliyeti bir tek-takas penceresini *(tek kısmi bitstream yükü)* temsil eder. M bölgesi için mesaj başına $N_{swap}(M)$ takas yapılan bir politika altında maliyetler hesaplanır (Eş. 5.6, Eş. 5.7).

$$E_{DFX(M)} = N_{swap(M)} \cdot E_{DFX}^{per-swap}(M) \tag{5.6}$$

$$t_{DFX(M)} = N_{swap(M)} \cdot t_{DFX}^{per-swap}(M) \tag{5.7}$$

Ölçülen 4-RM bitstream için tek-takas değerleri *(Çizelge 5.3)*: $t_{DFX}^{per-swap}(4) = 180{,}8$ ms; delta enerji $E_{DFX}^{per-swap}(4) = 11{,}75\ \mu Wh$; toplam pencere enerjisi 81,016 µWh. 2-RM politikasında $N_{swap}(2) = 1$ (yarım boy bitstream tek-atım PR) varsayımıyla $t_{DFX}^{per-swap}(2) \approx 90{,}4\ ms$ ve $E_{DFX}^{per-swap}(2) \approx 5{,}875$ µWh (delta) elde edilir; bu, aynı çalışma koşullarında ölçülen tek-takas penceresinin doğrusal ölçeklenmesiyle uyumludur.

$$t_{DFX}^{per-swap}(2) \approx 90.4\ ms$$

$$E_{DFX}^{per-swap}(2) \approx 5.875\ \mu Wh$$

Sonuçlar, her iki platformda mesaj-başı hesaplama ve enerji maliyetlerini ortak bir çerçevede birleştirir. Üretici tarafında (Zynq-7020) çevrim ve zamanlamalar ARM PMU/Global Timer'dan; enerjiler 8,5–9,5 V aralığında gözlenen yük üzerinde plato-temelli seri-şönt entegrasyonundan elde edilmiştir (Analog Discovery 3); 12 V ortalama akım PicoTest M3511A multimetre ile ölçülmüş ve yalnızca çapraz denetim amacıyla kullanılmıştır (Çizelge 5.1). Çözücü tarafında (Arduino Due) P = 21,6 mW sabit olduğundan enerji süreyle orantılıdır ve parse/KDF/AEAD ayrımı açıkça listelenmiştir (Çizelge 5.2). Pencere bölütleme UART, idle{+}crypto ve DFX katkılarını ayırır ve "AEAD delta (UART'a göre)"

artımsal şifreleme maliyetini ortaya koyar (Çizelge 5.3). Karşılaştırma, yalnız kripto HGÇ–ASCON ile ilave DFX'i ayırt ederek yeniden konumlandırma ek yükünü açıklar. Tek-atım kısmi yeniden yapılandırmada DFX doğrusal ölçeklenir. 2-RM politikasında ölçülen 4-RM pencerenin süre/enerjisinin yaklaşık yarısı söz konusudur (Çizelge 5.4).

Çizelge 5.3. Zynq üzerinde plato pencereleri (şönt yöntemi): Süre, akım, güç ve enerji

Pencere	Δt (ms)	V_shunt (V)	I (mA)	V_load (V)	P (W)	E (µWh)
UART (TX)	99.170	2.524	137.114	9.476	1.2993	35.792
HGÇ (tam pencere)	0.012	2.735	148.577	9.265	1.3766	0.004589
HGÇ delta (UART'a göre)	*0.012*	*+0.211*	*+11.462*	*−0.211*	*+0.07727*	*0.000258*
DFX (4-RM, tam pencere)	180.800	3.489	189.537	8.511	1.6132	81.016
DFX delta (4-RM, idle'a göre)	*180.800*	*+0.7466*	*+40.559*	*−0.7466*	*+0.23397*	*11.750*
DFX (2-RM, tam pencere)	90.400	3.489	189.537	8.511	1.6132	40.508
DFX delta (2-RM, idle'a göre)	*90.400*	*+0.7466*	*+40.559*	*−0.7466*	*+0.23397*	*5.875*

Not: Kripto ve DFX platoları ölçülen şönt-türevli çalışma noktasında değerlendirilmiştir; enerjiler pencere-özel $V_{load} = 12.0 - V_{shunt}$ ile hesaplanır. 2-RM DFX satırları, tek-atım kısmi yeniden yapılandırma penceresinin doğrusal ölçeklenmesiyle elde edilmiştir: $t \approx 90.4\,ms$, $E_{\Delta} \approx 5.875\,\mu Wh$, mutlak $E \approx 40.508\,\mu Wh$ (4-RM pencerenin yarısı). $t \approx n/R$ (işlem) ve $t_{PR} \approx \frac{n_{PR}}{R_{PCAP}}$ (PR) üst-sınırları denetim/handshake eklerini içermez; raporlanan süre ve enerjiler yalnız ölçüm sonuçlarıdır.

Çizelge 5.4. Kod koruma düzeneklerinin enerji/gecikme karşılaştırması (yalnız şönt ölçümü)

Çalışma	Platform	Uygulama	Enerji/Güç	Gecikme	Notlar
Soliman (2019)	Zynq-7020	5-şifre sıçrama	10 mW dyn.	10 ms/sw.	DPR tabanlı şifre sıçrama; ek bitstream gerektirir
Samir (2019)	Zynq-7000	Güç-uyarlamalı HSM	≤10 mW dyn.	50–100 ms/sw.	Enerji-uyarlamalı AEAD seçimi; bitstream savunması yok
JAMBU-128 (PRESENT)	Zynq-7020	AEAD (1×16 B)	0.139951 µWh/msg	366 µs	Sadece AEAD

Çizelge 5.1. (devam) Kod koruma düzeneklerinin enerji/gecikme karşılaştırması (yalnız şönt ölçümü)

TinyJAMBU-128	Zynq-7020	AEAD (1×16 B)	0.065769 µWh/msg	172 µs	Sadece AEAD
ACORN-128	Zynq-7020	AEAD (1×16 B)	0.191189 µWh/msg	500 µs	Sadece AEAD
Ascon-128a	Zynq-7020	AEAD (1×16 B)	0.002294 µWh/msg	6 µs	Sadece AEAD
HGÇ-ASCON (DFX yok)	Zynq-7020	İletişim dayanımı	0.004589 µWh/msg	12 µs	HGÇ sargı (XOF KDF, nonce, AAD)
HGÇ-ASCON (+DFX, 4-RM)	Zynq-7020	Kripto + DFX	11.755 µWh/msg	12 µs + 180.8 ms	HGÇ + DFX (sırlar + kod)
HGÇ-ASCON (+DFX, 2-RM)	Zynq-7020	Kripto + DFX (2-RM)	5.875 µWh/msg	12 µs + 90.4 ms	HGÇ + DFX (tek-atım 2-RM)

Not: DFX satırları, mesaj başına kısmi yeniden yapılandırma takası sayısıyla doğrusal ölçeklenir; tek-takas değerleri Çizelge 5.3'ten: 4-RM için t = 180,8 ms, $E_{\Delta} = 11{,}75\ \mu Wh$ (mutlak 81,016 µWh). 2-RM tek-atım için yaklaşık yarıya iner (t ≈ 90,4 ms, $E_{\Delta} \approx 5{,}875\ \mu Wh$).

DFX sıklık politikası

Yeniden konumlandırma, her paketle ilişkilendirilmek zorunda değildir. Her N mesajda bir kez yapılacak tek-atım 2-RM takası, her takas başına $\approx 90.4\,ms$ ve $\Delta E \approx 5.875\,\mu Wh$ ekler. Yalnız kripto yolu değişmeden kalır.

5.5. Güvenlik Analizi ve Saldırı Deneyleri

Bu bölümde HGÇ–ASCON; kaba kuvvet, sahtecilik/yeniden oynatma, yan kanal, zamanlama, donanım kurcalama, dinamik anahtar çözümleme ve sözlük saldırılarına karşı değerlendirilir. Oturum anahtarları, aygıt-içi S_{ROOT}, dev_{id} ve $POOL[idx]$ bağlamlarından ASCON-XOF128 ile türetilir; Politika B'de *(isteğe bağlı)* $(epoch, msg)$ eklenerek paket-başı ayrım elde edilir. Nonce'lar, anahtar başına tekillik sağlayan deterministik $pack128(dev, epoch, msg)$ biçimindedir; ASCON-128a altında kabul ölçütü yalnızca etikettir (128 bit). Politika A'da (dilim-başı), türetilmiş bir anahtarın ele geçirilmesi, rekey

yapılana kadar aynı (dev, idx) çiftini paylaşan paketleri etkiler; Politika B'de (mesaj-başı) etkilenim yalnızca ilgili paketle sınırlıdır. DFX tabanlı yeniden konumlandırma (relocation), prob/şablon saldırıları için fiziksel arama uzayını genişletir.

