University of West Attica

School of Engineering

Department of Informatics and Computer Engineering

Post-Graduate Studies Programme: CYBERSECURITY

Hardware-based Security Methods for Internet of Things (IoT), Internet of Everything (IoE) & Cyber-Physical Systems (CPS)

Tsagdis Anastasios Nikolaos

M.Sc. Thesis

Supervisor: Dr. Emmanouil T. Michailidis, *Adjunct Lecturer*

Egaleo, September 2022

Πανεπιστήμιο Δυτικής Αττικής

Σχολή Μηχανικών

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Πρόγραμμα Μεταπτυχιακών Σπουδών: ΚΥΒΕΡΝΟΑΣΦΑΛΕΙΑ

Hardware-based Security Methods for Internet of Things (IoT), Internet of Everything (IoE) & Cyber-Physical Systems (CPS)

Μέλη Εξεταστικής Επιτροπής συμπεριλαμβανομένου και του Εισηγητή

Η μεταπτυχιακή διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή:

Α/Α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1	**Εμμανουήλ Θ. Μιχαηλίδης**	**Ακαδημαϊκός Υπότροφος** Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών, Πανεπιστήμιο Δυτικής Αττικής/ Εισηγητής – Επιβλέπων	Emmanouil Michailidis Digitally signed by Emmanouil Michailidis Date: 2022.09.23 22:20:27 +03'00'
2	**Παναγιώτης Γιαννακόπουλος**	**Καθηγητής** Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών, Πανεπιστήμιο Δυτικής Αττικής / Μέλος Εξεταστικής Επιτροπής	Panagiotis Giannakopoulos Digitally signed by Panagiotis Giannakopoulos Date: 2022.09.29 12:23:56 +03'00'
3	**Χαράλαμπος Ζ. Πατρικάκης**	**Καθηγητής** Τμήμα Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών, Πανεπιστήμιο Δυτικής Αττικής / Μέλος Εξεταστικής επιτροπής	Digitally signed by Charalampos Patrikakis Date: 2022.09.24 00:36:55 +03'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Αναστάσιος Τσαγδής** του **Νικολάου**, με αριθμό μητρώου **Cscyb19022** φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών «**ΚΥΒΕΡΝΟΑΣΦΑΛΕΙΑ»** του Τμήματος **Μηχανικών Πληροφορικής και Υπολογιστών** της Σχολής **Μηχανικών** του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

ANASTASIOS TSAGDIS
23.09.2022 22:10

(Υπογραφή)

Acknowledgements

I would like to express my special appreciation and thanks to my advisor Dr. Emmanouil T. Michailidis, who has been a tremendous mentor to me.

Dr Michailidis, I would like to thank you for encouraging my research, for your patience, and motivation, for the useful comments, remarks and immense knowledge through the learning process and writing of this thesis. I would especially like to thank my good friend and former English language professor Io, whose valuable input and guidance affected my desire to write this Thesis in English.

Furthermore, I would like to thank all my friends, classmates and co-workers who have supported me in writing, and incited me to strive towards my goal over the last few months.

Special thanks to my family. Words cannot express how grateful I am to my mother-in-law, father-in-law, sister-in-law, my mother, father, and brother for all of the sacrifices that they have made on my behalf. Their prayers for my success were what sustained me thus far.

These acknowledgements would not be complete without thanking the most two important persons in my life for their constant support and care. That is why I would like to express appreciation to my beloved wife, Niki, who spent sleepless nights with and has always been my support in the moments when there was no one to answer my queries, also my little son Petros-Aggelos for making me so happy with his cute smile and through his good nature allowing me to carry out a lot of work at home.

Abstract

Hardware security has emerged as a prominent study area in the last decade, drawing academics, businesses and governments alike. Electronic data processing now controls documents and nearly all aspects of our lives. Bits become as important as actual resources on this planet. In today's information world, hardware security encompasses many essential criteria, including proper handling and storing electronic data anywhere, anytime, and efficiently using resources and energy.

Nowadays rapid technological growth has changed the way interactions, interfaces, communication, and electrical systems interact. This has resulted in us switching from the interconnection and communication of isolated devices to the ubiquitous things that, with the help of the internet, interact with each other and produce data for the extraction of information. This highly integrated global network structure is called the Internet of Things (IoT).

The Internet of Things (IoT) superset is called Everything (IoE) and defines the connection between people, processes, data and things. It unites all these ideas into a single world. A network intelligent system that connects people, things, and intelligent devices to share information and services. All cyber technologies (computing and communication) become a component of more sophisticated systems known as cyber-physical systems (CPS) when they are coupled with the physical environment. The cyber-physical system is one of the key technologies for establishing the Internet of Things (CPS).

Its components (IoT, IoE, and CPS) are critical infrastructure parts that must maintain its safety, dependability, and security while operating in real time. Their security is more challenging than standard IT systems. They must address security and privacy due to the variety of technologies. Traditional security primitives cannot be simply applied to IoT because of the diverse standards and communication stacks involved. A cyber-physical adversary can compromise a system. The sensitivity of the control system and the data it contains determines the level of security based on each cyber-physical system. They must consider aspects from both a software and hardware perspective.

IT security is critical for future tech progress. This research's main focus is on IoT, IoE, and CPS system security.

Περίληψη

Η ασφάλεια υλικού έχει αναδειχθεί σε εξέχουσα περιοχή μελέτης την τελευταία δεκαετία, προσελκύοντας ακαδημαϊκούς, επιχειρήσεις και κυβερνήσεις. Η ηλεκτρονική επεξεργασία δεδομένων ελέγχει πλέον τα έγγραφα και σχεδόν όλες τις πτυχές της ζωής μας. Αυτός ο κόσμος διέπεται πλέον τόσο από bits όσο και από φυσικούς πόρους. Στον σημερινό κόσμο της πληροφορίας, η ασφάλεια υλικού αποτελεί βασικό μέλημα των μηχανικών για την διασφάλιση της ακεραιότητας, εγκυρότητας και μετάδοσης των ψηφιακών στοιχείων.

Στις μέρες μας η ραγδαία τεχνολογική ανάπτυξη έχει φέρει στην ανθρωπότητα πολλές αλλαγές στον τρόπο επικοινωνίας και αλληλεπιδράσεων των ηλεκτρικών συστημάτων. Αυτό έχει ως αποτέλεσμα τη μετάβαση από τη διασύνδεση και την επικοινωνία μεμονωμένων συσκευών στα πανταχού παρόντα πράγματα που με τη βοήθεια του διαδικτύου, αλληλοεπιδρούν μεταξύ τους και παράγουν δεδομένα για την εξαγωγή και διανομή πληροφοριών. Το κέρδος για τον πολιτισμό μας, αυτής της εξαιρετικά ολοκληρωμένης παγκόσμιας δομής δικτύου, ονομάζεται Internet of Things (IoT).

Το υπερ-σύνολο IoT αποκαλείται Internet of Everything (IoE) και ορίζει τη σύνδεση μεταξύ ανθρώπων, διαδικασιών, δεδομένων και πραγμάτων. Ενώνει όλες αυτές τις ιδέες σε έναν ενιαίο κόσμο. Το IoE αποτελεί τον πυλώνα των IoT και περικλείει ένα ευφυές σύστημα δικτύου μέσω του οποίου συνδέονται άνθρωποι, πράγματα και έξυπνες συσκευές και μπορούν να μοιράζονται πληροφορίες και υπηρεσίες. Όταν όλες οι τεχνολογίες του κυβερνοχώρου (υπολογιστές και επικοινωνία) ενσωματώνονται στον φυσικό κόσμο, γίνονται μέρη πιο περίπλοκων συστημάτων που ονομάζονται Cyber-Physical Systems (CPS). Το CPS αποτελεί μία από τις βασικές τεχνολογίες για την υλοποίηση του Διαδικτύου των Πραγμάτων (IoT).

Τα στοιχεία (φυσικά υλικά) των IoT, IoE και CPS είναι κρίσιμα τμήματα υποδομής που πρέπει να λειτουργούν με αξιοπιστία και ασφάλεια σε πραγματικό χρόνο. Η ασφάλειά τους είναι πιο απαιτητική από αυτήν των τυπικών συστημάτων πληροφορικής. Έχουν να αντιμετωπίσουν την ασφάλεια και την ιδιωτικότητα της μεταδιδόμενης πληροφορίας σε μια μεγάλη ποικιλία τεχνολογιών. Οι παραδοσιακές πρωτόγονες αρχές ασφαλείας δεν μπορούν να εφαρμοστούν στο IoT λόγω των διαφορετικών προτύπων και των συστοιχιών επικοινωνίας που εμπλέκονται. Οι επίδοξοι εισβολείς μπορούν να θέσουν, ποικιλοτρόπως, σε κίνδυνο, τέτοια σύστηματα. Η ευαισθησία του συστήματος ελέγχου και των δεδομένων που περιέχονται στο υλικό καθορίζουν το επίπεδο ασφάλειας στο κάθε κυβερνο-φυσικό σύστημα. Τόσο οι προγραμματιστές όσο και οι αναλυτές απαιτήσεων, πρέπει να αξιολογούν τις πτυχές όχι μόνο από άποψη λογισμικού αλλά και από πλευράς υλικού.

Η ασφάλεια υλικού στα συστήματα πληροφορικής είναι κρίσιμη για τη μελλοντική τεχνολογική πρόοδο. Η κύρια εστίαση αυτής της έρευνας είναι η ασφάλεια υλικού στα συστήματα IoT, IoE και CPS.

Contents

Acknowledgements i

Abstract ii

Περίληψη iii

Contents vi

List of Figures x

List of Tables xiii

Abbreviations xiv

Chapter1 1

1. Introduction 2

1.1. Preface 2

1.1.1. Definitions & Differences 4

1.2. Hardware Security 9

1.2.1. Preface 9

1.2.2. Hardware Trust 11

1.2.3. Vulnerabilities and Countermeasures Against Attacks 14

1.2.4. Conflicts 17

1.2.5. Summarizes 17

Chapter 2 20

2. Internet of Things 21

2.1. Preface 21

2.1.1. Infrastructure 22

2.1.2. Traditional IT Security vs IoT Security 26

2.2. Security Challenges & Attacks 27

2.2.1. General 27

2.2.2. Hardware Security Challenges 32

2.2.3. Attacks 36

2.2.4. Classification of IoT Security Attacks 40

2.3. Hardware-based Security 43
2.3.1. General 43
2.3.2. Fake Replica 43
2.3.3. Side-Channel Attack 43
2.3.4. Reverse Engineering (RE) 44
2.3.5. Intellectual Property (IP) Hijacking 45
2.3.6. Trojans in Hardware 45
2.4. Hardware Trojans 46
2.4.1. Hardware Trojan Taxonomy 46
2.4.2. Insertion of a Hardware Trojan 48
2.4.3. The use of Hardware Trojans in Side Channel Attacks 50
2.4.4. Countermeasures 51
2.4.5. Multi-layer hardware Trojan protection framework (called RG-secure) 55
2.4.6. Security Technologies for Industrial IoT (Security Controller) 55
2.5. Physically Unclonable Functions (PUFs) 58
2.5.1. Types 60
2.5.2. PUF-Based hardware security solutions 63
2.5.3. Design challenges 65
2.5.4. PUF-Based Threats on IoT Devices 67
2.5.5. Using PUFs on IoT 77
2.6. Additional Hardware Security Methods for IoTs 79

Chapter 3 81

3. Internet of Everything (IoE) 82

3.1. Preface 82
3.2. Hybrid PUF 84
3.2.1. Preface 84
3.2.2. Security analysis 84
3.3. BlockChain 85
3.3.1. Preface 85
3.3.2. The need for both Device and Data Security in IoEIn most applications, communication devices are used more. 86
3.3.3. Communication & Network Security 91
3.3.4. Identity management & authentication 91
3.3.5. Reliable distributed consensus protocol 92

3.3.6. Decentralized cooperation & trust establishment 93
3.3.7. Types of BlockChain 93
3.3.8. Challenges 95
3.3.9. PUF Integration in Blockchain 97
3.3.10. PUFchain as a solution in Blockchain 97
3.3.11. Proposed novel POP 98
3.4. Smart Grid Metering Infrastructure with IoE 100
3.4.1. Preface 100
3.4.2. Smart Grid overview 101
3.4.3. AMI security requirements 105
3.4.4. Privacy in AMI 105
3.4.5. AMI privacy related works 106
3.5. Hardware devices and architecture for securing the IoE 108
3.5.1. Circuits, Devices and architectures for securing the IoE 108
3.5.2. Field-programmable things array (FPTA) 109

Chapter 4 111

4. Cyber-Physical Systems (CPS) 112

4.1. Preface 112
4.2. Cyber Physical Security overview 113
4.2.1. General 113
4.2.2. CPS workflow 113
4.2.3. Characteristics of Adversaries 118
4.3. Context-Wise security framework 122
4.3.1. Preface 122
4.3.2. Proposed CPS Security Requirements Engineering Framework 125
4.3.3. SRE Activities 126
4.4. Hardware Security for CPS 127
4.4.1. Security Issues around Sensor 128
4.4.2. Hardware-based exploits 129
4.4.3. Hardware Security Primitives and Countermeasures 130
4.4.4. PLC attacks 133
4.4.5. Device Fingerprinting 133

Chapter 5 135

5. Conclusions and Future Directions 136

5.1. Future Directions 136

5.1.1. IoT 136

5.1.2. IoE 138

5.1.3. CPS 140

5.2. Conclusions 141

References 146

List of Figures

Figure 1: Hardware security has evolved in recent decades.[3] 2
Figure 2: IoT presentation 5
Figure 3: IoE presentation 6
Figure 4: Security and trust in hardware [19] 11
Figure 5: Significant stages in the design and testing of electronic hardware [19] 12
Figure 6: Attack vectors and countermeasures for each IC phase [19] 13
Figure 7: SoC design is vulnerable to trust/integrity issues due to its long, globally distributed IP supply chain. [19] 14
Figure 8: Current state of system-on-chip security design and validation [19] 16
Figure 9: Examples of IoT applications [29] 21
Figure 10: IoT Trust Pyramid [31] 22
Figure 11: a piece of IoT hardware that detects and/or controls a "Thing"[31] 24
Figure 12: An attacker in an IoT environment [38] 28
Figure 13: Domains and security in IoT [38] 30
Figure 14: IoT Systems with Side Channel Attack (SCA) [31] 35
Figure 15: IoT attacks. The highlighted path highlights PUFs' IoT security focus.[38] 37
Figure 16: A man-in-the-middle attack [38] 39
Figure 17: IoT structure [38] 41
Figure 18: ICs are manufactured from design to application.[38] 43
Figure 19: Hardware-based attacks on semiconductor manufacturing entities [38] 45
Figure 20: Hardware taxonomy Trojan [90] 47
Figure 21: Hardware Trojan attacks on [91] 48
Figure 22: Hardware Trojan Detection. [91] 51
Figure 23: Comparing 3 architectures. Left: snapshot generation without extra security, TrustZone-based, and Security Controller-based [224] 57
Figure 24: A hybrid approach combining TrustZone and Security Controller [224] 58
Figure 25: PUF authentication modes [227] 59
Figure 26: A PUF-Based Authentication Process [240] 64
Figure 27: A PUF-Based Encryption Scheme [240] 65

Figure 28: A PUF-Based Decryption Scheme [240] 65
Figure 29: PUF Circuit Reliability in 65nm Technology with Aging CMOS [242] 66
Figure 30: Temperature Effects on PUF Circuit Reliability in 65 nm Technology [240] 67
Figure 31: PUF architecture [227] 70
Figure 32: Architecture based on the Arbiter PUF [255] 71
Figure 33: Architecture for a Ring Oscillator [24-7] 72
Figure 34: SRAM PUF structure [227] 73
Figure 35: TERO PUF constructions [262] 73
Figure 36: PUF protocols [227] 75
Figure 37: Obfuscated PUF structure [267] 77
Figure 38: Internet of Everything (IoE) [176] 82
Figure 39: Internet of Everything (IoE) [177] 83
Figure 40: Hybrid PUF Architecture [189] 84
Figure 41: Uses for blockchain technology [195] 86
Figure 42: Human-centric IoE vision requires device, person, location, and data privacy [194] 88
Figure 43: Blockchain-based IoT architecture [201] 89
Figure 44: Different types of Blockchain [195] 94
Figure 45: Blockchain consensus algorithms [194] 95
Figure 46: Issues with blockchain technology [194] 96
Figure 47: PUFchain working model [194] 96
Figure 48: PoP enrollment and authentication. (a) Device enrollment. (b) Steps. (c) Verification [194] 98
Figure 49: PUF Working Principle [206] 99
Figure 50: Smart Grid overview [208] 102
Figure 51: AMI metering infrastructure [208] 103
Figure 52: Privacy in AMI [208] 106
Figure 53: Related work hierarchy [208] 107
Figure 54: Field-programmable things array (FPTA) block diagram 110
Figure 55: Abstraction of CPS [113] 114
Figure 56: Security goals & threats [111] 115

Figure 57: Attacks [113] 116
Figure 58: Defining CPS security threats [133] 119
Figure 59: Solutions for CPS's security and privacy [133] 121
Figure 60: The context-aware security framework [113] 122
Figure 61: General context-aware security workflow [113] 123
Figure 62: Main security aspects [113] 124
Figure 63: SRE Framework for CPS [111] 126
Figure 64: Security issues [111] 129

List of Tables

Table 1: Differences between IoE & IoT [15] 7
Table 2: Summarizes Hardware attacks & countermeasures [19] 18
Table 3: Traditional IT security vs IoT security [36] 27
Table 4: Security differences between hardware and software [38] 29
Table 5: Types of Hardware attacks [31] 33
Table 6: Attacks classified by TCP/IP layer [38] 38
Table 7: Attack taxonomy for IoT layers [38] 42
Table 8: Untrusted party Trojan Models [91] 50
Table 9 : Memory technology needs in the present and the future [234-236] 61
Table 10: Comparison of PUFs [38] 61
Table 11: Examining the literature's various fuzzy extractor schemes [38] 63
Table 12: PUF intra-HD, inter-HD, FAR, and FRR comparison [189] 85
Table 13: Distributed IoT security threats and solutions [201] 90
Table 14: Comparison of consensus protocol types [204] 92

Abbreviations

Abbreviation	Meaning
3PIP	third-Party IP
AC	Access Complexity
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
AIK	Attestation Identity Key
AMI	Advanced Metering Infrastructure
AMR	Automated Meter Reading
API	Application Programming Interface
ARM	Advanced RISC Machine
ARNG	Random Number Generator
ASIC	Application-Specific Integrated Circuit
ATPG	Automatic Test-Pattern Generation
AV	Access Vector
BCH	Bose-Chaudhuri-Hocquenghem
BE	Best Effort
BEOL	Back End of Line
BFT	Byzantine Fault Tolerance
BISA	Built-In Self-Authentication
BLH	Battery-based Load Hiding
BMS	Battery Management System
BS	Based Score
CA	Certificate Authority
CAD	Computer-Aided Design

CDMA	Code Division Multiple Access
CEN	Comit Europen Normalisation
CEP	Complex Event Processor
CIA	Confidentiality, Integrity & Availability
CoAP	Constrained Application Protocol
CORDIC	Coordinate Rotation Digital Computer
COTS	Commercial-Off-The-Shelf
CPS	Cyber Physical Systems
CPU	Central Processing Unit
C-PUF	Controlled PUF
CRP	Challenge Response Pair
CUT	Circuit-Under-Test
CVSS	Common Vulnerability Scoring System
DAC	Digital-to-Analog Converter
DERs	Distributed Energy Resources
DES	Data Encryption Standard
DFA	Differential Fault Analysis/Attack
DFD	Design-For-Debug
DfS	Design-for-Security
DfT	Design-for-Test
DICE	Device Identifier Composition Engine
DMS	Distribution Management Systems
DoS	Denial of Service
DPA	DIfferential Power Analysis
DPoS	Delegated Proof of Stake
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EDA	Electronic Design Automation

EEPROM	Electrically Erasable Programmed Read-Only Memory
EGT	Electrolyte-Gated Transistors
EMA	Electromagnetic Analysis Attacks
EMS	Energy Management Systems
ETSI	European Telecommunications Standards Institute
FEOL	Front End of Line
FPTA	Field-Programmable Things Array
FSM	Finite State Machine
GC	Grouping and Choosing
GPU	Graphics Processing Unit
HAN	Home Area Network
HD	Hamming Distance
HDL	Hardware Description Language
HIP-DE/HIP-DEX	Host Identity Protocol-Diet Exchange
HSM	Hardware Security Modules
HT	Hardware Trojan
I/O	Input/Output
IANA	Internet Assigned Numbers Authority
IC	Integrated Circuit
IDoT	IDentity of Things
IEEE	Electrical and Electronic Engineers
IMD	Implantable Medical Devices
IoD	Internet of Digital
IoE	Internet of Everything
IoH	Internet of Human
IoT	Internet of Things
IP	Intellectual Property
LC	Leaky Circuit

LDPC	Low Density Parity Check
LLN	Low-Power Lossy Network
LoWPAN	Low-Power Wireless Personal Area Network
LP	Linear Programming
LP-WAN	Low-Power Wide-Area Network
LRR-DPUF	Learning Resilient & Reliable Digital PUF
LR-WPAN	Low-Rate Wireless Personal Area Network
LS	Lazy Stepping
M2M	Machine to Machine communication
MAC	Medium Access Control
MDMS	Data Management System
MERO	Multiple Excitation of Rare Occurrence
MF-R	Meta-Fog Redirection
MitM	Man in the Middle
ML	Machine Learning
MPU	Micro Processor Unit
MQTT	MQ Telemetry Transport
MRAM	Magneto Resistive Random-Access Memory
MTU	Maximum Transmission Unit
NAN	Neighborhood Area Network
NILL	Non-Intrusive Load Leveling
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
OB-PUF	OBfuscated PUF
OS	Operating System
OSN	On-line Social Network
P2P	Peer-to-Peer
PCB	Printed Circuit Board

PCH	Proof Carrying Hardware
PCR	Platform Configuration Registers
PE	Processing Elements
PKI	Public Key Infrastructure
PLC	Power Line Communication
PMU	Phasor Measurement Units
PoAh	Proof of Authentication
PoP	Proof of PUF
PoS	Proof-of-Stake
PoW	Proof-of-Work
PPUF	Public PUF
PRNG	Pseudo Random Number Generator
PUF	Physically unclonable functions
PWM	Pulse Width Modulation
RBAC	Role Based Access MANAGEMENT
RE	Reverse Engineering
RFID	Radio-frequency identification
RISC	Reduced Instruction Set Computer
RO	Ring Oscillator
RO	Ring Oscillator
ROM	Read-Only Memory
ROT	Roots of Trust
RSA	Rivest-Shamir-Adleman
RTL	Register Transfer Level
RTL	Register Transfer Level
SC	Security Controller
SCA	Side-Channel Analysis
SCADA	Supervisory and Data Acquisition System

SD	Secure Digital
SDLC	Software Development Life Cycle
SDN	Software Defined Network
SEA	SDN-based Evolution Architecture
SEM	Scanning Electron Microscope
SGIP	Smart Grid Interoperability Panel
Si	Silicon
SM	Smart meters
SOA	Service Oriented Architecture
SoC	System on Chip
SPA	Simplified Power Analysis
SRAM	Static Random-Access Memory
SS	Secure Sketch
SS	Secure Sketch
SSD	Solid-State Drive
TCB	Trusted Computing Base
TCG	Trusted Computing Group
TCPA	Trusted Computing Platform Alliance
TDC	Time-to-Digital Converters
TEE	Trusted Execution Environment
TERO	Transient Effect Ring Oscillator
TPM	Trusted Platform Module
TRNG	True Random Number Generator
TSA	Trojan Side Channels
TTP	Trusted Third Parties
UDS	Unified Diagnostic Service
UUID	Universally Unique IDentification
VR	Virtual-Reality

WAN	Wide Area Network
WBC	White Box Cryptography
WMD	Wearable Medical Devices

Chapter1

Introduction

1. Introduction

1.1. Preface

Over the last three decades, several hardware vulnerabilities and attacks have been discovered. Figure 1 shows the evolution of hardware security. Before 1996, hardware IP piracy was rare, mostly through IC cloning, IP watermarking, anti-piracy measures. The timing analysis attack [2] was introduced in 1996. This attack examines computation time to extract cryptographic hardware information. 1997 saw the first fault injection attack through analysis. [3]. The objective of the attack is to stress the system to undue strain and coerce it into leaking sensitive data. 1999 marked the publication of the first attack on a crypto device using power analysis as a side channel. [1].

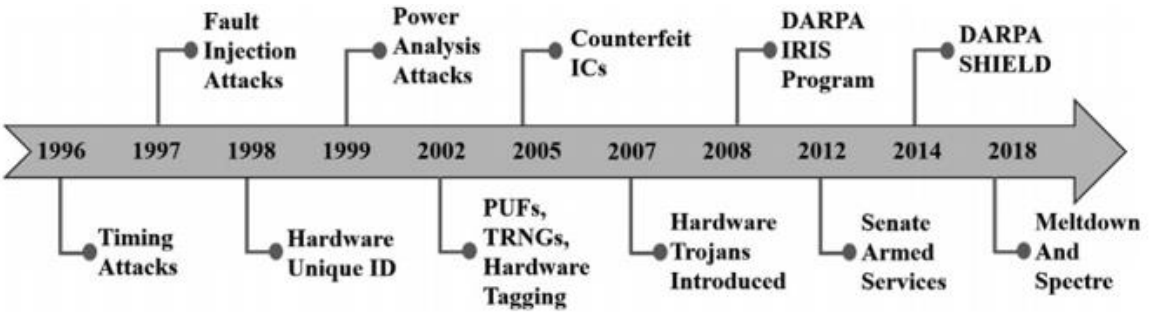


Figure 1: Hardware security has evolved in recent decades.[3]

Hardware security has become a major research area in the last decade, attracting academic, industrial, and government interest. Almost every aspect of our lives is now regulated, documented, and supported by electronic data. Whether it's electronic money, ID cards, car electronics, industrial controllers, or Internet routers, bits rule this world as much as resources do. To securely handle and store electronic data at any time and from any location, while using minimal resources and energy, is a critical requirement in today's information landscape [4].

Digital devices now pervade all aspects of life. Everyone is connected at any time and from anywhere, and the world is experiencing unprecedented technological advancements. ICs have become smaller, faster, and more powerful as semiconductor manufacturing techniques and technologies have advanced. [5]

The world has evolved from isolated systems to ubiquitous Internet-enabled 'things' capable of interacting with one another and generating data that can be analyzed for use. This highly connected global network structure, dubbed the Internet of Things, will improve everyone's quality of life, business productivity, and government efficiency. However, the Internet of Things (IoT) raises significant security and privacy concerns. Given the diversity of IoT standards and communication stacks, traditional security measures are difficult to implement. Traditional technologies are incompatible due to varying standards and communication stacks. Along with scalability and heterogeneity issues, IoT infrastructure includes resource-constrained devices like RFIDs and wireless sensor nodes. The IoTs includes simple scanners and wearables to complex systems like home appliances, cars, and smart highways and bridges. While these predictions show a more intelligent, efficient, and secure world of connected devices [6], some commentators fear a darker world of surveillance, privacy and security violations, and consumer lock-in. [7].

The Internet of Everything (IoE) is a subset of the IoT, which refers to the network of people, processes, data, and things. It unifies all of these concepts into a unified reality. Essentially, the IoE is based on the IoTs' pillars, which include network intelligent systems. Numerous technologies have evolved into a novel method of establishing the IoE. The Internet of Everything is a global network that connects people, things, and intelligent devices to share information and services.

We live in a networked world where software, system hardware, and sensors all communicate. This is what Cyber-Physical Systems (CPS) are [8]. Physical processes and software and communication dynamics are merged in Cyber-Physical Systems [9]. To design for computers, networking, and physical systems, new design technologies must be developed. The software is embedded in devices whose primary purpose is not computation, such as automobiles, medical devices, scientific instruments, and intelligent transportation systems [10].

CPS combine communication, computing, in order to achieve stability, performance, durability, and efficiency through control [11, 12]. While ongoing research is directed toward these goals, CPS security is largely ignored [13]. Because cyber-physical systems are widely integrated into many critical infrastructures, any security breach could have catastrophic consequences.

Aside from security, CPS privacy is a major concern. Cyber-physical systems collect massive amounts of data for analysis and decision-making. While data collection allows the system to make intelligent decisions using advanced machine learning algorithms, a data breach can occur

at any system of data collection, transport, processing, or storage. Again, most existing CPS design techniques ignore data protection, putting obtained data at risk [14].

The following chapters of this Thesis studies the hardware security effect and the solutions proposed in one of the categories mentioned above IoT, IoE, CPS, starting this chapter by given the definitions and differences between them and continuing with a brief reference to the general risks of Hardware security.

Chapter 2 studies the effects of hardware attacks within the IoT environment, the security challenges and the proposed solutions. Reference is made to hardware Trojans and the security challenges at PUFs. At the end of the chapter, indicative techniques for dealing with various hardware security threats are presented.

Chapter 3 presents the hardware security threats and how the intruders affect IoE. Indicatively, solutions are mentioned using Data analytics, hybrid PUFs, Blockchain, and appropriate smart Grid infrastructures.

Chapter 4 discusses ways to troubleshoot hardware security issues “under the umbrella” of CPS. The Context -Wise security Framework is presented, security issues in sensors, PLC, Device Fingerprinting etc.

In the end, in chapter 5, conclusions, future directions and critical issues that may arise in IoT, IoE and CPS have been gathered.

1.1.1. Definitions & Differences

1.1.1.1. Definitions

IoT

The Internet of Things (IoT) is a network of physically connected devices/objects that collect and exchange data via wireless networks. The IoTs is composed of two major components: the 'Internet', which serves as the backbone of connectivity, and 'Things,' which refers to objects/physical equipment. It applies the internet's capabilities for data processing, analytics, and decision-making to the physical world of physical items (Figure 2)

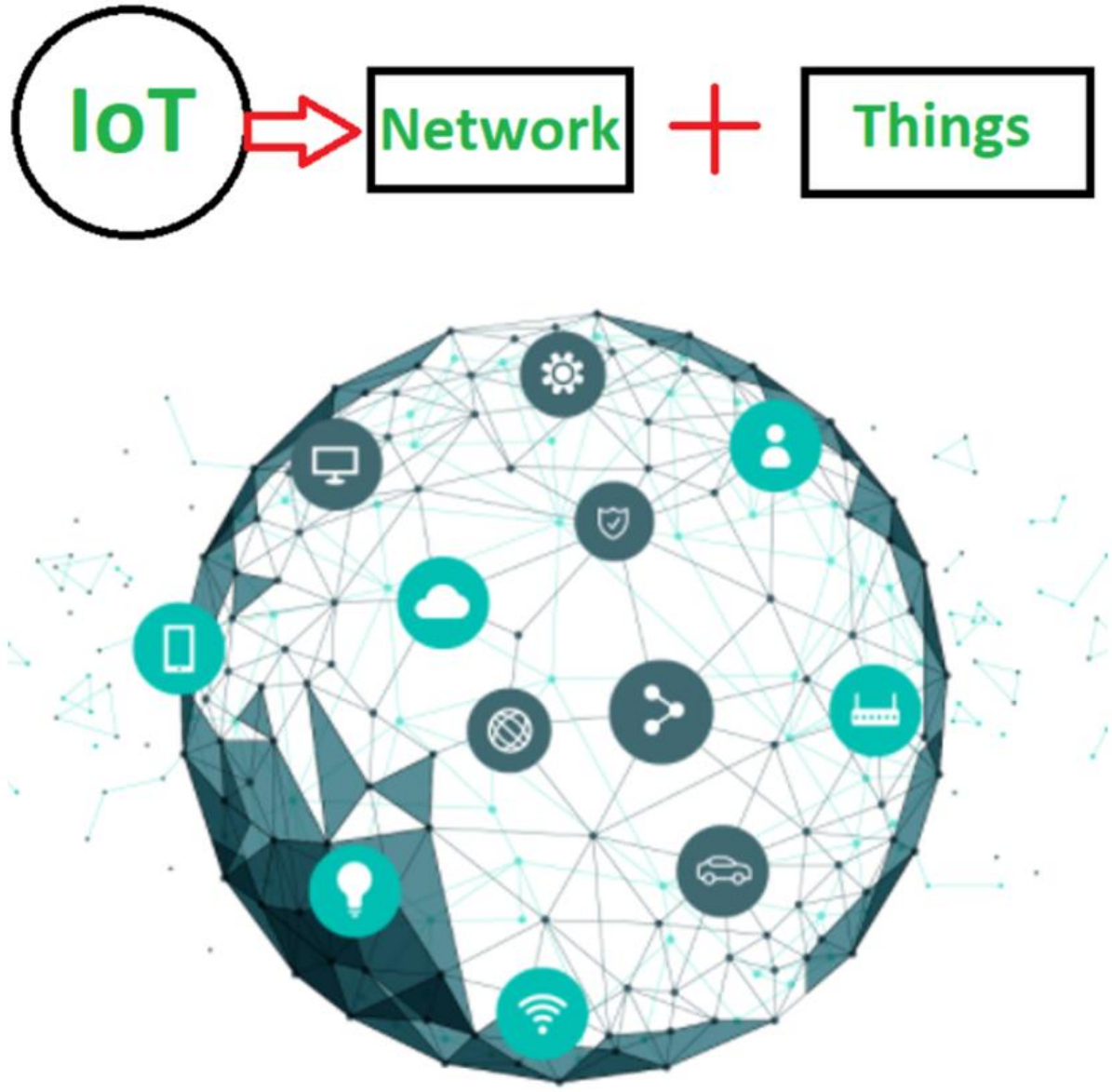


Figure 2: IoT presentation

IoE

"The Internet of Everything (IoE) is the intelligent connectivity of people, processes, data, and things". The IoE envisions a world in which billions of things are equipped with sensors that enable them to detect, measure, and assess their state; all of these devices are connected via public or private networks utilizing open or proprietary protocols (Figure 3).

The IoE Basis

- People: Increasing the relevance and value of human interactions.
- Data: Converting data to insight to enable more informed decision-making.

- Process: Delivering the correct data to the appropriate person (or computer) at the appropriate moment.
- Things: Physical devices and items that are connected to the Internet and one another to make intelligent decisions; frequently referred to as the IoT.

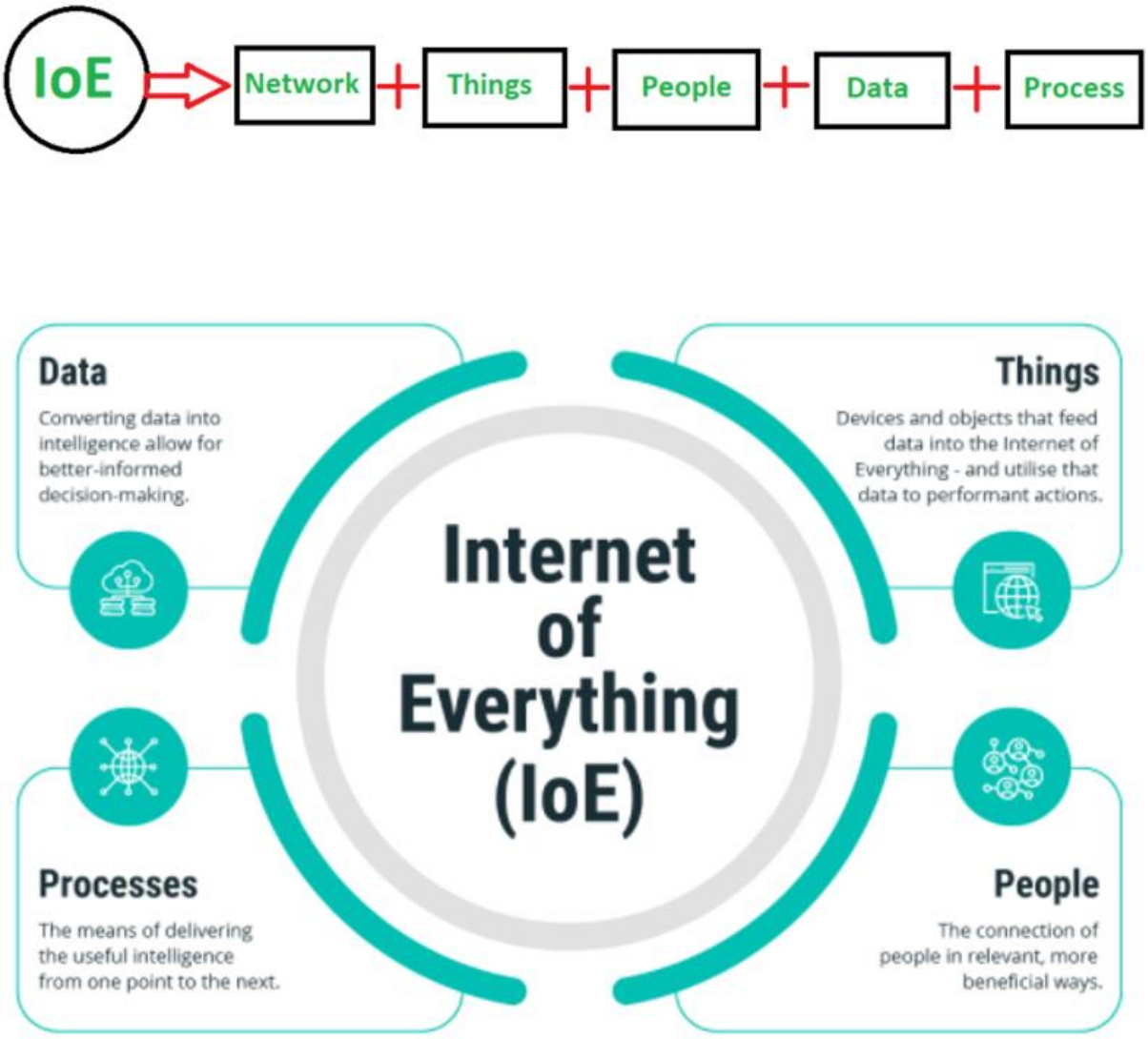


Figure 3: IoE presentation

CPS

Cyber-physical systems (CPS) are built systems in which the networked interaction of computational and physical components results in the emergence of functions and prominent

features. CPS technology aspires to develop the processes, networking, and technology necessary to integrate cyber and physical systems seamlessly.

1.1.1.2. Differences

Difference between IoE & IoT

Table 1: Differences between IoE & IoT [15]

No	Internet of Everything (IoE)	Internet of Things (IoT)
1.	CISCO coined the phrase "Internet of Everything	Kevin Ashton coined the phrase "Internet of Things" in 1999 while at Procter & Gamble (P&G).
2.	People, processes, data, and objects are intelligently connected through a 'web of things,' the next generation internet	Defining the IoT as a network of physical objects that collects and exchange
3.	Data-driven decision-making, new abilities, and better experiences are all goals of the IoE.	The IoT goal is to connect objects/physical devices. Alternatively, to create a connected network of Things
4.	In IoE, machines communicate with humans, and humans with machines.	In the IoT, machines communicate with one another.

No	Internet of Everything (IoE)	Internet of Things (IoT)
5.	It is more comlpicated than IoT since it encompasses IoD (Internet of Digital), IoH (Internet of Human), and IoT.	It's more intuitive than IoE because IoT is a subset of the larger IoE ecosystem.
6.	It is composed of four major components: people, processes, data, and things.	It has only one element in common: it is only concerned with physical objects.
7.	Including IoH, IoD, communication technologies, and the internet, the IoT is regarded to be the superset of all of these. It is often referred to as the IoT generation.	As a result, it is regarded as a precursor to the Internet of Everything.
8.	An example is: Building bridges between highways and hospitals, houses, and food will save more lives. Monitoring elder care.	The following are some examples: Smart monitoring, connected home appliances, autonomous farming, better energy management.

Difference between CPS & IoT

Increasing technical reliability, extending advancement opportunities, and exposing untapped potential are all shared goals.

While CPS and IoT are complementary, it is critical to grasp the distinctions between the two, including how they collaborate to achieve substantial breakthroughs in a range of disciplines and businesses.

"The Internet, whether it is used to connect people or objects, is merely a means of transmitting data. What distinguishes smart, connected products fundamentally from other types of products is not the internet, but the evolving nature of "things." The increasing capabilities of intelligent, linked products and the data they generate herald a new era of competitiveness"[17].

The **Internet of Things** — also known to as Industrial Internet is a technology that enables the internet-based interconnection of all types of devices to exchange data, optimize actuators, and work and community to generate results.

On the other hand, cyber-physical systems combine computing and control with physical processes, laying the groundwork for IoT and enabling improved efficiency and connectivity of devices, systems, and services across an infinite number of domains.

Future applications of the IoT and CPS will have a greater impact on society than the information technology revolution did in the preceding three decades.

By combining cutting-edge CPS technology and the IoT, CPSs pecialists can help millions of people around the world advance and improve their quality of life [16].

1.2. Hardware Security

1.2.1. Preface

Global cybersecurity issues have led to pro leaking counterfeiting piracy operations. That concern is growing in importance as the Internet's data volume grows exponentially. Many new system and application vulnerabilities are hardware-based, which is important in cybersecurity. Each new IoT application area introduces new attack surfaces and hardware requirements for secure and trusted system operation. Because electronic hardware components are more complex and distributed globally, untrusted actors design, manufacture, and distribute them. Counterfeiting, reverse engineering, and pirate activities are all part of this horizontal yet complex supply chain.

Computer and network security has been a major concern for both system designers and users since their inception. It has taken decades of research to protect systems and networks from attacks that corrupt/leak data or gain unauthorized access.

To become an expert in hardware security, you need to know many things. Understanding the math and underlying philosophy of cryptography is one goal. They must innovate to achieve efficiency.

Several cryptographic applications exist for information security. To build high-performance cryptographic algorithms, one must first understand the platforms. Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) are examples of modern ciphers. Internal algorithm structures and manipulations can reveal useful information. Their high performance makes them ideal test beds for cryptography engineering research and development.

Hardware security covers the entire lifecycle of electronic hardware (chips, PCBs, systems, and systems of systems). For an adversary, hardware security issues such as IP piracy and reverse engineering, counterfeiting, and micro probing attacks on ICs can act as a kill switch. [18] In addition, they cover the entire hardware component lifecycle from design to disposal, from chips to PCBs to systems. Hardware security includes threats, vulnerabilities, root causes, and responses [19].

Hardware security ensures safe and reliable software stack hardware. Secure storage of sensitive data and code from malicious software and networks [20]. Two vital issues:

a. A trusted execution environment protects code and data from untrusted programs in terms of confidentiality, integrity, and availability (the legal owner can use certain data/code) (TEE). These are the three pillars of secure hardware software execution. The hardware supports the isolation, while the software provides efficient policies and protocols.

b. The CIA's access control and information flow regulations for these assets are implemented appropriately in a SoC. (Figure 4).

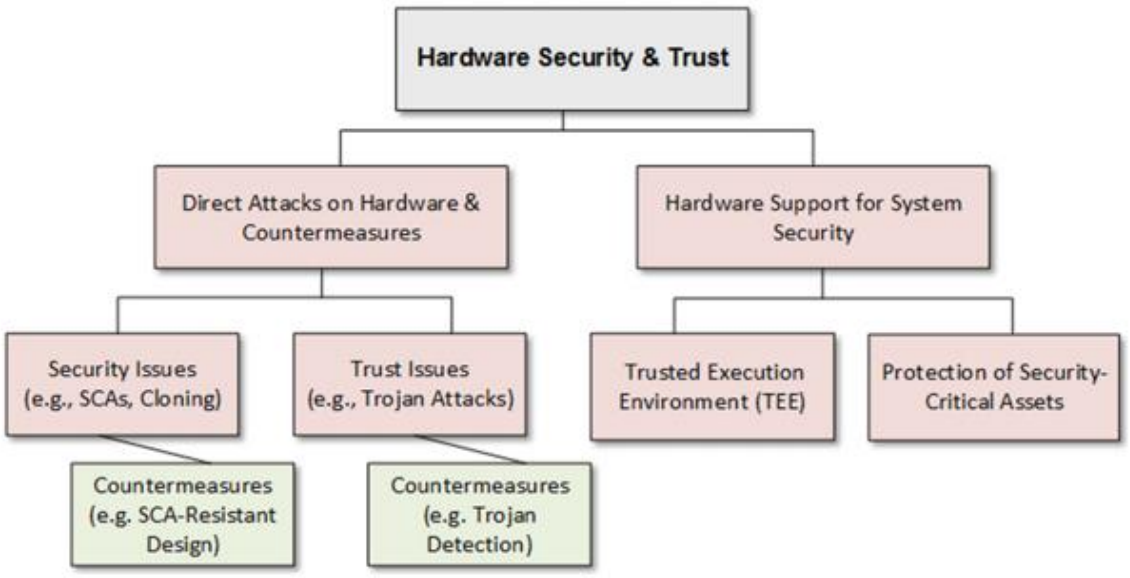


Figure 4: Security and trust in hardware [19]

1.2.2. Hardware Trust

Insufficient hardware support for software and system security causes hardware security issues. The use of untrusted IP, CAD tools, and fabrication and testing facilities can compromise hardware trust. They can jeopardize system security. Perhaps there will be issues with functionality. Dangerous implants from untrusted IP vendors can result in DoS or data leakage attacks in the field. Deficient parametric performance (low energy efficiency) can lead to reliability issues and safety concerns. Global supply chain changes and horizontal integration exacerbate hardware trust issues.