Yöntem, gömülü ürün dağıtımlarında ROM kazıma ve iletişim dinleme gibi tehditleri doğrudan ele alır. Bitstream şifreleme ve bellek bulanıklaştırma, kodun fiziksel ya da dolaylı yollarla çıkarımını zorlaştırırken; mesaj-başı anahtarlama olası sızmayı tek bir şifreli metinle sınırlar.

Yan kanal hususları iki boyutludur. Birincisi, yazılım tabanlı kimlikli şifre çözme (ASCON-128a), zamanlama değişkenliğini sınırlayan sıkı ve sabit bir denetim akışıyla yürütülür; sabit frekansta çalışma zamanlama yan kanallarını ayrıca azaltır. İkincisi, dinamik kısmi yeniden yapılandırma (DFX), BRAM destekli modülleri mesajlar arasında yeniden konumlandırarak kalıcı güç/EM şablonlarını bozar. R yer değiştirme yuvası ve M yeniden yapılandırılabilir modül varsayıldığında, mesaj başına konfigürasyon uzayı $P(R, M) = R! / (R - M)!$ olup yerleşim belirsizliği $H_{DFX} = \log_2 P(R, M)$ bit'tir. Örneğin R = 4, M = 2 için $H_DFX \approx \log_2(12) \approx 3{,}6\ bit$ tek-atım yerleştirme); 4-RM stres konfigürasyonu $\log_2(24) \approx 4{,}6$ bit sağlar. Bu bitler kriptografik anahtar entropusunu tamamlar, yer değiştirmez. Güvenlik iddialarında bağımsızlık varsayımları muhafazakâr ele alınmalıdır.

Doğruluk notu

DFX anahtar entropusuna ek yapmaz; yalnızca scraping/probing için fiziksel arama uzayını büyütür. Tek "gizli hücre" modelinde fiziksel konum belirsizliği: $H_{loc} = \log_2(M \cdot K)$. $M = 10\ BRAM$ ve K = 1024 adres için $H_{loc} \approx \log_2(10240) \approx 13{,}3$ bit. Gerçek veriler MK hücresinden k tanesine dağılmış, kalan hücreler yem (decoy) ile doldurulmuşsa, birleşimsel belirsizlik $H_{comb} = \log_2 C(M \cdot K, k)$ olur (örn. k = 10 ⇒ ≈ 111,4 bit; k = 50 ⇒ ≈ 451,7 bit). Uygulanacak model, dağıtım gereksinimlerine göre açıkça belirtilmelidir.

FPGA platformlarında donanım manipülasyonu *(ör. saat kurcalama veya donanım Truva atları)*, aygıta kilitli bitstream'ler ve silikon kimliği ile sınırlandırılır. Vivado Device DNA ve bitstream şifreleme, bitstream'leri hedef aygıta bağlayarak yetkisiz değiştirme/yeniden kullanım girişimlerini zorlaştırır (Johnson, 2017; Stolz vd., 2021). Zamanlama saldırıları; sabit saat kaynağı ve sabit-zaman AEAD doğrulama yolları ile ayrıca azaltılır (Johnson,

2017; Stolz vd., 2021). Dinamik anahtar çözümleme saldırıları, yüksek değişkenlikli girdilerden *(ör. rastgele kelime, zaman damgası, isteğe bağlı aygıt kimliği vb.)* mesaj-başı türetimle ele alınır; böylece mesajlar arasında anahtar kestirimi etkisizleşir; HGÇ'de KDF girdisi olarak "rastgele kelime/zaman damgası" kullanımı bu amacı destekler.

PR kanal bütünlüğüne ilişkin varsayımlar

Kısmi bitstream'ler, aygıta bağlı şifreleme ve bütünlük denetimleriyle PCAP üzerinden yüklenir. 7-serisinde Starbleed gibi bilinen sorunlara karşı kullanılan araç-zinciri sürüm/yama düzeyleri belgelenmiştir (Ender vd., 2020). PR kanalının ele geçirilmesini gerektiren tehditler kapsam dışıdır.

Terimler (güvenlik muhasebesi)

q: saldırgan denemelerinin/toplam sorguların sayısı *(doğrulama/sahteleme denemeleri)*. D: anahtar sayısı *(örn. aygıt/oturum)*; M: anahtar başına mesaj sayısı; Q = D · M *(filo genelinde toplam deneme)*. RO/PRF: XOF/KDF'in, kamu bağlam üzerinde S_{ROOT} ile anahtarlanmış bir PRF ya da Rastgele Orakl olarak modellenmesi.

AEAD için nicel sınırlar

t bit etikvdtında toplam Q denemede sahteleme olasılığıdır (Eş. 5.8).

$$\Pr[forge] \leq \frac{Q}{2^t} \tag{5.8}$$

Çok kullanıcılı durumda D aygıt ve aygıt başına M mesaj varsa ($Q = D \cdot M$) olasılık şu şekilde ifade edilir (Eş. 5.9).

$$\Pr[forge] \leq \frac{D \,.\, M}{2^t} \tag{5.9}$$

Sayaç tabanlı nonce'lar doğum günü çarpışması üretmez; rastgele L bit nonce kullanılsaydı çakışma olasılığı aşağıdaki gibi olurdu (Eş. 5.10):

$$P_{coll} \approx \frac{Q^2}{2^{L+1}} \tag{5.10}$$

Çok kullanıcılı örnek: $D = 10^4, M = 10^6 \Rightarrow Q = 10^{10}$.

$$t = 64: \Pr[forge] \leq \frac{10^{10}}{2^{64}} \approx 5{,}4 \times 10^{-10} \approx 2^{-30{,}8}$$

$$t = 128: \Pr[forge] \leq \frac{10^{10}}{2^{128}} \approx 2{,}9 \times 10^{-29} \approx 2^{-94{,}8}$$

Aygıt-başı örnek: $D = 1,\ M = 10^6 \Rightarrow Q = 10^6$.

$$t = 64: \Pr[forge] \leq \frac{10^{6}}{2^{64}} \approx 5{,}4 \times 10^{-14} \approx 2^{-44{,}1}$$

$$t = 128: \Pr[forge] \leq \frac{10^{6}}{2^{128}} \approx 2{,}9 \times 10^{-33} \approx 2^{-108{,}1}$$

Rastgele nonce başvurusu (kullanılmıyor): $L = 128,\ Q = 10^{10}$

çekiliş $\Rightarrow P_{coll} \approx \frac{(10^{20})}{2^{129}} \approx 1{,}5 \times 10^{-19} \approx 2^{-62{,}6}$

64-bit bloklar için veri sınırı

PRESENT tabanlı AEAD'de (JAMBU–PRESENT-128), tek bir anahtar altında işlenen toplam 64 bit blok sayısı doğum günü sınırı nedeniyle $\ll 2^{32}$ tutulmalıdır. Uygulamada güvenlik marjı için $\sim 2^{28..30}$ blokta anahtar döndürme önerilir. Döndürme noktası 2^{29} blok seçildiğinde, anahtar değişimine kadar izin verilen yaklaşık mesaj sayıları Çizelge 5.5'te verilmiştir.

Çizelge 5.5. Veri sınırları (64-bit bloklar)

Payload boyutu	Blok/mesaj	Maks. mesaj (2^{29} blokta)
64 B	8	2^29 / 8 = 67.108.864
256 B	32	2^29 / 32 = 16.777.216
1 KB	128	2^29 / 128 = 4.194.304
4 KB	512	2^29 / 512 = 1.048.576

Şifre takımının güvenliği ve adillik

JAMBU–PRESENT-128, 64 bit etiket üretir ($\frac{Q}{2^{64}}$); ASCON/ACORN/TinyJAMBU ise 128 bit etiket kullanır ($\frac{Q}{2^{128}}$). Raporlanan performans çizelgeleri bu güvenlik seviyesi farkını not eder. PRESENT-ECB yalnızca çekirdek hız karşılaştırması için kullanılmıştır. Pratik dağıtımlar bir AEAD kipini *(örn. JAMBU)* gerektirir.

Çizelge 5.6. Şifre türleri ve AEAD durumu

Düzenek	Tip	Etiket	AEAD?
ASCON-128a	Sünger AEAD	128-bit	✓
ACORN-128	Akış AEAD	128-bit	✓
TinyJAMBU-128	Hafif AEAD	128-bit	✓
PRESENT-128 (ECB)	Blok şifre çekirdeği	—	—
JAMBU-PRESENT-128	Kip + blok şifre (AEAD)	64-bit	✓

Yan kanal yöntembilimi ve PR-kanalı varsayımları

Kod yolları sabit denetim akışını izler. Uygun yerlerde tablo-tabanlı S-kutuları kullanılmaz; etiket karşılaştırması sabit zamandadır. DFX, BRAM-yerleşimli havuzları ve seçili kod sayfalarını yer değiştirerek kalıcı şablonları kırar; kriptografik entropiden bağımsızdır fakat saldırganın fiziksel arama uzayını artırır. PR kanal bütünlüğü varsayılır: aygıta bağlı, şifreli bitstream'ler ve doğrulanmış araç-zinciri yetkisiz yeniden kullanım/kurcalamayı sınırlar. Uygulamalar sabit-zaman kodlama izler; tablo-tabanlı S-kutuları kaçınılır *(PRESENT için bitslice önerilir)*. Gelecek çalışmalar sabit frekansta DPA/CPA ölçümleri, sızıntı katsayıları ve şablon dayanımını içerir. DFX kaynaklı yeniden konumlandırma, kalıcı şablonları istikrarsızlaştırmak üzere kullanılır.

Saldırı deneyleri (Arduino Due, 84 MHz)

Pratik kaba-kuvvet verimi ve sızıntı izolasyonu Arduino Due (SAM3X8E, 84 MHz) üzerinde değerlendirilmiştir. Günlüklerden ASCON-128a doğrulama maliyeti ≈ 224 µs (≈ 18.816 çevrim; $F_{CPU} = 84\,MHz$ ile uyumlu). Kaba-kuvvet deneyi, kripto-analitik başarı

değil etiket doğrulama verimini *(saniyedeki bütünlük denetimi)* ölçer (Çizelge 5.7). Verim denemeleri rastgele 128-bit anahtarlar üzerinde doğrulama denemelerini sayar; sızıntı denemeleri (i) tek bir oturum anahtarının K sızdığı Plain ASCON ile (ii) mesaj-başı anahtarların K_i sızdığı HGÇ–ASCON senaryolarını karşılaştırır (Çizelge 5.8). Ölçümler, Bulgular bölümünde raporlanan güç modeline (Due üzerinde P = 21,6 mW sabit) karşı makuliyet denetiminden geçirilmiş; enerji anomalisine rastlanmamıştır.

Çizelge 5.7. Saldırı deneyi 1: Arduino Due üzerinde kaba-kuvvet doğrulama (DFX kapalı)

Aygıt	AEAD doğrulama (µs / çevrim)	BF denemesi	BF ort. hız (/s)
Due @ 84 MHz	224 / 18,816	43,977	4,397.7

Not: Satır yalnızca doğrulama verimini raporlar (128-bit etiket/anahtar uzayında başarı yok). $\approx 4.4 \times 10^3$ deneme/s hızında, uniform 128-bit uzay için beklenen %50 başarı süresi astronomiktir ($ETA(50\%) \approx 1.2 \times 10^{27} yıl$).

Çizelge 5.8. Saldırı deneyi 2: Arduino Due üzerinde sızıntı izolasyonu (DFX kapalı)

Senaryo	AEAD doğrulama (µs / çevrim)	OK / 10,000	Süre / 10,000 (s)
Sade ASCON, K sızmış	224 / 18,816	10,000 / 10,000	2.270
HGÇ–ASCON, K_i sızmış	224 / 18,816	1 / 10,000	2.265

Not: 10,000 paket süreleri ölçülen ≈ 226.5 µs/paket değerini izler. Her paket doğrulandığından (erken çıkış yok), toplam süre yaklaşık 226.5 µs × N olarak ölçeklenir; N = 10,000 için ≈ 2.27 s olarak öngörülür. Sade tasarımda sızmış tek bir oturum anahtarı K tüm paketleri etkilerken; HGÇ altında sızmış tek bir mesaj-başı anahtar K_i yalnızca ilgili paketi etkiler ve zarar izolasyonunu gösterir.

Çalışmayla birlikte önerilen Hibrit Güvenlik Çerçevesi'nin uygulanabilir saldırı yöntemlerine karşı saldırgan etkinliği ve çerçevenin dayanımı, Çizelge 5.9'da özetlenmiş, ilgili yorumlar müteakkiben yapılmıştır.

Çizelge 5.9. Önerilen düzenek için saldırı dayanım özeti

Saldırı yöntemi	Saldırgan etkinliği	Dayanım
Kaba kuvvet	Çok Düşük	Çok Yüksek
Yan kanal	Düşük	Çok Yüksek
Zamanlama	Düşük	Yüksek
Donanım Manipülasyonu (HTH)	Orta	Yüksek
Dinamik Anahtar Çözümleme	Çok Düşük	Çok Yüksek
Sözlük Saldırısı	Düşük	Çok Yüksek

Saldırı türlerine göre notlar (Çizelge 5.9)

Kaba kuvvet: Mesaj-başı 128 bit anahtar ve 128 bit etiket; Due üzerinde ölçülen $\approx 4{,}4 \times 10^3$ deneme/s hızında arama uzayı pratikte erişilemez ($ETA(50\%) \approx 1{,}2 \times 10^{27} yıl$).

Yan kanal: Sabit frekans, sabit denetim akışı ve DFX kaynaklı yeniden konumlandırma güç/EM şablonlarını dengesizleştirir; kalıcı profil çıkarmayı zorlaştırır.

Zamanlama: Doğrulama yolu sabit zamanda ve sabit saatle yürür; veri-bağımlı dallanma/asgari-süre sızıntısı azaltılır.

Donanım manipülasyonu (HTH): Aygıta kilitli, şifreli bitstream'ler *(Device DNA bağlaması)* yetkisiz yeniden kullanımı zorlaştırır; ısrarlı yerleşim kurcalaması için PR kanalının ele geçirilmesi gerekir.

Dinamik anahtar çözümleme: $K_i = XOF(S_{ROOT}, dev, POOL[idx][, epoch, msg])$; tek bir K_i sızıntısı yalnızca ilgili paketi etkiler; mesajlar arası kestirim etkisizdir.

Sözlük: Kelime ya da düşük entropi bağımlılığı yoktur; çevrimdışı tahmin 128-bit AEAD anahtar uzayında kaba kuvvete indirgenir.

Yeniden konumlandırmanın rastgeleliği

Mesaj-başı slot-düzeyi yeniden konumlandırmada R kullanılabilir yuva ve M modülde tek-atım yerleşim sayısı $P(R,M) = \frac{R!}{(R-M)!}$ olup, yerleşim belirsizliği $H_{DFX} = \log_2\big(P(R,M)\big)$ bit'tir. M = 2 için R = 2 → 1.0 bit, R = 4 → 3.6 bit (P = 12), R = 10 → 6.5 bit (P = 90). Bu bitler 128-bit anahtar uzayından bağımsızdır ve toplanmamalıdır. Scraping/probing ve güç/EM şablon yeniden kullanımına karşı iş yükünü yükseltir.

Bölüm özeti

Arduino Due üzerinde ASCON-128a doğrulaması ≈ 226,5 µs/paket; 10.000 paket taramak ≈ 2,27 s sürer. Sade tasarımda sızmış bir oturum anahtarı K, 10.000/10.000 başarı verirken; HGÇ altında sızmış tek bir mesaj-başı anahtar K_i yalnızca 1/10.000 başarı üretir. Kaba-kuvvet verimi $\approx 4{,}4 \times 10^3 s^{-1}$ olduğundan 128 bit uzay pratikte erişilemez ($ETA \approx 10^{27} yıl$). Bu ampirik tablo, CryptoVerif ile doğrulanan biçimsel ayırt edilemezlik ve mesaj-başı anahtarlama varsayımlarıyla tutarlıdır (Blanchet, 2024).

5.6. Bulguların Özeti

Ölçümler, üretici (Zynq-7020) ve çözücü (Arduino Due) maliyetlerini ortak bir "mesaj-başı" çerçevede niceler. Üretici tarafında, PMU/Global-Timer zamanlamaları ve gözlenen 8,5–9,5 V yük aralığında plato-temelli şönt entegrasyonu ile aşama-başı süreler ve enerjiler elde edilmiştir (Çizelge 5.1). HGÇ–ASCON, yalnız kripto yolunu (KDF + AEAD) ≈ 12 µs'te tamamlarken; yalın ASCON-128a ≈ 6 µs'te tamamlar. Çizelge 5.1'deki karşılaştırma, 1×16 B yükte enerji sıralamasının gecikme ile aynı doğrultuda olduğunu gösterir: ACORN-128 ≫ JAMBU–PRESENT-128 ≳ TinyJAMBU-128 ≫ ASCON-128a. HGÇ–ASCON küçük bir KDF ek yükü getirse de mikro-saniye/µWh rejiminde kalır. Dinamik kısmi yeniden yapılandırma (DFX) ayrı, tek-atım bir yeniden konumlandırma penceresi olarak ortaya çıkar: ölçülen 4-RM takas ≈ 180,8 ms ve $E_\Delta \approx 11{,}75\ \mu Wh$ (mutlak ≈ 81,016 µWh) maliyetindedir; 2-RM tek-atım, aynı çalışma koşullarında ≈ 90,4 ms ve $E_\Delta \approx 5{,}875\ \mu Wh$ değerlerine ölçeklenir (Çizelge 5.3). Çözücü tarafında sabit P = 21,6 mW altında HGÇ–ASCON doğrulama/çözme süresi ≈ 962 µs olup alt-aşama enerjileri süreyle orantılıdır (Çizelge 5.2). Güvenlik değerlendirmeleri, mesaj-başı 128-bit anahtarın eksiksiz aramayı pratikte olanaksız kıldığını gösterir (Stallings, 2016); Arduino Due üzerinde ölçülen kaba-