1.2.2.1. Hardware trust issues

Figure 5: IC life cycle First, a design house develops functional (like data compression) and parametric (like operating frequency or standby power) specifications. This is then translated into logic gates, transistor-level circuits, and finally a physical layout. Performance, power, and other parameters are validated during transformation. Making a wafer (circular silicon disc) from the layout is done by lithography, etching, and other processes.

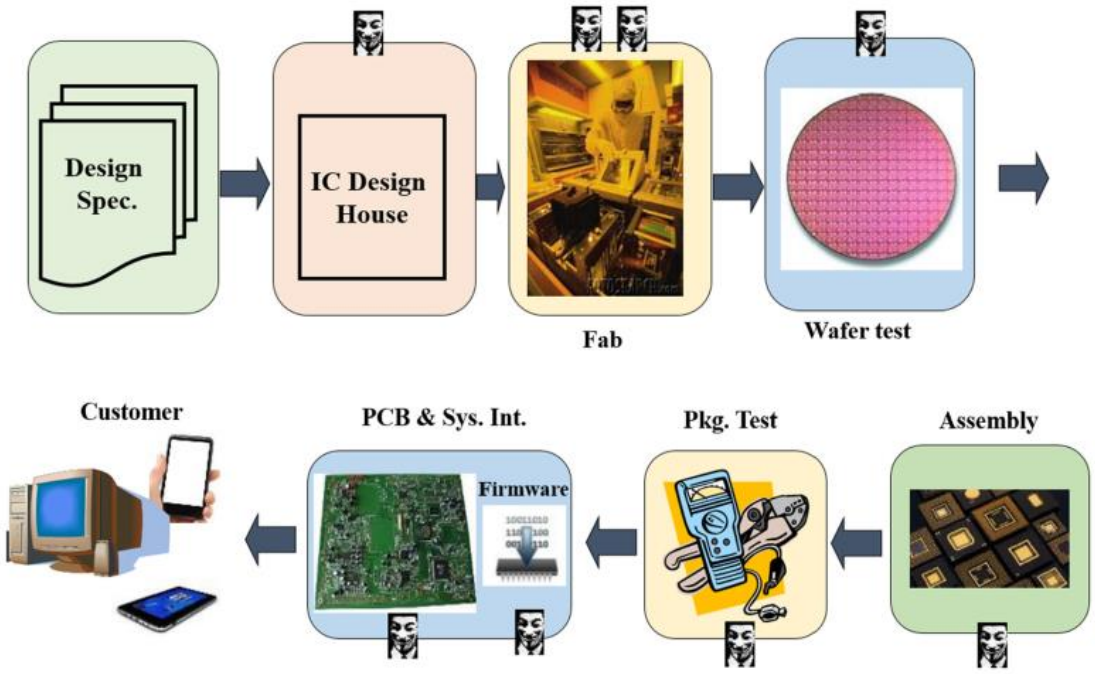


Figure 5: Significant stages in the design and testing of electronic hardware [19]

Then each integrated circuit on the wafer is tested for defects. At this level, ICs "die." The dies are packaged in ceramic or another material after diamond cutting the wafer. The packaged dies are then functional and parametrically tested in a manufacturing test facility using new test patterns. By doing so, non-functional or non-parametric defective chips are removed from the supply chain. Design-for-test and design-for-debug infrastructure test and debug complex ICs. These structures control and view difficult-to-access internal nodes. The goal is to make these nodes less visible and controllable.

Surviving chips are then distributed. Nowadays, OEMs obtain these chips from a production process, incorporate them into a PCB, flash firmware, or set up commercial components, then put the system together. An outside vendor is needed at every stage of hardware development. They are often unreliable and widely distributed. OEMs buy chips from a value chain, combine them into a PCB, flash firmware, and assemble the system.

Security problems caused by unreliable parties

Figure 6 depicts numerous critical security flaws caused by unreliable IC design, production, and testing procedures. They look into the integration of third-party IPs to satisfy functional and performance requirements. IP distributors. The researchers looked at SoCs used in mobile computing platforms (like phones) and identified common IP blocks [20].

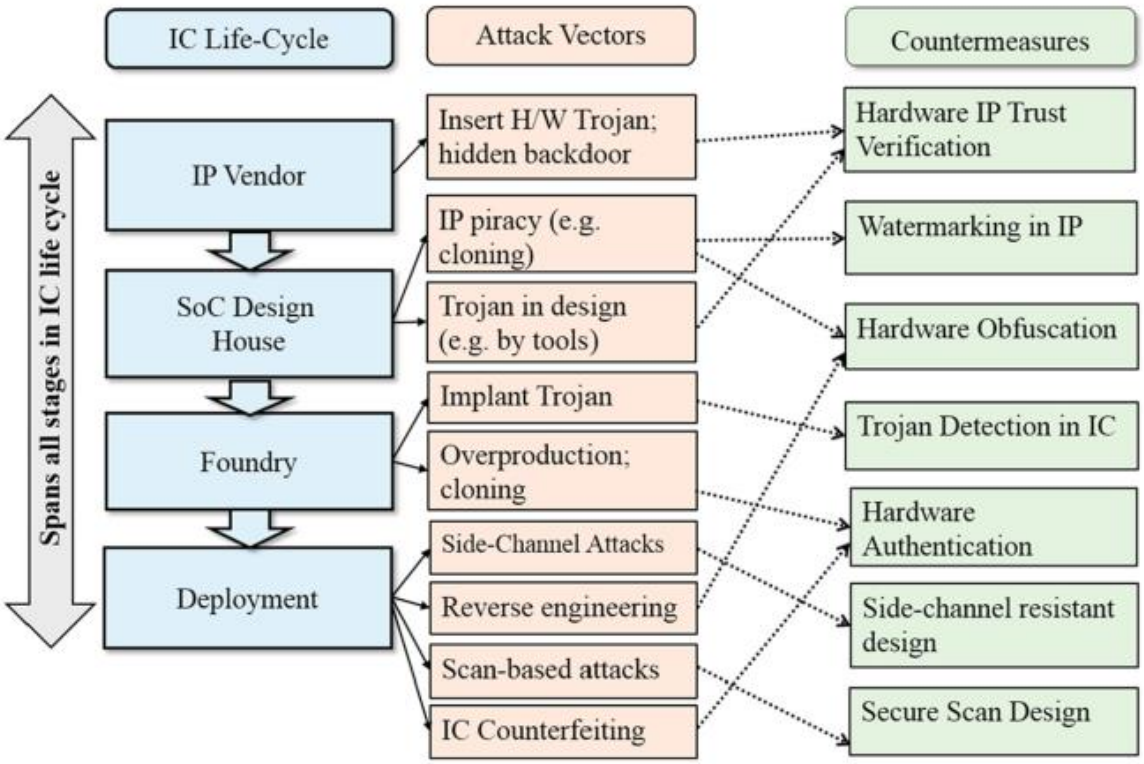


Figure 6: Attack vectors and countermeasures for each IC phase [19]

Figure 7 shows the IPs' possible sources. A studio that focuses on one type of IP. SoC IPs are often from dispersed third-party providers, making them unreliable. Foundries have access to unencrypted IP, interconnect, and DFT/DFD design files. Untrusted design, fabrication, and testing facilities face design theft, reverse engineering, and Trojan installation. According to Figure 6, certain design or testing solutions can mitigate security risks.

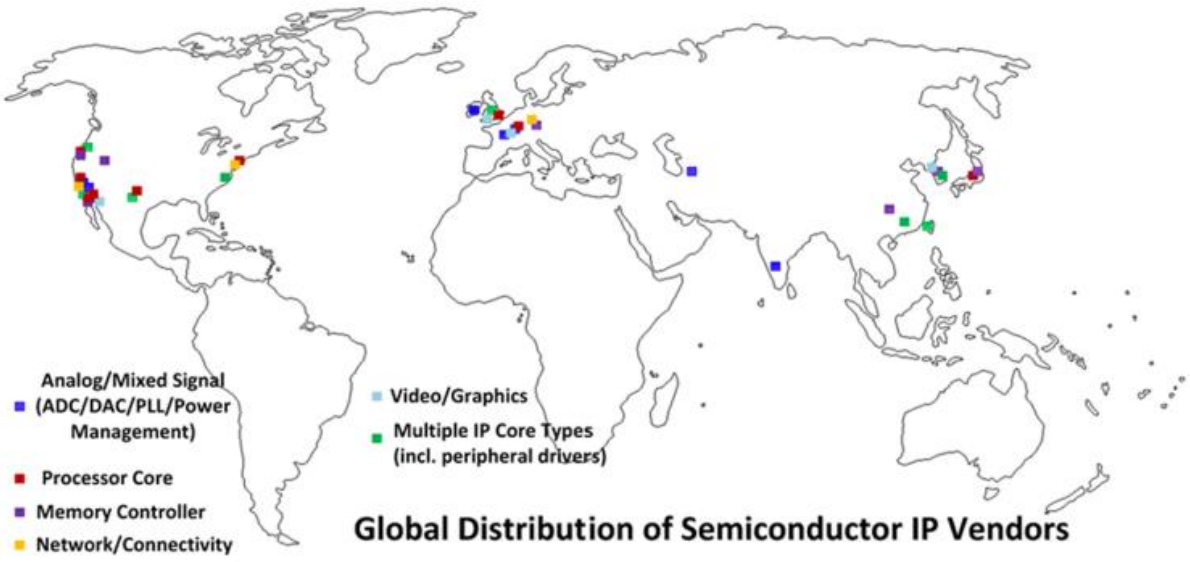


Figure 7: SoC design is vulnerable to trust/integrity issues due to its long, globally distributed IP supply chain. [19]

1.2.3. Vulnerabilities and Countermeasures Against Attacks

1.2.3.1. Attack surface

The attack surface includes all security threats. Anonymity refers to ignorance of a system's flaws. Attackers concentrate on beginning hardware attacks on more abstract attack surfaces. Lessening the attack surface is a common goal of counter Hardware security has three major attack surfaces:

Chip Level Attacks: Security characteristics of a chip can be hacked and reverse engineered [21-22]. If an attacker is able to replicate authentic chips, they can be sold as fakes. The supply chain may be compromised by Trojan-infected chips. Side-channel attacks can be used to steal data from a silicon. Using this flaw, a cryptographic chip or processor executing protected code or utilizing protected data can disclose confidential information.

PCB-Level Attacks: PCBs can be reverse-engineered more easily than integrated circuits. Most modern PCBs can be visually inspected (using techniques such as X-ray tomography) and efficiently process signals. These attacks reverse engineer the PCB to change it and produce fake units. Physically modifying PCBs can bypass DRM security (cutting traces, replacing components).

System-Level Attacks: combining hardware and software components. PCBs can be reverse-engineered more easily than integrated circuits.

1.2.3.2. Model of security

Attacks on hardware are diverse. Capabilities of the attacker, physical or remote system access, design assumptions, and usage scenarios all matter. An issue or solution's definition requires a security paradigm's definition. A security model consists of two parts:

c. Threat Model explains threats' goals and methods.

d. Trust Model: reliable parties or elements or components.

The threat model must specify the attacker's purpose, for example, leaking SoC secrets or disrupting its functionality, and how the attack is mounted, for example, by adding a Trojan that performs malicious memory writes under rare conditions. Trusted SoC designer and CAD tools.

1.2.3.3. Vulnerabilities

An attacker can exploit vulnerabilities in the hardware architecture, implementation, or design/test process. These defects might be functional or non-functional, depending on the system and usage. An attack typically finds and exploits one or more flaws. Finding flaws is usually the most challenging step. Here are some common hardware flaws:

Functional Bug: Bugs and poor design/testing are the main causes of vulnerabilities. Insecure cryptographic hardware and inadequate SOC asset protection are examples. They can detect these problems by observing the system's response to diverse inputs. Vulnerabilities can also be discovered by coincidence, making them easier to exploit.

Side-Channel Bug: These flaws cause side-channel data leakage from hardware components (such processors or crypto chips) [23]. Attackers can find these holes by monitoring hardware side-channel signals. Many powerful side-channel attacks use statistical methods to analyse side-channel parameter traces [24]. The amount of data lost determines the severity of a side-channel problem.

Test/Debug infrastructure: Testing and debugging the vast majority of hardware systemsallows designers and test engineers to ensure proper operation. They can also be used to debug devices by studying internal operations. By using the test/debug functionality, attackers can gain access to sensitive data or take control a machine.

Access control or information-flow challenges: System may be unable to distinguish authorized from unauthorized users. It could grant an attacker access to confidential knowledge and resources. Attackers can monitor data flow to decipher program control flow and protected zone memory address.

1.2.3.4. Countermeasures

As hardware attacks have evolved, so have responses. Countermeasures may be utilised during design or testing. Figure 8 shows current industry practise for SoCs in terms of :

a. ensuring the design contains security measures,

b. A system's protection from known attacks is confirmed (security validation).

The SoC manufacturing pipeline is divided into four phases:

1. exploration,
2. planning,
3. development, and
4. production.

The design space explored in the first two stages of the SoC life cycle yields a design that meets design goals (pre-silicon). Confirming and fabricating chips is part of the SoC development and production process. Exploration examines a SoC's assets, attacks, and requirements for secure software execution. This stage generates a list of security requirements. Cryptographic keys, memory locations, and configuration bits are protected. Secure architecture and implementation are validated before silicon. Post-silicon security testing ensures that manufactured chips are not vulnerable to known attacks. It covers a wide range of issues in terms of security, confidence, and scalability to large designs. Coding analyze and verification prior to silicon, followed by fuzzing and penetration testing [25].

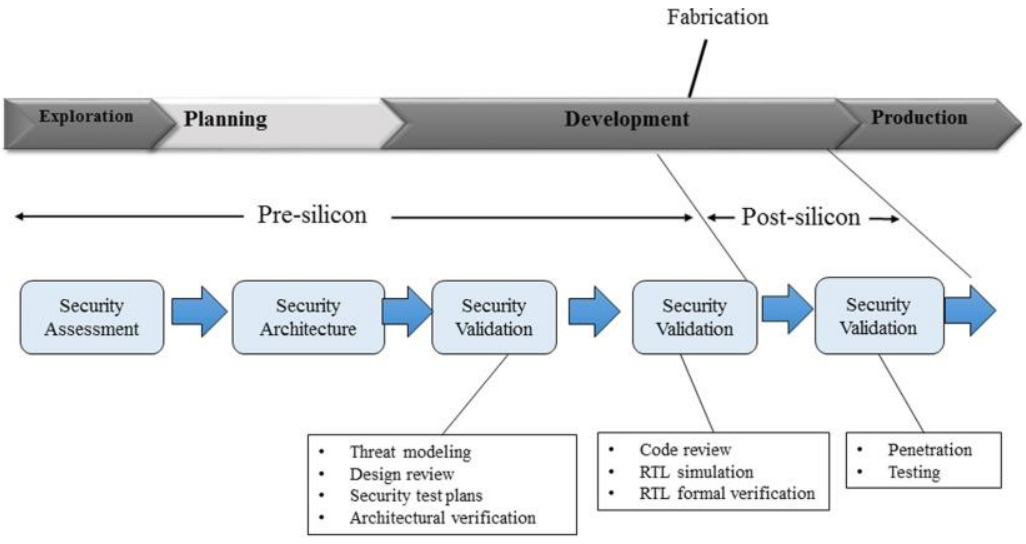


Figure 8: Current state of system-on-chip security design and validation [19]

Design solutions: Design-for-security methods work. DfS provides low-cost active and passive attack defense design. Trustworthy security primitives, side-channel reaction (masking and concealment), and Trojan installation hardening are effective DfS defenses. Software attacks on SoC platforms rely on resilient security design.

Test and verification solutions: Testing and validating can improve security and trust. Functional and formal pre-silicon verification and post-silicon fabrication testing have been used to find security vulnerabilities in chips, PCBs, and systems. Hardware-vulnerability protection.

1.2.4. Conflicts

The demands of security and test/debugging sometimes clash. Scan chains and DFD structures are essential post-manufacturing testing and debugging. On-silicon debugging requires internal IP block signals. For troubleshooting, security restrictions usually limit internal signal observability. Module locks, encryption keys, and firmware are crucial. These security assets make it hard to debug limited IP signals routed through high-security IP blocks. The problem is exacerbated by current industry practice. This is because security assets are determined per-IP. Debugging issues are discovered during actual silicon execution due to late identification. So, a silicon “respin” is required, requiring costly design changes and re-fabrication. Hardware must be designed to maintain DFT and DFD infrastructure for SoC test/debug.

1.2.5. Summarizes

Important hardware security issues and solutions are summarized in Table 2. There are two types of countermeasures: early and late in the hardware lifespan.

Table 2: Summarizes Hardware attacks & countermeasures [19]

ATTACKS				
Type of Attack	**What it is**	**Adversary**	**Goal**	**Life-cycle stages**
Hardware Trojan Attacks	Malicious design modification (in chip or PCB)	Untrusted foundry, untrusted IP Vendor, untrusted CAD tool, untrusted design facilities	• Cause malfunction • Degrade reliability • Leak secret info	• Design • Fabrication
IP Piracy	Piracy of the IP by unauthorized entity	Untrusted SoC Designer, untrusted foundry	• Produce unauthorized copy of the design • Use an IP outside authorized use cases	• Design • Fabrication
Physical Attacks	Causing physical change to hardware or modifying operating condition to produce various malicious impacts	End user, bad actor with physical access	• Impact functional behavior • Leak information • Cause denial of service	• In field
Mod-chip Attack	Alteration of PCB to bypass restrictions imposed by system designer	End user	• Bypass security rules imposed through PCB	• In field
Side-Channel Attacks	Observing parametric behaviors (i.e., power, timing, EM) to leak secret information	End user, bad actor with physical access	• Leak secret information being processed inside the hardware	• In field
Scan-based Attacks	Leveraging DFT circuits to facilitate side-channel attack	End user, bad actor with physical access	• Leak secret information being processed inside the hardware	• In field • Test-time
Microprobing	Using microscopic needles to probe internal wires of a chip	End user, bad actor with physical access	• Leak secret information residing inside the chip	• In field
Reverse Engineering	Process of extracting the hardware design	Design house, foundry, end user	• Extract design details of the hardware	• Fabrication • In field

COUNTERMEASURES				
Type of Countermeasure	**What it is**	**Parties involved**	**Goal**	**Life-cycle stages**
Trust Verification	Verifying the design for potential vulnerabilities to confidentiality, integrity, and availability	• Verification engineer	• Provide assurance against known threats	• Pre-silicon verification • Post-silicon validation
Hardware Security Primitives (PUFs, TRNGs)	Providing security features to support supply chain protocols	• IP integrator • Value added reseller (for enrollment)	• Authentication • Key generation	• Throughout IC supply chain
Hardware Obfuscation	Obfuscating the original design to prevent piracy and reverse engineering	• Design house • IP integrator	• Prevent piracy • Reverse engineering • Prevent Trojan insertion	• Design-time
Masking & Hiding	Design solutions to protect against side-channel attacks	• Design house	To prevent side-channel attacks by reducing leakage or adding noise	• Design-time
Security Architecture	Enable design-for-security solution to prevent potential and emerging security vulnerabilities	• Design house • IP integrator	Address confidentiality, integrity, and availability issues with design-time solution	• Design-time
Security Validation	Assessment of security requirements	• Verification and validation engineer	Ensure data integrity, authentication, privacy requirements, access control policies	• Pre-silicon verification • Post-silicon validation

Chapter 2

Internet of Things (IoT)

2. Internet of Things

2.1. Preface

The IoT enables billions of "things" to exchange data and communicate with one another to produce services with additional value. IoT technologies now have a huge impact on many aspects of people's lives. Variety of sensors and electronic devices collect data, including location and health data [27]. Figure 9 divides IoT applications into smart home, agriculture, security and privacy, healthcare, and wearables. IoT networks are limited in power, computing capacity, and mobility [28]. New products and technologies are introduced daily without adequate consideration for potential security concerns. This lack of consistency is a significant problem in IoT networks. Large-scale IoT networks face numerous challenges, including security , data fusion, data management [29], complexity [30], and spectrum scarcity . Any use of IPV4 for tackling in modern IoT technologies also limits scalability [26]

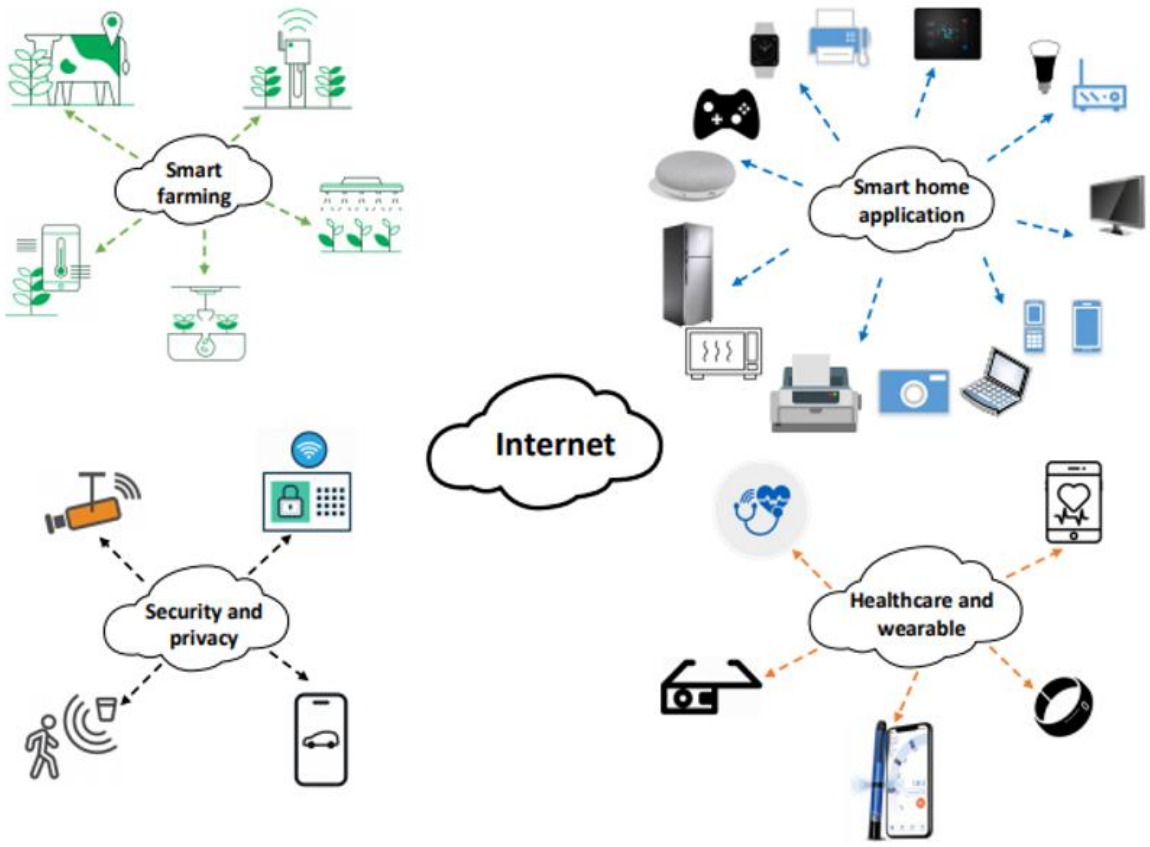


Figure 9: Examples of IoT applications [29]

2.1.1. Infrastructure

'Internet of Things' consists of two key terms: 'Internet' and 'Things'. To achieve this, IoT requires a network that is always accessible, i.e. a ubiquitous network. With regard to the IoTs, this could be any item with an unique id.

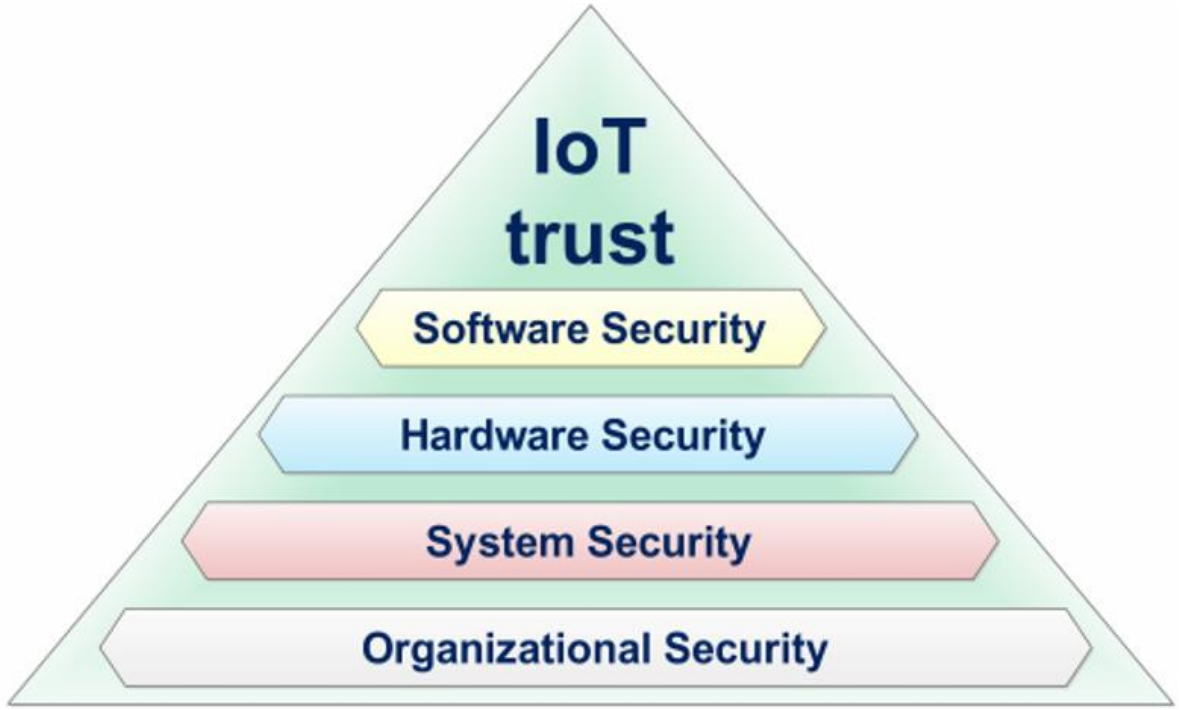


Figure 10: IoT Trust Pyramid [31]

The trust pyramid for the IoTs is depicted in Figure 10. Each of the security layers:

i. organizational,

ii. system,

iii. hardware, and

iv. software,

must be trusted in the IoT [33]. Encrypting messages between parties doesn't require hostile access to cryptographic computers [32]. Functional cyphers' mathematical security was required for decades. However, this premise [34] is no longer able to hold for many modern IoT applications, such as:

i. Smartcards,

ii. RFID labels

iii. electronic keys and door locks, etc.

IoT hardware is vulnerable to physical attacks. Strong opponents could thus physically access IoT devices. SCA, also known as a side-channel attack, is among the most prevalent physical attacks in these circumstances. [35]. These attacks exploit physical IoT hardware features to obtain internal device information. Some of the most well-known physical attack characteristics:

a. power utilizationand

b. electromagnetic radiation,

2.1.1.1. The IoT Hardware

In most cases, The IoT's hardware is a framework portion, made up of:

i. a Printed Circuit Board (PCB),

ii. multiple sensors/actuators,

iii. microelectronic components,

iv. physical or logical connection between devices in a network, and

v. a power supply.

PCB contains active and passive electronic components such as microchips, transistors, and diodes. Historically, all of those electronic components were trusted in the field. In response to cyber threats, the number of electronic components requiring security has increased dramatically [32].

2.1.1.2. Hardware architecture

In Figure 11, a general perspective of the IoT hardware configuration is provided from an architectural standpoint. The real-world environment in which the Things operate can either be observed by tracking a variety of physical quantities (agricultural temperature and humidity) alternative, influencing a variety of physical characteristics (temperature and humidity variations in precision agriculture). The following elements make up the IoT hardware architecture of an end device, also known as the Thing:

i. hardware modules for measuring the objective physical quantities,

ii. hardware components for driving the desired physical quantities,

iii. module for storing data

iv. interfaces for wireless communications,

v. interfaces for communications using wirelines,

vi. I/O (Input/Output) module for user,

vii. a module for power supply, and

viii. microprocessor unit locally (MPU).

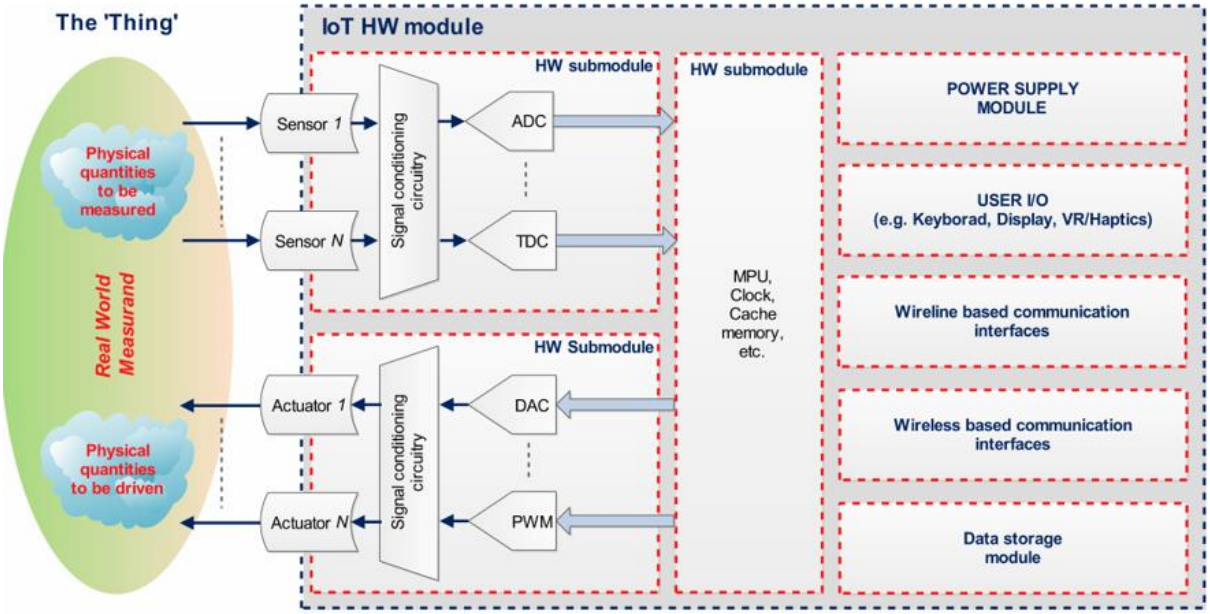


Figure 11: a piece of IoT hardware that detects and/or controls a "Thing"[31]

Typically, hardware submodules used/designed for measuring activities are made of the following:

i. sensors,

ii. circuitry for receiving signals, and

iii. Analog-to-Digital Converters (ADCs).

Sensor analogue signals are converted to digital signals using ADCs, and measurements of timing signals use Time-to-Digital Converters (TDCs). Typically, hardware submodules used/designed for actuation tasks are made of the following:

i. actuators,

ii. Conditioning circuits, and

iii. Digital-to-Analog Converters (DACs).

The digital signals that the MPU sends to the actuators are converted to analog signals by the DACs. A DC motor can also be driven using pulse width modulation (PWM) signals. If data (as a result of measurements or according to information received by the IoTs device in response to commands for actuation) must be stored locally; this can be achieved via:

i. flash drives,

ii. Secure Digital (SD) cards, and

iii. Solid-State Drives (SSDs).

Interfaces for wireless networks requires at least one of the following:

i. Wi-Fi,

ii. Bluetooth,

iii. SigFox/LoRa,

iv. ZigBee/RFID, and

v. cellular Narrow-Band IoT (NB-IoT), 5G.

Interfaces for wireless networks includes:

i. two twisted copper wires (in twisted pair wireline consists)

ii. Enhanced data rates and reduced signal attenuation by coaxial cable and

iii. Most expensive wired medium is fiber optic.

User I/O interfaces include:

i. Keyboards,

ii. displays, and

iii. Virtual-Reality (VR) or Haptics controller.

Whether IoTs devices are wired or battery-powered is a critical decision that will be made in the field application. The type and location of an IoT device will determine whether it has a rechargeable battery or a Battery Management System (BMS) [31].

2.1.2. Traditional IT Security vs IoT Security

The majority of IoT devices are "closed," meaning that once shipped from the factory, consumers cannot add security software. To be "secure by design," IoT devices must have security built in ("built-in security"). That is, IoT security must evolve from "add-on security," where security is simply applied to existing systems like servers or computers (traditional IT).

An IoT system is also composed up of nodes with limited hardware and software resources (like sensor or RFID nodes), whereas traditional IT is made up of resource-intensive devices. Thus, in the IoT era, only lightweight algorithms can strike a balance between security and capability.

It is also evident in each functional aspect (identification, sensing, communication, computing, service, and semantic) [37].

Additionally, in the IoT system model, the perception layer is the most difficult to protect because:

1. technological heterogeneity makes it difficult to use a single type of security technology; and
2. the perceptual environment is frequently open, and thus security strategies designed for closed environments may cause problems in an open environment.

However, application-layer privacy issues are more complex, as we use IoT applications every day that automatically collect our personal data. Decentralization of many IoT applications poses significant security risks.

Table 3: Traditional IT security vs IoT security [36]

Traditional IT Security	IoT Security
Add-on Security	Built-in Security
Complex algorithms	Lightweight algorithms for resource-constrained devices
User Control	Privacy issue: IoTs often collect automatically user private information
Small technological heterogeneity	Large technological heterogeneity and thus also large attack surface
Many security guards	Few security guards
IT devices are located in closed environments	IoT devices are also located in open environments

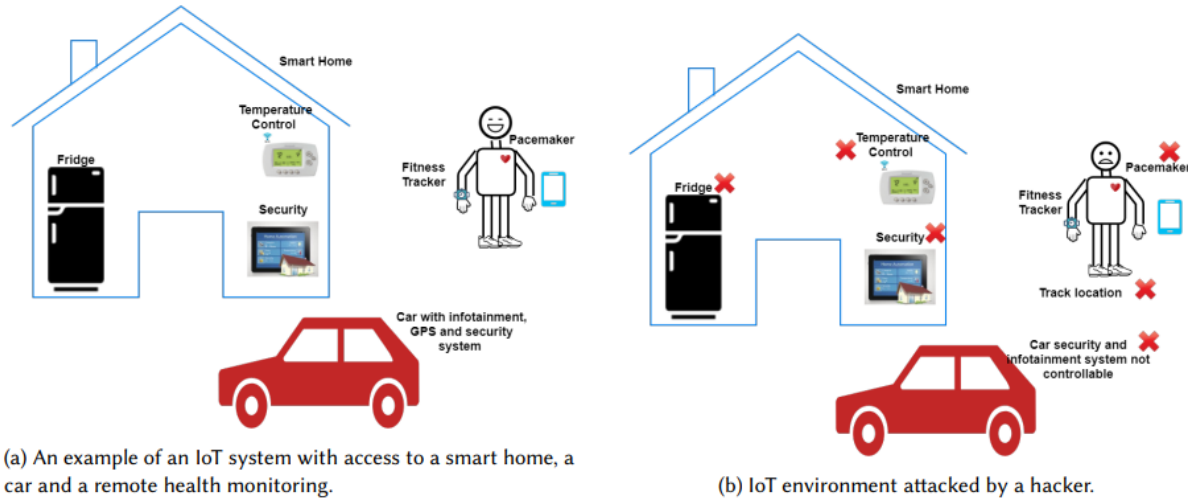
In summary, IoT devices are deployed in more risky and heterogeneous contexts with limited resources and security personnel. As a result, we must build lightweight solutions to deal with such potentially risky conditions that have a large attack surface. Table 3, summarizes the key distinctions between traditional information technology and IoTs security needs and application environments [36]

2.2. Security Challenges & Attacks

2.2.1. General

To steal confidential data or infiltrate the network to alter critical data, hackers usually plan and execute IoT attacks. Because IoT devices are rapidly expanding, cybercriminals increasingly target them. The limited energy and processing capabilities of IoT nodes, the network's massive number, and its heterogeneous composition [39]. Inefficient traditional security systems, especially those with many connected devices. Figure 12a shows online IoT devices. These devices can be hacked, causing major inconvenience to users. Hackers can cause heart attacks in pacemaker users, preventing them from driving. Figure 12b shows a hacking attempt on an IoTs device that is used to remotely monitor the heart rate of pacemaker patients, monitor smart homes, and navigate cars. The attacker wants to disrupt IoT communications and modify data

sent by stealing the encryption key. Nodes with low power consumption need tamper detection circuits to avoid attacks.



(a) An example of an IoT system with access to a smart home, a car and a remote health monitoring.

(b) IoT environment attacked by a hacker.

Figure 12: An attacker in an IoT environment [38]

Identity, authentication, encryption, confidentiality, jamming, cloning, and hijacking are among the IoT security issues. Many methods rely heavily on encryption to prevent hacker interceptions. Less data manipulation is possible when messages are encrypted. The security of IoT systems relies on cryptographic methods. All of these well-known cryptographic techniques use cryptographic keys: PKI, AES, DES, and ECC. A device's NVM stores these secret (private) cryptographic keys. On the other hand, physical attacks on NVMs are extremely vulnerable.

Table 4: Security differences between hardware and software [38]

	Hardware-based security	Software-based security
Circuit	Requires dedicated hardware and processor	Does not require new hardware
Cost	Expensive	Inexpensive
Power	Requires extra power for hardware	No extra power is required
Updates	Hard to update in order to accommodate new security threats	Easy to update to accommodate new security threats
Load on host Processor	No extra load	Extra load which might compromise efficiency
Key Protection	Protected by crypto-hardware	No protection for keys
Vulnerability	Difficult to extract the key	Easy to decode the algorithm and extract the key

The sensitivity of cryptographic keys complicates security. White Box Cryptography (WBC) utilizes software encryption to maintain key security and ensure data exchange. WBC is slow and only works with symmetric cryptographic algorithms, so it isn't suitable for IoT network security. HSMs are physical computing devices that store and manage digital keys. Heterogeneous equipment data (HSM) requires programming and Key generation, distribution, and storage are major security concerns as the number of IoT devices grows rapidly. Software-based and hardware-based IoT device protection strategies exist. Software-based security techniques protect messages only with software. Getting the secret keys is easy because they are based on algorithms. Because of this, these security methods are vulnerable to malware, phishing, and DoS. Software-based security methods store keys in NVM of vulnerable devices. Despite software-based security measures, modern hardware and computer advancements can make it possible for hackers to compromise systems like quantum computers. Hardware-based security encrypts and stores data on a dedicated hardware chip or processor. They can be used to protect data from attacks and restrict read-write access. But hardware-based security is vulnerable to MITM. For cryptographic processing and strong authentication, because they can manage and store digital keys, HSMs are employed. Messages have been encrypted using HSM, PKI, and AES.[40].

A vast network with mobility and heterogeneity is created by the connection of numerous sensors and devices to the Internet. Some traditional security techniques are ineffective in IoT

networks due to low energy and processing capabilities [41, 42]. IoT networks' heterogeneous makeup as a result of their diverse applications is a serious problem when using established security protocols. Determining the appropriate level of protection is critical. Most of these apps can be categorized by user association, openness, or heterogeneity. Due to IoT heterogeneity, key generation and exchange cryptographic approaches are susceptible to degradation [44]. Figure 13 depicts several different IoT security principles organized by application, architecture, connectivity, and data. Using these stack stages as a starting point, we may build a taxonomy of protection measures for the IoTs. [38].

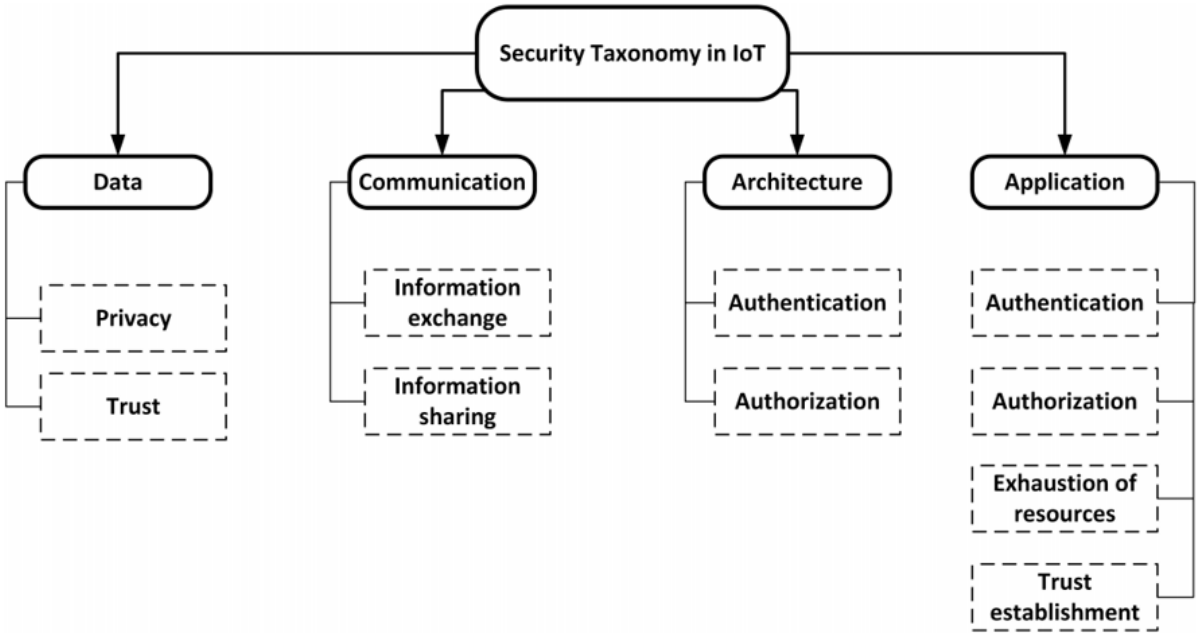


Figure 13: Domains and security in IoT [38]

2.2.1.1. Data

Cybersecurity for IoT networks must protect data privacy and confidentiality [46]. Generally, confidentiality prevents unauthorised users from accessing or stealing data. Managing large numbers of users and apps is challenging in IoT. Using secure key management can help secure IoT networks [47]. Only authorised users can view and update data, however. The term privacy is often confused with confidentiality [48, 188]. User acceptance of network security, privacy, and confidentiality is trust. Trust enforces privacy, confidentiality, and security across IoT layers, devices, and apps [49].

2.2.1.2. Communication

Data transfer among users, objects, or even IoT layers constitutes communication in an IoT network. IoT networks use multiple communication protocols, making them vulnerable to communication attacks [50]. Communications are thus vulnerable to Man in the Middle (MitM)[50] and Eavesdropping [51].

2.2.1.3. Architecture

There are no worldwide or particular IoT network topologies to evaluate authorisation and authentication security ideas. However, several architectures such as SDN [52], SEA [53], smart city [55], SOA [56], and black SDN [57] are suggested as a way to analyze both authentication and authorization.

2.2.1.4. Application

Aspects of an application's scope can be utilized to evaluate security mechanisms including authentication, authorization, exhaustion, and trust founding [58]. A number of solutions for authentication and authorization have been developed because IoT devices lack a defined architecture [59].Weaknesses in the IoT can have a wide range of consequences.

i. **Authentication:** For information access, devices must be authenticated. A key challenge in IoT systems is data integrity and unauthorised access via impersonating attacks. The authentication procedure ensures that nodes only transport genuine information from authorized sources.

ii. **Authorization:** The sensors and actuators that make up the IoTs can only gain access to private data if they have been given permission to do so. Consider the potential of IoT in medical settings. Sensitive information, such as patient health profiles, should be restricted to approved nodes or users in this scenario. The information should only be accessible to authorized users.

iii. **Exhaustion of resources:** Its energy consumption varies depending on the application [60]. Resource-exhaustion attacks are also a widespread IoT network threat that consumes a lot of resources, especially energy. For example, IoT networks' routing protocols are susceptible to exhaustion attacks because routing loops use up node energy. To attack in-range nodes, the attacker employs DoS. The battery's resources are depleted more quickly as each receiver reacts to the attacker.[61].

iv. **Trust establishment:** In an IoT network, there are three different ways that trust can be defined:

a. Trust for security at each separate layer: Security and privacy concepts need to be implemented at all levels to ensure the dependability, confidentiality, and integrity of data.

b. Trust between layers**:** levels of the IoTs need to preserve communication privacy and trust;

c. establishing the IoT network's and the end user's trust: Since the user and the IoT system share information, the trust idea need to be mutually beneficial. Additionally, each party's actions have an impact on how the other operates. Establishing a strong trust relationship between the user and the IoT system is therefore essential.

For authentication or encryption utilizing the aforementioned key-based security mechanisms, the digital secret key must be stored locally on each IoT node. Traditional non-volatile storage of digital keys such as Read-Only Memory or a one-time electronic fuse has been used. Any number of intrusive threats can be produced by an attacker using a scanning electron microscope (SEM). However, using these memory types requires extra fabrication processes.

2.2.2. Hardware Security Challenges

Attackers are increasingly interested in IoT devices due to their lack of hardware security [33]. Any IoT system must be secure to meet user privacy standards (see figure 10). Trusted IoT systems are also required in some cases, like healthcare or real-time traffic management.

Table 5 summarizes a literature-based taxonomy of hardware attacks. Here are some ways to attack an IoT device's hardware: [62]:

(i) a passive attack: when the tool is utilized in accordance with its operational guidelines; and

(ii) an active attack: when the device's inputs, outputs, and/or operating environment are changed, causing the device to behave abnormally.