kuvvet verimi ≈ 4,4 × $10^3 s^{-1}$'dir (Çizelge 5.7) ve 128-bit uzay için $ETA(50\%) \approx 10^{27}$ yıl kestirilir. Sızıntı deneyleri hasar izolasyonunu doğrular: sade tasarımda sızmış tek bir oturum anahtarı K tüm paketleri açarken, HGÇ altında sızmış tek bir mesaj-başı anahtar K_i yalnızca kendi paketini açar (Çizelge 5.8). Yan kanal dayanımı; sabit frekansta çalışma ve derli-toplu/sabit denetim akışlı AEAD doğrulama ile, kalıcı fiziksel yerleşimleri ortadan kaldıran DFX-temelli yeniden konumlandırmadan yarar sağlar; yerleşim belirsizliği, yuva sayısı R ile artar (örn. $H_{DFX} \approx 1{,}0;\ 3{,}6;\ 6{,}5\ bit$; R = {2,4,10}, M = 2); bu, anahtar entropusunu değiştirmeksizin scraping/probing iş yükünü yükseltir. Silikon kimliğine bağlı bitstream şifreleme (Device DNA), yetkisiz yeniden kullanımı engeller ve donanım manipülasyonuna karşı dayanımı destekler (Johnson, 2017; Stolz vd., 2021). Gözlemsel tablo, biçimsel analizlerle ortaya konan mesaj-başı anahtarlama ve ayırt edilemezlik varsayımlarıyla tutarlıdır (Blanchet, 2024). Çizelge 5.9'daki saldırı matrisi, ölçülen kaba-kuvvet verimi ve sızıntı-izolasyonu deneyleriyle uyumlu olup, belirtilen tehdit modeli altında yüksek dayanım göstermektedir.

Yöntemin güvenlik katkıları

(1) *Çok-boyutlu etki alanı ayrımı.* Anahtarlar; algoritma, aygıt, dilim ve (isteğe bağlı) mesaj boyutlarında dom ve (epoch, msg) ile ayrıştırılır.

(2) *Mesaj-başı anahtarlama (opsiyonel).* KDF'e (epoch, msg) dahil edilmesi, paket-başı ayrık anahtar üretir; maruziyeti yerelleştirir ve yanlış kullanım direncini kolaylaştırır.

(3) *İşletim sınırlarının nicemlenmesi.* Filo genelinde $Q/2\hat{}t$ sınırları, nonce politikası ve 64-bit blok veri sınırları performans sonuçlarıyla birlikte belirtilir; bu da risk-farkındalıklı dağıtımı mümkün kılar.

(4) *Şemalar arası dürüst raporlama.* Etiket boyu farkları (64 vs 128 bit) çizelgelerde açıkça işaretlenerek performansın güvenlik seviyesiyle hizalanması sağlanır.

6. TARTIŞMALAR

Değerlendirme, hafif anahtar türetimli mesaj-başı AEAD'in (HGÇ–ASCON) mikro-saniye ölçekli gecikme ve mikro-watt-saat ölçekli enerjiyle, olası ifşayı tek bir şifreli metinle sınırladığını göstermektedir. Üretici (Zynq-7020) tarafında, HGÇ–ASCON (yalnız kripto) ≈ 12 µs ve E ≈ 0,004589 µWh; yalın ASCON-128a ≈ 6 µs ve E ≈ 0,002294 µWh değerlerine ulaşır (Çizelge 5.1). Çözücü (Arduino Due) tarafında doğrulama/çözme maliyeti, sabit P = 21,6 mW altında zamanla doğrusal ölçeklenir (Çizelge 5.2). DFX, ayrı bir yeniden konumlandırma penceresi olarak ortaya çıkar: ölçülen 4-RM tek-atım ≈ 180,8 ms ve ΔE ≈ 11,75 µWh; 2-RM tek-atım ≈ 90,4 ms ve ΔE ≈ 5,875 µWh (Çizelge 5.3). Bu pencereler trafik göreli seyrek zamanlanabilir; böylece yeniden konumlandırma sıklığı tehdit modeline göre ayarlanabilir. Aşağıdaki başa-başa görünüm (Çizelge 6.1), yalın ASCON-128a ile HGÇ–ASCON (yalnız kripto) ve tek-atım 2-RM DFX içeren HGÇ–ASCON'u karşılaştırır.

Not: Daha önceki bir prototip, kod sayfası yükleyici için AES-128/CBC ile denenmiştir. Bu tezde raporlanan nihai ölçümler, Zynq-7020'nin PS (Cortex-A9) tarafında bütünlük ve yeniden-oynatma direnci için çalışan ASCON-128a (AEAD) + ASCON-XOF (KDF) ile yapılmıştır. Bu nedenle, önceki bant genişliği tabanlı süre üst-sınırları yalnızca sezgisel tavan olarak anılmış; tüm tablolar ve çıkarımlar doğrudan ölçüme dayandırılmıştır.

Çizelge 6.1. Değerlendirilen platformlarda başa-başa karşılaştırma

Kriter	ASCON-128a	HGÇ–ASCON	HGÇ–ASCON (+DFX, 2-RM)
Üretici gecikmesi (16 B)	≈ 6 µs	≈ 12 µs	≈ 12 µs + 90.4 ms (PR)
Üretici enerjisi	≈ 0.002294 µWh	≈ 0.004589 µWh	≈ 0.004589 µWh + 5.875 µWh (PR Δ)
Çözücü maliyeti (Due doğrulama)	≈ 710 µs (224 µs AEAD)	≈ 962 µs (476 ≈ µs KDF + AEAD)	Yalnız kripto ile aynı
Anahtarlama politikası	Uzun ömür/oturum anahtarı	Mesaj-başı (dom-ayrık)	Mesaj-başı (dom-ayrık)
Anahtar sızarsa hasar kapsamı	Çoklu paket	Tek paket	Tek paket
Fiziksel yerleşim istikrarı	Sabit	Sabit (DFX opsiyonel)	Hareketli hedef (BRAM relocation)
Uygulama farkı	—	+KDF, pool/index mantığı	+KDF/pool + DFX/bitstream'ler

Not: PR girdileri ölçülen 2-RM tek-atım penceresini yansıtır; 4-RM stres konfigürasyonu ≈ 180.8 ms ve 11.75 µWh ölçülmüştür (Çizelge 5.3). Yalnız kripto değerleri Çizelge 5.1, 5.2 ve 5.4'ten alınmıştır.

Güvenlik duruşu

Mesaj-başı 128-bit anahtarlar, tümden aramanın pratikte mümkün olmamasını sağlar (Stallings, 2016). Arduino Due üzerinde ölçülen kaba kuvvet doğrulama hızının $\approx 4{,}4 \times 10^3 s^{-1}$ olması, tekdüze bir 128-bit uzay için beklenen başarı süresinin $(ETA, \%50) \approx 10^{27}$ yıl olduğunu gösterir (Çizelge 5.7). Sızıntı deneyleri hasar yalıtımını doğrular: düz bir tasarımda sızdırılan oturum anahtarı, o oturumdaki tüm paketleri tehlikeye atarken, HGÇ altında sızdırılan bir mesaj-başı anahtar K_i yalnızca kendi paketini etkiler (Çizelge 5.8). Cihaza kilitli, şifreli bitstream'ler ve silikon kimliği (Device DNA), yetkisiz yeniden kullanımı sınırlandırır ve donanım kurcalamasına karşı koymaya yardımcı olur (Johnson vd., 2017; Stolz vd., 2021). Sabit frekanslı çalışma ve kompakt, sabit-kontrollü AEAD doğrulaması zamanlama varyansını azaltır; DFX, BRAM'de tutulan sırları ve seçili kod sayfalarını yer değiştirerek kalıcı güç/EM şablonlarını istikrarsızlaştırır. Ortaya çıkan tablo, biçimsel analizlerin desteklediği ayırt edilemezlik/mesaj-başı anahtarlama varsayımlarıyla uyumludur (Blanchet, 2024) ve daha önce önerilmiş DFX-tabanlı savunmaları—örneğin algoritma atlama ve enerji-uyarlamalı HSM'ler—tamamlar (Soliman vd., 2019; Samir vd., 2019).

Uygulanabilirlik

HGÇ–ASCON, düz ASCON'a kıyasla küçük bir XOF-KDF maliyeti ekler; ancak kısa paketler için aynı gecikme/enerji rejiminde kalır. Yeniden konumlandırma (relocation) ek yükü kripto patikasının dışındadır ve zamanlamayla amorti edilebilir ya da takvime bağlanabilir. Yaklaşım, diğer hafif AEAD'lerle (ör. ACORN, TinyJAMBU, JAMBU–PRESENT) uyumluluğunu korurken, mesaj-başı anahtarlama ve yeniden konumlandırmayı diklemesine sertleştirme katmanları olarak sunar. Statik yerleşimli ya da statik anahtarlı şifreleme algoritma tabanlarına kıyasla yöntem, makul bir hesaplama ek yükü ve ayarlanabilir bir yeniden konumlandırma maliyetiyle (Vipin ve Fahmy, 2018; Stolz vd., 2021) tehlike kapsamını mesaj düzeyine indirger ve literatürde belirtilen akademi-sanayi boşluğunu adresler.

AEAD temellerine karşı özet karşılaştırma

Çizelge 6.2, Zynq-7020 üzerinde yalnız kripto gecikme/enerjisini, değerlendirilen kurulumlardaki etkin tehlike kapsamıyla birlikte özetler; yeniden konumlandırma ek yükü Çizelge 5.3'de ayrı raporlanır.

Çizelge 6.2. Zynq-7020 (1×16 B): yalnız kripto gecikme/enerji ve ölçülen kompromi kapsamı

Yöntem	Gecikme (µs)	Enerji (µWh)	Verim sınıfı	Ölçülen kompromi kapsamı
HGÇ–ASCON-128a	12	0.004589	µs / µWh	Tek paket (mesaj-başı anahtar)
ASCON-128a	6	0.002294	µs / µWh	Oturumdaki tüm paketler
HGÇ–ACORN-128	505	0.193600	ms altı / 10^{-1} µWh	Tek paket (mesaj-başı anahtar)
ACORN-128	500	0.191189	ms altı / 10^{-1} µWh	Bu değerlendirmede oturum kapsamlı
HGÇ–JAMBU–PRESENT-128	372	0.142700	ms altı / 10^{-1} µWh	Tek paket (mesaj-başı anahtar)
JAMBU–PRESENT-128	366	0.139951	ms altı / 10^{-1} µWh	Bu değerlendirmede oturum kapsamlı
HGÇ–TinyJAMBU-128	178	0.068200	ms altı / 10^{-2} µWh	Tek paket (mesaj-başı anahtar)
TinyJAMBU-128	172	0.065769	ms altı / 10^{-2} µWh	Bu değerlendirmede oturum kapsamlı

Not: Kapsam, saldırı düzeneğindeki politika varsayımlarını yansıtır (sade AEAD: tek oturum anahtarı; HGÇ: mesaj-başı anahtar). AEAD'ler politika ile mesaj-başı da kullanılabilir; burada karşılaştırılabilirlik için sade sürümler oturum anahtarıyla ölçülmüştür. Enerji/gecikme değerleri Çizelge 5.1'den alınmıştır.

Güncel durum çalışmasıyla sentez

Çizelge 6.3, temsilî yaklaşımları karşıtlar. HGÇ satırı, ölçülen yalnız kripto rakamları ve 2-RM politika altındaki tek-atım DFX maliyetiyle (4-RM pencerenin yarısı; Çizelge 5.3) gösterilmiştir.

Çizelge 6.3. Farklı güvenlik yaklaşımlarının karşılaştırmalı analizi

Yöntem	Ölçülen Enerji/Güç Değerleri	Gecikme	Güvenlik Gücü	Üstünlükler	Sınırlılıklar
Algoritma sıçrama (Soliman vd., 2019)	Orta (≈10 mW dinamik)	Orta (≈10 ms/switch)	Orta	Şifre çevikliği; kaynak kullanımı verimli	Çoklu bitstream gerekir; yeniden yapılandırma yetkilendirmesi yok
Enerji-uyarlamalı HSM (Samir vd., 2019)	Orta (≤10 mW dinamik)	Yüksek (50–100 ms/switch)	Orta	Güç bütçesine uyum; çevik seçim	Bitstream/anahtar güvenliği yok; kullanılabilir gecikme açığı
LifeLine eş-bulanıklaştırma (Stolz vd., 2021)	Raporlanmamış (Alan/Zaman ek yükü odaklı - yüksek)	Yüksek (560 ms başlangıç)	Yüksek	Güçlü HW/SW bölüşümlü saklama; düşük kripto yükü	Güvenilir CPU gerektirir; dinamik güvenlik yok
Karanlık-silisyum bulanıklaştırma (Karam, 2016)	Yüksek (≈%2 ek)	Düşük (≈%13 gecikme ek)	Orta	Düşük alan ek yükü; uyumluluk	Statik anahtar; SCA'ya açık; çalışma zamanı esnekliği yok
DPReDO (Sunkavilli vd., 2022)	Raporlanmamış (Akım profili/zaman odaklı - yüksek)	Yüksek (PR için ~6 sn)	Yüksek	Gerçek-zamanlı Trojan azaltımı; düşük ek yük	Güvenli yeniden yapılandırma portu varsayar; prototip aşaması
HGÇ–ASCON (bu çalışma)	Çok yüksek (yalnız kripto ≈0.004589 µWh/msg; DFX 2-RM ΔE ≈ 5.875 µWh/msg)	Düşük (≈12 µs; + 90.4 ms PR)	Yüksek (AEAD + DFX)	Düşük güç; SCA ve MITM'e direnç; mesaj-başı hasar izolasyonu	DFX karmaşıklığı; şifreli pool depolaması; DFX enerji penceresi

Not: Sadece kripto işlemlerinin enerji harcaması (Zynq-7020): $E_{HSF}-ASCON \approx 0.004589\,\mu Wh$ (1×16 B). 2-RM tek-atım DFX değeri Çizelge 5.3'den türetilmiştir. Bu çizelgedeki literatür değerleri ağırlıkla Vivado/SAIF tabanlı PL-dinamik güç (mW) tahminleridir; bu çalışmadaki enerji (µWh/mesaj) ise 12V girişinden tüm kartın tükettiği toplam sistem enerjisidir. Dolayısıyla karşılaştırma, algoritmaların bire bir verimliliğinden ziyade, önerilen HGÇ yaklaşımının sistem seviyesindeki toplam verimliliğini vurgulamak amacıyla sunulmuştur.

Çizelge 6.3'te sunulan karşılaştırma, önerilen HGÇ yaklaşımının en dikkat çekici sonuçlarından birini ortaya koymaktadır. Çizelge notunda belirtildiği gibi, bu çalışmada ölçülen enerji değerleri tüm sistemin (PS, PL ve diğer çevre birimleri) tüketimini kapsarken, referans çalışmalardaki güç değerleri çoğunlukla yalnızca FPGA mantık bölümüne (PL)

odaklanmaktadır. HGÇ-ASCON'un mesaj başına harcadığı toplam sistem enerjisinin, diğer yaklaşımların sadece mantık devresi bölgesi (PL) için raporladığı dinamik güç seviyelerine göre dahi önemli ölçüde düşük olduğu görülmüştür. Bu, önerilen mimarinin yalnızca kriptografik olarak değil, aynı zamanda bütüncül bir sistem olarak da verimli olduğuna dair önemli bir somut göstergedir.

İlgili literatür notu

Önceki çalışmalar, gömülü/IoT bağlamında enerji bütçeleri ile anahtar yönetimi arasındaki gerilime vurgu yapar. Hafif ilkeller ve PRESENT sınıfı tasarımlar düşük alan/enerjiyi hedefler; ancak temel biçimlerinde sınırlı anahtar çevikliği sunarlar (Poschmann, 2009; Sadeghi vd., 2015). Genel amaçlı çekirdeklerde zamanlama/güç yan kanallarını sınırlamak için pratik sabit-zaman kodlama ve sızıntı azaltma teknikleri önerilir (Gueron, 2010; Nguyen vd., 2023). Tek-kullanımlık şerit/yalın XOR tarzı düzenler hesaplamayı önler; ancak anahtar dağıtımı ve yeniden-kullanım riskleri nedeniyle pratik değildir (Manifavas vd., 2015; Chen vd., 2023). Bu gözlemler, sıkı enerji kısıtları altında mesaj-başı AEAD anahtarları türetmek için küçük maliyetli bir XOF-tabanlı KDF kullanımını motive eder.

Sınırlılıklar ve gelecek çalışmalar

Bu çalışmanın bir sınırlılığı, 2-BRAM (2 RM) konfigürasyonuna ait DFX süre ve enerji değerlerinin doğrudan donanım üzerinde ölçülmek yerine, 4-BRAM (4 RM) konfigürasyonundan elde edilen verilerin doğrusal olarak ölçeklenmesiyle tahmin edilmiş olmasıdır. Kısmi bitstream boyutunun yarıya inmesiyle sürenin de yarıya ineceği varsayımı makul bir ilk yaklaşım olsa da, DFX sürecinin PCAP başlatma gibi sabit ek yükleri (overhead) nedeniyle gerçek donanım performansında küçük farklılıklar gözlemlenebilir. Gelecekteki çalışmalarda, farklı RM sayılarının DFX maliyetleri üzerindeki etkisinin doğrudan ölçülmesi, bu analizi daha da hassaslaştıracaktır.