The following methods can be used for both passive and active attacks:

(i) Non-invasive attacks, in which the attacker only accesses the communication interfaces and leaves the IoT device's structure unaltered (see figure 11),

(ii) Semi-invasive attacks, such as IC decapsulation, where the attacker modifies the IoT devices' hardware integrity, and

(iii) Attacks that are invasive have no limitations on what the attacker can do with the IoT.

Attackers attempt to take advantage of timing, power consumption, electromagnetic radiation, and other features of IoT devices in side-channel attacks, which are passive, non-invasive attacks. Not using probing or conventional read-out circuits to read IoT device memory. It is an invasive passive attack to probe unpacked IoT devices for data signals on PCB traces

Table 5: Types of Hardware attacks [31]

Hardware Security for IoT	**Passive Attacks** *The device is operated largely or even entirely within its specification*	**Active Attacks** *The device, its inputs, and/or its environment are manipulated in order to make the device behave abnormally*
Non-Invasive Attacks Device attacked as is, only accessible interfaces exploited (*relatively inexpensive*)	Side Channel Attacks (SCA): timing, power, EM radiation, cache trace, etc.	Insert fault in device without depackaging: clock glitches, power glitches, or by changing the environmental temperature, etc.
Semi-Invasive Attacks Device is depackaged but no direct electrical contact is made to the chip surface (*more expensive*)	Read out memory of device without probing or using the normal read-out circuits	Induce faults in depackaged devices with e.g. X-rays, electromagnetic fields, electrical pulses, light, etc.
Invasive Attacks No limits what is done with the device	Probing depackaged devices but only observe data signals (e.g. from PCB traces)	Depackaged devices are manipulated by probing, laser beams, focused ion beams, etc.

Unpacking IoT devices isn't required for non-invasive active attacks. Semi-invasive active attacks employing X-rays, electromagnetic fields, electrical pulses, and other invasive techniques to target unpacked IoT devices/ICs. The attacker is using focused ion beam or laser probing to reverse-engineer the IC architecture after unpacking the ICs.

As demonstrated above, The following are the primary issues with IoT hardware security:

(i) size,

(ii) limited computational capacity,

(iii) restricted power supply.

Due to their small size, the majority of IoT devices are vulnerable to passive and active attacks. Insufficient processing power in most IoT devices renders today's sophisticated security approaches ineffective. This is due to the fact that IoT devices have limited power and any additional hardware security module will require additional power to function.

2.2.2.1. Measurements & Instrumentation

Today, measurement methods and accompanying instruments are a popular focus of research and development in the field of IoT hardware security.

Implementing software security is insufficient to create trustworthy IoT devices. For instance, cyphers that are implemented on chips to increase the security of IoT systems are built on cryptographic algorithms. Cyphers that are built into software and hardware are currently operational.

- intricate mathematical constructions
- several iterations, and
- frequently requiring large-field math.

Nowadays, standard ciphers like AES and ECC are implemented on COTS microcontrollers AES and ECC are susceptible to conventional attacks, but the attacker points out that these are frequently ineffective. Attackers can use a method known as side-channel analysis to take advantage of both the built-in hardware and software implementation features of deployed cyphers as well as their mathematical structures and algorithms. These side-channel analysis attacks put several fundamental presumptions that a traditional cryptographer bases the design of cyphers on in terms of hardware and software security in jeopardy. The tester (or attacker) conducts "side-channel analysis" by taking synchronous measurements on the IoT hardware that is being attacked. A tester or an attacker seeks to quantify:

a. voltage declines,

b. Acoustic emissions, electromagnetic radiation, and,

c. power consumption,

d. heat,

e. execution period,

f. Emissions of light,

g. clock frequency,

h. rate of message error, and

i. The device's faulty outputs are being targeted.

Figure 14 depicts a general example of a SCA in an IoT system. The actions that are most frequently taken are on:

(i) differential power,

(ii) clock/timing,

(iii) electromagnetic radiation,

(iv) temperature of the device/IC.

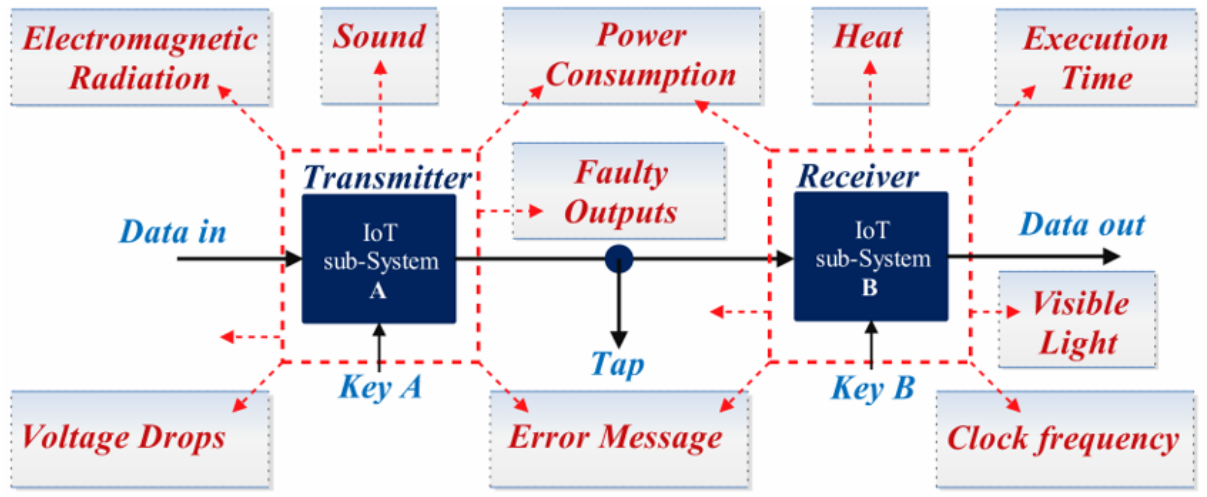


Figure 14: IoT Systems with Side Channel Attack (SCA) [31]

These and other side-channel data allow attackers to observe IoT device behavior. A cryptographic algorithm implemented in a microcontroller, ASIC, or FPGA [63]. An attacker could exploit a cypher's implementation by tracing power consumption. According to the power trace fluctuations, cryptographic algorithms run on dedicated hardware or microprocessor units. An attacker can deal with power in several ways. By connecting a resistor to the IoT board's power or ground rails, the voltage drop across the resistor can be measured [31].

2.2.3. Attacks

Attacks against IoT networks can be classified into two broad scenarios:

(i) The IoT gadget is not physically accessible to the attacker, hence remote access is only possible through software or network connections. When an adversary obtains the necessary cryptographic keys, they can subvert the authentication process.

(ii) In the second case, the chip or IoTs device is physically accessible to the attacker. For instance, the attacker may reverse-engineer it, make a fake copy, or take over the IP address [180].

The low cost, low power, and poor processing capability of IoT devices, as well as the network's heterogeneity and scale, restrict the use of common security measures. Due to advanced threats and security difficulties [65], new security methods are required in domains such as reliability, secrecy, identification/authentication, and non-renunciation.

2.2.3.1. Denial of Service Attack (DoS)

This attack's goal is to degrade network performance by using up all available resources. A distributed denial-of-service attack uses computational resources. DDoS attacks and individual DoS attacks are two types [66]. To exhaust the target entity's resources, an attacker uses a DoS attack. The target system is then bombarded with requests from multiple attackers or compromised users.

2.2.3.2. Sybil attack

Sybil attacks prefer networks with high user density. A node may have multiple IDs. The network will lose its ability to unite. Websites and social media platforms are prime targets for Sybil attacks. There is a growing concern that Sybil attacks may be launched against IoT devices as their use grows. A Sybil attack could lead to the submission of bogus data.

2.2.3.3. Changed or replayed routing information

This type of attack involves listening to a genuine transmitter and impersonating their identity. Then it delivers fraudulent data to the receiver, creating network loops [68].

2.2.3.4. Attacks based on Access-Level

Passive and active attacks are classified according to the level of network access.

Passive Attacks:

Most passive attacks simply eavesdrop on legitimate transmitter-receiver communications to steal data. [67]

Active Attacks:

Active attacks involve the intruder trying to disconnect by pretending to be someone else or changing the routing information [69].

2.2.3.5. Attacks in Communication Protocols

TCP/IP defines IoT network communication. Table 6 provides TCP/IP attack taxonomy.

The attacks on the IoT depicted schematically in Figure 15 have been documented before. Glowing arrows and dashed circles depict a route toward more adoption of PUF in the IoT [70].

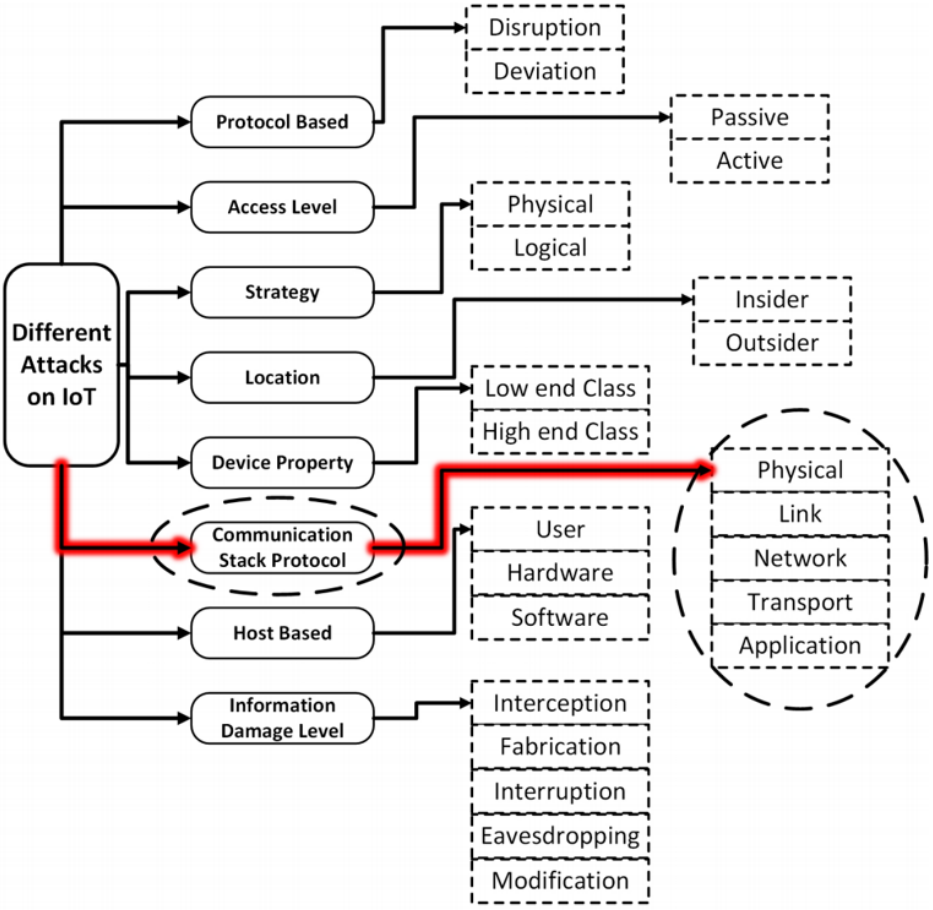


Figure 15: IoT attacks. The highlighted path highlights PUFs' IoT security focus.[38]

2.2.3.6. Attacks against devices based on their properties

Low-cost and high-cost IoT devices are available. Device attacks may vary based on these types. They may cause strange behavior or device failure [71]. High-end device attacks target the IoT network with powerful computers. To attack IoT networks, attackers can use CPUs or even GPUs [72]. Low-power, low-energy devices are used to attack IoT devices, unlike previous attack classes. The attacker attacks the IoT device via radio. Nowadays, smart watches and smart homes are very common. Miniature devices that manage your TV and security system via your smart home network. IoT devices can use smart home utilities [73].

Table 6: Attacks classified by TCP/IP layer [38]

Layer	Attacks	Attackers' Strategies
Physical	Jamming	With radio interference
	Tampering	Making fake nodes
Data Link	Collision	Transmit data at the same time in the same frequency channel until the node runs out of resource
	Exhaustion	Multiple collisions and continuous re-transmission until the node run out of resource
	Unfairness	Repeatedly ask for the channel to limit others' request
Network	Spoofed, or Replayed routing information	Routing loops, changing the source of the route, Repelling network from selected nodes
	Selective forwarding	Send selected information to the legitimate receiver
	SinkHole	Become the target of all nodes in order to gather all information
	Sybil	Create lots of pseudonymous identities to undermine the authorized system
	WormHoles	Re-transmit information to the IoT nodes
	Hello flood	Use Hello messages to flood the network with these tiny messages
	Acknowledgement spoofing	Spoof the link layer acknowledgement
Transport	SYN flooding	Resend request multiple times to fill the capacity of the transport layer
	De-synchronization	Reinitialize the connection in order to disrupt it
Application	Reliability attacks: Clock skewing, Selective message Forwarding, Data aggregation distortion	Impersonate itself as a reliable node in the IoT network and sends corrupted data

2.2.3.7. Attack strategies based on information transmission

Most IoT networks are built with the ability to observe and gather information about their surroundings. As a result, thousands of sensors are utilised to collect data. Additionally, there are numerous methods that may be used Against these sensors, six distinct groups of network attacks can be launched.

Man-in-the-middle attacks: A hacker is positioned in front of the authorized transmitter and receiver.The attacker, though, might keep data from both parties. Figure 16 depicts Alice and Bob fighting. To communicate, In this scenario, Bob receives a packet from Alice. Eve, on the other hand, accepts the packet, saves the crucial details, and gives it to Bob. The fact that Eve is logged into the network is a mystery to Bob and Alice. They think there is a clear connection [67].

Figure 16: A man-in-the-middle attack [38]

Message Replay attacks: An intrusion occurs when someone listens in on the channel and then retransmits the message to the intended recipient. They send packets with modifications to the IoT network.to breach it after the session. [74].

Fabrication attacks: sending several messages of incorrect authentication to the IoT network. Data exchange on IoT devices is harmed by this vulnerability [72].

Alteration attacks: A modification attack aims to compromise the IoT network's communication protocol. To prevent normal protocol operation, they distort and alter user information exchange [51].

Eavesdropping attacks: In these attempts, the attackers simply listen in on the genuine transmitter-receiver communication to gather vital user and network information. The network and the channel are no longer secure. If suitable processes are not used to protect the shared information, the confidentiality of IoT network users may be jeopardized [75].

Interruption attacks: Users' ability to communicate with other devices in an IoT network is disrupted in this attack. Users are actively attempting to connect and making heavy use of resources. This attack depletes vitality [41].

2.2.3.8. Host-Based Attacks

Here, the attacker focuses on the host system's software and hardware. Assuming the intruder has already gained access to the hospital, this form of attack aims to exploit vulnerabilities in the hospital's network infrastructure. Attacks directed at a specific host might be classified as hardware attacks, software attacks, or user-based attacks. Typically, the nodes that make up the IoTs are very small devices that contain some sort of software or application integrated within them. These three classes of IoT devices are the targets of attacks, and each class is compromised in a unique way [76].

Targeting the hardware: Attacks that target hardware involve tampering with components or attempting to crack driver software. Malicious code is occasionally installed on the microcontroller in order to launch an attack on the driver. In other cases, attackers will swap out benign components with malicious ones [77].

Targeting the software: The attackers in this type of attack use the victims' assets. For example, electricity, energy, or even the buffers and queues of different apps or protocols). Battery drain, buffer overrun, and full stack are all possible outcomes of this vulnerability [78].

Targeting the user: User secrets like passwords, hash lists, private keys, or protocol directives are stolen with this vulnerability. Keeping this information in one's head is a magnet for hackers [79].

2.2.4. Classification of IoT Security Attacks

They categorize common attacks by the IoT ecosystem. Although the IoT has no well-defined layered paradigm, Figure 17 depicts a three-layered model [80]. Perception, network, and application are examples.

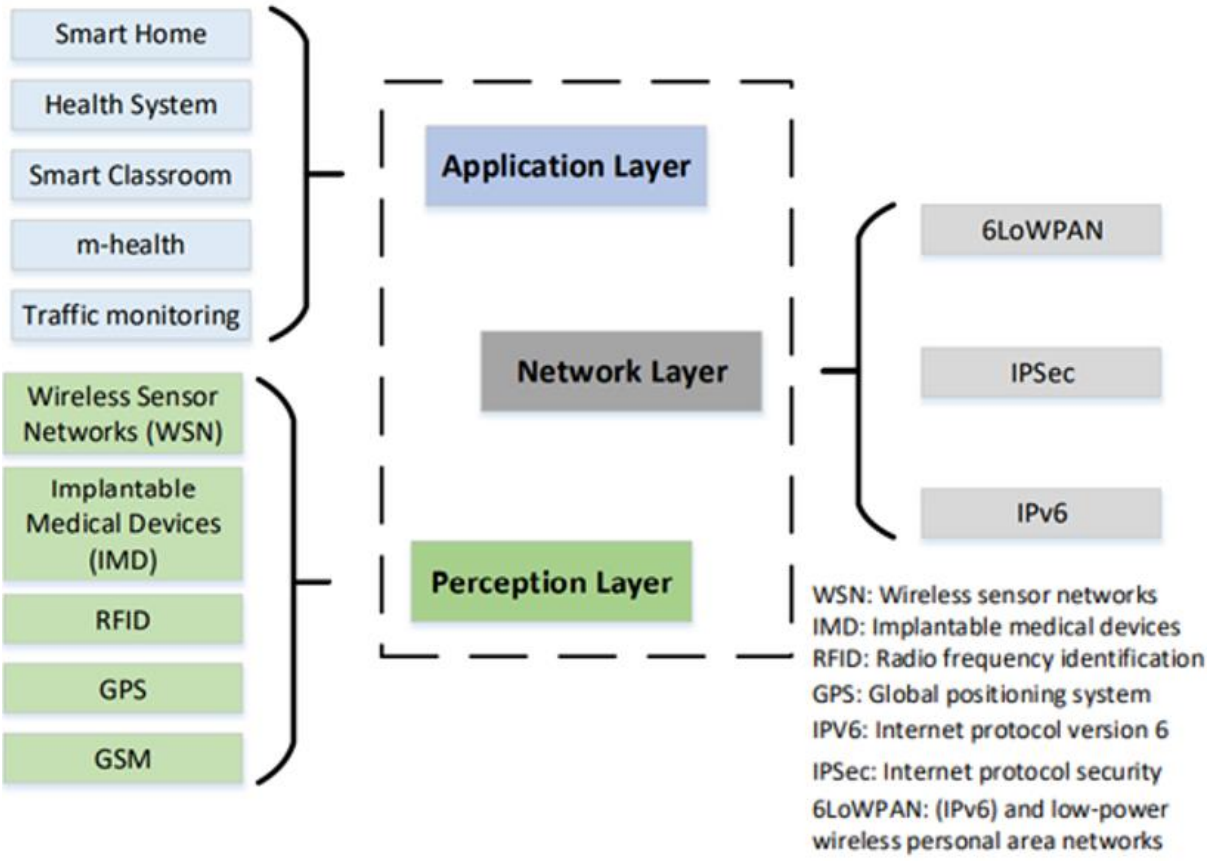


Figure 17: IoT structure [38]

2.2.4.1. Perception layer

The IoT network's perception layer connects the network's nodes. For example, Arduino boards may connect to the Internet via Ethernet, while Raspberry Pi can connect by Ethernet, WiFi, or Bluetooth. Each communication device should have a UUID (Universally Unique IDentification) [81]. These IDs are usually interchangeable.

2.2.4.2. Network layer

Interfaces are also part of the network layer. It is also in charge of network connectivity [84]. No standard network layer protocol exists for IoT devices. But among the most widely used protocols for the IoTs are MQTT 3.1 [82] and Constrained Application Protocol (CoAP) [83]. In this layer, packets are sent to and received from other network nodes (for example, the Internet or a sensor network). Due to resource constraints, this is especially important for IoT devices.

2.2.4.3. Application layer

The use of the same service by all network entities is ensured by this layer.This layer [80] processes data for various apps based on user requests. It can store data in a database for applications including smart transportation, smart home, and eHealth [80].

2.2.4.4. classification of IoT network attacks

Network backbones used by the IoT include Sensor networks that are wireless, the Internet, and RFID. Thus, It is crucial to group all threats into the aforementioned IoT levels. Some of these attacks are outlined in Table 7. PUF could be helpful in increasing the security of some tiny IoT devices because they lack encryption capabilities. [38]

Table 7: Attack taxonomy for IoT layers [38]

Encryption attack	Perception attacks	Network attacks	Application attacks
Side Channel attack	Node tampering	Sybil attack	Virus and worms
	RF interference	Route information attack	Spyware and adware
Man-in-the-middle attack	Node jamming	Sinkhole attack	Trojan horse
	Malicious node injection	RFID spoofing	Denial of service
	Physical damage	RFID cloning	
Crypto attacks	Social engineering	Man in the middle attack	Malicious script
	Sleep deprivation attack	Denial of service	

Due to quick and expanding IC developments, the worldwide supply chain may be attacked at various vulnerable locations. Fake copies, side-channel attacks, reverse engineering, IP hijacking, and hardware Trojans are a few manufacturing dangers.

2.3. Hardware-based Security

2.3.1. General

Hardware has defects in its design. Several places in the world's supply chain could be attacked because of swift and rising IC innovations. Fake copies, side-channel attacks, reverse engineering, IP hijacking, and hardware Trojans are all risks in manufacturing.

2.3.2. Fake Replica

This attack steals the original IP. Different. Pirates overbuild the IC. This can happen if a hacker gets design data during design or fabrication. During recycling, packaging, or with a new seller, a fake can appear [85]. Fake goods can harm an industry. Because the attacker uses the creator's reputation, they employ obsolete or expired designs. Selling counterfeit goods is how they hope to make more money. Additionally, he or she is capable of adding hazardous circuits like Trojans into ICs, endangering aircraft, autos, drones, and UAVs (Figure 18).

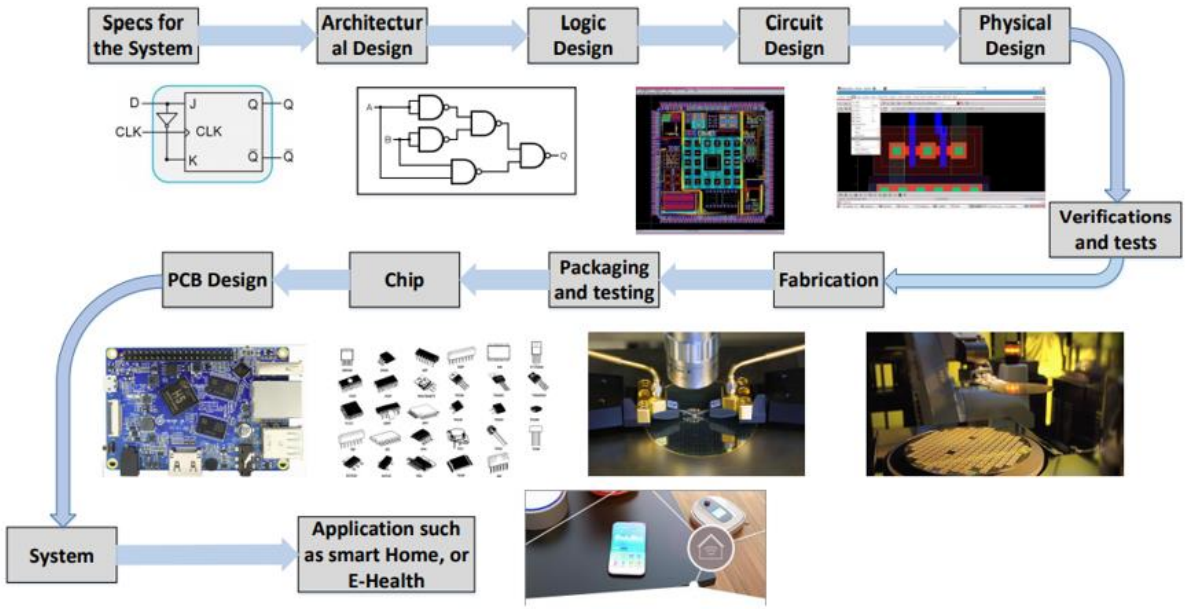


Figure 18: ICs are manufactured from design to application.[38]

2.3.3. Side-Channel Attack

An intruder can learn crucial information from physical states like power usage, timing, or electromagnetic reflection. With this knowledge, the attacker can test the program while it is

active. In public-key cryptosystems like RSA, these attacks [86] are common (RSA). Asymmetric cryptography encrypts and decrypts messages using public and private keys. The multiplication chain is calculated using two different methods [87]. Utilizing delay analysis, a timing side-channel attack. Exponential results are computed using several multiplications [88]. Additionally attacked are electromagnetic emissions, photonic emissions, systemic acoustic noise, and electricity use [89].

Here are various strategies for side-channel attacks:

Acoustic cryptanalysis key extraction attack: A whole 4096-bit RSA decryption key may use a sound produced by the computer when deciphering certain ciphertexts, information can be recovered from notebooks (of various brands) in an hour [70].

Differential Fault Analysis/Attack (DFA): It is a typical method of attack for breaking symmetric block ciphers. The final cycle of ciphers has a flaw introduced so that good and bad ciphertexts can be compared. [91].

Attacks on Power Metering Systems: In these attacks, the attacker monitors power utilization. Attacks can steal cryptographic keys. SPA visualizes power traces. This attack uses Differential Power Analysis (DPA) to bypass countermeasures like additional noise.

Electromagnetic Analysis Attacks (EMA): This attack uses a hardware device's electromagnetic radiation to acquire and analyse data. To recreate information displayed on a computer monitor, these emanations might be recorded remotely. In addition, tiny antennae placed close to the victim IC allow these attacks [90].

2.3.4. Reverse Engineering (RE)

In order to recreate, change, or insert a malicious circuit into an IC or IP, an attacker must go backwards from the application to the design point. This technique is known as reverse engineering. RE may entail a variety of actions, including:

- identifying the technology model being used during the design and manufacturing processes [94],
- removing several design components, including the gate, logic, circuit, and physical [92], and
- figuring out how the IP or IC works and witnessing it in action [93].

This includes appropriating, replicating, and exposing the design's technical details. Intruders can examine a circuit's behavior using a table of inputs and outputs. Thus, the attacker may confirm IP/gate-level design. By reverse engineering IP, the attacker may try to extract a gate, circuit, or physical design. Attacker can implant harmful circuits or copy and sell objects using abstract level.

2.3.5. Intellectual Property (IP) Hijacking

Designers for a company or anyone working on the fabrication process can steal design secrets. An attacker can even make more chips to resell. Untrustworthy people may steal design data and claim IP/IC ownership [95]. IP piracy examples: three. An attacker can steal another's IP address. Invasion may also take layout design. The intruder might also copy the IC design.

2.3.6. Trojans in Hardware

The definition of a hardware Trojan is an IC that has been maliciously altered. This Trojan could deceive communications or ruin control and processing systems. In this attack, a hacker can change or add a harmful circuit to a circuit. Hardware Trojans are difficult to find after wafer manufacture since testing methods are time-consuming and expensive. A large space inside ICs allows for Trojans to be implanted. Logic, circuit, and physical design points are examples [96]. Figure 19 summarizes these attacks and shows which entities in the semiconductor supply chain are vulnerable.

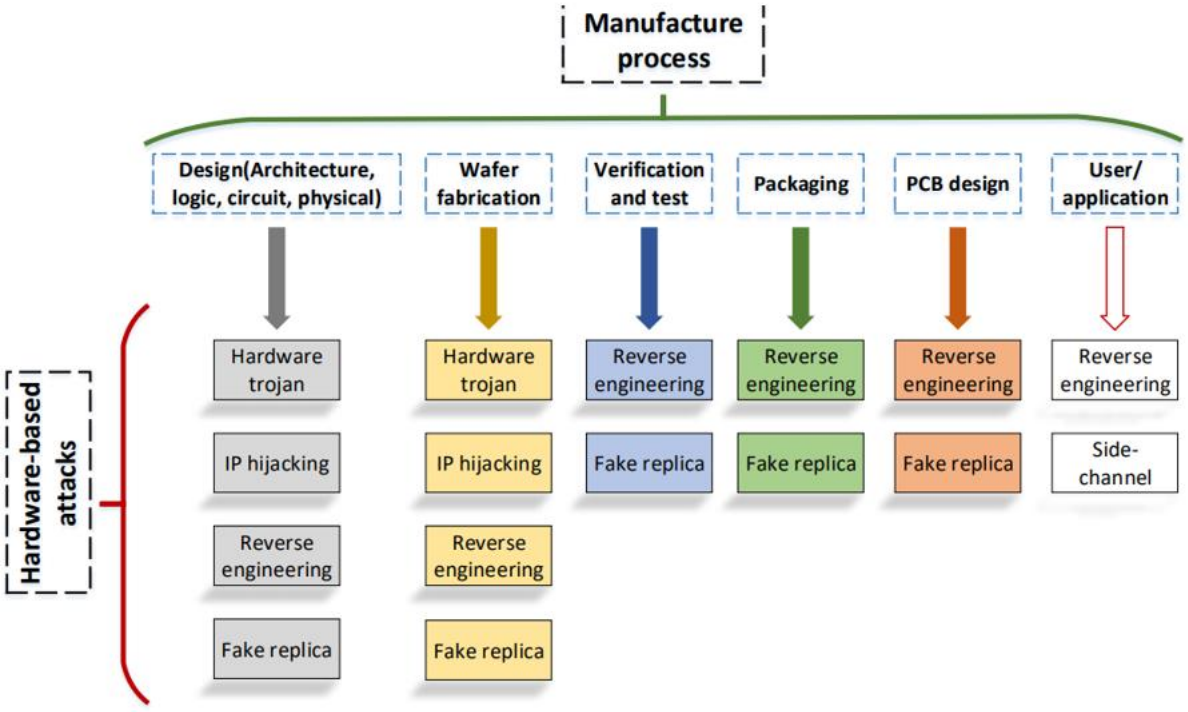


Figure 19: Hardware-based attacks on semiconductor manufacturing entities [38]

In the next section it analyzes further the effect of trojan in hardware.

2.4. Hardware Trojans

The hardware Trojans are introduced early in the IC design and fabrication process. They are latent for a short time after installation before being activated internally or externally Internally triggered mechanisms are circuit activations in combination and sequential. Inputs must be detected simultaneously to trigger a combination circuit. To activate sequential circuits, a succession of events must occur. Sequential circuit triggers have a larger state space than combination circuit triggers, for example. It takes a predetermined number of cycles to trigger a sequential circuit. In addition, external activations can occur at any time. A button or switch provided by the user can operate as a trigger. In some cases, an external component can trigger a trigger. The 'Always on' Trojan is a form of Trojan that requires no mechanism.

A hardware Trojan can change a device's behaviour. The hardware Trojans are designed to bypass, remove, or add logic to the IC, thus compromising its integrity. Adjusting functionality includes changing data or modifying a computing operation, among others. The hardware Trojans can also change parametric properties like clock and timing parameters or non-functional specifications. These Trojans mainly attack existing lines or transistors. Their presence limits the system's processing power and overall throughput. Data leakage and denial of service are the most prevalent Trojan-affected IC impacts. To transmit data from a computer system, Trojans modify the hardware of the modules. Interfaces like RS232 and JTAG commonly leak data.

Rapid data processing in high-performance systems is becoming increasingly important. Heterobolic functions like sinh, cosh, division and multiplication are also hardwired into today's computers. In DSP applications, the CORDIC processor's capacity to perform mathematical calculations has been demonstrated. It uses Jack's suggested Cordic Algorithm. This is done by shifting and adding operations on E. Volder's computer [97].

2.4.1. Hardware Trojan Taxonomy

To comprehend HTs, one must first comprehend their classification. Classification of hardware trojans is important for implementing effective solutions. [99] differentiates HTs based on Trojan activation: On/Off (internally or externally triggered). Based on their physical activation and action qualities, which refer to the Trojan's subsequent acts, Reference [100] categorizes Trojans. Trigger and payload mechanisms are used to categorize HTs into two types, as described in Reference [98]. Any kind of payload, digital, analog, or otherwise, is included. HT

can be broken down into its constituent parts according to its insertion phase, abstraction level, activation, effects, and geographical position. Figure 20 shows their combined and expanded HT taxonomy.

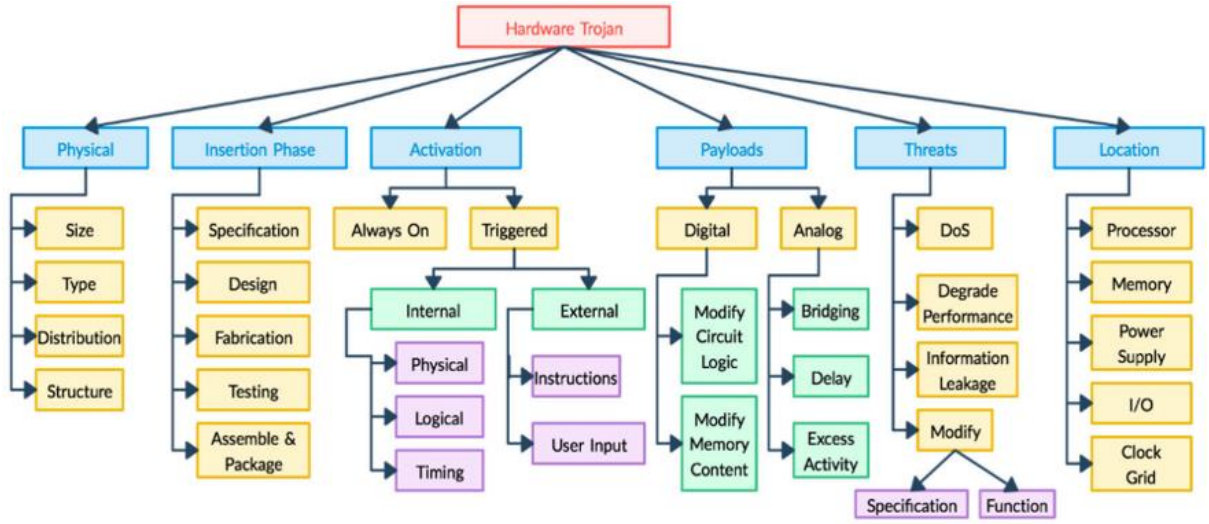


Figure 20: Hardware taxonomy Trojan [90]

Physical: Trojans are first categorized according to their size, type, distribution, and structure in the HT taxonomy.

- Type: It develops parametric Trojan types that are useful. While parametric trojans alter current wiring or logic, functional trojans are introduced by adding or removing gates or transistors [100].
- Size: A trojan's size depends on how many chips are added or destroyed. Smaller trojans have a higher activation rate [100].
- Distribution: This indicates the Trojan's position. Tight distribution describes a Trojan's topologically near components on the chip. Loose distribution is when a Trojan is spread across a chip [100].
- Structure: Adversaries aim to ensure that inserting a Trojan doesn't affect the circuit layout to avoid detection [100].

Insertion Phase: This is the point in IC design and fabrication where Trojans can be added. Specification, design, manufacture, testing, and assembly are HT insertion points [101].

Activation: HTs may be set off by system-wide or external events. A trigger is not necessary for some Trojans to stop a chip from functioning. Always-on Trojans are like this. Some triggers are internal or external. The trigger HT should not be constantly engaged because it can be detected [102]. Trojans can be activated by an internal logical state or counter value. Some Trojans are timed or delay activated. External triggers include remote instructions exploiting poor network security or user input supplied without awareness [99].

Payload: Payloads relate predictable events to Trojan activation. Once the trigger detects the anticipated condition, the payload is activated, and the Trojan begins malicious actions. Analog or digital payloads are used in HT. Digital trojans can enter the host computer and change the logic values of the payload nodes or the data held in memory. Performance, power, and noise are all things that analog Trojans can affect.

Threats: Classifying HTs by threat is possible. HTs can cause DoS attacks, performance issues, and data leaks. HT can also modify a chip's function by adding or subtracting logic, or its specifications, such as its delay [101]. This refers to the Trojan's circuit location. Processor, memory, power supply, I/O, and clock grid can contain HTs [103].

2.4.2. Insertion of a Hardware Trojan

The introduction of HT by unreliable manufacturers or architects is made possible by globalization and outsourcing chip manufacturing. .Figure 21 from [104] shows Trojan injection phases during IC lifespan:

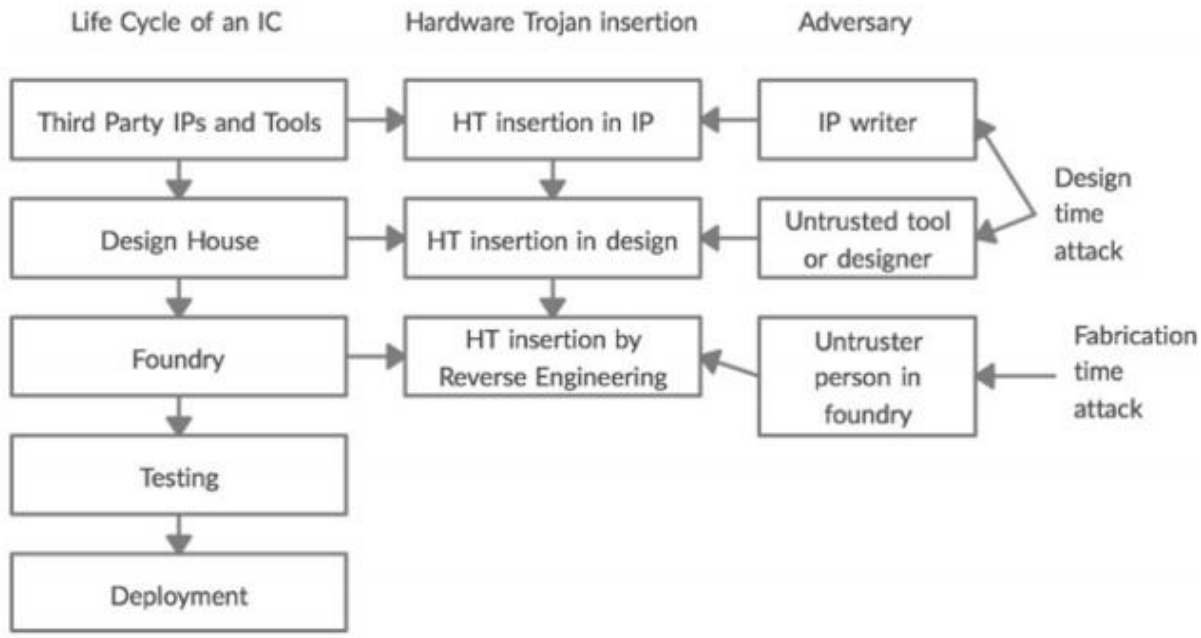


Figure 21: Hardware Trojan attacks on [91]

It is now standard phase for manufacturers to use third-party IP cores and design tools. An HT-infected third-party IP core can be used for design-time attacks. After the design phase, the chip is sent to a foundry for fabrication. These attacks can bypass both pre-silicon and post-silicon testing [104]. This attack uses dopant-level HT as an example. As there are no new gates or wires added, dopant-level Trojans are hard to detect.

To avoid detection during post-silicon validation, HTs must be stealthy in appearance. This is accomplished by designing Trojans that can only be activated by extremely uncommon occurrences, either in a combination or in a sequence. Sequential Trojans, like time bombs, are activated by a series of rare events. On the other hand, a rare combination of internal node logic values triggers a combination Trojan

Table 8 displays 6 HT attack concepts based on adversaries. It shows each model's untrusted party. Three entities are modeled: a foundry, a third-party IP (3PIP) manufacturer and a SoC developer. These are the first ones made:

- Model A: Because SoC developers can't create all IPs in-house, they purchase 3PIP cores with Trojans. Untrusted IP vendors are enemies.
- Model B: This model opposes untrustworthy design firms. Fabless design firms outsource fabrication to modern foundries. An opponent can plant HT in the foundry.
- Model C: Model C's adversaries are third-party EDA tools or fraudster designers. The rising complexity of SoC architectures requires These tools by designers or engineers.
- Model D: Commercial-off-the-shelf (COTS) items don't need special development and are off the shelf. They're cheap yet shady. No development stage is trusted in this concept.
- Model E: Except for the foundry, all supply chains are potential adversaries in this paradigm. A item could be designed or produced in a country that is not friendly or a counterfeiter could put a Trojan on the IC after cloning it.
- Model F: This model assumes that everyone in the supply chain, except for the SoC developer, is a competitor. It is the Model A+B. Some SoC designs use third-party IP cores, and the chips are made by third-party foundries. The only SoC developer you can trust.

- Model G: Companies that outsource ASIC design and manufacture use this model. The system design integrator and foundry are unreliable. After fabrication, testing, and packaging, clients receive chips.

Table 8: Untrusted party Trojan Models [91]

Model Number	Trojan Model	Untrusted Parties
A	Untrusted 3PIP (Third-party IP)	3PIP vendor
B	Untrusted fabless design house	Foundry
C	Untrusted SoC (System on a Chip) developer	SoC developer
D	Untrusted commercial off-the shelf (COTS)	3PIP vendor, foundry and SoC developer
E	Untrusted design	3PIP vendor and SoC developer
F	Untrusted outsourcer	3PIP vendor and foundry
G	Untrusted system integrator & foundry	System integrator and foundry

2.4.3. The use of Hardware Trojans in Side Channel Attacks

The insertion of some HTs serves to launch side-channel assaults. In a setting where side-channels aren't allowed, it advises creating a fake side channel to leak information [105]. Less than 100 Trojan side channels are required in these HTs. Weak TSCs are undetectable in standard testing. The fact that these HTs don't interfere with the device's functionality adds to their difficulty. The first TSC design uses CDMA communications. Since CDMA transports data in multiple code bits at once, code bits change much more quickly than data bits. SC's Pseudorandom Number Generator (PRNG) generates CDMA code sequence (PRNG). This sequence modulates data with an XOR gate. In order to create a hidden CDMA channel, the modulated sequence is forwarded to the Leaky Circuit (LC). Encoded 1 bits leak more than 0 bits. Avril uses these two code sequences to demodulate the higher correlation coefficient indicates more bit transfer. The TSC layout discloses a vulnerability in the key schedule of the AES-128 block cypher. The key

schedule is leaked using known input and key bits. With the correlation coefficient, a differential power analysis attack can tell between bad and good key bits.

A side-channel leakage HT can easily be used to retrieve a key. Manufacturers modify a few gates to improve path latency. It's difficult to detect because no logic changes.

[106] proposes HT-based AFA of HIGHT. HIGHT is a lightweight encryption algorithm for IoT. An adversary must place the HT into the Register Transfer Level (RTL). Solving the SAT system of integrated equations for the cypher and flaws yields the secret key.

2.4.4. Countermeasures

2.4.4.1. Trojan Detection

From the design phase to operation, security must be built in. Detecting Trojans is a common HT defense. An enemy's main goal is usually to launch a successful attack undetected. It has been determined from prior studies that the ability to remain undetected is a crucial feature for hardware Trojans. Changes are considered to preexisting designs and integrated circuits. Pre-silicon design validation is done, while post-silicon verification is done [107]. Figure 22 shows how they categorize Trojan Detection into pre-silicon and post-silicon. Post-silicon detection methods can also be destructive or nondestructive. The following details each pre- or post-silicon technique:

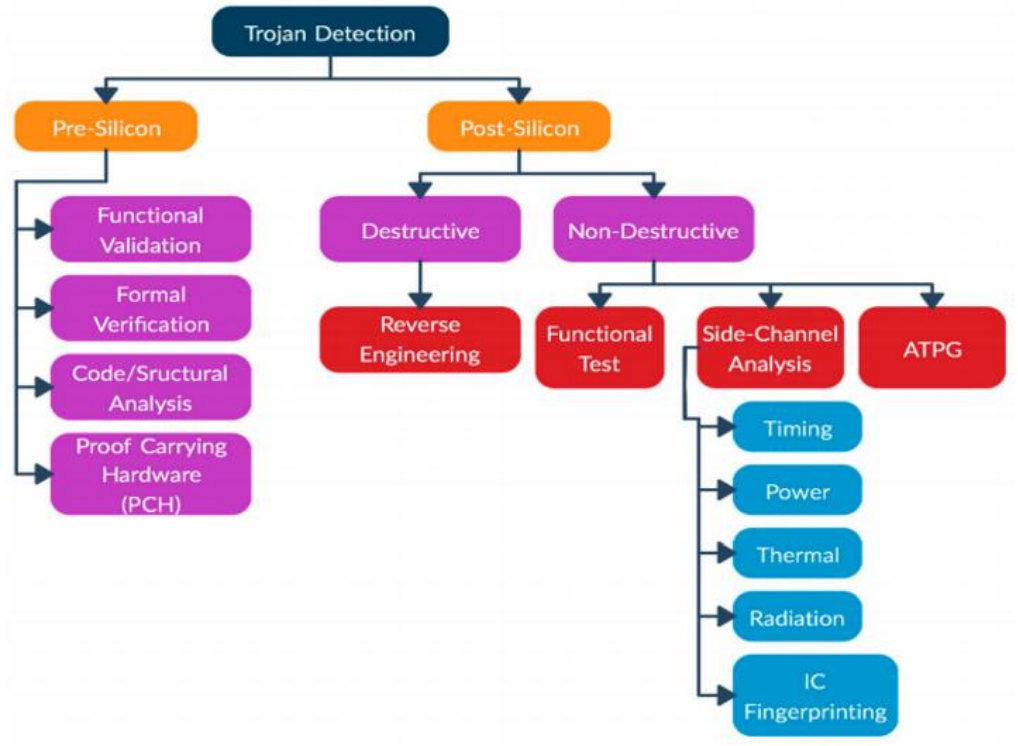


Figure 22: Hardware Trojan Detection. [91]

- **Post silicon Trojan revealing methods:** These methods can either be destructive or nondestructive:
  - **Destructive Methods:** Contaminated methods include reverse engineering, which involves repackaging an integrated circuit and reconstructing the product's DFT validation from each layer. This technique finds malicious IC alterations 100% of the time, but it costs money and takes a lot of time. After the process, the chip is useless and the data is unique.
  - **Nondestructive Techniques**: These techniques are used to validate produced integrated circuits from an unreliable foundry.