Yeniden konumlandırma sıklığı ayarlanabilir bir maliyet-fayda sunar: daha yüksek sıklık şablon istikrarsızlığını artırır; karşılığında ek DFX pencereleri doğar; daha düşük sıklık ise maliyeti amorti eder. Mevcut düzenek kısa iletileri değerlendirir; daha büyük yüklerde, seçilen AEAD'in bayt-başı verimi baskın hâle gelirken HGÇ maliyeti çoğunlukla sabit KDF+AEAD kurulumundan kaynaklanır. Yeniden yapılandırma kanalının güçlendirilmesi

(örn. kimlik doğrulamalı PR kontrolü) elzemdir (Johnson vd., 2017; Vipin ve Fahmy, 2018). Sabit-zaman kodlamanın ötesindeki yan kanal sertleştirmeleri (ör. güç dengeleme, maskeleme) sızıntıyı daha da azaltabilir (Gueron, 2010; Nguyen vd., 2023). Uzun ömürlü kök sır için bir PUF kullanılacaksa, kimi "güçlü PUF" ailelerine yönelik modelleme saldırıları dikkate alınarak etkileşimsiz ve dikkatli bir hazırlama önerilir (Maes, 2013; Rührmair vd., 2013; Ning vd., 2020). Son olarak, her yeniden konumlandırmada BRAM içeriğinin yeniden şifrelenmesi (Eş. 4.6'da özetlendiği gibi) çevrimiçi protokolü değiştirmeden değişimler arası şifreli metinleri ilgisizleştirmek için pratik bir yol sunar.

7. SONUÇ VE ÖNERİLER

Mesaj-başı kimlik doğrulamalı şifreleme ile hafif anahtar türetimini (HGÇ–ASCON) ve BRAM'de yerleşik sırlar ile seçili kod sayfalarının DFX yardımıyla yeniden konumlandırılmasını birleştiren hibrit bir koruma katmanı, gömülü/IoT hedefleri için gösterilmiştir. Kriptografik çekirdek AEAD arayüzüne (anahtar, nonce, AAD → şifreli metin, etiket) uyar (McGrew, 2008; Bellare ve Namprempre, 2000; Krawczyk, 2001) ve alan ayrımıyla (domain separation) sünger-tabanlı bir XOF üzerinden her mesaj için taze bir anahtar türetir (Bertoni, vd., 2007; Bertoni vd., 2011; NIST, 2015). ASCON-128a ve ASCON-XOF, NIST Hafif Kriptografi süreci ve algoritmanın yayımlanmış analiziyle uyumlu olarak seçilmiştir (Dobraunig vd., 2021; Turan vd., 2023). XOF-KDF tasarımı, gizli bir kökün yanına kamusal bağlamın (örn. aygıt tanımlayıcısı, indeks) alınmasına izin veren standart uygulamayı izler; bu, HKDF ve NIST KDF önerileriyle tutarlıdır (Krawczyk ve Eronen, 2010; Barker vd., 2011; Chen, 2009). Nonce tekilliği 128-bitlik sayaç yerleşimiyle sağlanır; bu, AEAD kipleri için yaygın olarak benimsenmiş yönergelere karşılık gelir (Dworkin, 2007). Bu tercihler Kerckhoffs ilkesine uygundur: "Güvenlik, algoritmanın değil anahtarın gizliliğine dayanır (Kerckhoffs, 1883)."

Ölçümler, yöntemin mikro-saniye ölçeğinde gecikme ve mikro-watt-saat ölçeğinde enerjiyle, olası ifşayı tek bir şifreli metinle sınırladığını gösterir. Üreten tarafta (Zynq-7020), yalnız kripto HGÇ–ASCON ≈ 12 µs'te ve E ≈ 0,004589 µWh (1×16 B yük) ile tamamlanırken, düz ASCON-128a ≈ 6 µs'te ve E ≈ 0,002294 µWh ile tamamlanır (Çizelge 5.4). Zamanlamalar, üretici dokümantasyonuna göre Cortex-A9 üzerinde ARM Global Timer/PMU ile elde edilmiştir (ARM Ltd., 2010; Xilinx (AMD), 2018). Çözücü tarafta (Arduino Due), doğrulama/çözme maliyeti sabit P = 21,6 mW'de doğrusal ölçeklenir ve ≈ 962 µs ölçülür (Çizelge 5.2). DFX, ayrı bir yeniden konumlandırma penceresi olarak görünür: tek-atım ölçümler 4-RM için ≈ 180,8 ms ve ΔE ≈ 11,75 µWh; özdeş çalışma koşullarında 2-RM tek-atım için ≈ 90,4 ms ve ΔE ≈ 5,875 µWh'tir (Çizelge 5.3). Düz ASCON-128a, HGÇ–ASCON *(yalnız kripto)* ve 2-RM tek-atım DFX içeren HGÇ–ASCON karşılaştırması Çizelge 6.1'de sunulur. ACORN-128, TinyJAMBU-128 ve JAMBU–PRESENT-128 ile ek yalnız kripto karşılaştırmaları Çizelge 6.2'de verilmiştir (Wu, 2016; Wu ve Huang, 2016; Wu ve Huang, 2019; Bogdanov vd., 2007).

Güvenlik değerlendirmeleri, amaçlanan tehdit modeliyle uyumludur. Mesaj-başı 128-bit anahtarlar, tümden aramayı pratikte olanaksız kılar *(Stallings, 2016)*; Due üzerinde ölçülen $\approx 4{,}4 \times 10^3 s^{-1}$ doğrulama hızı, tekdüze bir 128-bit uzay için $(ETA, \%50) \approx 10^{27}$ yıl anlamına gelir (Çizelge 5.7). Sızıntı deneyleri hasar yalıtımını doğrular. Düz bir şifreleme algoritma tabanında sızan uzun ömürlü bir oturum anahtarı tüm paketleri açarken, HGÇ altında sızan bir mesaj-başı anahtar yalnızca kendi paketini açar (Çizelge 5.8). Sabit-zaman doğrulama ve sabit frekanslı çalışma zamanlama sızıntısını azaltır; DFX, BRAM'deki sırları ve seçili kod sayfalarını yer değiştirerek güç/EM şablonlarını *(hem de fazladan entropi ihtiyacı/iddiası olmaksızın)* istikrarsızlaştırır. Cihaza kilitli, şifreli bitstream'ler ve silikon kimliği, yetkisiz yeniden kullanımı sınırlar ve kısmi yeniden yapılandırma kanalının donanım kurcalamasına karşı güvenli tutulması gerekliliğinin altını çizer (Johnson vd., 2017; Stolz vd., 2021). Bu gözlemler, kabul görmüş AEAD güvenlik kavramları ve mühendislik uygulamalarıyla tutarlıdır (McGrew, 2008; Bellare ve Namprempre, 2000; Krawczyk, 2001; Dworkin, 2007).

Yaklaşım, yerleşik hafif AEAD'lerle bütünleşen HGÇ'nin mevcut yöntemlere ortagonal bir güvenlik katmanı eklediğini deneysel olarak göstermiştir. Bu çalışma, HGÇ çerçevesinin sadece ASCON-128a ile değil, aynı zamanda ACORN-128, TinyJAMBU-128 ve JAMBU–PRESENT-128 gibi farklı yapılardaki AEAD algoritmalarıyla da başarılı bir şekilde entegre edilebildiğini ve her biri için performans maliyetini ölçerek genel bir çözüm olduğunu kanıtlamıştır. HGÇ, getirdiği makul ek yük karşılığında, temel alınan AEAD ne olursa olsun, tehlike kapsamını mesaj granülerliğine indirgemektedir (Çizelge 6.2) (Wu, 2016; Wu ve Huang, 2016; Wu ve Huang, 2019; Bogdanov vd., 2007; Dobraunig vd., 2021; Turan vd., 2023). DFX zamanlaması, trafik yoğunluğuna seyrek uyarlanarak tehdide göre ayarlanabilir bir maliyet sunmuştur. Bu tasarım, daha önceki DFX-tabanlı savunmalarla *(algoritma atlama, enerji-uyarlamalı modüller, dinamik obfuscation)* uyumlu biçimde, birincil kriptografik kontrol olarak AEAD tabanlı kimlik doğrulamalı şifreleme ile mesaj-başı anahtarlamanın sentezini sağlamıştır (Soliman vd., 2019; Samir vd., 2019; Sunkavilli vd., 2022; Karam, vd., 2016; Stolz vd., 2021).

KAYNAKLAR

Abideen, Z. U., Gokulanathan, S., Aljafar, M. J., Pagliarini, S. (2024). An overview of FPGA-inspired obfuscation techniques. *ACM Computing Surveys*, 56(12), 299.