- Functional test: For these tests, Trojans must be activated using test cases, and the responses must be compared to the anticipated outcomes. These tests can uncover hidden trojans that evaded detection during manufacturing tests. Functional tests may miss Trojans that do not alter the original circuit's operation. According to references [107], HTs chose idle nets to avoid accidental triggering and side-channel analysis (nets that spend the majority of their time in a single state). The likelihood of all nets switching identifies dormant CUT nets.

- Side-channel analysis: This method uses things like increased path latency, power, or heat caused by extra circuits or Trojans to find HTs [107]. Power, temperature, EM, and other side-channel data are used to generate noise-modelled IC fingerprints. According to the referenced study [108], IC fingerprinting can distinguish between legitimate and Trojan-infected chips to within 0.01 percent of the main circuit size. Most of these tests compare to gold standard integrated circuits, which are rarely available. Trojans utilize extremely little current, hence sensitive detectors are needed to detect leakage current.

- ATPG (automatic test-pattern generation) approach: This technology employs digital signals to activate and assess semiconductors. Several studies [108] employed ATPG to detect HT. MERO (Multiple Excitation of Rare Occurrence) generates experimental results that simultaneously excite rare nodes, increasing the possibility of an HT being activated and clearly based. The paper recommends merging this technology with side-channel detection.

- **Presilicon Trojan detection technique:** These methods are used by developers and engineers to test 3PIP cores and designs. There are three ways to find pre silicon:

- **Functional validation**: This technique is similar to the functional test outlined in the section on post silicon techniques. Operational checks are conducted on a tester by looking at how it responds to each pattern of input. Functional validation, on the other hand, is done through simulation using well-known functional testing methods [107].

- **Code/Structural analysis**: The analysis of codes looks for remarks or circuits that aren't needed contain Trojans.

- **Formal verification**: This method requires a set of security requirements to be met by the intended design before it can be logically validated. Beyond these stated security features, our method may be unable to detect further unexpected functionality supplied by a Trojan.

- **Proof Carrying Hardware (PCH):** A PCH framework verifies the security features of soft IP cores via an interactive theorem prover. Synthesizable cores delivered as a node or HDL code are known as soft IP cores.

2.4.4.2. Design for Trust

It is now impossible to use any method of HT detection that guarantees a 100% success rate. As a result, design-for-trust may be a more effective HT prevention strategy. For DfT to work, untrusted components must be trusted to perform trustworthy computation, and HTs must be prevented from being inserted.

For functional tests and side-channel analyses, increasing the sensitivity and likelihood of detecting HT is called for:

Runtime Monitoring: It is challenging to trigger and detect all Trojans, despite the availability of numerous presilicon and postsilicon detection approaches. The operations of the chip are constantly monitored throughout runtime in order to identify HTs. They can disable, bypass, or trigger extra security processes in response to a detected irregularity during run-time, reducing the consequences of HTs and assuring reliable operation. While the approach is effective at detecting HTs, it is slow. Compatibility-based parallel execution is a kind of adjustable security monitors. Here are the monitoring techniques:

- *Configurable Security Monitors*: Real-time functionality monitoring in a SoC using programmable logic is achieved by leveraging security monitors. Signals are sent to Security Monitors to be monitored. The setup of the Security Monitor for FSM has no effect on regular system functioning.

- *Parallel Execution Based on Variables:* This approach compares functionally equivalent variations on distinct Processing Elements (PEs). Every mismatch is used to find Trojan-infected PEs [107]. Creating variants fast improves efficiency. This method can enhance multi-core processors' confidence, but it costs in computation, performance, and power consumption.

- Hybrid Hardware/Software Architecture: Software-based Microprocessors can be protected by Trojans [109]. Design-verification tests identify unneeded circuits. Exception logic replaces suspect circuitry, preventing HTs [107].

2.4.4.3. Prevention

Design of Trust includes HT insertion prevention. Before inserting Trojans, attackers utilise reverse engineering to determine circuit functionality. Obfuscation, layout filler, camouflage, and trust modules can prevent HT insertion. In details:

- Obfuscation: This strategy involves hiding design functionality and structural elements to make HT insertion more difficult. Obfuscation is key-based. This approach changes a circuit's state transition function, permitting normal and concealed operation. Normal mode produces correct output, while obfuscated mode doesn't. Internal circuit nodes conceal genuine functionality. An adversary cannot insert Trojans unless they are aware of the input vectors and functionality of the circuit.Trojans can be active while hidden.

- Camouflaging is another layout obfuscation technique. In a disguised logic circuit, false connections between layers create indistinguishable layouts. Attackers are prevented from collecting a circuit's gate-level netlist from the layout by using dummy connections. The attacker is thus prevented from incorporating an HT.

- Layout Filler: Attackers frequently inject Trojans into circuits by placing them in unused blank spots. A built-in self-authentication (BISA) technique fills empty circuit layout areas with functional filler cells to prevent HT insertion. To prevent an attacker from replacing them with a Trojan without disrupting the circuit, filler cells are functional.

Most DfT techniques add or replace circuitry, affecting chip performance, processing complexity, area/time fixed cost, and power outflow.

2.4.4.4. Split Manufacturing

Split manufacturing reduces IC design risks. This approach secures ICs by obscuring design intent and prevents unwanted insertion. Foundries create FEOL and BEOL components. Unreliable foundry makes transistors and low-metal FEOL layers. The questionable foundry transports wafers.

2.4.5. Multi-layer hardware Trojan protection framework (called RG-secure)

With so many untrustworthy vendors producing integrated circuits, malicious circuits (hardware Trojans) can be implanted at any stage of an IoT device's lifecycle. With the globalisation of IoT device manufacturing technologies, scientists and IC manufacturers have long prioritized SoC security. Existing SoC high-level synthesis methodologies, such as formal verification and circuit characteristic analysis, cannot guarantee both register-transfer and gate-level security. In their proposal, they proposed RG-Secure, a multi-layer hardware Trojan protection framework for the Internet-of-Things perception layer. RG-Secure combines With scan-chain netlist feature analysis, a design based on third-party intellectual property can be trusted. Our RG-Secure includes light GBM, a distributed, lightweight gradient lifting technique useful in circuit design. The method can process high-dimensional circuit information quickly, enhancing hardware Trojan detection efficiency. Meanwhile, the F-measure is a commonly used metric to assess our strategy's efficiency. The tests show the RG-Secure framework can detect both register-transfer and gate-level hardware Trojans simultaneously. According to the trust-HUB benchmarks, the optimised light GBM classifier a true positive rate of 100% and a 94% true negative rate; it also achieves a 99.8% average F-measure and 100% accuracy. In most cases, an IP seller will sell 3PIPs. Malicious intellectual property sellers may include hardware Trojans in their products (Trojan IPs). They create a distributed HT by synchronising Trojan triggers. RTL is the stage of chip manufacturing where hardware Trojans can be implanted. [223]

2.4.6. Security Technologies for Industrial IoT (Security Controller)

Automation and industrial systems face considerable challenges as they transition to service-based business models. Factory outsiders or insiders can alter data to sabotage processes.

A remote, or software-based, attacker gains privileged access to the processor of industrial equipment via a network connection. The attacker lacks physical access to the equipment's hardware. To extract cryptographic secrets like the secret snapshot authentication key, a successful software attack must first gain arbitrary access to data authentication mechanisms.

Another example is a local attacker who has physical access to industrial equipment's electronic components. An insider, such as a disgruntled employee, or a malicious intruder could be the perpetrator. If successful, a local attacker can access the equipment's memory and steal cryptographic keys.

2.4.6.1. Architecture

They designed a system for the device snapshot authentication scenario to compare the TrustZone and Security Controller hardware security techniques. Our architecture is composed of the following five logical components:

1. *The snapshot generation* component gathers information about the equipment's status from sensors and other activities running on the equipment's host controller.
2. *The equipment snapshot* is comprised of the compiled data. The snapshot signing component accepts a snapshot as input and generates a digital signature using the key storage component's private key. This component makes use of cryptographic libraries that are platform-specific and may be hardware-accelerated, certified or both.
3. *The key storage* component saves the private authentication key permanently and makes it available to the snapshot signing component. Its level of protection is determined by the storage technologies used.
4. *The snapshot distribution* component distributes the snapshot to its recipients using a networking interface, such as an Internet-based central data acquisition server. Additionally, this component comprises the protocol stack required to connect via the specified interfaces, such as TCP/IP or Ethernet.
5. By examining the digital signature established by *the snapshot signing component*, the snapshot verification component confirms the snapshot's authenticity. After that, the snapshot is ready for further processing.

As shown in Figure 23, a system without TrustZone or SC implements and executes all five components in a standard execution environment (red world). In this case, there are two major security risks: Starting with the snapshot distribution component, a remote attacker with network access to the equipment may gain privileged access to all components. An attacker may modify other components' code (a breach of integrity) or retrieve unprotected keys from the Key store (confidentiality violation). Second, an adversary with physical access to the equipment host

controller can probe communication links or steal the key from physical memory to impersonate an industrial equipment instance or edit snapshot data maliciously.

For architectures with ARM TrustZone or a SC, system components must be assigned to the green or red worlds. To begin, they stored the sensitive authentication key in the green world. Second, the snapshot signing component stores sensitive data (the authentication key) in a green environment. As a result, the signing operation is completely isolated from the other components. Finally, to reduce TCB, they reduced the number of green world components. So the components responsible for creating and distributing snapshots, as well as the complex and thus attackable networking stack, are in the red. Verification is carried on a dedicated, possibly remote computer that is not part of this study. So, this component isn't partitioned. As shown in figure, our partitioning strategies are illustrated by the ARM TrustZone and Security Controller components.

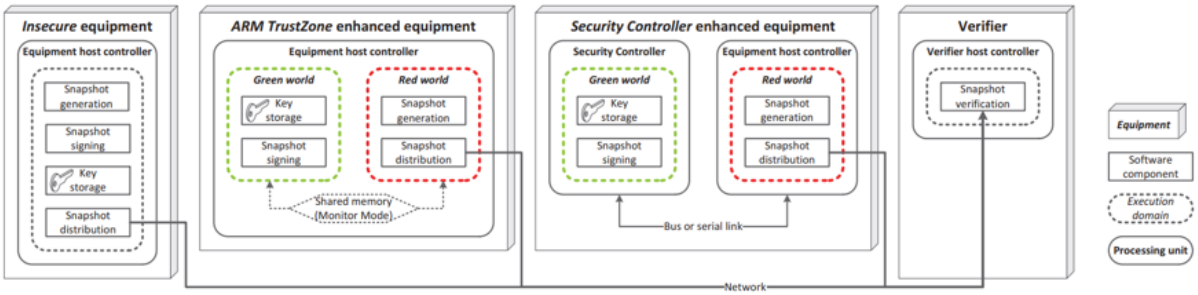


Figure 23: Comparing 3 architectures. Left: snapshot generation without extra security, TrustZone-based, and Security Controller-based [224]

2.4.6.2. Hybrid Approach: Combined ARM Trust Zone & Security Controller

While TrustZone offers greater flexibility and performance, a SC protects against physical threats. They believed that a hybrid method, as shown in Figure 24, can combine the best of both worlds while minimising the disadvantages of each solution. They only examined one potential option because the design space for coupled solutions expands greatly.

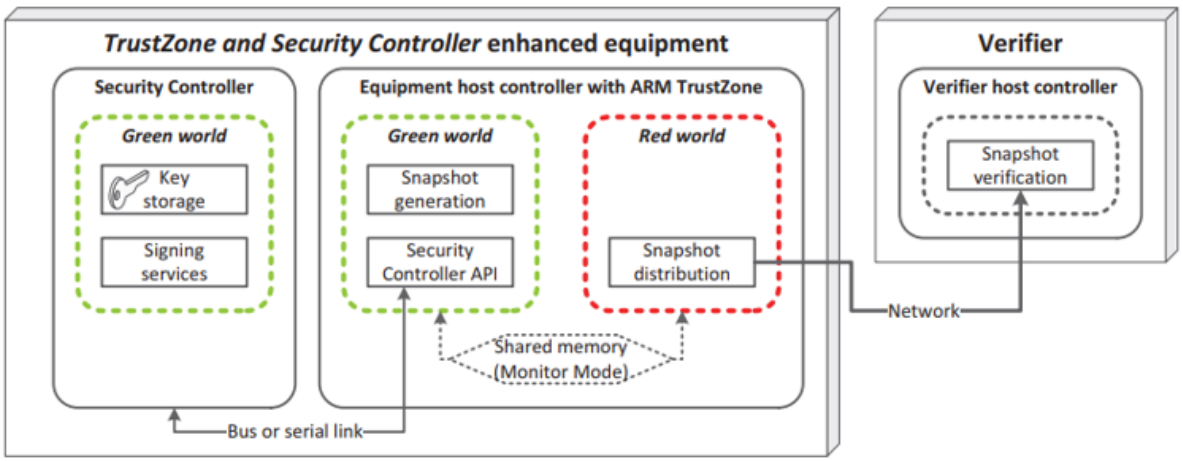


Figure 24: A hybrid approach combining TrustZone and Security Controller [224]

The Security Controller's NVRAM contains the most sensitive data, which is cryptographic key material. The SC offers basic cryptographic activities like digital signing directly to the green world of the host processor via its bus or serial interface. The hashing for a snapshot is saved in the TrustZone green world. The remote opponent cannot compromise the snapshot's integrity or confidentiality. The verifier remains.

For enhanced performance and implementation flexibility, a TrustZone based processor is used instead of a SC [224].

2.5. Physically Unclonable Functions (PUFs)

Physically Unclonable Functions extract a fingerprint from a device based on fabrication and manufacturing variance. External stimuli measure device properties like threshold voltage and critical dimensions. When the device is first measured, the server records it as a challenge. It is a response when the same parameter is measured in response to the same stimuli. The device's authenticity is verified by comparing the challenge and response. An I/O is the difference between a PUFi's challenge and the number of probable CRPs.

Depending on how many CRPs they can hold, PUFs are classified as “strong” or “weak.” A hardware device's "fingerprint" is digitized using production variability. The number of CRP generator components determines how many answers there are in a weak PUF [225]. PUFs have strong CRPs. Although a brute force attack may gain temporary access to the system, the high number of unique CRPs prevents an attacker from applying all responses and gaining access. Strong PUFs are thus generally used for authentication [225]. Longer cryptographic keys require

more PUF responses. An independent CRP is one where one CRP is known but the others in the PUF are unknown. Like PUFs, PUFs are unpredictable. Explicit PUFs are those that go by a different name, like coating PUFs. They are referred to as implicit PUFs if the randomization was created via manufacturing variances [226].

The first PUF-based authentication scheme [228] is as follows: PUF chip is directly attached to server during registration (see Figure 25a). The PUF circuit responds to server-issued challenges. The server keeps CRPs in a table. The IoT device is then fitted with the chip. The server sends a random PUF challenge during the authentication phase (Figure 25b). The response measures the PUF and responds in bits. The device is then authenticated.

Figure 25: PUF authentication modes [227]

Two statistical properties known as intra-distance and inter-distance in [188] govern the utility of PUF:

- "Intra-distance": describes the difference between two distinct Hamming or fractional Hamming responses to a PUF challenge [229].
- "Inter-distance": refers to the difference in response between two PUFs that are distinct, as measured by the fractional or Hamming distance.[229].

These measures provide information on the reproducibility and uniqueness of PUFs.

The device's unique CRPs are used by PUF-based security methods. To utilise any cryptographic technique with a PUF device, it has to be signed up with the server first. During registration, the server checks the client's PUF in order to get a unique answer. This pair of challenges and answers is in the server's memory. The server obtains the matched response during authentication by applying the same challenge used to generate the client's PUF.

2.5.1. Types

Distinct components of the device can extract a fingerprint, resulting in various PUF types. The first technology to extract a fingerprint using randomness was an optical PUF. The optical PUF disperses light particles randomly. When hit by a laser, it makes a random pattern [230]. On top of an IC, a network of metal wires is randomly doped with dielectric particles to create a coated PUF.Doping distribution, dielectric strength, and wire size differences will all affect capacitance between wire pairs. This PUF is used on the top layer of ICs to protect the underlying circuitry from an attacker. Coating removal alters capillary capacitance. PUFs are used by RFID tags [231].

PUFs are now designed based on intrinsic variation. PUFs exploit the inherent unpredictability of IC production. In 2002, they created the first silicon based PUF. Differentiating a circuit produces distinct leakage currents. Due to variations in component production, even a silicon PUF with a comparable appearance will have significant delays. The arbitrator and the ring oscillator are two examples of PUFs that use delays to their advantage. Large component parts are needed to safely assemble these PUFs. These PUFs may not be suitable for IoT nodes due to the demand for chip space and the potential of side channel attacks from heat leaking.

2.5.1.1. Memory based PUF's.

Due to variations in component production, even a silicon PUF with a comparable appearance will have significant delays. The arbitrator and the ring oscillator are both delay based PUFs. These PUFs require extremely large component components to be secured. Due to the need for chip space and the risk of side channel attacks from heat leaking, these PUFs might not be appropriate for IoT nodes. It could be a while before we get a binary number or a voltage reading.in SRAM and ReRAM PUF. PUFs are strong and weak. Side channel attacks on SRAM-based PUFs commonly use a signal analyzer to measure power [232]. SRAM PUFs' CMOS architecture facilitates system integration, but they leak side channel information when switching states. So can the laser's wavelength. So hackers can copy the device [233]. PUFs with Re-RAMs are faster than Flash and operate at low noise levels, making them immune to side channel attacks. Due to the use of resistance, Re-RAM and MRAM consume less power than Flash memory. As compared to Flash and MRAM, Re-RAM consumes roughly 10pJ/bit. Table 9 compares Flash, ReRAM, and MRAM.

Table 9 : Memory technology needs in the present and the future [234-236]

Operation	Flash	ReRAM	MRAM
Program parameter	NOR Vds = 5V; NAND Vgb = 15V	Vset = +100mV	Current: 500uA
Program power required	1mJ/bit	10pJ/bit	100pJ/bit
Program speed (ns)	5000ns/block	2-20ns	2-20ns
Read parameter	Voltage: 10mV	Current: 1-20uA	Current: 1-20uA
Read power required	10 pJ	1pJ	1pJ
Read speed (ns)	50ns	2-20ns	2-20ns

2.5.1.2. A comparison and use of various types

The different PUFs and their features are displayed in Table 10. The main goal of PUFs is to make network nodes safer and entities by combining physical and application security. They can make a cross-layer structure by combining PUF with the other two layers. However, the structures and components that are already there can be utilised to improve the security of the levels above (physical and application). To solve numerous IoT security concerns, PUFs are being created. For authentication and authorization, PUFs are employed.

Table 10: Comparison of PUFs [38]

	Type	Name	Werak/Strong	Comment
Special Fabrication	coating		Weak	Smaller number of CRP
	Optical		Strong	Difficult to evaluate the uniqueness
Silicon PUF	Delay based	Arbiter	Strong	Vulnerable to attacks
		Ring Oscillator	Weak	Needs large power and space
	Memory based	Re-RAM	Strong	Very sensitive to environmental and voltage fluctuations
		NOR	Weak-Strong	averaging reads increases reliability of PUF response
		Butterfly	Weak	unstable adjoining will effect PUF response
		SRAM	Weak	Vulnerable to side-channel attacks

2.5.1.3. PUF Responses' Robustness

Noise and temperature changes can affect the analog physical parameters used to determine a device's fingerprint. Any parameter change can affect the PUF's digital fingerprint. Using differential design strategies, a PUF's stability can be applied to eliminate environmental dependencies.

The PUF output will become noisy due to the environment change. Due to noise, the output bits of a PUF can be flipped, leading to a failed authentication. Perfect repeatability can be achieved by the implementation of mechanisms that shorten the intra-hamming distance of PUF solutions. To improve PUF repeatability, numerous error-correcting coding strategies are applied. These methods should be made to prevent client device confusion and lower client PUF noise, which can lead authentication failures. These actions ought to lower false positives while enhancing PUF quality. various codes for fixing errors were used to make PUF responses more similar. Adding redundant information can help detect and correct errors in challenge-response authentication (parity bit or assistance data). PUFs used 2-D Hamming codes and linear block codes [38].

Fuzzy Extractors generate PUF cryptographic keys

Users can generate a key without keeping it by using PUF-based secret key generation. PUFs prevent device cloning and reduce hacker access. PUFs solve significant security issues with distribution and storage. In the past, PUFs were employed to create cryptographic keys. Reproducible PUF keys can be created in a variety of methods. [237] advocates using BCH codes and RNGs to create fuzzy extractors. In the FE's "secure sketch" phase, BCH codes assist in reassembling a PUF estimate from jittered PUF data. Decoder-free simple scheme Error-correction is thus constrained. Use this authentication method when a very small error margin is allowed. Error-free key generation is necessary for data encryption and decryption. Table 11 contrasts the outcomes.

Table 11: Examining the literature's various fuzzy extractor schemes [38]

Fuzzy extractor construction	Key length	Helper data bits	Failure probability	Flipping probability
Reed Muller Generalized Multiple Concatenated coding	128	13952	10^{-6}	15%
BCH Repetition Code	128	2052	10^{-9}	13%
Generalized Concatenated (GC) Reed Muller	2048	2048	$5.37.10^{-10}$	14%
GC Reed Solomon	1024	1024	$3.47.10^{-10}$	14%
Polar Codes with SC	128	896	10^{-6}	15%
Polar Codes with Hash-Aided SCL decoder	128	896	10^{-9}	15%
Serially concatenated BCH-Polar Codes with SC decoder	250	262	10^{-8}	15%
Serially concatenated BCH-Polar Codes with Belief Propagation decoder	250	262	10^{-10}	15%

The study [238] recently released a FE structure for SRAM PUFs depending on Polar codes. The use of a potent Hash-Aided SC decoder made sure that keys will always be repeatable. The key had a 109 failure probability and was created with 896 helper bits. In [239], they demonstrated how to read SRAM PUF fingerprints using a unique Arduino shield. Then, in order to produce trustworthy keys and forecast FAR and FRR, they evaluated a variety of fuzzy extractor algorithms. Helper data bits that are lengthy and sophisticated decoder structures are suggested as ways to obtain low failure likelihoods. The ideal approach for PUF-based keys may not be to increase the FAR and FRR of decoders created to remove the message from channels that are too loud [38].

2.5.2. PUF-Based hardware security solutions

PUFs are generally used for two types of security applications [241]:

2.5.2.1. PUF-Based Authentication

An IoT device's identification must be trusted. During Enrolment, a Trusted Third Party with access to the PUF-enabled IoT device issues random challenges and stores responses in a secure database. Then it's usable. After that, the authority selects an unknown challenge, collects the IoT device's PUF response, and compares it to a previously recorded value in its database. The

IoT device is authentic if both values match. Reusing challenges prevents MITM attacks. So the verifier must collect a lot of CRPs during enrollment (Figure 26).

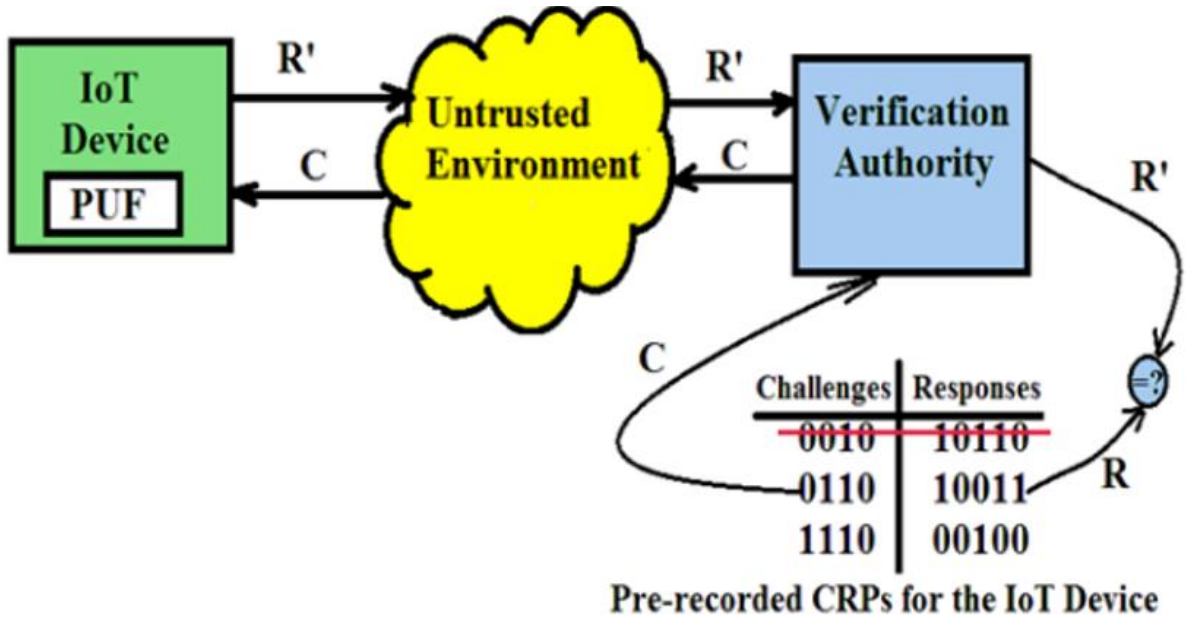


Figure 26: A PUF-Based Authentication Process [240]

2.5.2.2. PUF-Based Encryption/Decryption

Commonly used encryption methods ensure privacy and security. On-chip PUF devices can generate encryption keys. This reduces the risk of side channel attacks.

Due to noise, PUF outputs may vary between evaluations for the same IC and inputs. Consistency of PUF responses is required to obtain a solid encryption key. Figures 27 and 28 show PUF-based symmetric key encryption. This response (R) uses the decrypted PUF (C). Error correction decoding algorithm with pre-computed syndrome bits (S). Correcting noise-related PUF response errors yields a stable encryption key. Ciphertext is then XORed with plaintext. This key is generated using pre-computed CRPs and syndrome bits (Figure 28). Using this method, the receiver can access many pre-recoded PUF CRPs. The IoT device acts as a slave broadcaster, while the base station acts as a master receiver. In this case, the receiver should initiate. C) and syndrome bits (S). Those compute the key and encrypt the message. A channel eavesdropper can receive the challenges (C), but not the key. Periodic key changes reduce this risk.

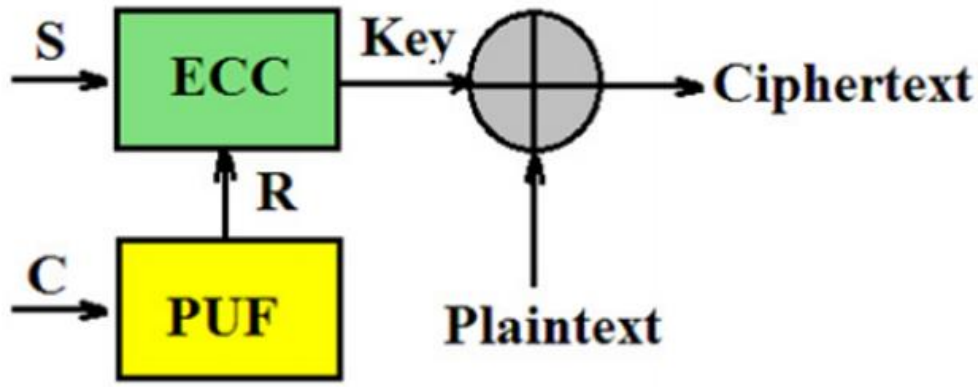


Figure 27: A PUF-Based Encryption Scheme [240]

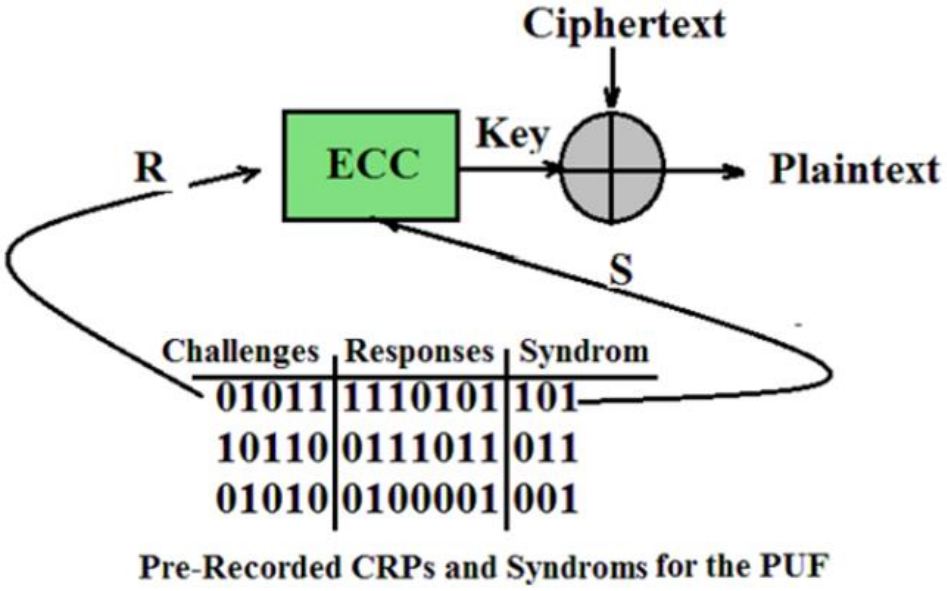


Figure 28: A PUF-Based Decryption Scheme [240]

2.5.3. Design challenges

Many reliability and security issues must be addressed before PUF devices can be used. Transistor ageing is a major source of dependability difficulties. With time, the electrical properties of CMOS devices (e.g. threshold voltage) might alter irreversibly [242]. On the other hand, as illustrated in Figure 29, the reliability metrics of various PUF designs are estimated over time and normalised to the dependability value of new circuits. As seen in Figure 30, variations in operational responses like temperature might affect PUF response accuracy and dependability. In some cases, the PUF may respond differently based on operational conditions and/or time intervals. As a result, an IoT device that relies on PUF authentication may be denied service. The

ECC block in figure 30 may also cause more errors than it can fix. So the PUF cannot produce solid encryption keys.

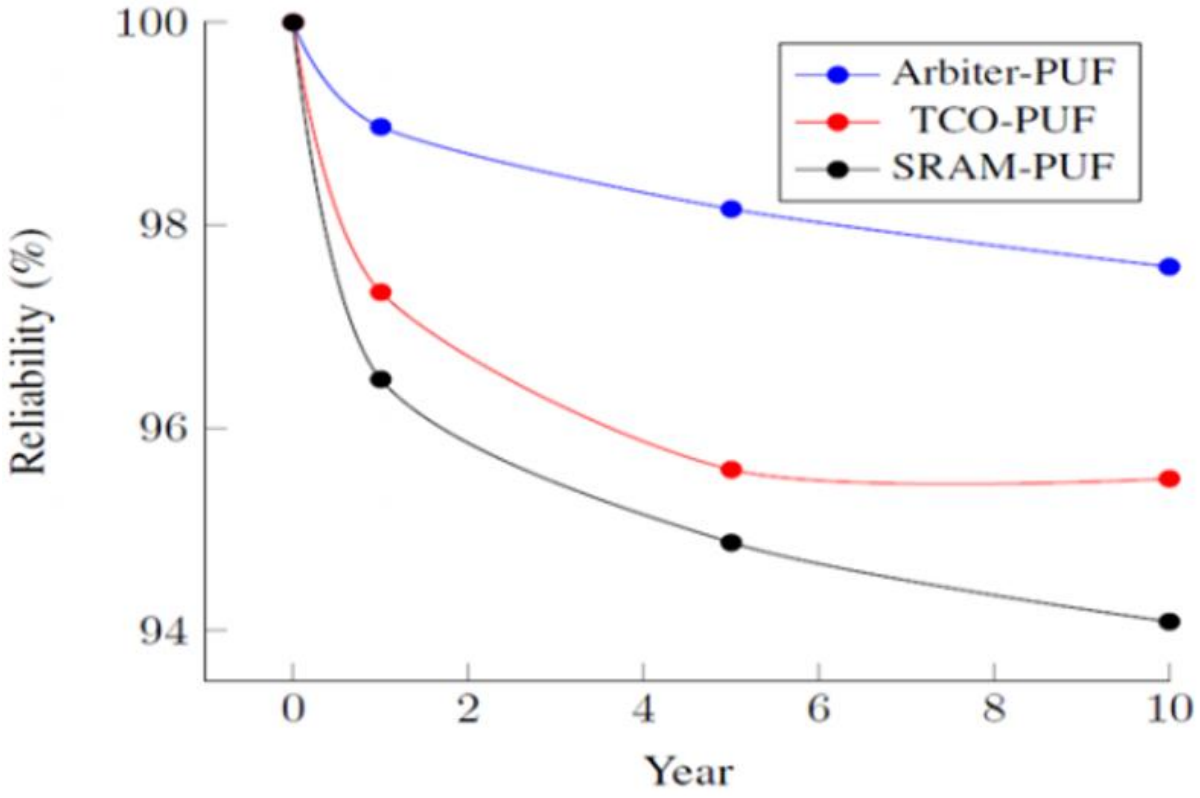


Figure 29: PUF Circuit Reliability in 65nm Technology with Aging CMOS [242]

Modeling attacks compromise the security of PUF technology. With enough PUF CRPs, an adversary can create a numerical model that can accurately imitate/predict PUF responses to arbitrary challenges. An adversary can thus gain unauthorized access to an IoT network. Machine learning (ML), linear programming (LP), and algebraic methods [243] have all been proven to develop numerical PUF models. Moreover, most IoT nodes are located in unprotected remote locations, making CRP collection very easy. PUFs' vulnerability to modelling attacks may limit their use in IoT security applications if not addressed.

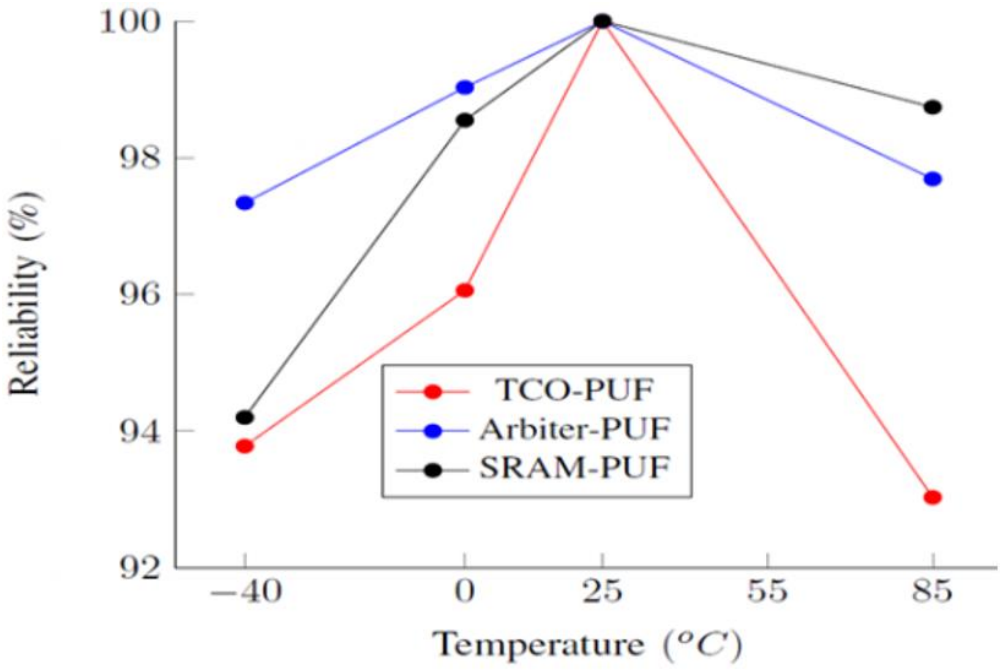


Figure 30: Temperature Effects on PUF Circuit Reliability in 65 nm Technology [240]

2.5.4. PUF-Based Threats on IoT Devices

The main new threat to a PUF-based security system is an attacker gaining the response to correctly respond to a challenge. The PUF can be physically cloned [244] or a modeling attack can be used to predict CRPs [245]. They saw two types of attackers. One can intercept device communication. One opponent physically gains access to the device.

2.5.4.1. Man in the Middle Attack

With the help of this method, an attacker can capture and save CRPs from a server-device channel. Through indirect means, CRPs can be used to teach an algorithm how to learn that predicts future CRPs by modeling the PUF [244].

IoT devices dynamically link to unidentified devices, which facilitates man-in-the-middle attacks. By positioning a Raspberry Pi close by and connecting to the same (potentially encrypted) wireless network, an adversary can attack a device. The right defenses must be put in place because this area is vulnerable to attacks.

2.5.4.2. Side Channel Attack

The attacker in this case has direct physical access to the target device. Two orthogonal axes can be used to classify side channel PUF assaults [245]. There is a spectrum from least to most invasive attacks. Active and passive attacks make up the second axis.

Invasive, Semi-invasive, and Non-Invasive

Invasive attacks compromise chips and get access to internal parts. PUFs were once thought to be immune to such attacks [244] since they would render the PUF ineffective. A chip's secret key cannot be obtained by an intruder [246]. The hacker needs to bring the IoT device to a lab with expensive scientific supplies. This method is less appealing for IoT devices in public spaces because it is impractical to bring them to a lab**.**

Semi-invasive attacks can only be carried out if the passivation on the chip is not destroyed. Reports of PUF assaults involving photonics and electromagnetic probing have been documented [247]. Even though Invasive or semi-invasive side channel attacks need direct access to the PUF. Even though semi-invasive attacks aren't as hard to plan as invasive ones, they still need special lab tools.

Non-invasive attacks use data mining techniques to discover concealed evidence without compromising system integrity (e.g., power usage, delay time). Lightweight and easily transportable [248], the assault gear can be brought near the IoT devices being attacked. Machine learning is used in non-intrusive attacks to assess CRPs and replicate them with great precision [249].

Active and Passive

The active attack modifies the system, such as the supply voltage or operating temperature [250], to conduct PUF attacks. Passive attacks, on the other hand, quietly observe a PUF's temperature or energy use. Both attack methods require physical access to the device and are effective against PUFs.

2.5.4.3. Defense Strategy

Let's start by reviewing side channel attack defense tactics.

Direct access to the PUF is necessary for invasive and semi-invasive side channel attacks. The enemy must take the gadget to a research facility to launch a complex attack on the system. Inexpensive gyroscope sensors can detect when a device is in motion, which can help lower the threat level, but they can't prevent all forms of side channel attacks. However, it is important to remember that these attacks are usually costly, which may discourage inexpensive IoT devices from becoming vulnerable.

These attacks can be carried out without compromising the device's physical defenses using low-cost equipment outside of a lab. Passive non-invasive attacks test the PUF by monitoring external factors like power consumption, making it difficult for machine learning to

build a model of the PUF. Tens of thousands, if not hundreds of thousands, of challenges are needed for this [248]. However, the PUF can be configured to accept just certain types of challenges, making it hard to collect enough, or to slow down the attack by accepting fewer challenges per second [251].

The fact that PUFs have variable responses depending on the context of operation is also exploited by active, non-invasive attacks. Attack modeling can be simplified if they alter these conditions to make fewer CRPs available. Increasing the PUF's resilience to external conditions is a more robust defense technique that can be taken in response to the modelling attack, as it converts active attacks into passive ones. Avoiding CRP reuse [240] is a well-known MIM attack method. Two options: First, CRPs can be encrypted for privacy [252]. With weak PUFs, the number of CRPs is limited, but this requires more computational resources, negating our initial motivation for using PUFs in the IoT. They could also use a (strong) PUF with enough CRPs to avoid reusing one. The most common number to increase CRPs is to increase device processing number [250]. Due to constraints in IoT device processing power, this approach may prove ineffective.

Advanced man in the middle attacks model and forecast CRPs using intercepted CRPs as input. CRPs are needed for attacks. To counter these attacks, PUF's non-linearity was increased [241] Sadly, this method failed [253].

A secure IoT system based on PUF was suggested. It would have to deal with side channel and man in the middle attacks. Protect yourself from direct and indirect attacks by putting up a physical barrier. One common form of IoT side channel attack involves manipulating external conditions. By limiting these types of attacks, PUF design becomes more secure. This emphasizes the significance of carefully selecting a PUF architecture. It is necessary to use PUF in conjunction with an authentication protocol to prevent man-in-the-middle attacks.

2.5.4.4. PUF Architecture for IoT

Picking the optimal PUF architecture for IoT use cases is not easy. The most important considerations in making this decision are as follows.:

i. Resistance to potential attacks.

ii. Cryptographic applications require strong statistical qualities (CRPs' singularity and homogeneity).

iii. Relationship between CRP density and developed area (In strong PUF, the amount of CRPs grows exponentially with computational resources, while in weak PUF it grows linearly).

iv. Ease of implementation on FPGA (after deploying IoT devices, the PUF can be modified and updated to account for emerging threats).

The proposed PUF architectures for IoT applications are summarized in Figure 31, together with their respective strengths and weaknesses

PUF Architecture	Strength	Weakness	FPGA compatibility
Arbiter PUF	(Partly) Resilience against machine learning attack.	Delay path must be identical	No
Ring Oscillator PUF	Easy to implement	Environmental sensitivity	Yes
SRAM PUF	Good statistical properties	Low number of CRPs	No
Public PUF	No secret info	Need further investigations	Need further investigation
TERO PUF	Environmental resiliency	Need further investigations	Need further investigations
Other new PUF (Interpose PUF, LRR-DPUF, array PUF)	Resiliency against machine learning attack	Need further investigations	Need further investigations

Figure 31: PUF architecture [227]

Arbiter PUF

An Arbiter PUF [254] generates a binary value of 0 or 1 based on a comparison of two delay routes of equal size, as shown in Figure 32. Due to subtle differences, one method is faster than the other even though they should have the same potential. The challenge is used to select the two dynamic pathways. To do this, the separate components of the challenge are fed into a series of interconnected multiplexers. Each multiplexer determines which multiplexer to switch its output to base on the input. This results in an infinite number of alternative paths.

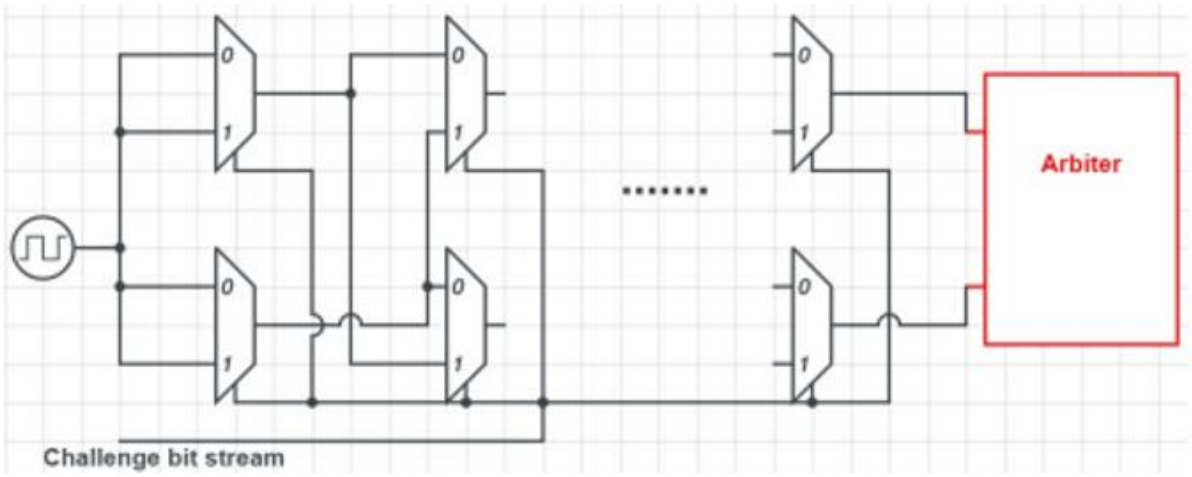


Figure 32: Architecture based on the Arbiter PUF [255]

It is a strong PUF (Criterion 3). In order to achieve acceptable statistical qualities (Criterion 2), all delay paths must be identical in length. Although an Arbiter PUF can be implemented on an FPGA [255], it is time consuming [252] and better suited to ASICs.Many independent ring oscillators work together to form a PUF.

Ring Oscillator PUF

A Ring Oscillator PUF is made up of numerous equal ring oscillators (see Figure 33). Consider there must be two m ROs, the first of which has a frequency of fa and the second of which has a frequency of f2m (for the last RO). A "0" or "1" is obtained by comparing the two RO frequencies. The RO pair employed is determined by the incoming challenge. Although theoretically all ROs should oscillate at the same frequency, some discrepancies arose due to differences in the manufacturing process. RO PUFs' strong advantage for FPGA-based IoTs is how straightforward they are (Criterion 4). (Criterion 2) [256] They exhibit favorable statistical characteristics.

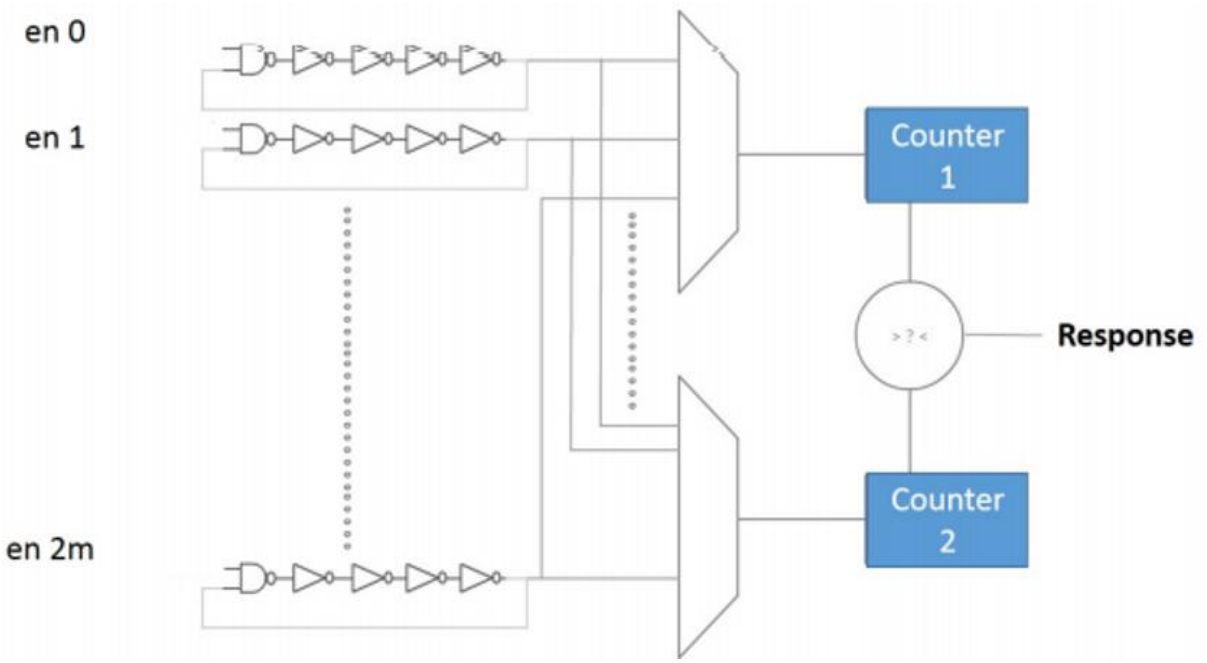


Figure 33: Architecture for a Ring Oscillator [24-7]

SRAM PUF

These PUFs have excellent statistical qualities (Criterion 2) and dependability [252] (figure 34). For this design, static random-access memory (SRAM) blocks are used as the foundation [258]. Once an SRAM PUF is activated, each SRAM cell's value is either "0" or "1". This initialization state will be distinct from other devices [257]. As a result, the challenge can be used as an SRAM cell address, as can the initial value of the SRAM PUF response. SRAM PUFs can be implemented on microcontrollers, but not on FPGAs (Criterion 4). As a result, they are unpopular for IoT PUFs [259]. A limited number of CRPs makes SRAM PUFs weak PUFs (Criterion 3). As a result, it requires obfuscated interfaces, which add extra security layers (Criterion 1).

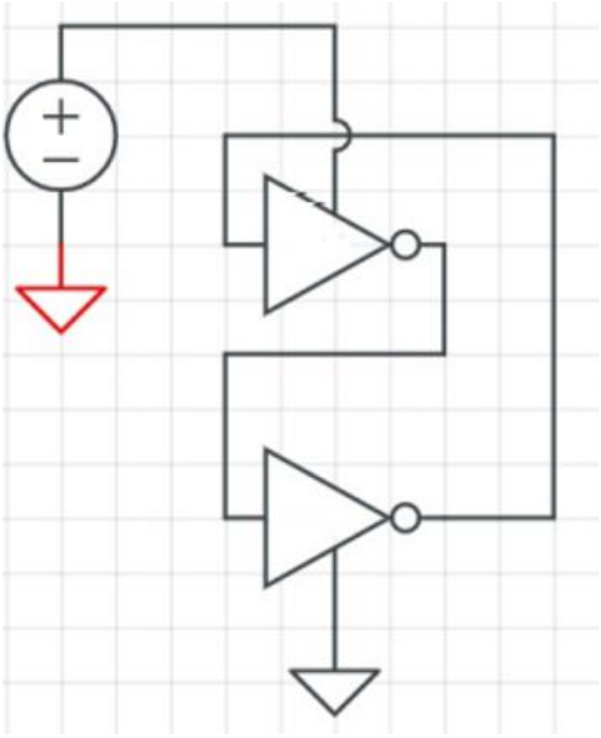


Figure 34: SRAM PUF structure [227]

Newer PUF Architectures

Alternative PUF architectures for IoT devices have been presented recently. In place of RO PUFs, TERO PUFs might be used [260]. Please refer to Figure 35. Preliminary findings [260] rule out electromagnetic analysis as a viable cloning method for this design. TERO cells, which make up a TERO PUF, can either be transitory or stable. An RS flip flop is a TERO PUF. Assigning a value of one to the init signal triggers a short oscillation.

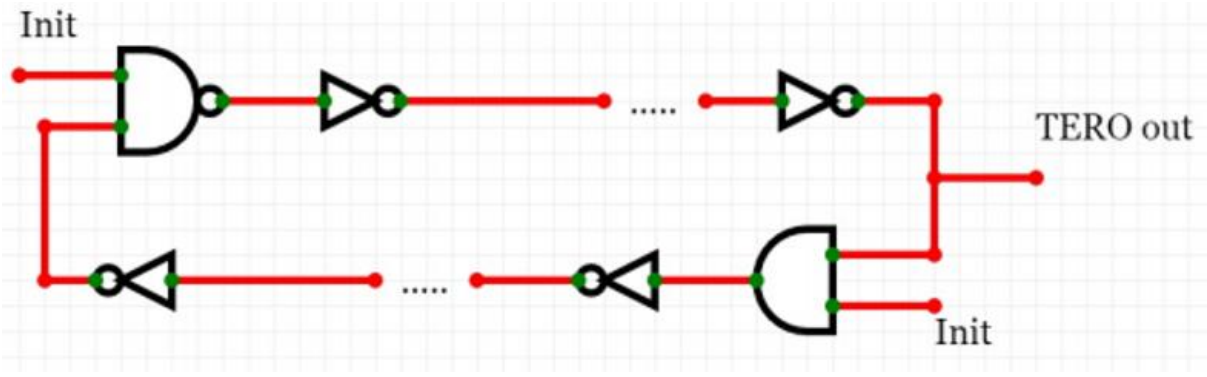


Figure 35: TERO PUF constructions [262]

A hybrid ring oscillator arbiter PUF and Public PUF (PPUF) [263] have also recently been proposed for IoT use. Hybrid ring oscillator arbitrator PUFs combine ring oscillator and arbitrator PUFs. They are perfect for battery-powered IoT devices due to their low power consumption [264]. PPUFs are a type of arbiter PUF that utilizes XORs rather than multiplexers. Public key protocols that are low-power, area-efficient, and immune to side-channel attacks can be made using PPUFs [263].

They are still in their infancy, and their inherent strengths and weaknesses are unknown. These novel PUF architectures must demonstrate their suitability for IoT devices based on FPGA or whether they are more appropriate for ASIC technologies. Also, the following steps are advised:

i. Putting these new PUFs together with authentication protocols to meet IoT authentication needs.

ii. FPGAs are used to put the solution together with limited computing resources, such as Xilinx's Artix 7 or Spartan families.

iii. Assuming a detailed report on the necessary computing power, the term "lightweight solution for the authentication procedure" is evaluated.

iv. Working together with third parties to reassess the findings obtained. Thus, these new PUF architectures need more study before they can be implemented in practical IoT systems.

2.5.4.5. PUF Protocols for IoT

PUFs are used in numerous authentication schemes. Figure 36 summarizes. Protocols are frequently not independent of PUF strength. Protocols using weak PUFs frequently rely on cryptography to compensate for CRP scarcity (e.g., hashing, encryption, etc.). This effectively eliminates PUFs in IoT.

The original premise of an unclonable or modellable PUF] was used in the first PUF authentication protocol More research revealed this was incorrect. Machine learning analysis of CRPs revealed PUF fragility]. A PUF authentication protocol must therefore be robust to machine learning attacks.

PUF Protocols	Strength	Weakness	Compatible to which PUF?	FPGA Compatibility	Further Notice
Early protocols	(Partly) Resilience against machine learning attacks	Cryptographic primitives utilization	Integrated into different PUF architectures	No	Occupy considerable amount of computational resources
Mutual authentication protocol	Mutual authentication feature	Assumption of perfect PUF architecture	Not specified	Need further investigation	Implementation overhead has not been reported
Obfuscated challenge response protocol	Resiliency against machine learning attacks	Dependency on random number generation	Arbiter PUF	Need further investigation	Computational resource utilization & FPGA compatibility need to be clarified
Lockdown protocol	Resiliency against machine learning attacks	Need further investigation	XOR Arbiter PUF	Need further investigation	Computational resource utilization needs to be clarified

Figure 36: PUF protocols [227]

The ability to deal with (partially) unstable PUFs is another important PUF protocol feature. Protocols generally require error correction mechanisms to account for responses that are not completely stable, contrary to initial PUF definitions [265].

Finally, IoT systems require mutual authentication. IoT sensors should be protected from sending sensitive user data to unreliable (unauthenticated) services that might accept manufactured sensor information from attackers unless both parties' identities are confirmed. An ideal server would be for mutual authentication not only between devices but also between devices and servers, given the frequency of direct IoT connections.

Early Protocols

After machine learning attacks were revealed, many writers devised countermeasures. The protocols process the response using cryptographic primitives such as hashing or encryption techniques and send it back to the server. NVM, on the other hand, contradicts one of the primary goals of using PUFs, namely that no secret should be stored on the device. Insufficient resources and susceptibility to physical attacks make IoT systems particularly susceptible. The IoT wasn't a consideration during the development of early protocols.

Mutual Authentication Protocol

A PUF-based mechanism for reciprocal authentication of IoT devices was proposed in reference [266]. The protocol allows for two-way authentication between devices and between devices and the server. The current method of authentication between devices and servers is as follows: Every gadget connected to IoT has its own special code. To begin, the device sends the server its ID and a random nonce N1. To generate a random number, the server picks a CRP (Ci,Ri).

For a wide feature of IoT applications, this protocol's primary benefit is mutual authentication. Not only did the authors continue to assume an ideal PUF, but it was robust to machine learning attacks. We didn't consider PUFs that are unstable. The protocol, like others, used cryptographic primitives, which went against the PUF's basic premise. The additional effort required for hardware implementation is unknown.

Protocol for Obfuscated Challenge Response

The obfuscated challenge-response protocol is resistant to machine learning attacks, in contrast to the prior approach. No mutual authentication is supported, so IoT devices must authenticate with a server. Partially unstable PUFs are ignored. The protocol only relays a subset of the whole challenge to an OB-PUF. To hide the correlation between the two figure 37 illustrates the three components of an OB-PUF: a random number generator, a control block for challenges, and a PUF, which in this case is an arbiter PUF.

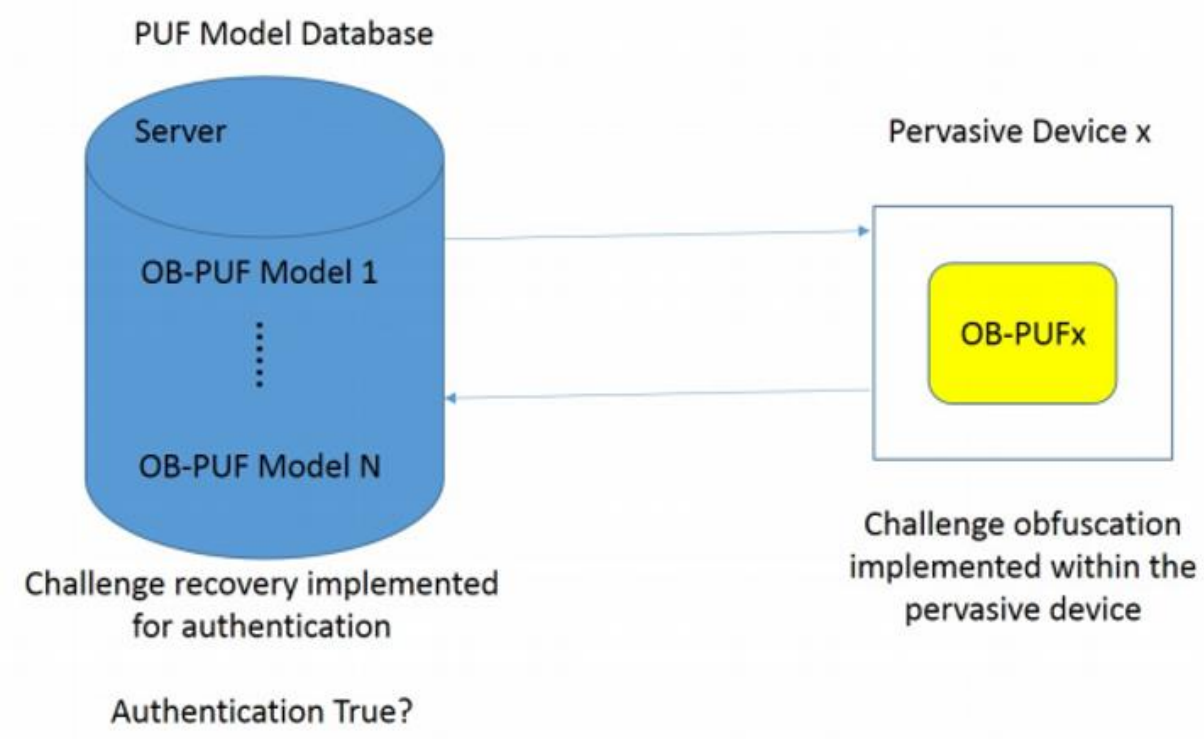


Figure 37: Obfuscated PUF structure [267]

Lockdown Protocol

Unlike the previous protocol, the lockdown protocol [268] does not use cryptographic primitives. It allows server-to-server authentication but not device-to-device. Unlike previous methods, lockdown can deal with unstable PUFs. They presented both strong and weak PUF versions, but we focus on the strong PUF version.

2.5.5. Using PUFs on IoT

Traditional PUF architectures were vulnerable to machine learning attacks and changed the physical features of their physical environment while promising unclonable fingerprints. In recent decades, modern constructions have achieved great progress. To assess their relative advantages and demerits, they must be viewed from a range of aspects. They are perfect for battery-powered IoT devices due to their low power consumption [264]. PPUFs are a type of arbiter PUF that utilizes XORs rather than multiplexers. Public key protocols that are low-power, area-efficient, and immune to side-channel attacks can be made using PPUFs [263]. However, these two approaches account for most suggested protocols.

Typically, an IoT device will request authentication from a server. Mutual authentication is offered by just a minority of methods. Identity management is crucial in the IoTs. Is it possible for PUF to help here? Authentication is difficult, if not impossible, without a server. A server acts as a mutually trusted mediator in the Mutual Authentication Protocol, which is the sole protocol that provides a device for devising authentication.

While an XOR arbiter PUF is robust, it is better suited to static ASICs than dynamic FPGAs. These PUFs can't be easily upgraded, which is a feature in the IoT environment. For consistent use, ring oscillator PUFs will need to be strengthened so that they are less susceptible to environmental fluctuations.

Encryption may be unnecessary when simply authentication is required (e.g., smart locks). They needed to connect securely in many cases (e.g., to update IoT devices' firmware), so encryption was required. In this instance, PUFs have multiple applications, including key exchange and random number generation.

Repeated usage of CRPs is not allowed in protocols, and robust PUFs are necessary to prevent replay attacks. It's impossible to estimate how many IoT devices will be needed due to their long-life expectancies. CRP options are constrained by the need for low cost and low power consumption in IoT devices. The consequences of losing access to CRPs while the device is operational are unclear. Small PUFs have a number with CRPs, which can be solved by techniques like [269].

Another issue is device management. A server must read Before transferring CRPs to a customer, store a large number for each device. There is around one terabyte of data for every million PUFs. Their loss renders authentication impossible. The system becomes vulnerable if data is compromised. Deploying PUFs on FPGAs makes device since they can be modified afterward.

Finally, PUFs may be compromised if the attacker has physical access. There is no doubt that the attacker can read, but if they keep the PUF, they can use it to bypass hurdles. Restriction of attacker's ability to read CRPS quickly may be useful if attacker only has temporary access to PUF As a result, secure hashing is usually used or valid authentication is severely slowed.

2.6. Additional Hardware Security Methods for IoTs

The level of security required by each device determines the extent to which multiple security measures can be combined effectively.

Device Identity

Device identification is required for accountability. Organizations employ digital certificates and Public Key Infrastructure (PKI) to safeguard digital communication. Device authentication, host-to-host communication security, and TLS/SSL security all make use of digital certificates. Because of its scalability, PKI is well-suited for authenticating and verifying the authenticity of electronic devices. Manufacturers of electronic equipment need to couple PKI with a workable certificate management system because digital certificates expire [274].

Hardware Security Module (HSM)

In order to protect cryptographic keys and data encryption operations.it is necessary to use tamper-resistant hardware security modules. Cryptographic keys, trade secrets, and other confidential information are all safeguarded by HSM. This HSM device can be used on its own, in conjunction with a server, or as part of another piece of hardware. Our devices' validity could be increased by implanting a semiconductor chip with a unique identification. A key injection occurs here. HSMs act as a Root of Trust for this procedure, ensuring the integrity of the key injection process. Using HSMs to generate, secure, and manage these keys [275].

Trusted Platform Module (TPM)

The Trusted Computing Platform Alliance (TCPA) created TPM to solve issues of confidentiality and safety. The TPM is a secure microprocessor that offers cryptographic capabilities and can be implemented in hardware or software. The TPM hardware serves as the hub of the platform's trust infrastructure. TPM features an RSA engine capable of 2048-bit RSA encryption/decryption for digital signatures and key wrapping. TPM also features a built-in hash mechanism for computing tiny data hash values. Multiple PCR are used by TPM, each of which stores a single cryptographic hash and can be accessed by third-party programs. To do this, each boot chain binary computes and stores the hash of the next binary. To guarantee the measurement log's integrity, compare the measured values' sequence to current PCR values. Encryption of the TPM attestation key is also given. Platform authentication is done via Attestation Identity Keys (AIK). AIKs are created using TPM certificates [276]. Three different types of certifications are available for TPMs:

(1) Endorsement Key. Each TPM has its own unique public/private key pair, which is used to verify its identity.

(2) The platform certificate certifies platform security.

(3) A third-party certifies the platform's security [276].

TPM establishes identification using EK and AIK. In addition, TPM supports software attestation and authorization. The Trusted Execution Environment (TEE) of TPM provides protection from both software and hardware risks. TPM adoption in IoT devices has drawbacks, including additional expense and size and power consumption [276].

Roots of Trust (RoT)

The operating system implicitly trusts hardware/software components as roots of trust. It gives a cryptographic system confidence. Cryptographic keys, device authentication, and software validation are all areas where it should be implemented in hardware and relied upon. Secure central processing unit (CPU), runtime memory (RAM) that protects data in use, tamper resistance, tamper-proof TRNG (True Random Number Generator), TCXO (Tamper-Proof Counter), and safe storage are all examples of such devices [277], as a result, the host is alerted to an attempted harmful insertion.

Device Identifier Composition Engine (DICE)

Deployment, firmware attestation, and data encryption are all part of the DICE security standard from the Trusted Computing Group (TCG). For resource constrained IoT security and privacy devices, its small hardware requirements make it ideal for production hardware into security products. An easy-to-implement, one-way hashing algorithm suitable for use in a microcontroller. The DICE design is designed to improve security in IoT devices where TPM are not feasible due to resource constraints. The boot is layered. Each layer computes a unique "secret" that is kept secret. The structure is centered on the Unique Device's secret (UDS). The secrets change when a new code is booted. If a vulnerability reveals a secret, the device is immediately re-keyed, protecting the secrets. [278]

Chapter 3

Internet of Everything (IoE)

3. Internet of Everything (IoE)

3.1. Preface

This approach was developed in response to the rise of wireless-based technologies and the IoTs [174]. This approach integrates data, processes, things, and people, bringing previously distinct pieces together.

To put it simply, the Internet of Everything concept is a scenario in which Millions of protocols are used to connect billions of things through public or private networks (standard and proprietary) and are capable of sensing and/or reacting with their environment (actuators). To summarise, the IoE paradigm has four separate aspects (data, processes, things, and people), rather than just one (things) [173].

IoT connects people, data, things, and processes. As shown in figure 38, loT is totally composed up of "things." IoE also improves people's life through commercial and industrial activities. Independent devices such as M2M, P2M, and P2P systems are now connected to the Internet. Figure 38 depicts the IoE encirclement of people, processes, data, and things [176].

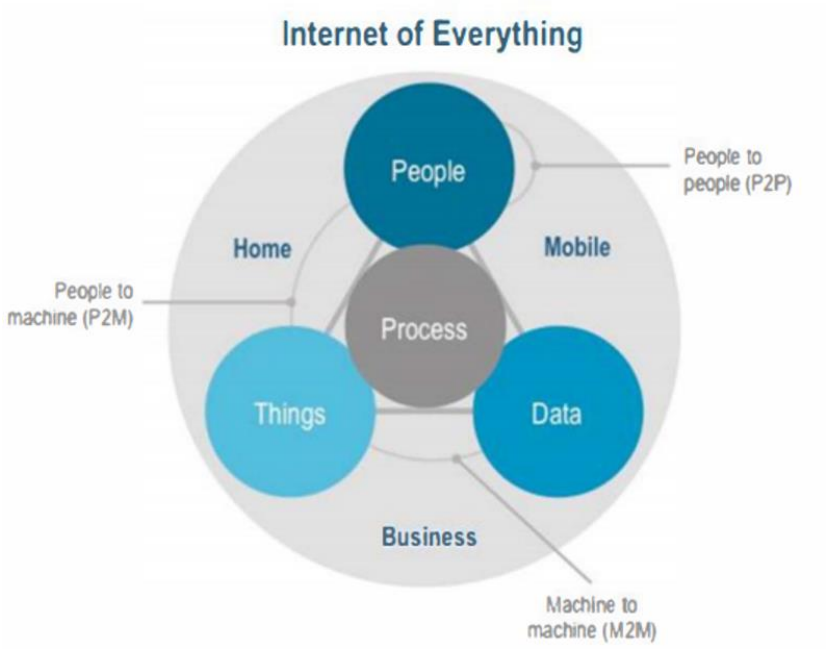


Figure 38: Internet of Everything (IoE) [176]

It affects people, corporations, and industrial processes. Real-time data from various sensors is aggregated and applied to automated human-centered operations [178]. The IoE aids in achieving societal, financial, environmental, and political objectives. Additionally, it is used for automation, e-learning, remote monitoring, smart grids, traffic control, and fossil fuel mining. Figure 39 depicts the IoE's framework. We are the sum of our actions. This results in daily production of billions of bytes of data. While typical data management systems can manage IoE data, they are not always efficient. To assure the security of IoE generated data, research is underway.

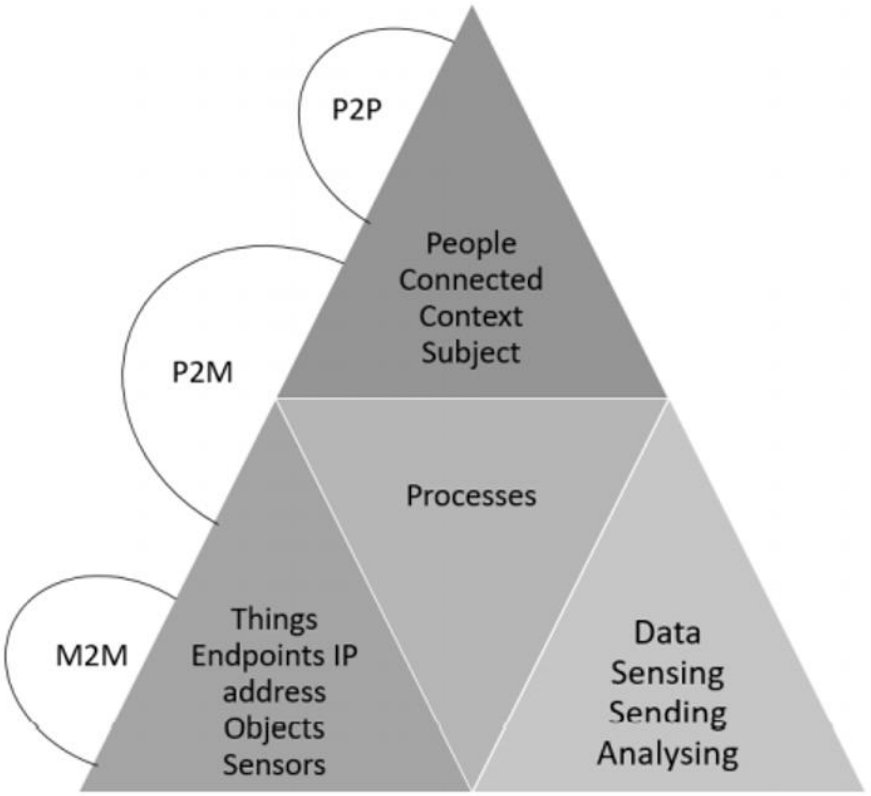


Figure 39: Internet of Everything (IoE) [177]

New concepts like the IoE enable machine-to-machine and person-to-person connection. For software processing, IoE and large data systems must be converged. IoE collects data from multiple sensors. Big data analytics is the gathering of collecting, aggregating, and utilising data to improve one's lifestyle [179]. Throughout this process, ensure privacy and security.

3.2. Hybrid PUF

3.2.1. Preface

IoE will continue to expand as new materials and manufacturing technologies are developed [190]. Printed electronics will introduce billions of new networked devices to the IoE. This increases demand for security solutions, particularly low-power devices. An NVM is currently necessary to store secrets like unique IDs or cryptographic keys. NVMs are vulnerable to physical attacks since their memory content is permanent. To overcome these challenges, researchers propose leveraging random inherent changes of circuits as a source of entropy to create reproducibly unique IDs [191]. PUFs. Anticounterfeiting and cryptography are common target applications.

3.2.2. Security analysis

3.2.2.1. Basic Architecture

Printing an M-inverter array hybrid PUF core the inverters use electrolyte-gated transistors and inkjet-printed resistive loads. Peripheral logic circuitry (Si) houses the printed PUF core (10). Figure 40 depicts the scalable hybrid PUF concept's basic architecture. Inverter output voltages vary due to random transistor threshold values, which are compared to generate unique identities.

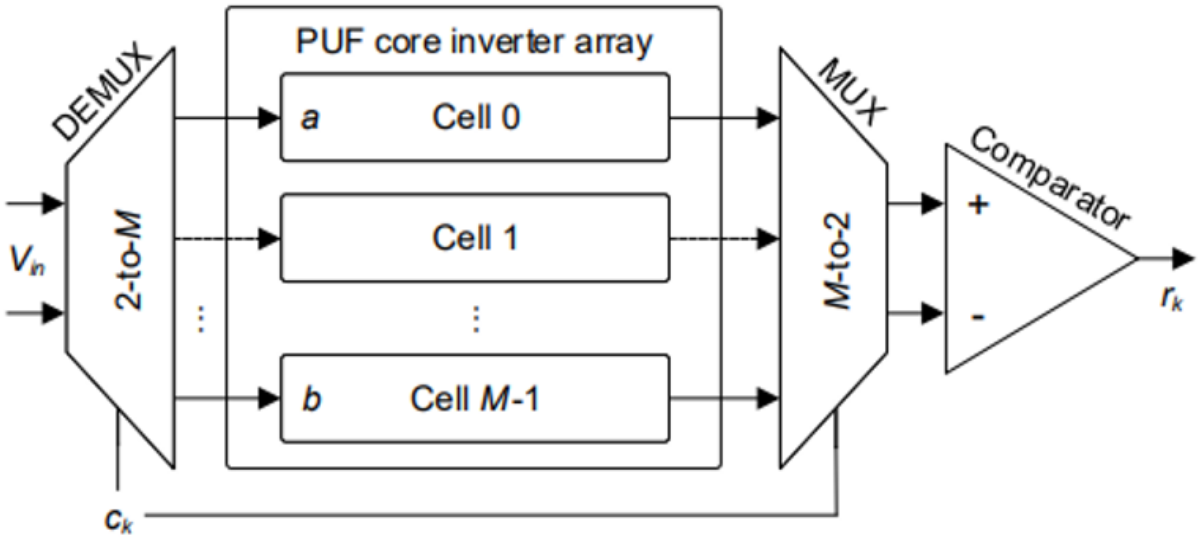


Figure 40: Hybrid PUF Architecture [189]

3.2.2.2. Intra- and Inter-Hamming Distance

The intra- and inter-HD are important performance indicators:

1) Intra-HD: When a fixed challenge is used, the difference in reaction times between two PUF responses from the same PUF instance. It assesses the reproducibility of PUF responses under varying conditions.

2) Inter-HD: HD difference between different PUF responses to the same challenge. The inter-HD measures the PUF answers' uniqueness.

3.2.2.3. Performance Comparison

Table 12 compares the work with other Si-optimized PUFs. This hybrid PUF outperforms the PUF sensor [192] and the RO-PUF [193] in values of FAR and FRR. The hybrid PUF is the first electrically novel material PUF to be tested. The hybrid PUF has the best intra and inter values, demonstrating that our PUF's reactions are naturally entropy-rich. The hybrid PUF's low FAR and FRR values make it incompatible with cryptography but good for job identification. This makes the hybrid PUF an appealing alternative for a low-cost, lightweight hardware intrinsic security primitive.

Table 12: PUF intra-HD, inter-HD, FAR, and FRR comparison [189]

PUF type	Intra-HD	Inter-HD	FAR†	FRR†	Ref.
	μ_{intra}	μ_{inter}			
This work*	1.50 %	50.01 %	-3.61	-3.61	
Pseudo-LFSR	2.72 %	62.73 %	-2.96	-3.05	[15]
SRAM-PUF	-	-	-4.01	-4.03	[16]
PUF sensor	7.16 %	31.00 %	-6.04	-6.02	[17]
RO-PUF	1.53 %	49.60 %	-6.06	-6.20	[14]

Note: *Printed/hybrid PUF. †$log_{10}(\cdot)$ of the value.

3.3. BlockChain

3.3.1. Preface

E-commerce grew faster because of e-commerce transactions [195]. But in all cases of E-commerce, the business deals were handled by a single organization. The risk of a single point of

failure increased, and the integrity issue persisted. The central entity's presence also adds to transaction delays [195]. Blockchain technology can solve many issues. In this way, all transactions have full or partial ledger, ensuring complete transparency. Since its debut in 2008, it has been tested in numerous applications (See Figure 41). The IoT is a subset of the larger IoE. Vulnerabilities increase in number to the number of devices in IoE environments. [196] Each new device represents a new attack vector. Cryptographic hash functions ensure the blockchain's security and consistency. Many IoT devices are vulnerable, and data attack is a major concern. The blockchain may be a viable solution for IoT architectures [197].

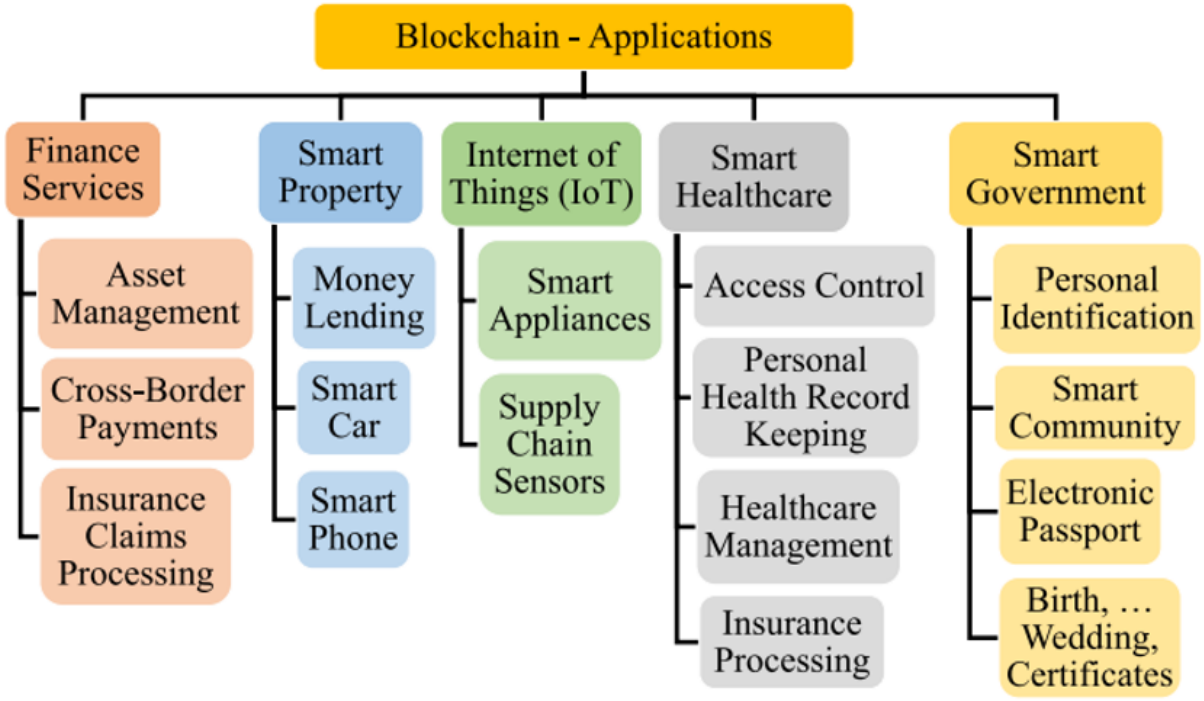


Figure 41: Uses for blockchain technology [195]

3.3.2. The need for both Device and Data Security in IoEIn most applications, communication devices are used more.

Smart cities, smart healthcare, and smart transportation are all built on the IoTs. The IoTs enables the "3Is" of a smart city: Instrumentation, Interconnections, and Intelligence [198]. The Internet of Components is a component of the IoE. An edge layer helps process data before it is sent to the cloud in the IoTs network. With such environments becoming more common in most application domains, the use of communication devices has increased. The network of Things

(IoT) is gaining traction. [199] People, data, processes, and things are the four main components of an IoE environment (see Figure 42).

IoE people are network nodes. Traditional electronic devices, whether handheld or desktop, allowed people to connect with the Internet and the world. With the IoE, people now have access to an infinite number of new communication methods. For example, implantable medical devices like pacemakers send data to a server for diagnosis. Worn on the body, wearable medical devices (WMDs) can monitor heart rate [200]. Implantable medical devices (IWMDs) include these. A traditional IoT network sends data in its entirety. With an Edge layer, not all data is sent.

Data collection can be done in many methods, including crowdsourcing, which involves people. The cloud only stores data that can be analyzed further. Edge layer devices process raw data. In an IoE environment, data processing to information allows for faster decision-making. The data collected is used to make intelligent decisions in our daily lives.

Process helps get data where and when it's needed. This controls network data flow. An IoE network includes people, devices, and the cloud. Data extracted by devices from the environment or people must be processed to extract information. This unprocessed data is sent to the cloud or used to make decisions.

Things collect data. Things have changed a lot. The devices can communicate wirelessly or wiredly and transmit environmental data.

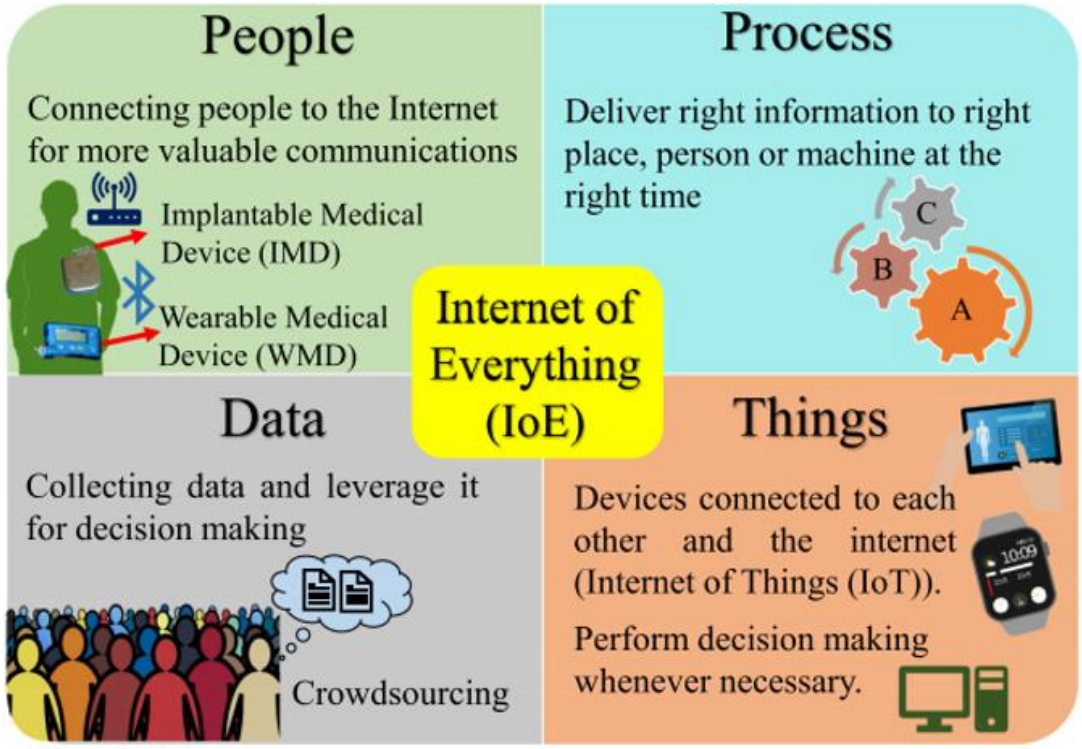


Figure 42: Human-centric IoE vision requires device, person, location, and data privacy [194]

IoT paradigm unites the countless diverse objects and sensors that surround us and allows information exchange amongst all parties (also referred to as nodes). For example, increased device heterogeneity and various data formats may render typical isolated IoT solutions inefficient.

For starters, centralised architecture has significant connectivity and maintenance costs, limiting scalability further. As networks grow, centralised systems become more vulnerable to targeted attacks [202].

Figure 43: Decentralised IoT using blockchain technology may address the challenges based above. This is due to basically three factors. To begin, an autonomous decentralised system allows trustworthy members to participate independently, improving the system's task-processing capability. Second, multiparty collaboration ensures node consistency, preventing system failure. Third, by cloning the blockchain ledger, nodes could synchronise the complete system state, decreasing computational and storage requirements. Smart cities and health care are only two recent examples of blockchain-based IoT architecture's benefits [203].

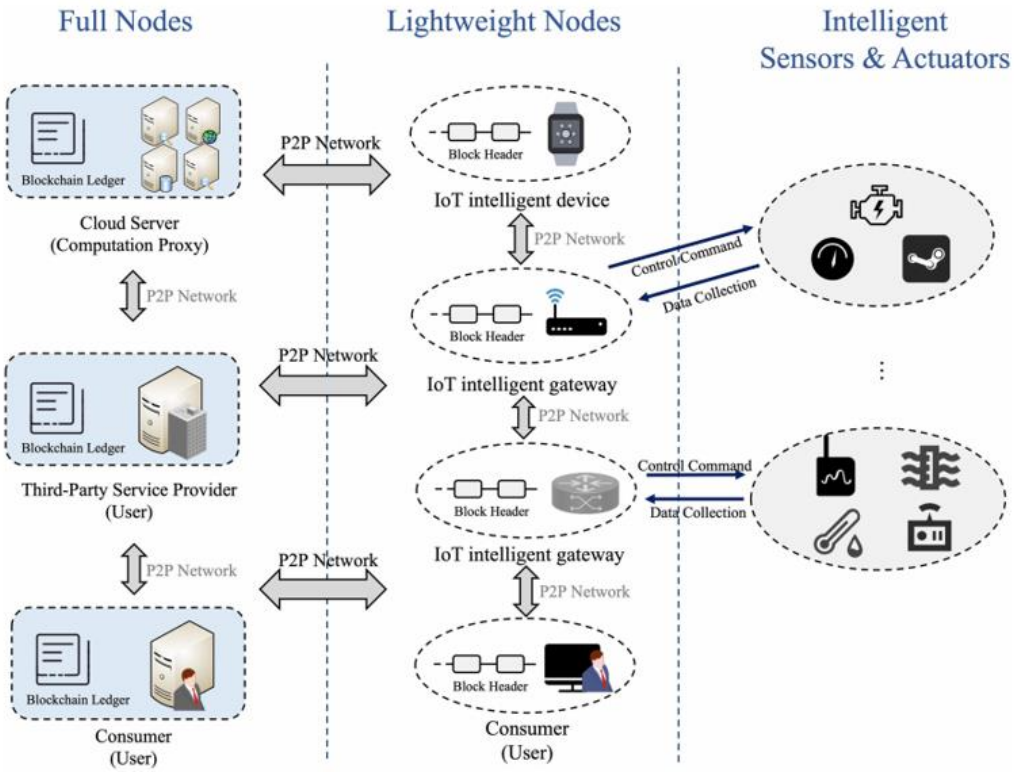


Figure 43: Blockchain-based IoT architecture [201]

Environnementally, the IoTs has generated severe security issues about blockchain technology. Table 13 summarises system design issues and promising solutions for categorised layers.

Table 13: Distributed IoT security threats and solutions [201]

Layers	Attack Types	Challenges in System Design	Promising Solutions
Physical Layer	Physical damage Jamming Firmware replacement attack	Device connection Device recovery Condition monitoring of devices	State detection scheme Data recovery mechanism
Network & Transport Layer	Eavesdropping Information stealing	Cryptography algorithm for messages	Lightweight cryptography algorithm
	Message spoofing	Data privacy and security	Zero-knowledge proof Homomorphic encryption technology access control
	Byzantine attack Denial of service Collusion	Consensus protocol	Hierarchical consensus protocol Dynamic committee
Application Layer	Sybil attack Identity forgery Selfish attack	Identity management Authentication and authorization Admission control Coalition and trust establishment	Lightweight authentication protocol based on lightweight cryptography algorithm Customized smart contracts for admission Reputation assessment model based on blockchain data

These are the following security concerns that arise when designing distributed IoT systems:

(1) communication and network security;

(2) identity management and authentication;

(3) reliable distributed consensus protocol;

(4) decentralized cooperation and trust establishment;

(5) transaction data privacy and security.

3.3.3. Communication & Network Security

A distributed IoT pattern relies heavily on peer-to-peer (P2P) communication, allowing nodes to interact without a central server platform. In an open network, P2P mode promotes inter-personal collaboration. Transparency, though, might be harmful. By eavesdropping, node capture, or message spoofing, the adversary can jeopardise the IoT system's stability.

Cryptographic algorithms are one of the key technologies for tackling the security issue. ECDSA is something that the National Institute of Standards and Technology (NIST) recommends very highly [4].

Despite this, several IoT sensors and smart devices are energy-constrained, with limited computing and storage capacities, respectively. ECDSA's ability to handle computational and memory demands while remaining energy efficient remains uncertain.

The authors found that ECC-based algorithms can balance energy and memory use, which is important given the limited energy resources of IoT devices. Further testing should place on applying cryptographic methods to a realistic and complicated IoT context. Cryptographic techniques are used in the complicated and real-world IoT trial situation because of the devices' unique features.

3.3.4. Identity management & authentication

PKIs manage and authenticate identities in the IoT. Today's most widely used PKIs are CAs and privacy-based trust webs. This may not be enough to meet the identity management methods of the IoE, increasing device count that will necessitate significant computing and storage resources for continuous message exchange and authentication. The original blockchain platform, which only uses "address" to represent a node, due to the significant degree of heterogeneity across IoT devices, it cannot be used to these devices successfully. To overcome these obstacles, combining smart contracts and lightweight encryption algorithms is a promising research direction. Every time an IoT device joins the network, they can implement reliable identity management by creating dedicated smart contracts that cover registration (device type and manufacturer), identity verification, information update (firmware updates, expiration dates, reporting of device loss), and obsolescence (device erasure).[204]

3.3.5. Reliable distributed consensus protocol

Security, scalability, and practicality are all important factors in a P2P network. A majority vote may be vulnerable to 51 percent attacks and selfish mining. To meet the increasing demands for security and efficiency, experts and academics have proposed new consensus protocols based on the latest technology. An overview of consensus protocols is provided in the table 14. They also compared their efficiency, security, and scalability performance. Stake-based and hybrid consensus protocols are the most prevalent among them.

Table 14: Comparison of consensus protocol types [204]

Consensus Protocol	PoW	PoS	PoS+PoW	BFT-based	Raft-based	DPoS	Hybrid
Prominent Platform	Bitcoin Ethereum	Cardano Algorand	PPcoin Blackcoin	Hyperledger Tendermint	R3 Corda Tangaroa	Bitshare EOS	N/A
System Type	Permissionless	Permissionless	Permissionless	Permissioned	Permissioned	Permissioned	Permissionless
Energy Consumption	High	Low	Medium	Low	Low	Low	Low
Block Confirmation	Probabilistic	Probabilistic	Probabilistic	Deterministic	Deterministic	Deterministic	Deterministic
Transaction Rate	Low	Medium	Medium	High	High	High	High
Committee Election	N/A	Dynamic	Dynamic	Static	Static	Static	Dynamic
Fault Tolerance	<=50%	Unknown	Unknown	<=33%	N/A	Unknown	Unknown
Anonymity	High	Medium	Medium	Low	Low	Low	High
Scalability	Medium	Medium	Medium	Low	Low	High	High
Openness	High	Medium	Medium	Low	Low	Medium	High
Fairness	Medium	Medium	Medium	Low	Low	Low	Medium

In comparison PoS uses less energy than PoW because it does not rely on the node's computing power. Simulation tests show that Ouroboros has a distinct personality.