Adetomi, A., Enemali, G., Arslan, T. (2017). *Relocating Encrypted Partial Bitstreams by Advance Task Address Loading*. 2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Napa, CA, USA, 188-191.

Albartus, N., Hoffmann, M., Temme, S., Azriel, L., Paar, C. (2020). DANA: Universal dataflow analysis for gatelevel netlist reverse engineering. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(4), 309-336.

Al-Haija, Q. A., Enshasy, H., Smadi, A. (2017). *Estimating energy consumption of diffie hellman encrypted key exchange (DH-EKE) for wireless sensor network*. 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Srivilliputtur, India, 1-6.

Al-Meer, A., Al-Kuwari, S. (2023). Physical Unclonable Functions (PUF) for IoT devices. *ACM Computing Surveys*, 55(14s), 314.

AMD. (2023). *OS and Libraries Document Collection (UG643)*. Santa Clara, USA: AMD.

AMD. (2025). *Vivado Design Suite User Guide: Dynamic Function eXchange (UG909)*. Santa Clara, USA: AMD.

AMD. (2025). *Vivado Design Suite User Guide: Programming and Debugging (UG908)*. Santa Clara, USA: AMD.

ARM Ltd. (2010). *ARM Cortex-A9 MPCore Technical Reference Manual (r3p0)*, Cambridge, UK: ARM Ltd.

Azriel, L., Speith, J., Albartus, N., Ginosar, R., Mendelson, A., Paar, C. (2021). A survey of algorithmic methods in IC reverse engineering. *Journal of Cryptographic Engineering*, 11(3), 299-315.

Barker, E., Chen, L., Davis, R. (2011). Recommendation for key derivation through extraction-then-expansion (NIST Special Publication 800-56C). *National Institute of Standards and Technology*, Gaithersburg, MD.

Becher, A., Pirkl, J., Herrmann, A., Teich, J., Wildermann, S. (2016). *Hybrid energy-aware reconfiguration management on Xilinx Zynq SoCs*. 2016 International Conference on ReConFigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 1-7.

Bellare, M., Namprempre, C. (2000). *Authenticated encryption: Relations among notions and analysis of the generic composition paradigm*. Advances in Cryptology – ASIACRYPT 2000, Kyoto, Japonya, 531-545.

Bertoni, G., Daemen, J., Peeters, M., Van Assche, G. (2007). *Sponge functions*. ECRYPT Hash Workshop 2007.

Bertoni, G., Daemen, J., Peeters, M., Van Assche, G. (2011). *Duplexing the sponge: Single-pass authenticated encryption and other applications*. Selected Areas in Cryptography (SAC 2011), Toronto, Kanada, 320-337.

Biryukov, A., Shamir, A., Wagner, D. (2001). *Real time cryptanalysis of A5/1 on a PC*. Fast Software Encryption: 7th International Workshop, FSE 2000, New York, USA, 1-18.

Blanchet, B. (2024). *Dealing with Dynamic Key Compromise in Crypto Verif*. 2024 IEEE 37th Computer Security Foundations Symposium (CSF), Enschede, Netherlands, 495-510.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C. (2007). *PRESENT: An ultra-lightweight block cipher*. Cryptographic Hardware and Embedded Systems - CHES 2007, Viyana, Avusturya, 450-466.

Bommana, S. R., Veeramachaneni, S., Ershad, S., Srinivas, M. B. (2025). Mitigating side channel attacks on FPGA through deep learning and dynamic partial reconfiguration. *Scientific Reports*, 15, 13745.

Chakraborty, R. S., Saha, I., Palchaudhuri, A., Naik, G. K. (2013). Hardware Trojan Insertion by Direct Modification of FPGA Configuration Bitstream. *IEEE Design & Test*, 30(2), 45-54.

Chen, H., Che, M., Seiki, N., Shiramizu, T., Yano, T., Mikami, Y., Ueda, Y., Kato, K. (2023). Physically encrypted wireless transmission based on XOR between two data in terahertz beams. *Electronics*, 12(12), 2629.

Chen, L. (2009). Recommendation for key derivation using pseudorandom functions (NIST SP 800-108). *National Institute of Standards and Technology*, Gaithersburg, MD.

Ciylan, F., Ciylan, B., Atak, M. (2023). FPGA-based chaotic image encryption using systolic arrays. *Electronics*, 12(12), 2729.

Diffie, W., Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644-654.

Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M. (2021). Ascon v1.2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34(3), 33.

Dworkin, M. (2007). Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC (NIST SP 800-38D). *National Institute of Standards and Technology*, Gaithersburg, MD.

Ekdahl, P., Johansson, T. (2003). Another attack on A5/1. *IEEE Transactions on Information Theory*, 49(1), 284-289.

Ender, M., Moradi, A., Paar, C. (2020). *The Unpatchable Silicon: A Full Break of the Bitstream Encryption of Xilinx 7-Series FPGAs*. 29th USENIX Security Symposium (USENIX Security 20), 1803-1819.

Ender, M., Swierczynski, P., Wallat, S., Wilhelm, M., Knopp, P. M., Paar, C. (2019). *Insights into the mind of a trojan designer: the challenge to integrate a trojan into the bitstream*. Proceedings of the 24th Asia and South Pacific Design Automation Conference (ASPDAC '19), 112-119.

Engels, S., Hoffmann, M., Paar, C. (2022). A critical view on logic locking security. *Journal of Cryptographic Engineering*, 12(3), 229-244.

Faraj, M., Gebotys, C. (2021). Quiescent photonics side channel analysis: Low cost SRAM readout attack. *Cryptography and Communications*, 13(3), 363-376.

Gueron, S. (2010). *Intel Advanced Encryption Standard (AES) new instructions set (Rev. 3.01)*. Santa Clara, Intel Corporation.

Guin, U., Huang, K., DiMase, D., Carulli, J. M., Jr., Tehranipoor, M., Makris, Y. (2014). Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain. *Proceedings of the IEEE*, 102(8), 1207-1228.

Güneysu, T., Kasper, T., Novotný, M., Paar, C., Rupp, A. (2008). Cryptanalysis with COPACOBANA. *IEEE Transactions on Computers*, 57, 1498-1513.

İnternet: Kıyak, C.B. URL1: https://github.com/tkopter/proposd_hsf, Son Erişim Tarihi: 01.07.2024.

İnternet: Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N. (1999). Performance comparison of the AES submissions. *schneier.com*. URL2: https://www.schneier.com/wp-content/uploads/2016/02/paper-aes-performance.pdf, Son Erişim Tarihi: 01.01.2025.

Janakiraman, S., Vinoth Raj, R., Sivaraman, R., Sridevi, A., Upadhyay, H. N., Amirtharajan, R. (2025). Integrity-verified lightweight ciphering for secure medical image sharing between embedded SoCs. *Scientific Reports*, 15, 7465.

Johnson, A. P., Patranabis, S., Chakraborty, R. S., Mukhopadhyay, D. (2017). Remote dynamic partial reconfiguration: A threat to Internet-of-Things and embedded security applications. *Microprocessors and Microsystems*, 52, 131-144.

Karam, R., Hoque, T., Ray, S., Tehranipoor, M., Bhunia, S. (2016). *Robust bitstream protection in FPGA-based systems through low-overhead obfuscation*. 2016 International Conference on ReConFigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 1-8.

Kataria, J., Housley, R., Pantoga, J., Cui, A. (2019). *Defeating Cisco trust anchor: A case-study of recent advancements in direct FPGA bitstream manipulation*. 13th USENIX Workshop on Offensive Technologies (WOOT '19), Santa Clara, USA.

Kerckhoffs, A. (1883). La cryptographie militaire. *Journal Des Sciences Militaires*, 9, 5-38, 161-191.

Kerling, P. (2019). *Conceptual design and realization of a dynamic partial reconfiguration extension of an existing soft-core processor*. Yüksek Lisans Tezi, Technische Universität Ilmenau, Ilmenau.

Kharidu, H. K., Sudha, V. (2025). FPGA implementation of EEG based hardware optimized data encryption technique for IoT applications. *Integration*, 102, 102381.

Krawczyk, H. (2001). *The order of encryption and authentication for protecting communications (or: How Secure Is SSL?)*. Advances in Cryptology – CRYPTO 2001, Santa Barbara, USA, 310-331.

Krawczyk, H., Eronen, P. (2010). *HMAC-based extract-and-expand key derivation function (HKDF) (RFC 5869)*. Internet Engineering Task Force.

Maazouz, M., Toubal, A., Bengherbia, B., Houhou, O., Batel, N. (2022). FPGA implementation of a chaos-based image encryption algorithm. *Journal of King Saud University - Computer and Information Sciences*, 34(10), 9926-9941.

Maes, R. (2013). *Physically unclonable functions: Constructions, properties and applications*. Berlin/Heidelberg: Springer.