In [205] proposes a hybrid consensus protocol that splits the consensus agreement into two layers: small-scale committee election and Byzantine fault tolerance (BFT). The committee is made up of P2P network nodes and uses the BFT algorithm to pack, verify, and distribute the new block.

Nonetheless, the distributed IoT network's complexity and heterogeneity necessitate real-time state synchronization.

3.3.6. Decentralized cooperation & trust establishment

On the other hand, given that every node at any time, the system, establishing reliable cooperation and trust in an IoT system is critical. With the IoTs applications like vehicular ad hoc networks and crowdsourcing, the reputation assessment model has proven to be effective at fostering cooperation and trust between nodes.

They could create a global reputation assessment model using blockchain technology to improve the above trust models. One possibility is to use irreversible and transparent block data to assess nodes. To begin, using blockchain technology, any node can extract other nodes' behavior data from the block. The nodes' credibility would be calculated automatically after entering their behavior data into the reputation assessment model The node's performance over time is reflected in the type of behavior and timeliness that it exhibits.

3.3.6.1. Transaction data privacy & security

Personal privacy security is critical for system security. All blockchain transactions are public, and the transaction history may reveal transaction frequency, content, and destination, allowing adversaries to deduce participants' true identities. Blockchain and IoE allow adversaries to access IoT IP or physical address data. A denial-of-service attack or node capture will cause device failure and system instability. As a result, sensitive information about nodes should be protected, especially when transaction data is recorded.

Security features such as zero-knowledge proof and homomorphic encryption may help protect blockchain privacy.

Zero knowledge proofs help keep public blockchain transactions private. On one hand, sensitive data like transaction amount and destination address could be protected anonymously. However, users can still validate a transaction with hidden information [201]

3.3.7. Types of BlockChain

Types of Blockchain Technology (see Figure 44) [195]. On the blockchain, each node has a copy of the ledger, or network of ledgers, locally stored. A blockchain network has no central authority. A consensus algorithm compensates for the blockchain's lack of a central entity [195]. For validating transactions, all network participants confirmed on a consensus algorithm. The network's "miners" must put the consensus algorithm through its paces and see if the results hold.to add them to the blockchain.

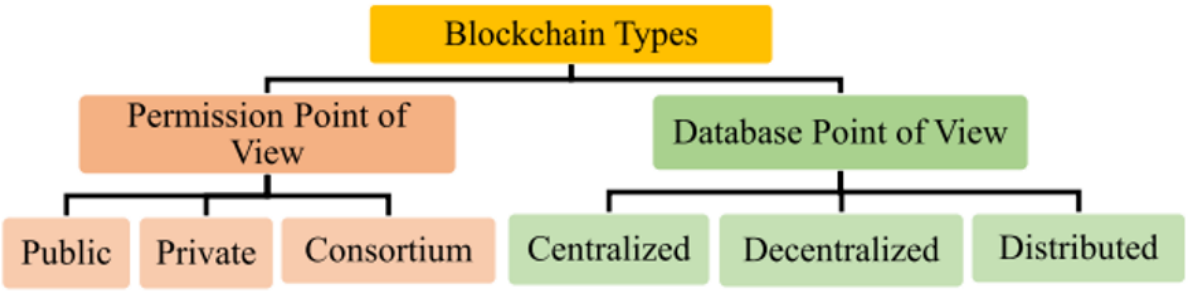


Figure 44: Different types of Blockchain [195]

Consensus techniques create and validate blocks in various ways. They divided into three groups:

1) validation,

2) voting, and

3) authentication.

Bitcoin is a PoW money, Etherium is a PoS currency, and Link is a delegated PoS currency. Consensus algorithm: many transactions join to generate blocks in the blockchain. Once validated, blocks are linked to the blockchain cryptographically. The blockchain's consensus mechanism needs the most computing power. PoAh is a basic compromise method designed for IoT designs. A cryptographic signature verifies PoAh mining.

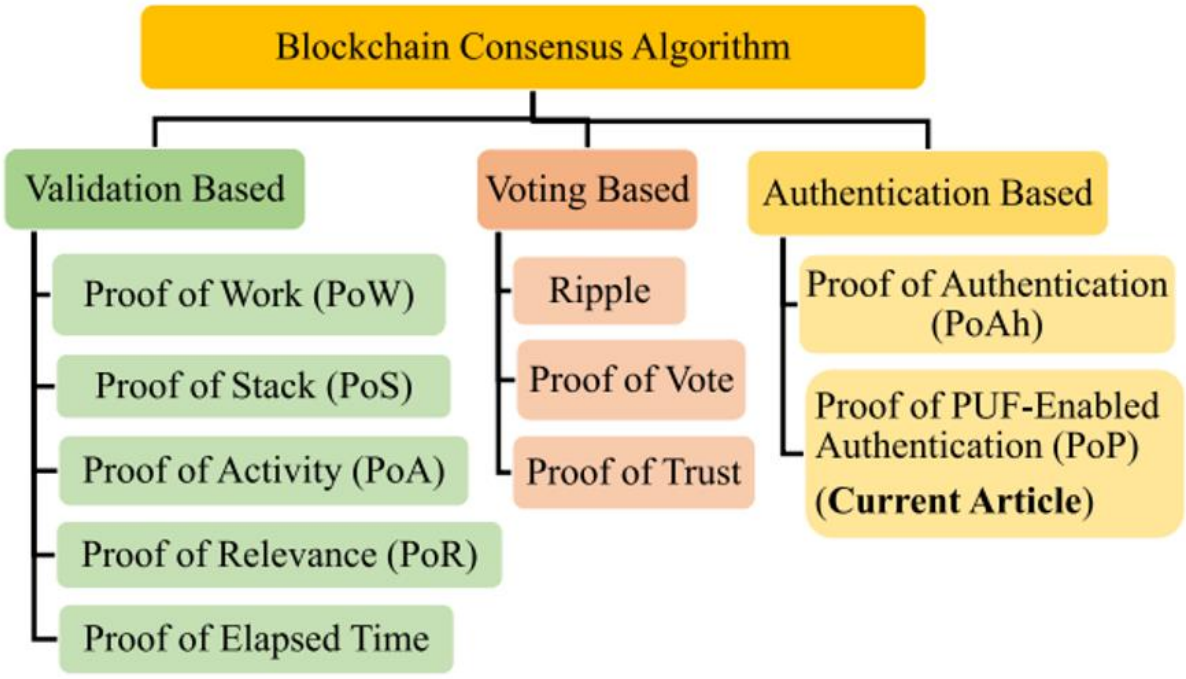


Figure 45: Blockchain consensus algorithms [194]

3.3.8. Challenges

Despite its many uses, the blockchain faces many hurdles (Figure 45) [195] A block added to the blockchain cannot be changed or removed. The ledger/chain is broken when data is updated on blocks added to the blockchain. Blocks are formed by blockchain transactions. After the network creates blocks with transactions, the mining process begins, validating the blocks and transactions. Computational power and specialised hardware required for mining consume many processes. The dedicated hardware requirements further impede scalability [198].

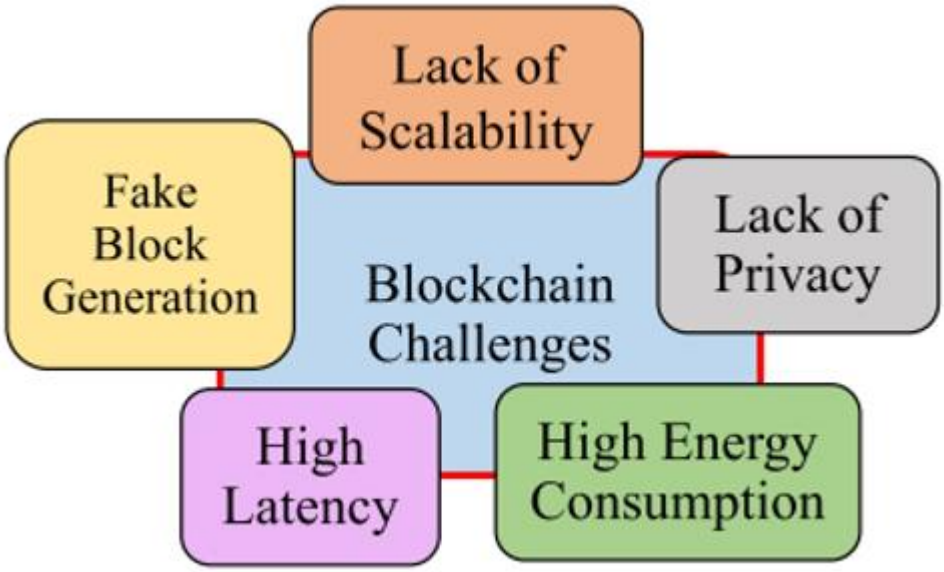


Figure 46: Issues with blockchain technology [194]

The P2P network's data and nodes grow, and so does its latency. As the amount of transactions increases, so does the validation time, causing more problems. Using a distributed ledger also makes it harder to disguise the user's identity. The transactions reveal a user's true identity. Untrue blocks can be generated in blockchain attacks.

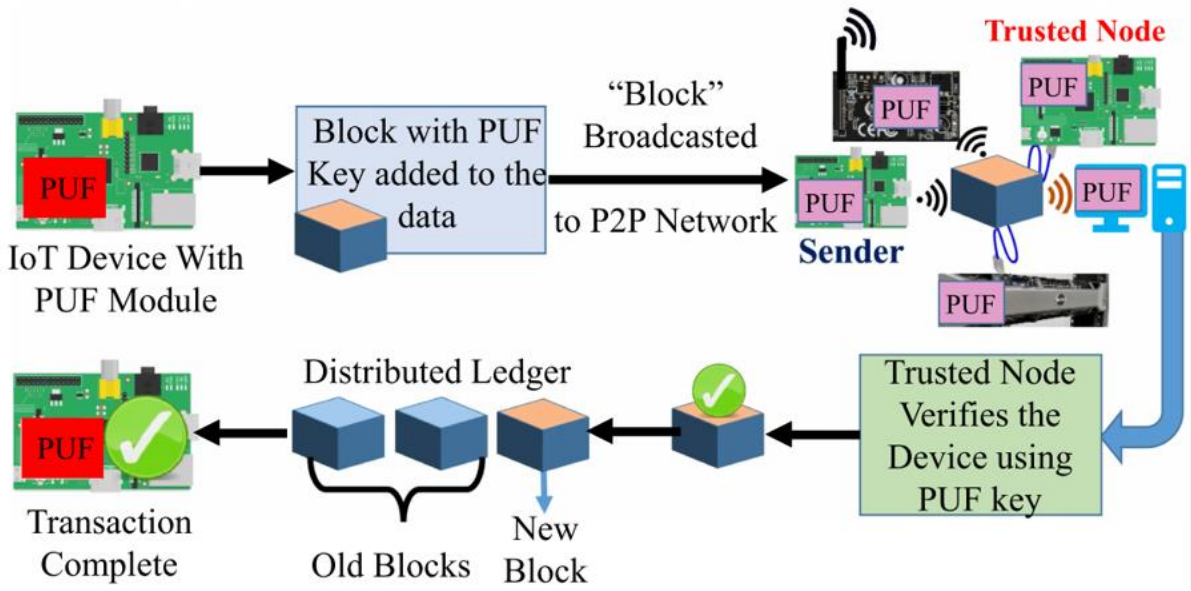


Figure 47: PUFchain working model [194]

Figure 47 shows an energy efficient, scalable, and low-latency PUFchain. For key generation and hashing, a PUFchain node includes an IoT device and hardware module The IoT device collects weather data. The PUF and hashing module are installed to the IoT gadget. Reducing the IoT device's computing load Security and performance of PUFchain are not affected by IoT device specs. Cryptographic processor and PUF module make up PUF and hashing. The IoT device and PUF modulate feed the cryptographic processor. The cryptographic processor does the hashing. The IoT device computes the hash and sends it to the network.

3.3.9. PUF Integration in Blockchain

The PUF chain uses blockchain consensus to make IoT networks that are low-power and small. The PUFchain network is client and trusted nodes. Client-side data collection and network Privileged nodes mine and validate data collectors In the IoT, a PUF gives an IoT device its identity A PUF module can generate unique keys. The PUF module's response changes depending on the PUF key's output. Uncloned or generated PUF module keys are not allowed. It was a "physically unclonable function." Memory of IoT devices lacks PUF keys. A key generation module that can also hash keys. Depending on the PUF architecture, changing the input can produce multiple keys. Change the PUF key output to avoid various security threats.

3.3.10. PUFchain as a solution in Blockchain

3.3.10.1. Blockchain Bottlenecks

Most "things" in an IoE are low-power, low-performance. The blockchain has always required a lot of processing power. Because of this, integrating it into IoE environments is not without its challenges.

3.3.10.2. PUFChain

Solutions include the "PUFChain", a novel blockchain called for an IoT environment with limited resources. PUFs made security and the ability to grow easier [206]. The main processor doesn't have to do as much work with a PUF and Hashing module, so it's good for most situations. The power overhead can be reduced significantly with ultralow power PUF designs.

3.3.10.3. A Consensus Algorithm: POP

The blockchain has a PUF module and a hashing module. The PUF module makes unique keys that are used by the cryptographic hashing function. To strengthen the algorithm, the PUF module generates keys that serve as unique identifiers for each device.

3.3.10.4. The ways that PUFchain works

There are two ways to use PUFChain: the PUF mode and the PUFchain mode. As the name implies, PUF mode employs only the system's PUF module for cryptographic purposes. PUF keys can be used for a variety of purposes, including the allocation of device identifiers and the encryption of data either stored locally or transmitted while in communication. The other configuration is called PUFChain, and it is used to implement the blockchain in the network. This configuration uses the entire module, which includes the PUF module and the hashing module.

3.3.11. Proposed novel POP

The proposed consensus algorithm is designed for low-power IoT networks. In PoP, the PUF module creates the device's unique ID. the PUF key generated by the device's PUF module. No other device can produce the same key. Figure 48 shows the algorithm's phases.

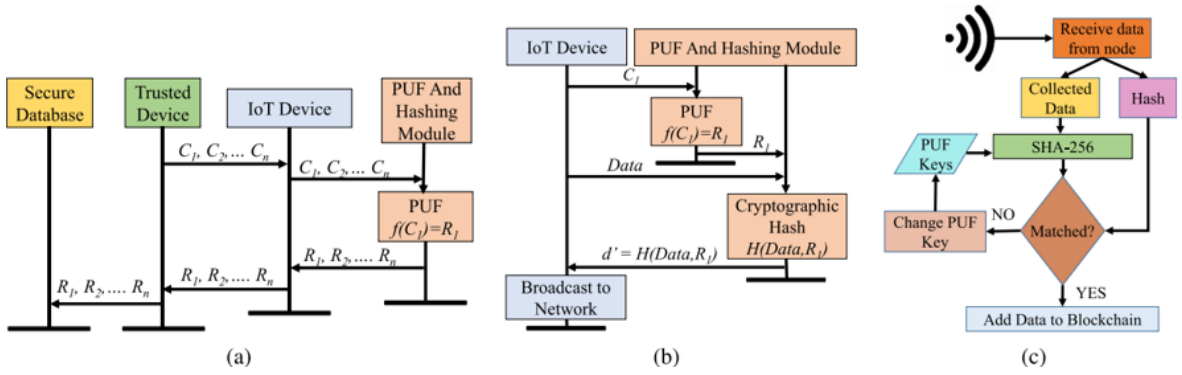


Figure 48: PoP enrollment and authentication. (a) Device enrollment. (b) Steps. (c) Verification [194]

3.3.11.1. Device Enrollment Phase

PUF chain consensus network is closed to unenrolled devices. Firstly, the new device's PUF module generates responses to challenge inputs. The challenges should meet certain criteria to be considered as PUF inputs. Only trusted network nodes can access the CRPs database.

3.3.11.2. Initiating a Transaction

The data collection process begins once the device is connected to the network. Only trusted network nodes can access the CRPs database. The PUF receives a challenge input and generates a response. The response is hashed and added to the data block. The PUFchain network receives this block.

3.3.11.3. Device Authentication Phase

The trusted node extracts the data and hash from the block. There are PUF keys associated with each device that broadcasts a block. The trusted node extracts the data and hash from the block.The trusted node extracts the data and hash from the block. The device is validated if the generated hash matches the block's hash. This is done for all keys associated with that device if the hashes do not match. If no hash matches, the block is discarded.

3.3.11.4. Hardware security PUF

PUF is a vital part of the PUFChain and PoP components. A PUF reveals the nano-electronic manufacturing variations from silicon wafer devices. PUF outputs are a device's fingerprint. [207].

3.3.11.5. PUF Working Principle

During IC fabrication, nano electronic manufacturing variations are introduced. ICs have no duplicate devices. They're PUF modules. Figure 49 shows the idea behind how the PUF works. A PUF module's input is a "challenge," and its output is "response." Cryptographic keys are created by comparing circuit outputs.

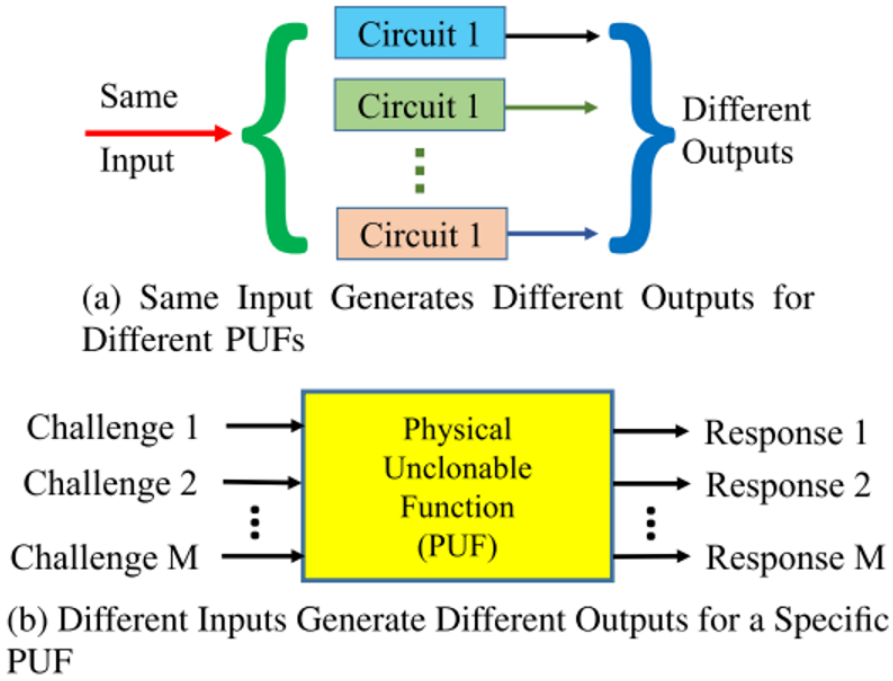


Figure 49: PUF Working Principle [206]

3.3.11.6. Figures-of-Merit (FOMS) of PUF

Uniqueness, reliability, and randomness are required before CRPs can be used in applications [206]. The PUF design's keys are unique because of the property uniqueness. PUF keys are module-specific. PUF must generate the same keyunder varying power supply conditions or ageing effects. The Hamming distance between generated keys determines their uniqueness and reliability. The FoM also emphasizes randomness. The key must have an equal number of 0s and 1s.

3.4. Smart Grid Metering Infrastructure with IoE

3.4.1. Preface

Renewable energy integration and rising energy demand have put the old electrical network under strain. To address these issues, The IoE created a Smart Grid. The Advanced Metering Infrastructure (AMI) allows for bidirectional communication between smart meters and utilities regarding energy use, outages, and tariffs. Thanks to these new AMI features and privileges, cybercriminals can now remotely exploit these smart devices without physical access. Because of the interconnectedness of many smart devices and the data they transmit, Consumers' right to privacy and security is now a top priority. The application of big data, intelligent infrastructure, and market economics has changed society. Energy distribution, control, and monitoring are transformed by IoE [209].

Smart Grid technology is used to optimize energy distribution between customers and suppliers. Data from smart meters, grid sensors, PMUs, and fault detectors are fed into AMI [210]. Smart meters and the utility's back office can communicate swiftly, allowing for on-demand or periodic energy readings and fine-grained data. Having access to detailed information about energy usage allows for more accurate forecasts, identification of problems, load balancing, pricing, and demand response.

However, consumers' fears about their privacy have been heightened by the two-way relationship. Data, pricing, and information on energy consumption are greatly aided by applications on both the grid and the customer side. Motivating users to reduce load and save energy during peak periods is useful for several applications. In the process, however, customer privacy has been compromised, specifically consumer profiling.

3.4.2. Smart Grid overview

Modernised electrical system that uses information technology to assist efficient energy distribution and transmission between customers and providers. Smart grids can transmit data and electricity at twice the speed of conventional power grids. Providers and end-users alike can benefit from the smart grid because of its ability to link intelligent assets to a network design that includes advanced metering infrastructure (AMI) for energy management. The smart grid integrates data and computing across electricity generation, transmission, distribution, and consumption to create a more sustainable, reliable, secure, and cost-effective network.

Smart grid benefits include [211, 212]:

- Better power quality and reliability.
- Increased capacity and efficiency of current power grids.
- Increasing resistance to shock.
- The ability to forecast and self-heal system responses.
- Alternative energy use increases
- Power distribution.
- Maintenance and operation automation.
- Electric vehicles and new power sources reduce greenhouse gas emissions.
- reducing the need for inefficient production during times of high demand.
- Change to electric cars and new technologies for storing energy.
- Growing consumer choice

IEEE P2030, EU-M490 for Smart Grid, ETSI, and ECSS (CEN) have done work toward establishing common ground in smart grid design and conceptual reference models at the highest levels of abstraction.

NIST's smart grid design [108] includes logical domains for bulk generation, transmission, distribution, customers, markets, service providers, and operations. The domains of generation, transmission, Information and power flow between distribution and customers, while data is gathered by markets, service providers, and operations.

3.4.2.1. Customer domain

Since consumers use the generated electricity, they are the smart grid's principal shareholder. Customer is tied to Distribution, Operations, Market, and Service Provider. This website lets users track energy use and manage accounts. Figure 50 shows how the Customer domain's two essential elements serve as an interface to other domains via AMI or the internet. ESI and utility meters (ESI). The customer premise display device offers remote load management, energy monitoring, and non-energy meter reading for managing the electricity account and cyber security.

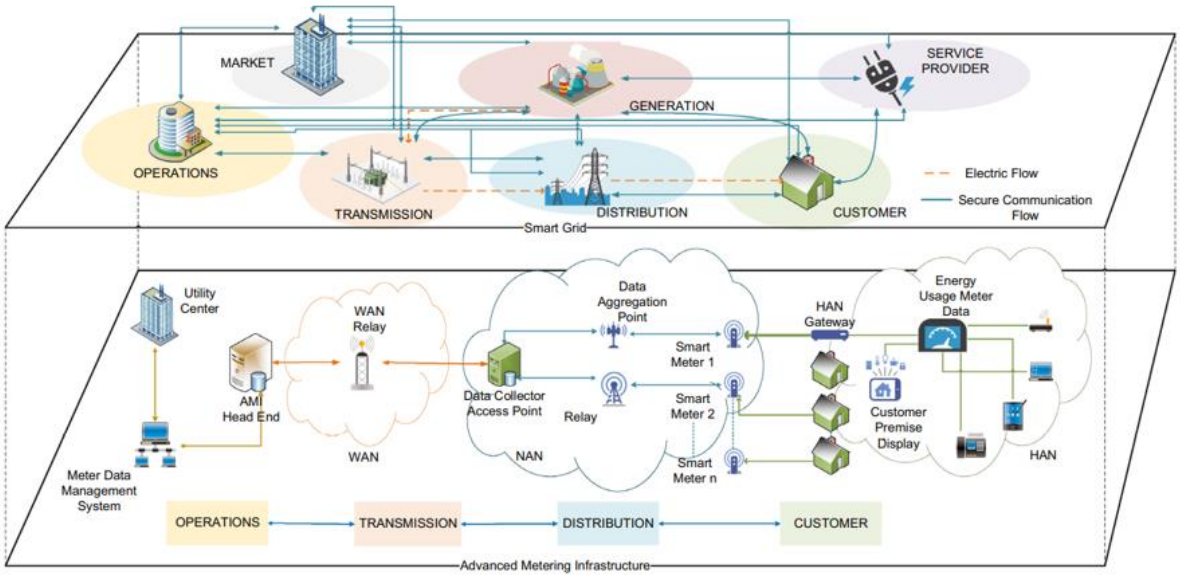


Figure 50: Smart Grid overview [208]

3.4.2.2. Distribution domain

Distribution via smart grid is second. Structure of the distribution domain is impacted by smart grid and infrastructure type. The Distribution domain is electrically coupled to the Customer and Transmission domains (Figure 51). The Distribution domain's trustworthiness is influenced by actors, structure, and communication between domains. The Distribution domain controls the real-time power flow connected using the operation domain as a bridge, to the Market domain. The Market domain also interacts with Distribution, affecting local consumption and electrical generation.

3.4.2.3. Transmission domain

The Transmission domain, as depicted in Figure 51, oversees moving power to the Distribution domain, moving on from the Generation domain. The transmission domain's primary

objective is to balance supply and demand to ensure grid stability. Transmission domains may include DERs like energy storage or generation. The transmission network is also watched and controlled by the SCADA system.

3.4.2.4. Operations domain

The operation domain analyzes and controls the process of sending and distributing energy. The Operations domain is in charge of controlling and monitoring the network, as well as handling faults and research into how well a system works and how reliable it is.

3.4.2.5. Advanced Metering Infrastructure (AMI)

Smart Grid's Automated Meter Reading (AMR) replaced, which collected user data including energy use. People recognized how crucial it was for the utility and its customers to communicate. AMI combines technology to connect customers to the utility's back office. The AMI network connects client equipment, including meters, to the utility center in a smart grid. AMI analyzes energy usage, sets real-time prices, sends outage notifications, updates firmware, and changes system settings. Consumer gadgets and the utility center communicate and receive sensitive information and commands, raising security and privacy concerns. Integrity, privacy, and availability are crucial security goals. AMI network architecture should prioritize integrity and privacy.

3.4.2.6. AMI network infrastructure

The AMI network infrastructure was designed with a WAN utility substation to headend, headend to smart meters, and home appliance to smart meter via home area network (figure 51).

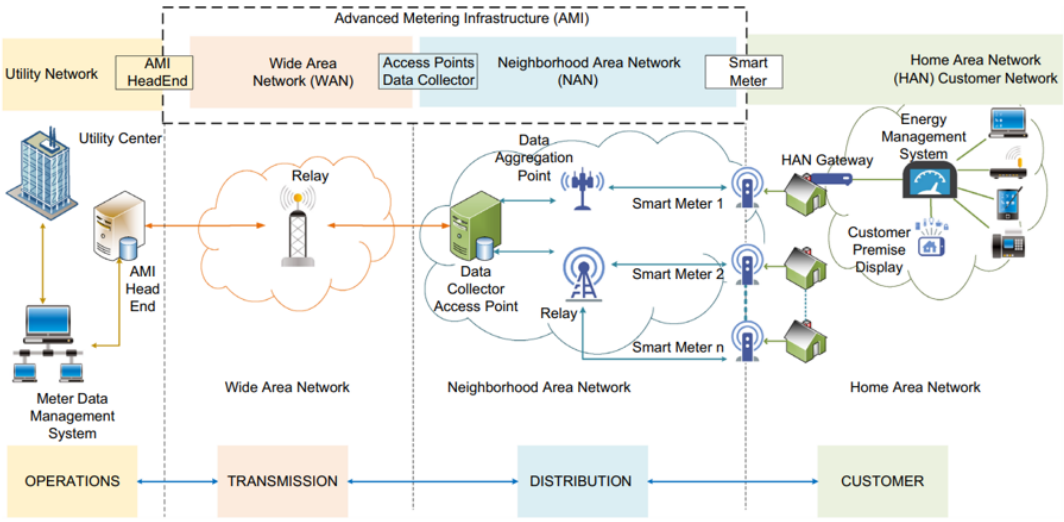


Figure 51: AMI metering infrastructure [208]

3.4.2.7. Home Area Network (HAN)

appliances with HAN connectivity. HAN manages equipment using ZigBee, Ethernet, Wi-Fi, RFID, GPRS, PLC, and Bluetooth. Demand for electricity at its peak, current energy use, and device performance. HAN includes applications that communicate electrical data in homes and buildings.

3.4.2.8. Local Area Network (NAN)

NAN connects WAN and HAN (WAN). It controls and regulates power. NAN gets data from HANs. The data concentrator is NAN's gatekeeper. Smart metering and demand response require long-distance 100 kbps to 10 Mbps communication infrastructure (10 km). NANs use ZigBee, Wi-Fi, PLC, WiMAX, LTE, DSL, and coaxial cable networks.

3.4.2.9. Wide Area Network (WAN)

Utility networks are connected to NAN via WAN. A significant percentage of the security is controlled, monitored, and prioritized by WAN applications. Due to the significant volume of data transferred, The data rate needs to be between 10 Mbps and 1 GB, and the service area needs to be up to 100 km. Optical, cellular, WiMAX, and satellite communications are often used.

3.4.2.10. AMI metering infrastructure

AMI uses a variety of metering devices to ensure data flow between the home appliances of the customer and the utility center. There are different kinds of AMI meters.

3.4.2.11. Smart Meters (SM)

AMI needs smart meters. Smart meters are solid-state devices that can be programmed and can send and receive data from utilities. Unlike physical meters, smart meters can be physically tampered with. The smart meter connects the utility network and home automation devices.

3.4.2.12. Additional metering systems

[213] is a different non-metering infrastructure.

- HAN gateway connects Smart Grid's Customer and Distribution domains.
- Installation at the end-location. user's A user-friendly display that shows consumers their energy consumption and associated expenditures in real time.
- Energy management system. System energy with utility billing and real-time pricing systems. Data gatherer A data collector collects data from smart metres and sends it to utility's administrative hub.

- Utility, billing, and metering systems. Metering and charging in the operational domain are handled by the utility hub.
- Headend for AMI. The MDMS and the AMI network can talk to each other through the AMI Headend.Metering data management (MDMS) The MDMS gathers and organizes data from smart grid meters. The MDMS collects, checks, estimates, and makes changes to meter data, including energy use, generation, and metre logs. In the interim, an MDMS saves the data and makes it available to permitted systems. [213].

3.4.3. AMI security requirements

The power grid system has long required that consumers have access to electricity. Integrity and confidentiality have become critical requirements as information technology and customer participation in energy efficiency have grown. In this way, communication was vital to the functionality, infrastructure, and architecture of the power grid system. NIST's Smart Grid Interoperability PanelDetailed guidelines for cyber security in the smart grid have recently been published, defining the "CIA trinity" of interconnected security priorities [214].

In the CIA trinity, confidentiality is paramount. To protect individual privacy and proprietary information, confidentiality is the practice of preventing unauthorised access to and disclosure of information.

For data non-repetition and validity, integrity prevents unauthorised data manipulation and deletion. Data integrity is vital to smart grid data security. Examples of modifying AMI data include changing pricing information and commands. A smart grid's information availability is critical. Availability ensures an appropriate actor has timely access to and use of information. Availability concerns can vary depending on the data sent across smart grid systems.

3.4.4. Privacy in AMI

The term "privacy" refers to the privacy of one's possessions, actions, and decisions. This section addresses data privacy. Information privacy refers to an organization's ability to manage personal data collection, dissemination, and use [215]. Personal data control is a critical ethical and human rights issue in the digital age [216]. In Figure 52, we see four main types of privacy [59]:

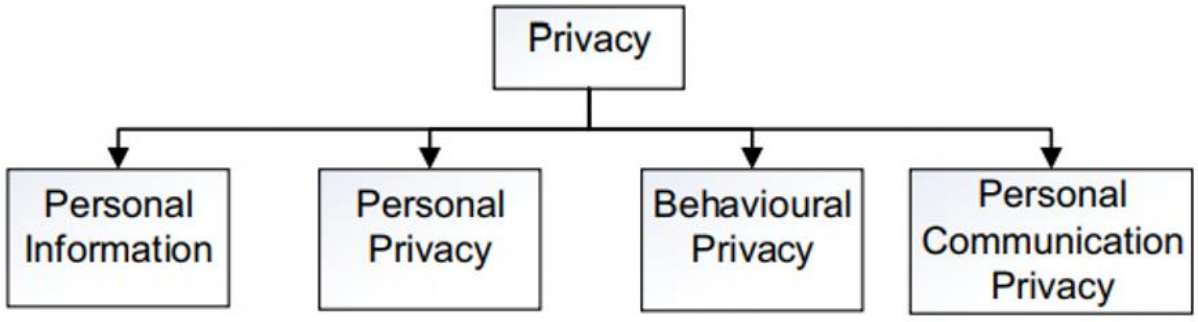


Figure 52: Privacy in AMI [208]

- Personal data. This is a formal definition of personal data. A person's identity can be identified directly or indirectly using a unique identifier or a combination of biological, psychological, socioeconomic, racial, ethnic, or national characteristics.
- Personal privacy. The right to control one's own body. It addresses physical requirements, health issues, and the usage of medical devices.
- Behavioral privacy. Individuals have the right to make their own decisions and to keep certain personal habits private.
- Discretion in personal communications. freedom from excessive monitoring, censoring, or surveillance of communications.

3.4.5. AMI privacy related works

Various academics have developed novel privacy-preserving AMI techniques in response to privacy concerns associated with fine-grained sensitive energy data. As shown in Figure 53, these privacy-preserving techniques fall into two categories: non-cryptography and cryptography [217].

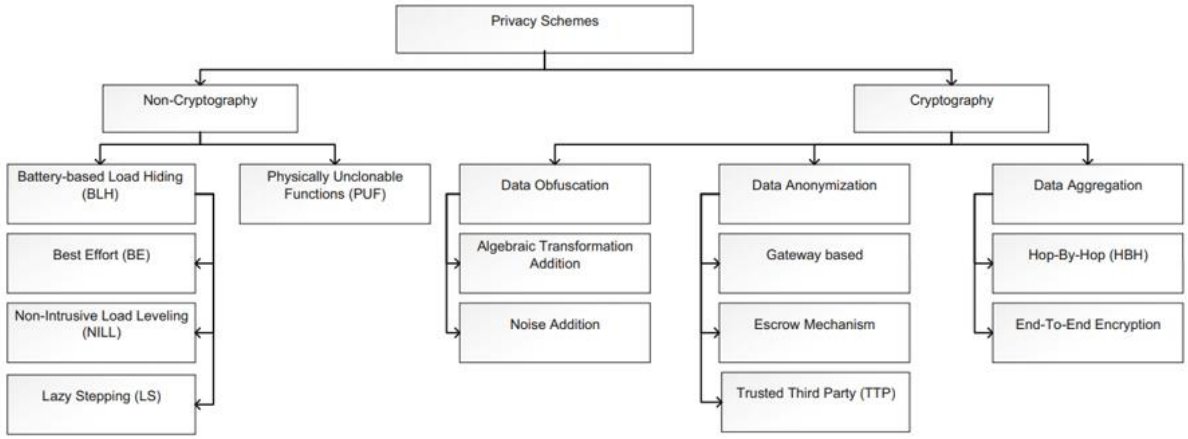


Figure 53: Related work hierarchy [208]

3.4.5.1. a non-cryptographic strategy

These techniques conceal a consumer's actual energy consumption using non-cryptographic methods. BLH and PUC are non-cryptographic techniques (PUF).

Battery-based load hiding (BLH).

Load Hiding via Batteries is the first non-cryptographic approach (BLH). A well-known method called the BLH makes use of a rechargeable battery to partially satisfy energy demand while concealing actual use. The Non-Intrusive Load Leveling (NILL), the Lazy Stepping (LS), and the Best Effort (BE) schemes are recent examples of BLH methods or algorithms.

Physically unclonable functions (PUF)

A second non-cryptographic method is called PUF. PUF devices are low-cost, impersonation-proof hardware solutions for authentication and integrity. The PUF use the building's one-way feature to protect user anonymity. A PUF's signature is completely up to the user and is determined by its complex physical features. Unique PUFs can never be replicated because of their randomness.

3.4.5.2. Cryptographic approach

Cryptography is the second way of privacy protection. The information lost by distributed computation is just that which can be inferred from the output of the algorithm thanks to the cryptographic method [219]. Smart grids use three types of cryptography: Data Masking, Anonymization, and Aggregation Based on the literature, they also assessed the benefits and drawbacks of current cryptographic work.

Data obfuscation

Data on energy use at the micro level can be masked by adding noise or undergoing an algebraic transformation.

Data anonymization

Anonymization of data is the second way to protect privacy in cryptology. Anonymization de-identifies customers' energy consumption data [220]. Enough data will be received to calculate required data but not enough to identify a meter or user. A trusted infrastructure can also help achieve these goals.

Data aggregation

Data aggregation is the third method for preserving privacy. Data aggregation is the process of concatenating and summarizing data packets from multiple devices using network aggregators. While data aggregation reduces data transmission, it compromises privacy by requiring plaintext data access [208].

There is a trade-off that occurs between making metering data available for operational objectives like Privacy for end users is protected by using decentralized methods of state estimation, demand response, and billing. The privacy of sensitive data such as energy usage data may impair routine billing processes for a utility. The utility of the data is diminished because of the need to preserve individuals' privacy, which in turn might cause inaccurate aggregate results. Therefore, it is crucial to evaluate these trade-offs in terms of information flow and context retention. According to the literature review, unique privacy-preserving strategies are based to address prior work's shortcomings while also meeting the AMI's objectives.

3.5. Hardware devices and architecture for securing the IoE

How long can the exponential growth of communications networks, especially wireless networks, be sustained? It's useful to look back on the history of wireless to envision possible futures. While history is not destiny, it is frequently true that established and recurring patterns from the past can be used to predict the future.

3.5.1. Circuits, Devices and architectures for securing the IoE

Largest network "attack surface" for terascale networks. They may face serious consequences if the IoE is secured using current Internet security approaches. Becoming an IoE

product has resulted in many things lacking even basic safeguards. We need answers now. A new beginning, an opportunity to include security without the limits of current practise, the IoE represents a new beginning. Malware payloads are typically measured in kilobytes and have remained largely consistent in size over time. Anti-malware tools, on the other hand, are hundreds of megabytes in size. That the response is diverging from the threat indicates a flawed strategy. It is a problem that software-only solutions are used. In the IoE age, a few simple hardware-based solutions can greatly improve security. Use a processor architecture that prevents the usual "overflow exploit" from being successful. Adding an ARG to cryptographic engines is another. Inherently spoofable all-digital RNGs Alternatively, an analogue RNG can be developed such that the source of the random numbers can be easily identified. Unlike an all-digital approach, an ARNG is sensitive to supply voltage and temperature. The ARNG can also be physically verified at low cost if it is on a different chip. Wireless interrogation and/or powering of the ARNG can help with testing and validation. They were designing low-power, basic ARNGs for IoE client devices as part of a comprehensive security approach that addresses the issue from hardware, firmware, and software perspectives. While no approach can ensure 100% security, a multidisciplinary strategy can greatly reduce risks at low expense.

3.5.2. Field-programmable things array (FPTA)

The terascale's pressures show more issues: With only a decade or two to go, there may not be enough engineers on the planet to construct even a tiny fraction of a trillion devices! Determining the NRE associated with designing and manufacturing a large number of unique mask sets would quickly render the enterprise unprofitable.

Improved CAD tools that function as a "workforce multiplier" could help narrow the gap and enhance economics. This strategy should reduce the number of engineer-dollars spent on each design by an order of magnitude. Sensing, calculating, communicating, and actuating are essential IoE device capabilities. By using this high-level commonality, designers may construct a design family faster and cheaper.

Develop a fabric that explicitly acknowledges in its hardware architecture that many IoE devices will share that high-level likeness. That rationale led to the FPTA.

An FPTA number family would be analogous to today's FPGA products, with different capabilities and pricing. Mask and design costs might therefore be amortised over a large number of units, allowing for low-cost production in a short time. For example, field programmability enables for dynamic bug fixes and security updates after manufacture. The FPTA would incorporate security into its “best practises,” relieving designers of the need to become security

experts. To improve the "impedance match" between problem and solution, the FPTA would move much of the design effort from the scarce hardware engineer to the plentiful software engineer (figure 54).

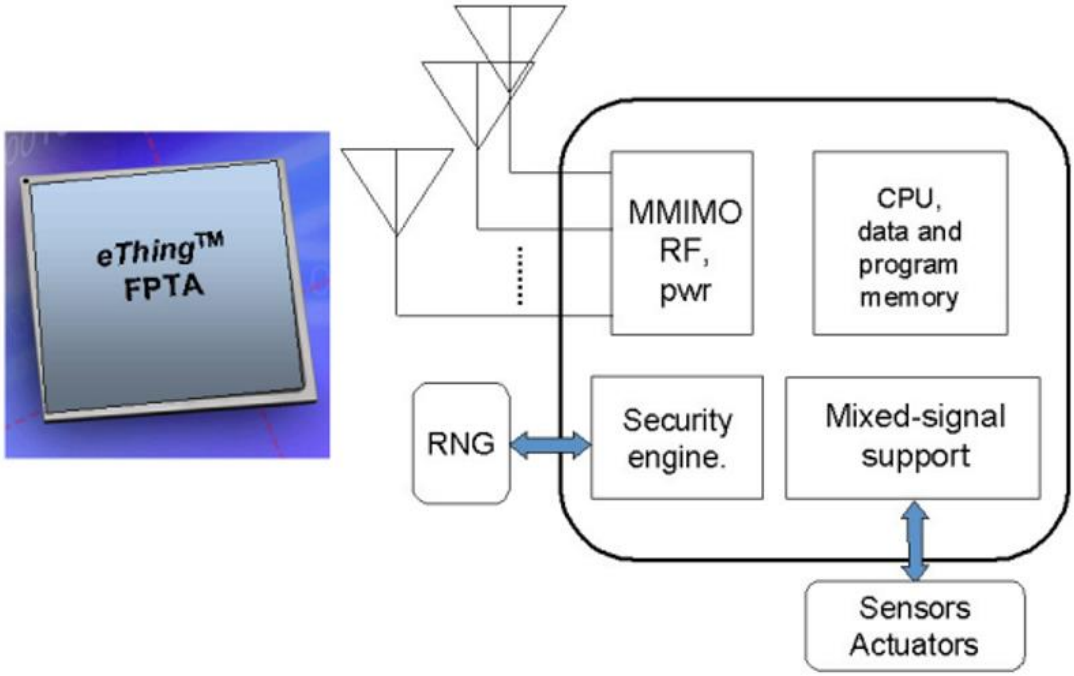


Figure 54: Field-programmable things array (FPTA) block diagram

Chapter 4

Cyber Physical Systems (CPS)

4. Cyber-Physical Systems (CPS)

4.1. Preface

We live in a networked world where software, system hardware, and sensors all communicate. This is what CPS are [111]. CPS typically has two components: a physical process and a cyber system. Most of the time, the cyber system watches or controls the physical process. A cyber system is a networked collection of small devices that can sense, compute, and communicate (usually wirelessly). But as physical and cyber systems become more intertwined, cyber security flaws make physical systems more vulnerable. Sensor networks, which are already well-established and share the networked functioning and limited capability characteristics of CPS [114], seem to be the focus of most mapping efforts, as they are with any new subject.

The widespread impact of CPS on society, economy, and environment has recently sparked academic, industrial, and government interest in CPS research. Security breaches could have disastrous repercussions because there are no effective countermeasures. For instance, If a power grid's communication lines are down, it may collapse and experience widespread blackouts. Along with security, CPS privacy is a major concern. Cyber-physical systems collect massive amounts of data for analysis and decision making. Along with security, CPS privacy is a major concern. Cyber-physical systems collect massive amounts of data for analysis and decision making. Data collection enables machine learning algorithms to make decisions. If a data collection flaw occurs, massive amounts of private or sensitive national security data could leak. Breach events can occur during data collection, transmission, operation, and storage. As a result, most current CPS data do not protect collected data [115].

These systems' heterogeneity highlights the importance of security. In addition to software, Developers and requirement analysts must take hardware into account, such as sensor and network security, in their evaluations. Many examples of safe software engineering procedures exist, but they focus primarily on software. To enable security requirements procedures, we require a framework for the security requirements of CPSs. When defining security needs, many existing frameworks pay little to no attention to sensor, hardware, network, and third-party components.

While cyber-physical systems require security requirements engineering, there is no standard process for developing secure software. Although there are many methods and frameworks for creating software, they may constantly be improved [116]. Software security engineering provides tools, methodologies, procedures, and best practices for safe system design [117]. Software security is not well understood and should be addressed early in the Software Development Life Cycle (SDLC) [118]. Embedding security into system design is now standard [119].

Thus, incorporating security needs from the start ensures secure software while saving the software development team time and effort [111].

4.2. Cyber Physical Security overview

4.2.1. General

These days, new cyber disasters seem to happen every other day, making cyber security a top priority in the age of information. Indeed, cyber events may have directly impacted many [120]. Notably, the recent cyber-attack on Target impacted up to one-third of the US population. In this case, hackers used vendor credentials to attack the system. The attack on Target and its customers is one of many cyberattack strategies [114].

4.2.2. CPS workflow

A typical CPS workflow looks like this:

1. **Monitoring:** CPS must constantly monitor physical processes and the environment. It is also used to provide feedback on prior CPS actions and to monitor future operations. The CPS's major physical goal is designed by the physical method.

2. **Networking**: Data aggregation and dissemination. CPS may have numerous sensors. These sensors can generate real-time data, which must be aggregated or distributed by analyzers. Simultaneously, multiple programmes must network with each other.

3. **Computing:** To decide if the physical process fits predefined criteria, the computing stage uses logic and data collected during monitoring. If the conditions are not met, corrective action is advised. For example, a datacenter CPS may have a model that predicts temperature rises in response to various scheduling approaches.

4. **Action:** This stage executes the calculations performed during the computing phase. Actuation can be utilised to correct the CPS's cyber behavior and change the physical process. For example, a medical CPS could administer medication.

Figure 55 shows the CPS workflow. In this example, y represents sensor data, z represents data aggregation within the network, u shows the controller's correct calculation of the physical system states., and v represents the actuator's control commands [113].

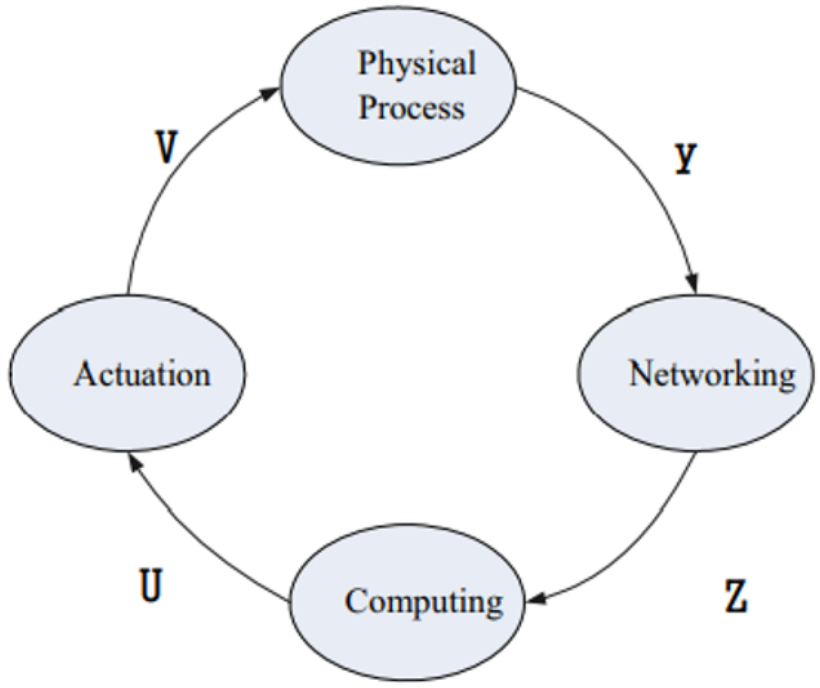


Figure 55: Abstraction of CPS [113]

4.2.2.1. Threats, Security Goals, and Risk Assessment for CPS

All potential threats to a system, not just those related to a specific need, can be uncovered through a thorough risk analysis. Insufficient risk assessment when implementing CPS security requirements may result in unexpected and undesirable system behavior, as developers may overlook important requirements. Determining the risk of security requirements within the CPS framework is their goal. The software and other CPS components determine the hazards. Additionally, they investigated the main security goals and dangers for a CPS. The foundation of this study is a set of matrices obtained during the development of an automotive smart parking system. The CPS and any related software or hardware determines the risks. The main security objectives and threats to a CPS were both examined. This study is based on information obtained

while developing an automotive smart parking system. In Figure 56, we see a visual representation of the CPS's security goals and potential dangers.

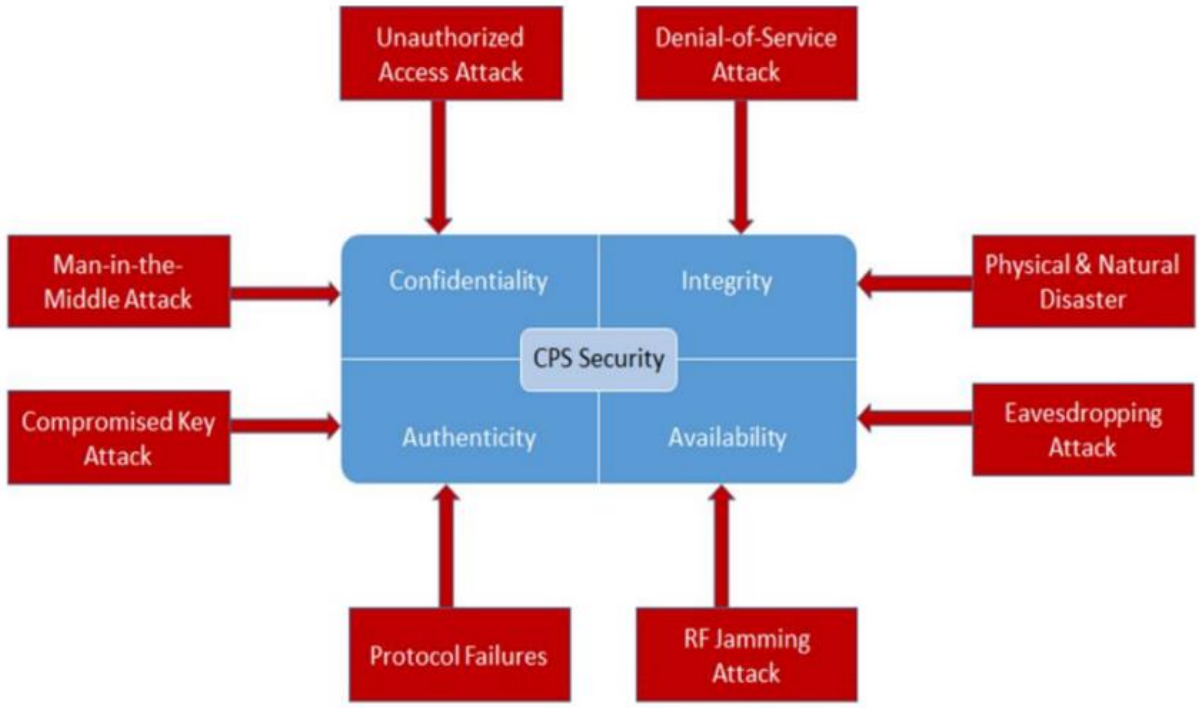


Figure 56: Security goals & threats [111]

4.2.2.2. CPS Security Goals

Trust in CPSs is a prerequisite for social acceptance. Users' trust can only be earned by presenting acceptable security objectives. Risk mitigation is one of the system's security objectives. They want us to understand CPS security objectives better. Determining the reliability of sensor-driven systems involving multiple sensor nodes requires security concerns. This demonstrates the importance of authentication, availability, integrity, and secrecy in CPS security. Here are critical security goals:

Integration

Changing data or resources without permission. When an adversary accidentally or maliciously changes or deletes vital data and receivers get false data, integrity is violated. CPS integrity is the ability to prevent, detect, or resist deception attacks on sensor, actuator, or controller data.

Authentication

Adding nodes (sensors) to the network requires authorization [121]. Authentication in a CPS is difficult because it may require heterogeneous network authentication.

Availability

Availability ensures that the authorized user always has access to the data. Service interruptions due to hardware, software, or power failures must be adjusted when a DoS attack occurs [123].

Confidentiality

Unauthorized users cannot access networked data. Unauthorized users can access network information, compromising confidentiality [122].

4.2.2.3. CPS Challenges

Anything that poses a risk to cyber-physical systems is a danger. These are the main CPS dangers [124]. Figure 57 shows different assaults on CPS.:

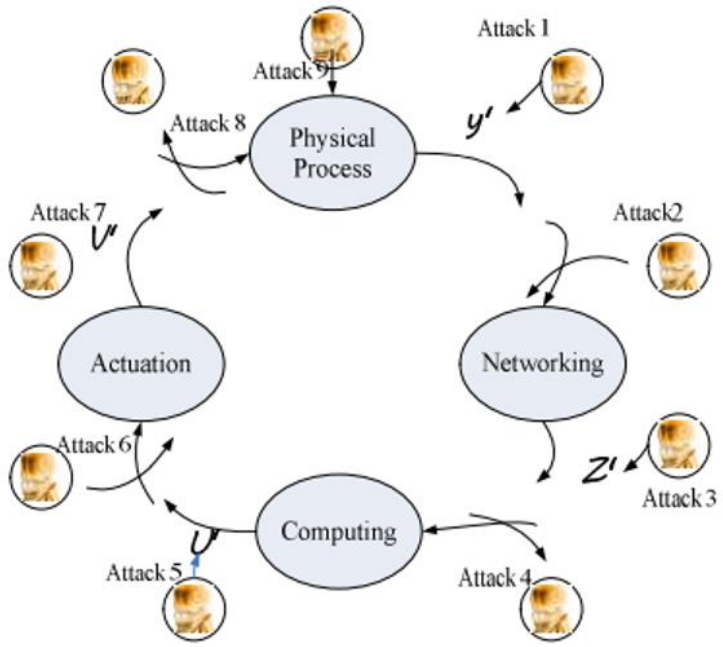


Figure 57: Attacks [113]

Eavesdropping

Eavesdropping is an attack where an attacker intercepts all data exchanged by a system [125]. The term "passive attack" refers to an attacker who observes rather than interferes with

the system's operation. CPS is especially susceptible to traffic analysis eavesdropping, such as using sensor network monitoring data to listen in on conversations. Eavesdropping violates the privacy of users and patients' health information sent over the network. In Figure 57, attack 4 intercepts data aggregation processes, while attack 8 intercepts controller requests. Industrial espionage can also cause significant damage to a company.

Compromised-Key Attack

Unlocking secure data requires a key, a secret code. A key is compromised once an attacker has it [126]. A message can be decrypted without the sender or recipient knowing. The attacker can decrypt or change data, or make more keys to get to other encrypted resources or communications. A key can be obtained by an attacker, but it takes time and resources to do so. Like attack 9 in figure 57, an attacker may take control of the sensors and use reverse engineering to discover the keys they store, or They may pose as a sensor node to agree on keys with other sensors.

A compromised-key attack allows the attacker to alter data by obtaining the system's key. The attacker has access to further system components. Without the users' knowledge [55].

Man-in-the-Middle Attack

MIM attacks send incorrect data. Not recognizing it as false impairs the system's function. [127]. An attack on a system that controls railway switches could cause train faults or crashes.

In attack 7, for instance, the attacker sends V' to signal a system change, but V' isn't the real actuation instruction. Unintended consequences may happen when an operator attempts to fix a problem while adhering to regular operating procedures. The change and replay of control data can alter the system's performance. This attack type is also described by attacks 1, 3, and 5.

Denial-of-Service Attack

DoS attack [128] prevents a system from processing or responding to legitimate network traffic or resource requests. This type of attack overloads the network with data, preventing normal service delivery. A denial-of-service attack prevents normal system operation. After gaining access to the cyber-physical network, the attacker can do one of the following:

- Overload a controller or sensor network with traffic until the controller or sensor network shuts down.
- Fail to send valid data to the controller or system networks.

– Send wrong information to the controller or system networks, which can end service or make it stop working.

To disrupt normal network traffic, attackers may flood the whole sensor network with jamming data, as shown in Figure 57.

Physical Attack & Natural Disaster

A physical device, such as hardware, sensors, cameras, or terminals, may be harmed during this assault. It is necessary to prevent attacks like this one that could harm human life. External attacks on these systems could cause significant damage. Natural disasters can kill people, disable sensors or actuators, and greatly increase costs.[13].

Unauthorized Access

A genuine concern, unauthorized data access should be addressed early in the SDLC. An attacker can easily access user data in many ways. This data may be sent via sensors or network connections [129].

Radio Frequency Jamming

Radio frequency (RF) jamming disables physical interaction. This may cause issues with sensor-to-PLC or gateway communication. Radio frequency jamming occurs when electromagnetic waves or high-level signal traffic are used to detach the tag [130].

Protocol Failures

In the event of protocol failure, network communications and hardware may fail. These threats and their implications demonstrate the importance of cyber-physical system security in their development. Attackers can access data and abuse systems in any way they want. Unimaginable dangers would occur. In this way, cyber-physical systems lose practically all their usefulness [131].

4.2.3. Characteristics of Adversaries

This section discusses the various enemies.:

i. Hackers that are skilled can identify specific software flaws and develop exploit codes,

ii. Unrestricted access to a target system is frequently obtained by displeased insiders with harmful intentions, allowing them to cause system damage or steal system data without extensive knowledge of cyber intrusions [132].

iii. They may design the system to bring down such important cyber-physical infrastructures as aeronautical systems or power grid systems.

Attackers can respond with policies or techniques tailored to the attack. Researchers can also better understand enemy traits and anticipate adversaries to develop threat models.

The physical environment can be irreparably damaged by manually attacking or aggressively attacking each layer of the CPS. Also, the CPS's physical sensors and networks are vulnerable to internet-based attacks. [134]. At the perceptual layer, attacks on sensors, actuators, and the IoTs are possible. At the transport and application layers, malware and counterfeit attacks threaten user privacy. CPS security threats are shown in Figure 58. Physical, cyber, and system security are all part of CPS security. Intelligent buildings, cities, industries, healthcare and smart grids are just a few examples of the importance of smart environments.

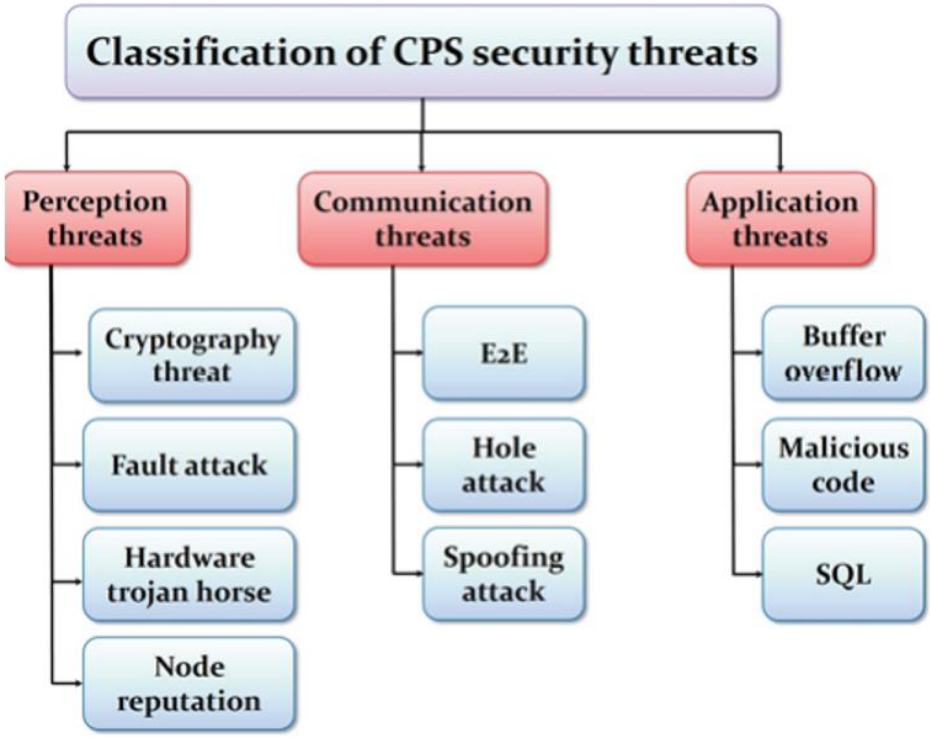


Figure 58: Defining CPS security threats [133]

4.2.3.1. Perception Threats

The perceptual layer includes the physical world and memory functions like sensors and RFID. These devices are generally installed outside, resulting in physical attacks during component or device replacement [135]. Moreover, despite the importance of the hardware layer, the majority of CPS security research ignores it in favor of the cyberspace.There are several types

of physical threats: attacks on equipment or lines; interference; fault attacks; denial of service attacks; and threats to node reputation.

Cryptography threat: Unsafe software or hardware can be used to steal, transmit, process, and store information meant to keep a computer or other networked device safe. There is a heightened risk of hardware attacks on endpoints in unattended settings [136]. Lightweight cryptography is used in embedded devices because of their limited CPS terminal resources. The Add Round Key and SubBytes outputs of AES are vulnerable to a data-parsing attack [137].

Fault attack: Unintentionally generated fault actions on the target device are used in a fault attack to recover the data password and figure out how the inside circuitry works. Every system that experiences a problem has a circuit control assault on the critical route. Furthermore, a fault attack may employ local or global techniques [138].

Hardware Trojan horse: The Trojan horse's physical, activation, and behavioral properties are classified. It shows how many components have been added, destroyed, or corrupted as well as changes in the chip's physical structure. Troy's activation and deactivation are governed by the activation property. The behavior of Trojan horses can help identify the disruptive behavior they introduce.

Node reputation: Node reputation can be harmed by node capture, forgery, and outages. Encryption keys are obtained or relinquished by nodes, jeopardizing system security [139]. The fake node sends dangerous data by joining the network. Additionally, it compromises data security and launches DoS assaults leveraging the system's node power. By suspending the node's service, a node abort attack jeopardizes availability and integrity. The secrecy, availability, integrity, and stability of the target node are the targets of numerous node reputation attacks.

4.2.3.2. CPS Security Solutions

CPS security protects data, network, physical, and application settings. Figure 59 organizes CPS security solutions into four groups: device protection, network access detection, malicious code detection, and application protection. We are going to focus to device protection. The others are not in the area we want to discuss

Figure 59: Solutions for CPS's security and privacy [133]

Device protection: In order to provide a safe hardware infrastructure, one must first build a secure hardware platform with a resilient architecture. In [140] proposed a high-security SoC architecture based on hardware anchors, or OS-hardware connections. To prevent malicious tampering, hardware anchors constantly track bus activity. There was little performance overhead in their proposed architecture. In [141] proposed a hardware/software architecture for safe OS system. They made Ian, an emulator-based model. This experiment identified and removed all harmful rootkits, and found no false positives for positive modules. In [142] developed a PUF-based CPS security paradigm. The framework investigated new methods for combining PUF security properties. It is a digital fingerprint used to identify semiconductor devices like microprocessors. It differs from other semiconductors due to spontaneous physical changes that occur during semiconductor production. The PUF is frequently used for encryption in high-security integrated circuit applications. In [143], An integrated CPS solution was built on a multi-agent WSO2 complex event processor system. The solution analyzes each communication to determine whether it is encrypted or not. It also exemplifies the ideal CPS, which ensures confidentiality, privacy, and availability. A strategy for preventing adversaries from thwarting assaults employing sophisticated cryptographic primitives like AES, RSA, ECC, and HMAC was put out in [137]. When defending against strikes outside of the expected range, this sort of attack defense is thought to be more successful than indiscriminate attack [133].

4.3. Context-Wise security framework

4.3.1. Preface

Context-aware security architecture for cyber-physical systems. As shown in figure 60, they used context-sensitive security information in authentication, encryption, key agreement protocol, and access control. With context coupling, cyber-physical system security mechanisms may adapt dynamically to their physical environment. This type of security mechanism is known as a "context-aware security framework."

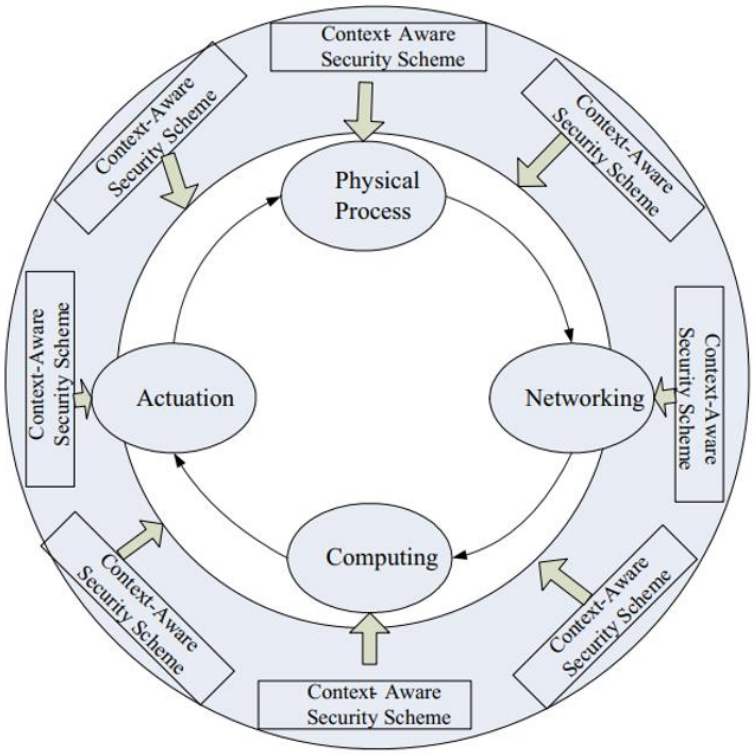


Figure 60: The context-aware security framework [113]

Conditions and constraints that define how an application should behave or set up an event [144]. The setting may be one of comfort or pleasure. Location, mood, past medical history, lighting conditions, weather, and temporal context (e.g. (e.g. time). With the aid of our framework, they concentrated on security-relevant context, which is a collection of contextual characteristics describing an entity's situation and value to select suitable controls (measures) or their configuration to protect data and systems from unauthorized access, use, disclosure, disruption, modification, or destruction to maintain confidentiality, integrity, availability, and availability.

Contextual variables include attack model and adversary types. Threats to confidentiality, integrity, and availability are affected by context-specific factors. Value-based controls and configurations can reduce dangers. Figure 61 depicts context-aware security workflow.

$$\{\text{Security Processes}\}_{\text{context}} \rightarrow \text{Secure Measures}$$

Consider the following scenario: While within the hospital, a surgeon has permission to examine his patients' medical records. However, while outside, the access control system notices the context has changed and rejects the request.

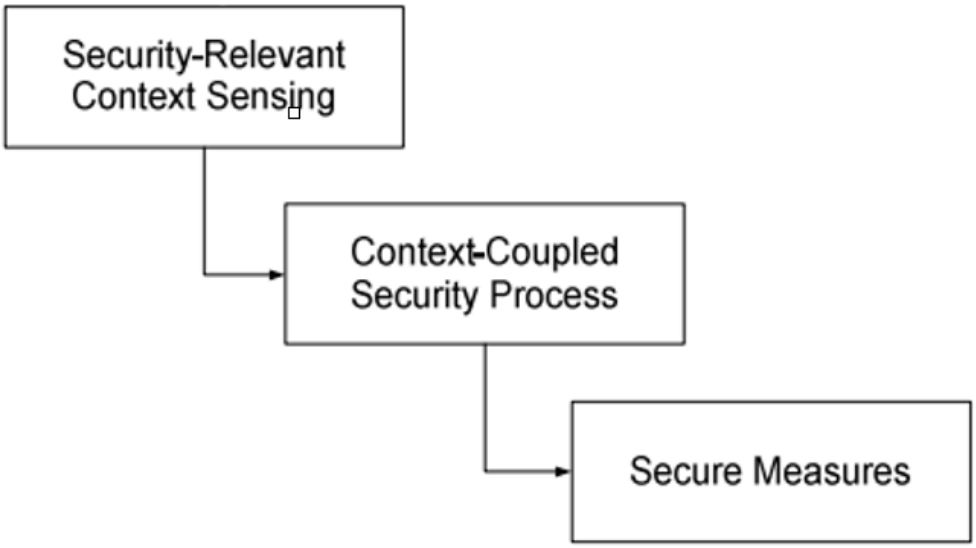


Figure 61: General context-aware security workflow [113]

Figure 62 shows the three major components of the context-aware security architecture for CPS: sensing, cyber, and control security.

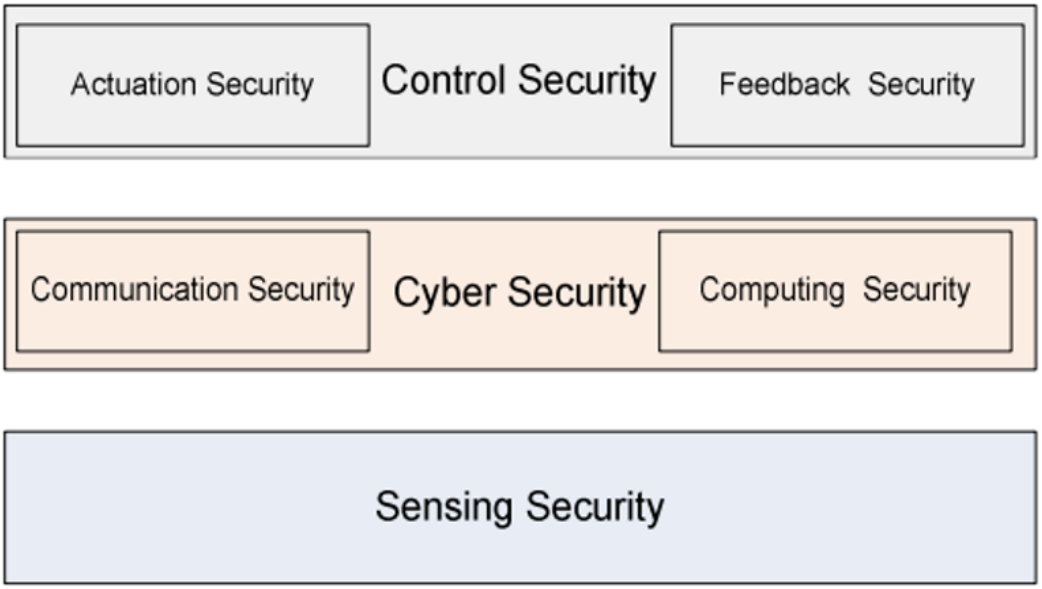


Figure 62: Main security aspects [113]

Sensing security: If the security configuration is context-sensitive, the context information must be reliable. They proposed using Trusted Platform Module From discovery to acquisition to communication. In order to develop trusted software systems, TPM [145] is required. It makes use of the TPM's capacity to guarantee to a distant verifier the integrity of software operating on a sensor. A TPM can provide a trusted boot by cryptographically hashing all code loaded at boot time. For increased security, they used the ARM11 [146] processor's trust zone feature. The processor chip contains memories, cryptographic eliminators, and the master key, in addition to authenticating all data from sensors to verifiers.

Cyber security Encompasses both security and security. In addition to establishing a network for the fusion of data and its distribution to back-end businesses, CPS is networked. In this way, they can protect inter- and intra-CPS communication from active (interferers) and passive (eavesdroppers) attackers alike. Among its components are a context-sensitive key management system, mutual authentication, and privacy protection. This can lead to future errors or disruptions.

Control Security: It has two categories.:

(1) actuation security and

(2) feedback security.

Actuation security makes sure that only individuals with permission can actuate. As CPS's requirements change, authorizations will be updated. "Feedback security" protects CPS control systems that provide actuation feedback. Modern security solutions focus on data security, but their effects on estimation and control algorithms must be studied to protect against CPS attacks [113].

4.3.2. Proposed CPS Security Requirements Engineering Framework

Framework for Security Needs Engineering (SRE) helps determine the security needs. Because of their heterogeneity and adaptability, CPSs have no complete framework for developing security requirements. For this reason, they proposed a framework for security requirements engineering that describes how to determine security needs during That part of the process wherein security needs for CPSs are elicited and settled upon is known as requirements engineering. These operations establish security to prevent attacks on a CPS. During the requirements engineering phase, this framework can help develop early security ideas. This pursuit results in RE approaches that address security concerns early in software development. The proposed framework integrates key CPS security objectives, threats, and risk assessments. Figure 63 shows their eight primary activities and one critical method called abuse case. An example of an abuse situation is one that would normally prohibit the system from working properly [147]. This method works for all processes related to primary activities. This CPS paradigm also assists practitioners and academics in determining security requirements. The framework identifies the processes required by requirement analysts to develop CPS security requirements.

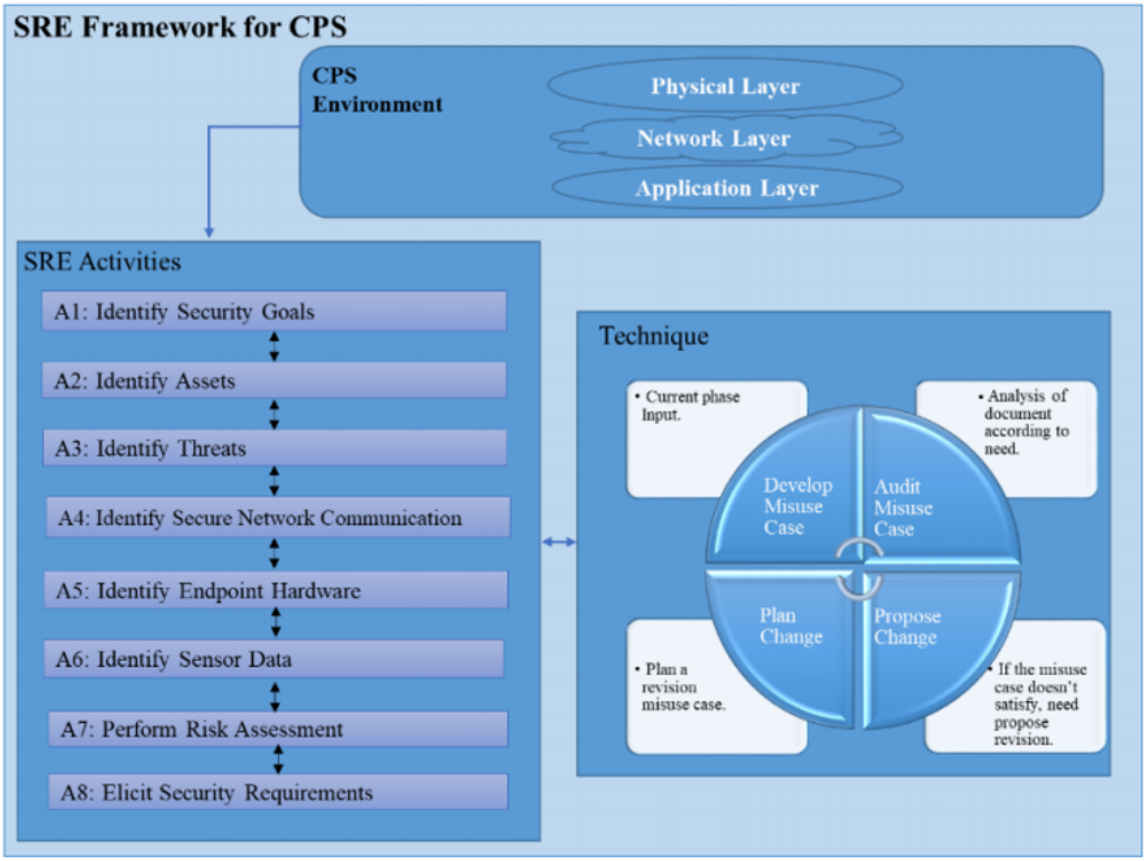


Figure 63: SRE Framework for CPS [111]

4.3.3. SRE Activities

The CPS framework has eight actions (A1 to A8). These activities were based because of their relevance to security requirements engineering [148] and cyber-physical systems [13,149].

4.3.3.1. Identify Security Goals

Security requirements are determined by business objectives and quality attributes. To establish security objectives, this is done. Confidentiality, integrity, and availability are the main security objectives.

4.3.3.2. Identify Assets

People, money, software, hardware, and sensors are all examples of assets. Thus, the goal of this action is to identify all CPS assets. In addition, the organization's environmental and physical assets are assessed. Human capital, data, network, sensors, and physical components are common examples.

4.3.3.3. Identify Threats

This task is to identify cyber-physical system threats. A misuse case is used to classify threats. They classified threats as software, network, and physical.

4.3.3.4. Identify Secure Network Communication

Participate in this exercise to develop a protocol for secure network communication. Devices in a wireless sensor network needs to be certified before it can be used. Implement industry-standard security protocols like TLS, DTLS, IKE/IPsec, and HIP-DE (HIP-DEX). This module implements a secure wireless sensor network communication protocol.

4.3.3.5. Identify Endpoint Hardware

Hardware Recognition Authenticated endpoint hardware is advised. Identifying supporting hardware such as sensors, routers, servers, or smart devices.

4.3.3.6. Identify Sensor Data

Identify the generation and exchange of sensor data Actuators and sensors exchange information with the outside world. The sensor gathers information about the thing it touches. These data are acquired via an API and routed to a PLC and SCADA system (SCADA). Certain higher-level sensors enable data broadcasting from the consolidated cloud. These sensors use M2M protocols to communicate. In order to properly analyze sensors, they recommended identifying all of the different mediums.

4.3.3.7. Perform Risk Assessment

This activity analyzes risk. Procedure reveals security risks. A system's assets and threats are identified. On a scale from low to high, the risk's influence on the asset is evaluated from 0 to 4. From there, the risk impact factor is calculated. The impact cost is the sum of the risk costs.

4.3.3.8. Perform Risk Assessment

Obtain Security Requirements Information

This task elicits, analyzes, and establishes security requirements. The requirements are organized and documented clearly [111].

4.4. Hardware Security for CPS

Aspects of CPS security are discussed in this section, including vulnerabilities and security challenges inherent in system hardware such as ICs, sensors and actuators, and PCBs. Historically,

The CPS has been developed on hardware that wasn't necessarily created or intended for use with the CPS, as well as on existing designs and architectures. CPS control, resource management, reliability, integrity, and security design issues require extra attention to vulnerabilities and attacks [115].

4.4.1. Security Issues around Sensor

Networks Given the reliance on sensor networks for many cyber-physical systems, their security is vital to avoid physical damage. Figure 64 shows physical, network, and application security. Widely-used sensor networks. They include industrial and military machine monitoring. Data security is critical because it is processed via networks. A typical network security solution does not meet the unique security requirements of sensor networks. In part, this is because the sensors are in open, public spaces. This makes them more vulnerable to vulnerability. This is an important network of sensor network security that should be considered for each component [150]. Unprotected components are vulnerable without it.

Sensor networks face a similar problem with confidentiality. Unwanted networks could be used to spy on people [150]. Example: long-term surveillance of people or vehicles. As shown in Figure 64, attacks on the sensor network's physical environment and gateway to the controller/server compromise sensor network security.

The sensor receives a powerful signal intended to interfere with physical network communication. For security-critical CPS, this can be disastrous. Military applications are vulnerable to such attacks. The very nature of networks may provide some protection against these attacks.

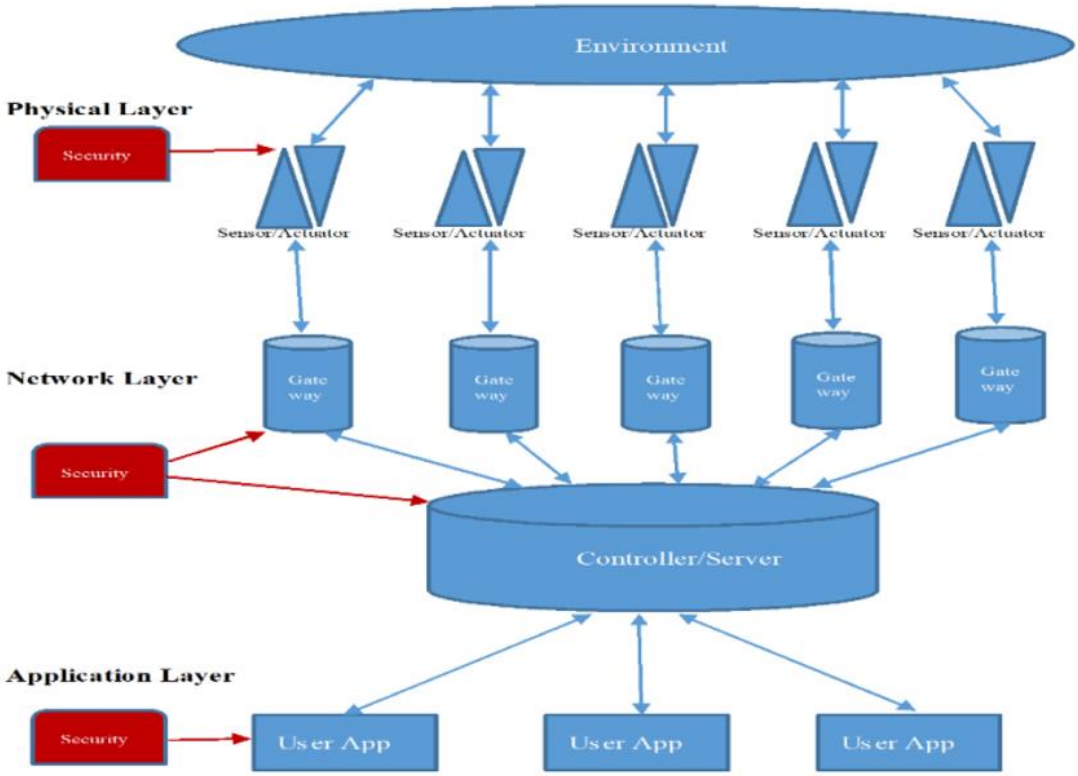


Figure 64: Security issues [111]

4.4.2. Hardware-based exploits

Large-scale systems with multi-functional hardware connected via networking are hard to verify, especially when legacy components are significant. Historically, heavyweight software and cryptographic protocols covered higher system abstraction layers [152]. Also, CPS-oriented cyber security must cover real-time communication between embedded systems and sensors, data communication layer, and controlling and processing units not built to comply with such security rules. The task-level security levels' lack of attention to hardware security exacerbates the vulnerabilities. Examples of hardware-based attacks.

4.4.2.1. Cryptographic Keys

The communication layer of a cyber-physical system ensures privacy and integrity by using public/private encryption schemes and cryptographic keys, as well as extra HSM. They are typically kept in persistent memory, that might be compromised without proper safety precautions. After obtaining these crucial keys, the attacker can carry out a number of deadly cyberattacks against the system. Comparable to smart card identity theft, but involving both hardware and software-aided attacks, and the possibility of cross-layer data flow, this attack may be more sophisticated.

4.4.2.2. Device Identity

As previously mentioned, attackers can access the system by stealing the device ID. Take advantage of vulnerabilities in cyber-physical systems to launch attacks like relay and replay [153]. For system, a remote sensor's ID can be stolen and used to compromise By impersonating a stolen ID, malware can compromise system security .This is similar to the 2007 nuclear reactor shutdown incident.

4.4.2.3. Physical Tampering of System Elements

Unchecked physical alteration puts device security and integrity, CPS performance, and cost at risk by disclosing backdoors to an attacker. For instance, an energy meter that has been physically interfered with may show lower energy usage than is actually the case, costing the supplier money. Elements that are Fake and Lack Security

4.4.2.4. Elements of Counterfeit with Low/No Security

Legacy components in cyber-physical systems frequently require additional maintenance and replacement. Ineffective supply chain management increases the risk of counterfeit components entering the system. It is possible that these counterfeit elements have a short life span, perform poorly, have defects or are out of specification. They may also have backdoors for remote attacks [154]. The use of a counterfeit integrated circuit with a short lifespan and/or performance that is out of specification in a critical application (Nuclear power plant radioactivity sensor with shutdown interrupt) poses a big risk to the system as a whole.

These hardware-oriented vulnerabilities and attacks may be consistent across layers while varying in coverage and threat-levels. When ensuring CPS hardware security, all possible attacks and weaknesses must be considered.

4.4.3. Hardware Security Primitives and Countermeasures

4.4.3.1. General

Secure hardware is required to maintain CPS integrity and ensure security from the inside. To upgrade all hardware is impractical due to lack of flexibility provided by software/firmware patches, higher labor and hardware costs, as well as the fact that many systems are built from legacy components that have been ad hoc integrated and evolved into CPS. The security of devices and systems is ensured by various hardware security primitives. Many of the issues facing CPS are unique to hardware-based security primitives such as PUFs and TRNGs, which cannot be addressed easily by software-based higher-level abstractions alone. Current hardware security primitives and potential remedies for CPS security are outlined below:

4.4.3.2. Physical Unclonable Function (PUFs)

They exploit inherent physical differences in parts like as transistors and interconnects to generate non-deterministic keys/signatures. As an alternative to systems that rely on maintaining keys in non-volatile memory, PUFs are volatile, less expensive, and more secure. Using cryptographic protocols and authentication systems that use secret keys and unique device identities to secure the communication layer is required since CPS heavily relies on multidimensional element interactions. PUFs can help avoid crypto-key and device identity theft by generating appropriate keys and authentication IDs without requiring any on-device key storage weaknesses. An embedded sequence of challenge-response numbers can also be used to provide authentication using PUFs [155]. A composite system-level PUF made just for the authentication protocol can be used to make PUF responses.

This PUF would feature an integrated cyber defence structure, according to the authors of [156]. System-level security characteristics are based by explaining the composite system of several PUFs. It comprises of an embedded component system with PUF circuits and a cluster of readers acting as cluster heads. In order to validate the system's integrity, each component's integrity must be validated individually. Next, it verifies the entire system's components by retrieving the system-level PUF response (a compilation of element-level PUF responses). Without component-level authentication, the system must determine which components failed authentication. This approach might easily be used to check the integrity of a PCB with several components [64]. Lesser-known PUFs in the elements can produce individual authentication IDs.

4.4.3.3. True Random Number Generators (TRNGs)

A true random number generator (TRNG) is used to make things like nonces, one-time pads, LFSR seeds, and cryptographic keys, among other things 158]. In most circumstances, a cryptographic conditioning unit is also included in a TRNG entropy source, extraction/sampling unit. A TRNG's key component is its entropy source. Unlike pseudo-random number generators, a TRNG derives its entropy from intrinsically unpredictable electrical and/or thermal processes. Possible causes include RTN in scaled transistors, power supply noise, radioactive decay, latch metastability, and ring oscillator jitter.After that, the entropy extraction/sampling unit samples the analogue entropy source. A voltage comparator comparing an RTN-prone signal to a reference voltage and generating a digital output is one example [159]. TRNGs may find specialized security applications in CPS. As cyber-physical systems (CPS) are comprised of multiple interconnected components, TRNGs are useful for generating random keys for one-time pads in different crypto-protocols or for generating session keys that safeguard CPS against unauthorized access (and cyber-attacks). For high-speed applications where the key might be used by multiple parts, a true

random number generator (TRNG) with low dependency on cryptographic hash functions and high throughput is required.

4.4.3.4. Design for Anti-Tamper

CPS is vulnerable to both software and hardware-based cyber-attacks, both remote and local. Denial of service attacks against CPS and theft of secrets (cryptographic keys or other sensitive data) require design-for-anti-tamper. Adversaries can conduct invasive, semi-invasive, or noninvasive attacks. A thorough understanding of the threat model is required to develop system-specific defense mechanisms [160]. Data leakage or system failure due to remote attacks on hardware (e.g., system power supply and clock glitches) or attack channels that are hidden from plain sight (e.g., cache timing attacks, etc.). For this reason, it can be challenging to implement real-time remote attack resistance strategies at both the system and device levels in a CPS, given the wide variety of devices that make up the system. Monitoring performance (e.g., PUF error, throughput, TRNG randomness), etc.) can indicate out-of-spec operations and potential security breaches to the trusted authority [160]. PUF and TRNG performance is highly dependent on operating conditions (power supply, temperature, etc.).

Reverse engineering and probing are two examples of semi-invasive and invasive assaults on large-scale CPS. Advanced tamper-sensing technology must be implemented to avoid physical tampering. While researchers have created silicon-level defenses against passive and active assaults, other crucial components like sensors and actuators continue to be exposed [69]. Active sensor networks can detect unwanted intrusions at the device and system level with appropriate extensions. There is no universal architecture for anti-tampering CPS design because CPS come in many micro and macro designs for various applications. [161].

4.4.3.5. Design for Anti-Counterfeit

Today's counterfeit integrated circuits (ICs) pose a serious threat to the functioning of the CPS. Most commercial and industrial CPSs use antiquated hardware that requires periodic upgrades. They also raise questions about security and compatibility with newer systems. As a result, users frequently rely on commercial components. A high risk of counterfeiting exists for these components due to their lack of traceability. Contaminated counterfeit chips that have been repurposed, cloned, or malfunctioning pose a serious threat to critical CPS systems (transportation, military, health, etc.). Counterfeit integrated circuits are often detected by identifying counterfeiting faults. A lengthy and complex physical inspection process may be required. Embedded sensors can detect previous usage of recycled integrated circuits [162]. Older system components may lack such integrated mechanisms. A sophisticated detection technique is required. Traditional systems like SCADA [163] and the IEEE P1711 standard for

legacy serial lines have attempted to secure data, but these efforts are insufficient to protect large-scale CPS. To reduce potential negative outcomes, it is necessary to implement a comprehensive plan for identifying and avoiding counterfeits.

Vulnerabilities exist in both the physical and cyber domains of CPS, and cannot be avoided solely through hardware security primitives. Also, threats and vulnerabilities at various levels of abstraction present unique challenges for CPS security and necessitate distinct solutions.

Thus, the application of the defense strategies is diverse. A more thorough examination of CPS abstractions regarding hardware interface and application-specific security protocols is necessary to determine a threat and attack model as well as potential defense scenarios [115].

4.4.4. PLC attacks

It is challenging to protect data from malicious actors, especially those that have complete control over PLCs. Consider the [165] study, which discovered that many PLCs suffer from authentication problems. The discovered vulnerability allows for complete remote control of the PLC via the internet. Using COTS devices and a software backdoor in a CPS can give complete control over PLCs [166]. To take over a system's controllers and issue commands to other devices without permission, the authors of [167] exploited the Modbus protocol's lack of authentication Stuxnet is a well-known malware attack that hijacked PLCs and changed their settings. Attackers have taken advantage of security holes in internet-connected PLC software by launching denial-of-service (DoS) and PLC reset attacks. Various malware and network-based attacks targeting PLCs have been developed and executed recently. As a result, non-invasive CPS device authentication is required.