Manifavas, H., Hatzivasilis, G., Fysarakis, K., Papaefstathiou, Y. (2015). A survey of lightweight stream ciphers for embedded systems. *Security and Communication Networks*, 9(10), 1226-1246.

McGrew, D. (2008). An interface and algorithms for authenticated encryption (RFC 5116). *RFC Editor*.

Miller, V. S. (1985). *Use of elliptic curves in cryptography*. Conference on the Theory and Application of Cryptographic Techniques (CRYPTO'85), Santa Barbara, USA, 417-426.

Moradi, A., Barenghi, A., Kasper, T., Paar, C. (2011). *On the vulnerability of FPGA bitstream encryption against power analysis attacks: extracting keys from xilinx Virtex-II FPGAs*. Proceedings of the 18th ACM conference on Computer and communications security (CCS '11), 111-124.

Moradi, A., Oswald, D., Paar, C., Swierczynski, P. (2013). *Side-channel attacks on the bitstream encryption mechanism of Altera Stratix II: facilitating black-box analysis using software reverse-engineering*. Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays (FPGA '13), 91-100.

Moraitis, G., Dubrova, E. (2020). Bitstream modification attack on SNOW 3G. *Cryptology ePrint Archive*, Paper 2020/038.

Nabeel, N., Habaebi, M. H., Islam, M. D. R. (2021). Security Analysis of LNMNT-LightWeight Crypto Hash Function for IoT. *IEEE Access*, 9, 165754-165765.

National Institute of Standards and Technology [NIST]. (2015). SHA-3 standard: Permutation-based hash and extendable-output functions (FIPS 202). *U.S. Department of Commerce, NIST*, Gaithersburg, MD.

Nguyen, H., Hoang, T., Tran, L. (2023). Efficient hardware implementation of elliptic-curve Diffie–Hellman ephemeral on Curve25519. *Electronics*, 12(21), 4480.

Ning, H., Farha, F., Ullah, A., Mao, L. (2020). Physical unclonable function: Architectures, applications and challenges for dependable security. *IET Circuits, Devices & Systems*, 14(4), 407-424.

Poschmann, A. (2009). *Lightweight Cryptography - Cryptographic Engineering for a Pervasive World*. Cryptology ePrint Archive, Paper 2009/516.

Proulx, A., Chouinard, J.-Y., Fortier, P., Miled, A. (2023). A Survey on FPGA Cybersecurity Design Strategies. *ACM Transactions on Reconfigurable Technology and Systems*, 16(2), 1-33.

Raghunath, B. H., Aravind, H. S. (2023). An Efficient FPGA-Based Dynamic Partial Reconfigurable Implementation. *International Journal of Intelligent Systems and Applications in Engineering*, 11(1s), 183–192.

Rivest, R. L., Shamir, A., Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120-126.

Rührmair, U., Sölter, J., Sehnke, F., Xu, X., Mahmoud, A., Stoyanova, V., Dror, G., Schmidhuber, J., Burleson, W., Devadas, S. (2013). PUF modeling attacks on simulated and silicon data. *IEEE Transactions on Information Forensics and Security*, 8(11), 1876-1891.

Sadeghi, A.-R., Wachsmann, C., Waidner, M. (2015). *Security and privacy challenges in Industrial Internet of Things*. Proceedings of the 52nd Annual Design Automation Conference , San Francisco, CA, USA, 1-6.

Samir, N., Gamal, Y., El-Zeiny, A., Mahmoud, O., Shawky, A., Saeed, A. (2019). *Energy-Adaptive Lightweight Hardware Security Module using Partial Dynamic Reconfiguration for Energy Limited Internet of Things Applications*. 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 1-4.

Sasdrich, P., Moradi, A., Mischke, O., Güneysu, T. (2015). *Achieving side-channel protection with dynamic logic reconfiguration on modern FPGAs*. 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), Washington, DC, USA, 130-136.

Shafique, A., Naqvi, S. A. A., Raza, A., Ghalaii, M., Papanastasiou, P., McCann, J., Abbasi, Q. H., Imran, M. A. (2024). A hybrid encryption framework leveraging quantum and classical cryptography for secure transmission of medical images in IoT-based telemedicine networks. *Scientific Reports*, 14(1), 31054.

Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4), 656-715.

Silva, C., Cunha, V. A., Barraca, J. P., Aguiar, R. L. (2023). Analysis of the Cryptographic Algorithms in IoT Communications. *Information Systems Frontiers*, 26(4), 1243-1260.

Sivaranjani Devi, C., Amirtharajan, R. (2025). A novel 2D MTMHM based key generation for enhanced security in medical image communication. *Scientific Reports*, 15(1), 25411.

Soliman, S., Jaela, M. A., Abotaleb, A. M., Hassan, Y., Abdelghany, M. A., Abdel-Hamid, A. T., Salama, K. N., Mostafa, H. (2019). FPGA implementation of dynamically reconfigurable IoT security module using algorithm hopping. *Integration*, 68, 108-121.

Song, K., Zhu, Z., Yang, H., Ni, T., Xu, W. (2022). *MobileKey: A fast and robust key generation system for mobile devices*. Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers , Atlanta, USA and Cambridge, UK, 427–431.

Stallings, W. (2016). *Cryptography and network security: Principles and practice* (7. Baskı). Harlow, England: Pearson.

Stolz, F., Albartus, N., Speith, J., Klix, S., Nasenberg, C., Gula, A., Fyrbiak, M., Paar, C., Güneysu, T., Tessier, R. (2021). LifeLine for FPGA protection: Obfuscated cryptography for real-world security. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(4), 412–446.

Sunkavilli, S., Chennagouni, N. G., Yu, Q. (2022). *DPReDO: Dynamic Partial Reconfiguration enabled Design Obfuscation for FPGA Security*. 2022 IEEE 35th International System-on-Chip Conference (SOCC), Belfast, UK, 1–6.

Swierczynski, P., Becker, G. T., Moradi, A., Paar, C. (2016). *Bitstream Fault Injections (BiFI) – Automated Fault Attacks against SRAM-based FPGAs*. Cryptology ePrint Archive, 2016/641.

Turan, M. S., McKay, K., Chang, D., Bassham, L. E., Kang, J., Waller, N. D., Kelsey, J. M. (2023). Status report on the final round of the NIST Lightweight Cryptography Standardization Process (NIST IR 8454). *National Institute of Standards and Technology*, Gaithersburg, MD.

Van Herrewege, A. (2015). *Lightweight PUF-based key and random number generation.* Doktora Tezi, KU Leuven, Leuven.

Vipin, K., Fahmy, S. A. (2011). *Efficient region allocation for adaptive partial reconfiguration*. 2011 International Conference on Field-Programmable Technology, New Delhi, India, 1-6.

Vipin, K., Fahmy, S. A. (2018). FPGA dynamic and partial reconfiguration: A survey of architectures, methods, and applications. *ACM Computing Surveys*, 51(4), 1–39.

Wanderley, E., Vaslin, R., Crenne, J., Cotret, P., Gogniat, G., Diguet, J.-P., Danger, J.-L., Maurine, P., Fischer, V., Badrignans, B., Barthe, L., Benoit, P., Torres, L. (2011). Security FPGA Analysis. *Security Trends for FPGAS: From Secured to Secure Reconfigurable Systems* (s. 7-46). Dordrecht: Springer Netherlands.

Wei, Z., Cui, Y., Chen, Y., Wang, C., Gu, C., Liu, W. (2020). *Transformer PUF: A Highly Flexible Configurable RO PUF Based on FPGA*. 2020 IEEE Workshop on Signal Processing Systems (SiPS), Coimbra, Portugal, 1-6.

Wu, H. (2016). *ACORN v3: Lightweight authenticated cipher*. CAESAR Competition Submission (Round 3).

Wu, H., Huang, T. (2016). *JAMBU: A lightweight authenticated encryption mode (v2.1)*. CAESAR Competition Submission (Round 3).

Wu, H., Huang, T. (2019). *TinyJAMBU: A family of lightweight authenticated encryption algorithms*. NIST LWC Submission (Round 1).

Xia, E., Hu, B.-J., Shen, Q. (2024). A survey of physical layer secret key generation enhanced by intelligent reflecting surface. *Electronics*, 13(2), 258.

Xiao, Q., Zhao, J., Feng, S., Li, G., Hu, A. (2024). Securing NextG networks with physical-layer key generation: A survey. *Security and Safety*, 3, 2023021.

Xilinx. (2018). *Zynq-7000 All Programmable SoC Technical Reference Manual (UG585)*. San Jose, CA: Xilinx.

Yao, L., Liang, H., Han, Q., Zhang, H., Huang, Z., Jiang, C., Yi, M., Lu, Y. (2022). M-RO PUF: A portable pure digital RO PUF based on MUX unit. *Microelectronics Journal*, 119, 105314.



Gazili olmak ayrıcalıktır