4.4.5. Device Fingerprinting

A device's fingerprint is a collection of identifying features in its hardware, software, or both. The principles of device fingerprinting have been tested and proven effective in numerous settings.[168] discusses remotely fingerprinting a computer based on clock skew. A fingerprint is made using microscopic clock errors [60, 61]. You can locate devices on a small campus network using this technique. [169] develops a smartphone fingerprint using hardware flaws discovered during sensor manufacture. Actuator fingerprints were recently based in CPS. It was necessary to study industrial-grade sensors because device fingerprinting techniques for authentication and passive detection of attacks have been proven effective in information technology. One non-intrusive method for authenticating sensors that transmit data to PLCs is the NoiSense [170] technology. The sensors in a CPS, however, are not sufficiently equivalent in terms of functionality or computation to display the fingerprints [171]. They therefore look for a response to the

following: Do actual CPS sensors have their own distinct fingerprints? Manufacturing flaws in hardware exhibit distinct physical properties useful for profiling and fingerprinting [169]. They discovered that sensor noise (measurement errors) is highly dependent on manufacturing flaws. Changes in sensor noise patterns are difficult to mimic because they are unique to each device.

NoiSense extracts time and frequency domain information from sensor noise to create a sensor fingerprint. One sensor can be distinguished from another using machine learning. At experiments were out in a functioning water treatment and distribution facility, various sensors were used [172]. Sensor identification accuracy ranges from 97% to 90%. The proposed system also maintains the sensor fingerprint over time and scales to tens of sensors. Is each sensor imprinted with a unique fingerprint? The testbeds for water utilities have a finite number of sensors.To test for fingerprints on a large number of the same type and model ultrasonic sensors, extra low-cost ultrasonic sensors have been added. Ten identical dual transducer ultrasonic sensors (HCSR04) were used to detect a fingerprint. The identical tank held all ten sensors. We spent three hours gathering and studying the material [164].

Chapter 5

Conclusions & Future Directions

5. Conclusions and Future Directions

5.1. Future Directions

With the IoT's rapid growth, its security is becoming more important.

A safe IoT environment requires both software and hardware security. Sadly, the scientific literature lacks a comprehensive analysis of all IoT hardware security issues. Trust management and security must be put in place, beginning with a description of the threats at each level of the IoT system model. The perception layer is the most vulnerable because IoT devices are physically exposed, have limited resources, and use different kinds of technology. The diversity of IoT resources further complicates efforts to build a robust worldwide system for IoT layer protection.

These devices are vulnerable and unprotected. Part of the reason is the lack of IoT devices need to have safe hardware and software design, development, and deployment. The diversity of IoT resources further complicates efforts to build a robust worldwide system for IoT layer protection.

The vastness of the IoE makes it impossible for the community of researchers to solve all relevant problems. That said, we have consciously chosen to address first the most conspicuous impediments to getting to the terascale, and to devise research activities intended to remove those impediments. Solving the problems of powering, securing and designing a billion devices would have the highest impact of all the IoE activities we've identified.

For these reasons the researchers suggested a variety of future research themes and difficulties that must address to create scalable, reliable, and efficient IoT, IoE and CPS Hardware and no only security solutions. It follows a taxonomy of some open challenges for future research according the IoT, IoE, CPS environments

5.1.1. IoT

5.1.1.1. General

Several research investigations could be directed for the following purposes:

i. Identify the appropriate, current measurement techniques,

ii. Identify the measurement techniques that attackers use,

iii. evaluate the IoT hardware security needs according to the applications they are used for,

and apply cutting-edge security methods for IoT hardware

5.1.1.2. PUfs

Future uses of PUFs must focus on obtaining replicable schemes from Noisy PUFs and understanding each PUF's behavior in diverse environmental and physical situations. This problem involved devising systems that correspond to the anticipated error in each device, avoiding under- or over-correcting PUF replies, as well as lowering the Rates of False Rejection and False Authentication

Overall, we believe IoT researchers should revisit the primary aims behind employing PUF and re-evaluate the original use cases. PUFs could help implement cryptography more effectively using electronic signatures, enhanced random numbers, or encryption keys. Newer PUF work focuses on this. Also recommended for IoT

5.1.1.3. Resource limitations

The IoT's resource-constrained architecture has made it difficult to develop robust security mechanisms. Cryptographic algorithms, unlike normal paradigms, must be constrained to work within these constraints. To successfully deploy security and communication protocols for IoT, any required broadcasts or multicasts must meet both storage and energy requirements

5.1.1.4. Heterogeneous devices

A multi-layer security architecture is set up for different kinds of devices, from small sensors to powerful servers. Before providing services to end users, the framework should adapt to existing resources and make methods about IoT tier security. This type of dynamic security framework requires intelligence, which is standardized in IoT infrastructures.

5.1.1.5. Hardware/firmware vulnerabilities

Weaknesses in hardware may become more common as low-cost, low-power devices emerge. Before IoT is put into use, the security algorithms in the hardware, routing, and packet processing methods must all be checked. If a flaw is exploited after distribution, it is harder to find and fix Thus, a standardized verification protocol is required to leverage IoT security.

5.1.1.6. Trusted updates and management

How to manage and update software on millions of IoT devices in a scalable and trusted manner is an important open research question for the future. Concerns about secure and trusted ownership and management of IoT devices, the supply chain, and data privacy are open research questions that the research community needs to answer to help IoT become widely used. Blockchain technology may help create IoT security solutions. Scalability, efficiency, arbitration/regulations, and key collision are significant research issues with blockchain technology.

5.1.1.7. Blockchain vulnerabilities

They are still vulnerable [284] despite the robustness of blockchain technologies. To host the blockchain, the attacker must first compromise the consensus process, which relies on the miner's hashing power. Compromised blockchain accounts can also be exploited using randomized private keys. However, effective strategies for protecting transaction privacy and preventing race attacks that could lead to double spending are still unknown

5.1.1.8. Security protocol interoperability

The protocols must communicate via conversion mechanisms to standardize a global security framework for the IoTs. It is possible to combine the security requirements at each tier in a way that is appropriate using architectural restrictions found within the global mechanism.

5.1.1.9. Single points of failure

The IoTs paradigm is more susceptible than other systems to single points of failure paradigms due to network, infrastructure, and protocol heterogeneity. There is still much to learn about ensuring IoT availability, especially for mission-critical applications. These systems and standards must create redundancy while considering the overall infrastructure's cost-benefit ratio.

5.1.2. IoE

5.1.2.1. PUfs

A low-power PUF integration design can be pursued, and different consensus techniques can be investigated.

5.1.2.2. AMI

A variety of services including periodic billing, distributed state estimations, and real-time pricing can’t be provided without access to AMI data. Each service's data collection frequency and

accuracy vary. While doing so, they considered privacy use cases and consumer data privacy as they researched cryptographic and non-cryptographic methods for protecting user privacy in AMI. While some existing systems provide these services while maintaining customer privacy, our research shows that they have several flaws.

Privacy preserving data

A few systems protect data during transmission, while others at the smart meter level, conceal datal. Other methods use trusted third parties to hide data from utility companies. BLH is an excellent approach, but it has some significant drawbacks. Battery recharging and discharge may interfere with the dynamic price.

Existing solutions can't provide differential privacy and save money. Using escrow services or trusted third party services to anonymize data is insufficient because they must be trustworthy regarding true identities.

One common method of data aggregation is homomorphic encryption. Due to its computational complexity and cost, homomorphic encryption is considered unsuitable for smart grids. Homomorphic encryption raises issues of differential privacy and error tolerance. The adoption would rise with the creation of completely homomorphic cryptosystems that are computationally efficient.

In order to guarantee that the aggregator is only aware of the total meter readings,simple multi-party communication (SMPC) approaches can be utilized. SMPC, on the other hand, has a high computational cost and requires node interaction throughout the computation phase. Reduced engagement costs may make it an appealing option

AMI services

Real-time pricing and distributed state estimation, and periodic or on-demand metering are all services that leverage AMI data in transit. In one or more of the aforementioned scenarios, the non-cryptographic and cryptographic works seek to safeguard the privacy of the user. These methods frequently don't meet the demands for the desired services. By concealing the data's source and presenting a generalized reading of energy usage, for instance, data anonymization safeguards the privacy of consumers. Utility companies might not be able to offer consumer-specific services despite the fact that this strategy offers privacy and permits distributed state estimate. Therefore, both customer privacy and effective service delivery must be maintained.

5.1.2.3. Technology Spread

As with other similar technologies, the proposed IoE's main barrier is user adoption.

While it is conceivable to construct a new network of devices that operate by the proposed IoE paradigm, we may significantly mitigate this issue by incorporating the IoE network into existing wireless networks (e.g., IoT and mobile). This process, which enables us to optimise the IoE potential, can be facilitated by using several tactics, including the ones listed below.:

i. developing a simple and transparent procedure for integrating the required IoE functionalities into existing tracker devices, for example, by integrating them as a service in new devices, using a simple firmware/software upgrade process, or creating an app, when the trackers or entities are built into devices that work with this solution (e.g., smartphones, tablets, etc.);

ii. conducting effective information campaigns emphasizing the benefits to each user who joins the IoE network, emphasizing the gained opportunity to exchange information with a large community of users, an enormous amount of valuable data that they can exploit in a variety of contexts, including the one of security discussed in that paper;

iii. rewarding users who connect their devices to the IoE network as trackers, thereby enabling the system to conduct entity detection and distributed-ledger registration duties. Such benefits could include unrestricted access to some IoE network services, such as those utilised for remote data storage.

The interaction between entities and trackers can be implemented using custom (e.g., wearable) or standard (IoT, smart phone, and tablet) devices, but the IoE's potential can be enhanced by adding routers, access points, hotspots, and others to the network.

5.1.3. CPS

Since many devices connect to the same centralized network, IoT open issues in the CPS are problematic. Even if one of thousands of IoT devices is compromised, it's connected to the data center network. The centralized network housing many IoT devices could also be hacked. Using IoT to attack humans is a bigger concern. Unlawful use of personal data is a CPS data center issue. Due to the CPS data center's enormous collection of data in many forms for producing analytical conclusions. The data center does not include personal information.

AIs open questions have become a threat to humanity. AI can threaten factory control, medical diagnosis, editing, and creativity. The Internet allows the Fourth Industrial Revolution to collect and receive data quickly. It can be implemented into CPS through data analysis, learning, and AI. IoT devices are spread throughout a physical system. The IoT network manages data.

Because many cyber and physical IoT devices are technological, 4IR security threats are emerging. Threats rise, but control expands. More CPS space means more risk. Therefore, it is not advisable to apply the Fourth Industrial Revolution technological risk analysis. Conduct a CPS risk analysis if all IoT are linked and operational.

Closing I would like to add three more future research for study:

- Standardized abstractions and architectures are needed to modularize cyber-physical systems.
- CPS applications have complicated, connected physical environments. Reliability and security provide unique difficulties, requiring new frameworks, algorithms, and tools.
- Future cyber-physical systems will need highly reliable, adaptable, and in many cases, certifiable hardware and software, and system-level trustworthiness

5.2. Conclusions

In today's world, the IoTs is inescapable. To avoid material or even human losses, it is imperative to safeguard the IoT environment immediately. The enormous benefits of new technologies are dramatically compromised by a series of security concerns raised by an increasing number of people seeking an unfair advantage. Kidnappings, fraud, and theft are some of the traditional security concerns that plague modern societies.

This thesis reviewed the literature on hardware security issues and numerous security challenges in IoT devices and IoE and CPS environments. The researchers classified these issues into high, middle, and low-level IoT components. The IoT security taxonomy takes into account all aspects of security, including data, connection, architecture, and application.

Modern security paradigms do not fully exploit powerful technologies like wireless smart devices or the IoTs, which has millions of active devices, or blockchain-based distributed ledgers, which allow the certification of a series of events. Blockchain technology solves the issues associated with centralized IoT systems, such as single-point attacks, privacy leakage, and limited scalability. However, limited resources, heterogeneity, and network topology mobility in IoT environments have created new challenges.

To create a secure IoT ecosystem, developers must secure both software and hardware. Threats from software and hardware are interdependent. Evidence from the literature suggests

that not all hardware security issues facing IoT systems have definitive answers. In addition, attackers' present measurement methods and equipment are unknown. In addition, attackers' present measurement methods and equipment are unknown. The IoT's hardware variety is one of its most appealing features. Their diversity makes them vulnerable to outside side-channel attacks, making their security a significant concern that must be addressed quickly. We can assist prevent aftermarket side-channel attacks by creating new techniques and tools for analyzing side-channels in IoT products in the lab.

IoT devices can't be secure without safe hardware. If the gadget has an embedded HT that can destroy it at any time, all the investment and labor could be wasted. IoTs must have embedded hardware security to protect the "identity" of devices, prevent tampering, and secure their data. Understanding HT taxonomy can help researchers identify and stop distinct types of HT. Researchers can better recognize and prevent exploitable windows of time in an IC's lifecycle if they have a firm grasp of the different points at which HT insertion might occur. IoT data security requires embedded hardware security. TPM and DICE add protection to the cryptographic keys needed to protect the system's integrity, secrecy, and authenticity.

An attack on the device or its data should not only secure guard the device and its data, but also keep the user's privacy. Not all security is equal. This sector demands acceptable security requirements, especially during design, as most techniques depend on post-silicon security. Achieving high hardware security requires protecting the environment and developing appropriate regulations to assure secure chip manufacture while respecting third-party privacy. The thesis discusses the HT, the most serious danger to hardware security, and how side-channel analysis helps. The thesis discusses the HT taxonomy in full. Researchers can benefit from a deeper understanding of the HT taxonomy so that they can implement more effective, new approaches.to identify and deter HT. Researchers should also study the stages of HT insertion throughout an IC's lifecycle to detect and prevent HT insertion at these vulnerable times. Hardware security solutions such as HSM and TPM are offered, as among others, detection techniques, a design for trust framework, and a split manufacturing for trust model.

Many steps in the chain leading up to the completion of the hardware design for ICs were examined as well. Multiple hardware flaws are introduced at different points in this chain.

PUFs are one answer to hardware-based attacks. PUFs., are new types of primitive security with a simple structure that makes them ideal for low-cost IoT network security. A low-cost authentication and encryption system using PUF is demonstrated here. In spite of this, PUF technology has some reliability and security concerns. Different responses to PUFs pose issues. Overcorrecting noisy PUF replies with FE techniques might increase the likelihood of inaccurate

authentication or erroneous rejection rates, which is briefly discussed. In order to address the issues, researchers are looking into ways to reduce the IoT devices' energy research.

The IoT's future depends on our ability to provide cost-effective, reliable security solutions; otherwise, many IoT nodes will be connected to the internet with little or no protection, leaving them vulnerable to previously inconceivable security threats. Because most IoT devices will be low-power and lightweight, using traditional cryptographic algorithms will be prohibitively expensive.

Furthermore, this thesis addressed a new security paradigm called IoE, which combines wireless-based device capabilities with distributed ledger certification capabilities. Entities and trackers are billions of new or existing devices that can communicate in IoE.

Blockchain and IoE convergence require technologies like cryptography, authentication, consensus, and reputation assessment. Computationally intensive consensus algorithms for distributed device authentication and data validation. For this reason, addressed a novel blockchain architecture called the PUF chain. Future research could focus on ultra-low-power PUF integration and alternative consensus algorithms.

It also offered an overview of the huge data created by interactions between people, machines, and sensors in the IoE and explored the viability of establishing Big data analytics-based IoT security.

From the other side Due to its novelty and differences from the established network landscape, the field of CPS security has seen relatively little attention thus far. CPS transmission mediums include sensors, data formats, real-time data generation, process analysis, and application interactions.

Future IT will broaden CPS security by integrating the IoTs and other sensors. It is therefore necessary to protect the security of the system through communication with other systems in a variety of settings.

Considering the pervasive use of CPS in diverse "smart" environments like the "smart home," "smart city," "smart industry," "smart healthcare," and "smart grid," CPS security should be a constant worry. As the number of connected devices grows, the importance of continuously improving CPS security measures rises.

Likewise, this thesis was given cross-layer security challenges inherent in current cyber-physical systems. There is a framework for securing CPS. While cyber-physical systems require security, safe development techniques are lacking. Many security requirements techniques exist, however they are all software-centric and do not support cyber-physical systems. addressed a complete security requirement engineering framework for cyber-physical systems that may aid practitioners and researchers when identifying security requirements. In order to achieve this security vulnerabilities and countermeasures were established at the system, device, and hardware levels. Using passive sensor measurements, a fingerprinting technique for the sensor and processing noise is shown to be effective in identifying it. We develop upper bounds on state deviation in the presence of a stealth attack. The results show that utilizing the noise fingerprint, sensors may be identified with up to 98 percent accuracy. This is a high rate of true positive and negative detections. An argument for security against stealthy attacks. The proposed technique detects a strong enemy.

They noted that in addition to the cyber infrastructure, there are a number of physical processes to secure. Physics-based techniques can identify CPS attacks as well. This strategy, however, has limits. Ensuring physical systems is a difficulty that can be solved by precisely modeling typical processes. They also emphasized data integrity over data confidentiality in CP. The goal is to detect attacks accurately while minimizing false alarms. An improvement to the system model using a bank of observers' scheme is possible. Detecting an attack is the first step to recovery. Significant effort has been made by CPS to develop models for detecting attacks. Model-based attack detection systems, on the other hand, are unable to spot covert and multi-point attacks and can disrupt normal operations.

Sadly, the scientific literature lacks a comprehensive analysis of all IoT, IoE and CPS hardware security issues. Future IoT will depend on our ability to find cost-effective, reliable security solutions; otherwise, many IoT nodes will be connected to the internet with little or no protection, leading to unprecedented security attacks.

Cyber-physical systems are predicted to play a prominent part in future engineering systems with far greater autonomy, functionality, usability, reliability, and cyber security. Close partnerships between academic disciplines in computing, communication, control, and other engineering and computer science disciplines, combined with grand challenge applications, can speed CPS research.

References

1. Bhunia, S., & Tehranipoor, M. (2018). Hardware security: a hands-on learning approach. Morgan Kaufmann.

2. Kocher, P. C. (1996, August). Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Annual International Cryptology Conference (pp. 104-113). Springer, Berlin, Heidelberg

3. Voas, J. (1997). Fault injection for the masses. Computer, 30(12), 129-130.

4. Mukhopadhyay, D., & Chakraborty, R. S. (2014). Hardware security: design, threats, and safeguards. CRC Press.

5. Tuyls, P. (2010). Towards hardware-intrinsic security: foundations and practice. Springer Science & Business Media.

6. Benjamin Kleine, Bethany Lobo, Amanada Levendowski March 2015 Internet of Things: The new frontier for data security and privacy (Part 1).

7. Iqbal, M. A., Olaleye, O. G., & Bayoumi, M. A. (2017). A review on internet of things (IoT): security and privacy requirements and the solution approaches. Global Journal of Computer Science and Technology.

8. ur Rehman, S., & Gruhn, V. (2017, May). Recommended architecture for car parking management system based on cyber-physical system. In 2017 International Conference on Engineering & MIS (ICEMIS) (pp. 1-6). IEEE.

9. Shi, J., Wan, J., Yan, H., & Suo, H. (2011, November). A survey of cyber-physical systems. In 2011 international conference on wireless communications and signal processing (WCSP) (pp. 1-6). IEEE.

10. Zhang, F., Szwaykowska, K., Wolf, W., & Mooney, V. (2008, November). Task scheduling for control-oriented requirements for cyber-physical systems. In 2008 Real-Time Systems Symposium (pp. 47-56). IEEE.

11. Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010, June). Cyber-physical systems: the next computing revolution. In Design automation conference (pp. 731-736). IEEE.

12. U.S. Energy Information Administration, U.S. Department of Energy, “International Energy Statitics,” [Online]: http://www.eia.gov/.

13. Kyoung-Dae Kim and P.R. Kumar, “Cyber-physical systems: A perspective at the centennial,” Proceedings of the IEEE, vol. 100, no. Special Centennial Issue, pp. 1287–1308, 2012.

14. Konstantinou, C., Maniatakos, M., Saqib, F., Hu, S., Plusquellic, J., & Jin, Y. (2015, May). Cyber-physical systems: A security perspective. In 2015 20th IEEE European Test Symposium (ETS) (pp. 1-8). IEEE.

15. Satyabrata_Jena (2022, Jan) “Difference between IoE and IoT”, [ONLINE]: Https://www.geeksforgeeks.org/difference-between-ioe-and-iot/

16. Vanderbilt Engineering Graduate Admissions Team, (2022, Feb), [ONLINE]: https://blog.engineering.vanderbilt.edu/what-is-the-difference-between-cps-and-iot

17. Michael E. Porter and James E. Heppelmann (M2014, Nov), [ONLINE]: Https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition

18. M. Tehranipoor, F. Koushanfar (2010), A Survey of Hardware Trojan Taxonomy and Detection, IEEE Design and Test of Computers

19. Bhunia, S., & Tehranipoor, M. (2018). Hardware security: a hands-on learning approach. Morgan Kaufmann.

20. S. Ray, E. Peeters, M.M. Tehranipoor, S. Bhunia, (2018), System-on-chip platform security assurance: architecture and validation, Proceedings of the IEEE 106 (1) 21–37.

21. M. Tehranipoor, U. Guin, D. Forte, (2015). Counterfeit integrated circuits, Counterfeit Integrated Circuits 15–36.

22. R. Torrance, D. James, (2011). The State-of-the-Art in Semiconductor Reverse Engineering, ACM/EDAC/IEEE Design Automation Conference (DAC) 333–338.

23. F. Koeune, F.X. Standaert, 2005. A tutorial on physical security and side-channel attacks, in: Foundations of Security Analysis and Design III, pp. 78–108.

24. P. Kocher, J. Jaffe, B. Jun, , 1999. Differential power analysis, in: CRYPTO

25. F. Wang, (2004). Formal Verification of Timed Systems:A Survey and Perspective, Proceedings of the IEEE 1283–1305

26. Shamsoshoara, A., Korenda, A., Afghah, F., & Zeadally, S. (2020). A survey on physical unclonable function (PUF)-based security solutions for Internet of Things. Computer Networks, 183, 107593.

27. M. Bhayani, M. Patel, C. Bhatt, 2016. Internet of Things (IoT): In a way of SmartWorld, in: Proceedings of the international congress on information and communication technology, Springer, pp. 343-350.

28. R. Kloti, V. Kotronis, P. Smith, Openflow, 2013, A Security Analysis, in: Network Protocols (ICNP), 2013 21st IEEE International Conference on, IEEE, pp. 1-6.

29. A.Shamsoshoara, Y. Darmani, (2015). Enhanced Multi-route ad hoc On-demand Distance Vector Routing, in: Electrical Engineering (ICEE), 2015 23rd Iranian Conference on, IEEE, , pp. 578-583.

30. Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, K. Leung, (2013). A Survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities, IEEE Wireless Communications 20 (6) 91-98.

31. Tudosa, I., Picariello, F., Balestrieri, E., De Vito, L., & Lamonaca, F. (2019, June). Hardware security in IoT era: The role of measurements and instrumentation. In 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT) (pp. 285-290). IEEE.

32. G. A. Fink, 2015. “Security and privacy grand challenges for the Internet of Things,” in Proc. of CTS, ., pp.27–34

33. M. Wolf, A. Weimerskirch, “Hardware Security Modules for Protecting Embedded Systems," Application Note, [Online]. Available: www.escrypt.com

34. A.Kalnoskas, “How IoT and mixed-signal designs will drive SiP tech in 2016," [Online]. Available: www.analogictips.com

35. Y. Jin, Y. Makris, “Hardware trojan detection using path delay fingerprint,” in Proc. of HOST, 2008, pp. 51–57

36. Frustaci, M., Pace, P., Aloi, G., & Fortino, G. (2017). Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet of things journal, 5(4), 2483-2495.

37. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, 2015. “Internet of Things: A survey on enabling technologies, protocols, and applications,” IEEE Commun. Surveys Tuts., vol. 17, no. 4, pp. 2347–2376, 4th Quart.

38. Shamsoshoara, A., Korenda, A., Afghah, F., & Zeadally, S. (2019). A survey on hardware-based security mechanisms for internet of things. ArXiv. org.

39. Farzad Samie, Vasileios Tsoutsouras, Lars Bauer, Sotirios Xydis, Dimitrios Soudris, and Jörg Henkel, 2016. Computation offloading and resource allocation for low-power IoT edge devices. In internet of Things (WF-IoT), IEEE 3rd World Forum on. IEEE, 7–12.

40. Hiroyuki Akinaga and Hisashi Shima. 2010. Resistive random-access memory (ReRAM) based on metal oxides. Proc. IEEE 98, 2237–2251.

41. Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. 2017. Internet of Things security: A survey. Journal of Network and Computer Applications 88 (2017), 10–28.

42. DiegoMMendez, Ioannis Papapanagiotou, and Baijian Yang. 2017. Internet of things: Survey on security and privacy. arXiv preprint arXiv:1707.01879, (2017).

43. Kai Zhao and Lina Ge. 2013. A survey on the internet of things security. In Computational Intelligence and Security (CIS), 2013 9th International Conference on. IEEE, 663–667.

44. Kai Zhao and Lina Ge. 2013. A survey on the internet of things security. In Computational Intelligence and Security (CIS), 2013 9th International, Conference on. IEEE, 663–667

45. Shamsoshoara, A., Korenda, A., Afghah, F., & Zeadally, S. (2019). A survey on hardware-based security mechanisms for internet of things. ArXiv. org.

46. Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. 2014. On the integration of cloud computing and internet of things. In 2014 International Conference on Future Internet of Things and Cloud. IEEE, 23–30.

47. "Srdjan Capkun, Levente Buttyán, and Jean-Pierre Hubaux. 2003. Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on mobile computing 1 (2003), 52–64."

48. Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. 2017. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet of Things Journal 4, 5 (2017), 1125–1142.

49. Laurent Eschenauer and Virgil D Gligor. 2002. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM conference on Computer and communications security. ACM, 41–47.

50. Ibrahim Abaker Targio Hashem, Victor Chang, Nor Badrul Anuar, Kayode Adewole, Ibrar Yaqoob, Abdullah Gani, Ejaz Ahmed, and Haruna Chiroma. 2016. The role of big data in smart city. International Journal of Information Management 36, 5 (2016), 748–758.

51. Pavan Pongle and Gurunath Chavan. 2015. A survey: Attacks on RPL and 6LoWPAN in IoT. In Pervasive Computing (ICPC), 2015 International Conference on. IEEE, 1–6.

52. Angel Leonardo Valdivieso Caraguay, Alberto Benito Peral, Lorena Isabel Barona Lopez, and Luis Javier Garcia Villalba. 2014. SDN: Evolution and opportunities in the development IoT applications. International Journal of Distributed Sensor Networks 10, 5 (2014), 735142.

53. Sanaz Rahimi Moosavi, Tuan Nguyen Gia, Amir-Mohammad Rahmani, Ethiopia Nigussie, Seppo Virtanen, Jouni Isoaho, and Hannu Tenhunen. 2015. SEA: a secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways. Procedia Computer Science 52 (2015), 452–459.

54. Mališa Vučinić, Bernard Tourancheau, Franck Rousseau, Andrzej Duda, Laurent Damon, and Roberto Guizzetti. 2015. OSCAR: Object security architecture for the Internet of Things. Ad Hoc Networks 32 (2015), 3–16

55. Aditya Gaur, Bryan Scotney, Gerard Parr, and Sally McClean. 2015. Smart city architecture and its applications based on IoT. Procedia computer science 52 (2015), 1089–1094

56. Michael Galleso. 2016. Samsung Gear S3 Classic and Frontier: An Easy Guide to Best Features. Lulu Press, Inc

57. Shaibal Chakrabarty and Daniel W Engels. 2016. A secure IoT architecture for Smart Cities. In 2016 13th IEEE annual consumer communications & networking conference (CCNC). IEEE, 812–813.

58. Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems 29, 7 (2013), 1645–1660.

59. Dong Chen, Guiran Chang, Lizhong Jin, Xiaodong Ren, Jiajia Li, and Fengyun Li. 2011. A novel secure architecture for the Internet of Things. In Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on. IEEE, 311–314.

60. Eleonora Borgia. 2014. The Internet of Things vision: Key features, applications and open issues. Computer Communications 54 (2014), 1–31.

61. Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. 2016. Integration of cloud computing and internet of things: a survey. Future Generation Computer Systems 56 (2016), 684–700.

62. Y. B. Zhou, D. G. Feng, “Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing," IACR Eprint archive, 2005. [Online]. Available: https://eprint.iacr.org/2005/388

63. N.Adochiei, V.David, F.Adochiei, I.Tudosa, 2011, “ECG waves and features extraction using Wavelet Multi-Resolution Analysis," Proc. of E-Health and Bioengineering Conference (EHB), Iasi, pp.1-4.

64. C. Lesjak, D. Hein, J. Winter, 2015, Hardware-security Technologies for Industrial IoT: TrustZone and Security Controller, in: IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society, IEEE, pp. 002589{002595.

65. Shancang Li, Li Da Xu, and Shanshan Zhao. 2015. The internet of things: a survey. Information Systems Frontiers 17, 2 (2015), 243–259.

66. Ebraheim Alsaadi and Abdallah Tubaishat. 2015. Internet of things: features, challenges, and vulnerabilities. International Journal of Advanced Computer Science and Information Technology 4, 1 (2015), 1–13

67. Sahabul Alam and Debashis De. 2014. Analysis of security threats in wireless sensor network. arXiv preprint arXiv:1406.0298 (2014).

68. Deepak Nandal Sushma and Vikas Nandal. 2011. Security threats in wireless sensor networks. IJCSMS International Journal of Computer Science & Management Studies 11, 01 (2011), 59–63.

69. Ulya Sabeel and Saima Maqbool. 2013. Categorized security threats in the wireless sensor networks: Countermeasures and security management schemes. International Journal of Computer Applications 64, 16 (2013).

70. Daniel Genkin, Adi Shamir, and Eran Tromer. 2014. RSA key extraction via low-bandwidth acoustic cryptanalysis. In Annual Cryptology Conference. Springer, 444–461.

71. "Md Mahmud Hossain, Maziar Fotouhi, and Ragib Hasan. 2015. Towards an analysis of security issues, challenges, and open problems in the internet of things. In Services (SERVICES), 2015 IEEE World Congress on. IEEE, 21–28."

72. Yang Lu and Li Da Xu. 2018. Internet of Things (IoT) cybersecurity research: a review of current research topics. IEEE Internet of Things Journal 6, 2 (2018), 2103–2115.

73. Huichen Lin and Neil Bergmann. 2016. IoT privacy and security challenges for smart home environments. Information 7, 3 (2016), 44.

74. Tuhin Borgohain, Uday Kumar, and Sugata Sanyal. 2015. Survey of security and privacy issues of internet of things. arXiv preprint arXiv:1501.02211 (2015).

75. Sathish Alampalayam Kumar, Tyler Vealey, and Harshit Srivastava. 2016. Security in internet of things: Challenges, solutions and future directions. In 2016 49th Hawaii International Conference on System Sciences (HICSS). IEEE, 5772–5781.

76. Steven T Eckmann, Giovanni Vigna, and Richard A Kemmerer. 2002. STATL: An attack language for state-based intrusion detection. Journal of computer security 10, 1-2 (2002), 71–103

77. Rajendra Billure, Varun M Tayur, and V Mahesh. 2015. Internet of Things-a study on the security challenges. In Advance Computing Conference (IACC), 2015 IEEE International. IEEE, 247–252.

78. "Zhi-Kai Zhang, Michael Cheng Yi Cho, Chia-Wei Wang, Chia-Wei Hsu, Chong-Kuan Chen, and Shiuhpyng Shieh. 2014. IoT security: ongoing challenges and research opportunities. In Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on. IEEE, 230–234."

79. Ian G Harris. 2016. Social Engineering Attacks on the Internet of Things. Pridobljeno iz http://iot. ieee. org/newsletter/september-2016/socialengineering-attacks-on-theinternet-of-things. html (16. 4. 2017) (2016).

80. Yuanjun Song. 2013. Security in Internet of Things

81. Paul Leach, Michael Mealling, and Rich Salz. 2005. A universally unique identifier (uuid) urn namespace. Technical Report. Network Working Group.

82. Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. 2008. MQTT-SâĂŤA publish/subscribe protocol for Wireless Sensor Networks. In Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on. IEEE, 791–798.

83. Zach Shelby, Klaus Hartke, and Carsten Bormann. 2014. The constrained application protocol (CoAP). Technical Report. IETF.

84. Xue Yang, Zhihua Li, Zhenmin Geng, and Haitao Zhang. 2012. A multi-layer security model for internet of things. In Internet of things. Springer, 388–393.

85. Farinaz Koushanfar, Saverio Fazzari, Carl McCants, William Bryson, Peilin Song, Matthew Sale, and Miodrag Potkonjak. 2012. Can EDA combat the rise of electronic counterfeiting? In DAC Design Automation Conference 2012. IEEE, 133–138

86. Pankaj Rohatgi. 2009. Improved techniques for side-channel analysis. In Cryptographic Engineering. Springer, 381–406

87. Prerna Mahajan and Abhishek Sachdeva. 2013. A study of encryption algorithms AES, DES and RSA for security. Global Journal of Computer Science and Technology (2013)

88. Wikipedia. 2019. RSA (cryptosystem) - Wikipedia. https://en.wikipedia.org/wiki/RSA_(cryptosystem). (Accessed on 04/26/2019).

89. Alexander Schlösser, Dmitry Nedospasov, Juliane Krämer, Susanna Orlic, and Jean-Pierre Seifert. 2013. Simple photonic emission analysis of AES. Journal of Cryptographic Engineering 3, 1 (2013), 3–15

90. Sidhu, S., Mohd, B. J., & Hayajneh, T. (2019). Hardware security in IoT devices with emphasis on hardware Trojans. Journal of Sensor and Actuator Networks, 8(3), 42.

91. Breier, J.; He, W, 2015. Multiple Fault Attack on PRESENT with a Hardware Trojan Implementation in FPGA. In Proceedings of the IEEE International Workshop on Secure Internet of Things (SIoT), Vienna, Austria, 21–25 September 2015; pp. 58–64

92. Randy Torrance and Dick James. 2011. The state-of-the-art in semiconductor reverse engineering. In 2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC). IEEE, 333–338

93. Samah Mohamed Saeed, Xiaotong Cui, Robert Wille, Alwin Zulehner, Kaijie Wu, Rolf Drechsler, and Ramesh Karri. 2017. Towards reverse engineering reversible logic. arXiv preprint arXiv:1704.08397 (2017).

94. Yu Bi. 2016. Enhanced Hardware Security Using Charge-Based Emerging Device Technology. University of Central Florida, Thesis in Ph.D. (2016). [20] Rajendra Billure,

Varun M Tayur, and V Mahesh. 2015. Internet of Things-a study on the security challenges. In Advance Computing Conference (IACC), 2015 IEEE International. IEEE, 247–252.

95. Jarrod A Roy, Farinaz Koushanfar, and Igor L Markov. 2010. Ending piracy of integrated circuits. Computer 43, 10 (2010), 30–38

96. Mohammad Tehranipoor and Farinaz Koushanfar. 2010. A survey of hardware trojan taxonomy and detection. IEEE design & test of computers 27, 1 (2010), 10–25.

97. Syed, A., & Lourde, R. M. (2016, December). Hardware security threats to DSP applications in an IoT network. In 2016 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS) (pp. 62-66). IEEE.

98. Bhunia, S.; Narasimhan, S.; Chakraborty, R. Hardware Trojan: Threats and emerging solutions. In Proceedings of the 2009 IEEE International High Level Design Validation and Test Workshop, San Francisco, CA, USA, 4–6 November 2009; pp. 166–171

99. Koley, S.; Ghosal, P. Addressing Hardware Security Challenges in Internet of Things: Recent Trends and Possible Solutions. In Proceedings of the 12th IEEE International Conference on Advanced and Trusted Computing (UIC-ATC-ScalCom-CBDCom-IoP), Beijing, China, 10–14 August 2015; pp. 517–520.

100. Wang, X.; Tehranipoor, M.; Plusquellic, J. Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions. In Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 9 June 2008; pp. 15–19.

101. Venugopalan, V.; Patterson, C. Surveying the Hardware Trojan Threat Landscape for the Internet-of-Things. J. Hardw. Syst. Secur. 2018, 2, 131–141. [CrossRef]

102. Kounelis, F.; Sklavos, N.; Kitsos, P. Run-Time Effect by Inserting Hardware Trojans, in Combinational Circuits. In Proceedings of the Euromicro Conference on Digital System Design (DSD), Vienna, Austria, 30 August–1 September 2017; pp. 287–290.

103. Kumar, S.; Mahapatra, K. How to Protect Your Device from Hardware Trojans. In Proceedings of the Real World IoT Security Conference, Bangalore, India, 20 June 2017

104. Bhunia, S.; Hsiao, M.; Banga, M.; Narasimhan, S. Hardware Trojan Attacks: Threat Analysis and Countermeasures. Proc. IEEE 2014, 102, 1229–1247. [CrossRef]

105. Lin, L.; Kasper, M.; Guneysu, T.; Paar, C.; Burleson, W. Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering. In Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems CHES, Lausanne, Switzerland, 6–9 September 2009; pp. 382–395

106. Chen, H.; Wang, T.; Zhang, F.; Zhao, X.; He, W.; Xu, L.; Ma, Y. Stealthy Hardware Trojan Based Algebraic Fault Analysis of HIGHT Block Cipher. Secur. Commun. Netw. 2017. [CrossRef]

107. Xiao, K.; Forte, D.; Jin, Y.; Karii, R.; Bhunia, S.; Tehranipoor, M. Hardware Trojans: Lessons Learned after One Decade of Research. J. ACM Trans. Des. Autom. Electron. Syst. (TODAES) 2016, 22, 6. [CrossRef]

108. Ranjani, R.S.; Devi, M.N. Malicious Hardware Detection and Design for Trust: An Analysis. Elektrotehniski Vestn. 2017, 84, 7–16

109. Wang, X. Hardware Trojan Attacks: Threat Analysis and Low-Cost Countermeasures through Golden-Free Detection and Secure Design. Master's Thesis, Case Western Reserve University, Cleveland, OH, USA, 2014

110. Rehman, S. U., & Gruhn, V. (2018). An effective security requirement engineering framework for cyber-physical systems. Technologies, 6(3), 65.

111. Wang, E. K., Ye, Y., Xu, X., Yiu, S. M., Hui, L. C. K., & Chow, K. P. (2010, December). Security issues and challenges for cyber physical system. In 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing (pp. 733-738). IEEE.

112. "J. A. Stankovic, I. Lee, A. Mok, and R. Rajkumar, "Opportunities and obligations for physical computing systems", IEEE Computer, 38(11):23–31, November 2005."

113. Wurm, J., Jin, Y., Liu, Y., Hu, S., Heffner, K., Rahman, F., & Tehranipoor, M. (2016). Introduction to cyber-physical system security: A cross-layer perspective. IEEE Transactions on Multi-Scale Computing Systems, 3(3), 215-227.

114. Ramachandran, M. Software security requirements management as an emerging cloud computing service. Int. J. Inf. Manag. 2016, 36, 580–590. [CrossRef]

115. Ur Rehman, S.; Khan, M.U. Security and Reliability Requirements for a Virtual Classroom. Procedia Comput. Sci. 2016, 94, 447–452. [CrossRef]

116. Shahzad, M.; Shafiq, M.Z.; Liu, A.X. A large-scale exploratory analysis of software vulnerability life cycles. In Proceedings of the 34th International Conference on Software Engineering, Zurich, Switzerland, 2–9 June 2012; pp. 771–781.

117. Salini, P.; Kanmani, S. Survey and analysis on security requirements engineering. Comput. Electr. Eng. 2012, 38, 1785–1797. [CrossRef]

118. D. DiMase, Z. A. Collier, K. Heffner, and I. Linkov, “Systems engineering framework for cyber physica l security and resilience,” Environ. Syst. Decisions, vol. 35, no. 2, pp. 291–300, 2015.

119. Ericsson, G.N. Cyber security and power system communication essential parts of a smart grid infrastructure. IEEE Trans. Power Deliv. 2010, 25, 1501–1507. [CrossRef]

120. "Oh, S.-R.; Kim, Y.-G. Security Requirements Analysis for the IoT. In Proceedings of the 2017 International Conference on Platform Technology and Service (PlatCon), Busan, South Korea, 13–15 February 2017; pp. 1–6."

121. Lund, M.S.; Solhaug, B.; Stølen, K. Model-Driven Risk Analysis: The CORAS Approach; Springer Science & Business Media: New York, NY, USA, 2010

122. Vanessa Fuhrmans, “Virus Attacks Siemens Plant-Control Systems”, TheWall Street Journal, july 22, 2010.

123. Leavitt, Neal, “Researchers Fight to Keep Implanted Medical Devices Safe from Hackers”, Computer, Volume 43, Issue 8, Pages: 11-14, August 2010.

124. "K. Chalkias, F. Baldimtsi, D. Hristu-Varsakelis and G. Stephanides, “Two Types of Key-Compromise Impersonation Attacks against One-Pass Key Establishment Protocols”, Communications in Computer and Information Science, Volume 23, Part 3, 227-238, 2009."

125. Zeng, K. Physical layer key generation in wireless networks: Challenges and opportunities. IEEE Commun. Mag. 2015, 53, 33–39. [CrossRef]

126. "Pelechrinis K., Iliofotou M., “Denial of Service Attacks in Wireless Networks: The case of Jammers”, UC Riverside Department of Computer Science and Engineering, 2006."

127. Farooq, M.U.;Waseem, M.; Khairi, A.; Mazhar, S. A critical analysis on the security concerns of internet of things (IoT). Int. J. Comput. Appl. 2015, 111. Available online: http://www.pcporoje.com/filedata/592496. pdf (accessed on 10 July 2018).

128. Khoo, B. RFID as an enabler of the internet of things: Issues of security and privacy. In Proceedings of the Internet of Things (iThings/CPSCom), 2011 International Conference on

and 4th International Conference on Cyber, Physical and Social Computing, Dalian, China, 19–22 October 2011; pp. 709–712.

129. "Jürjens, J. UMLsec: Extending UML for secure systems development. In Proceedings of the «UML» 2002—The Unified Modeling Language, Dresden, Germany, 20 September 2002; pp. 1–9. 9. Leitão, P.; Colombo, A.W.; Karnouskos, S. Industrial automation based on cyber-ph"

130. Kelly O'Connell, "CIA Report: Cyber Extortionists Attacked Foreign Power Grid, Disrupting Delivery", Internet Business Law Services, http://www.ibls.com/internet_law_news_portal_view.aspx?id=1963&s=latestnews, 2008

131. Kim, N. Y., Rathore, S., Ryu, J. H., Park, J. H., & Park, J. H. (2018). A survey on cyber physical system security for IoT: issues, challenges, threats, solutions. Journal of Information Processing Systems, 14(6), 1361-1384.

132. Y. Peng, T. Lu, J. Liu, Y. Gao, X. Guo, and F. Xie, "Cyber-physical system risk assessment," in Proceedings of 2013 9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China, 2013, pp. 442-447

133. Q. Shafi, "Cyber physical systems security: a brief survey," in Proceedings of 2012 12th International Conference on Computational Science and Its Applications (ICCSA), Salvador, Brazil, 2012, pp. 146-150.

134. W. He, J. Breier, S. Bhasin, and A. Chattopadhyay, "Bypassing parity protected cryptography using laser fault injection in cyber-physical system," in Proceedings of the 2nd ACM International Workshop on CyberPhysical System Security, Xian, China, 2016, pp. 15-21

135. P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," Journal of Cryptographic Engineering, vol. 1, no. 1, pp. 5-27, 2011

136. F. Khelil, M. Hamdi, S. Guilley, J. L. Danger, and N. Selmane, "Fault analysis attack on an FPGA AES implementation," in Proceedings of 2008 New Technologies, Mobility and Security, Tangier, Morocco, 2008, pp. 1-5.

137. R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of Things (IoT) security: current status, challenges and prospective measures," in Proceedings of 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 2015, pp. 336-341.

138. Y. Jin and D. Oliveira, “Trustworthy SoC architecture with on-demand security policies and HW-SW cooperation,” in Proceedings of the 5th Workshop on SoCs, Heterogeneous Architectures and Workloads (SHAW-5), Orlando, FL, 2015

139. D. Oliveira, N. Wetzel, M. Bucci, J. Navarro, D. Sullivan, and Y. Jin, “Hardware-software collaboration for secure coexistence with kernel extensions,” ACM SIGAPP Applied Computing Review, vol. 14, no. 3, pp. 22-35, 2014.

140. O. Al Ibrahim and S. Nair, “Cyber-physical security using system-level PUFs,” in Proceedings of 2011 7th International Wireless Communications and Mobile Computing Conference (IWCMC), Istanbul, Turkey, 2011, pp. 1672-1676

141. L. Vegh and L. Miclea, “Secure and efficient communication in cyber-physical systems through cryptography and complex event processing,” in Proceedings of 2016 International Conference on Communications (COMM), Bucharest, Romania, 2016, pp. 273-276.

142. Feng Gui, “Development of a New Client-Server Architecture for Context Aware Mobile Computing”, PHD Thesis, Florida International University, 2009.

143. Escrypt whitepaper, “Trusted Computing Technology for embedded Systems”, 2009.

144. Yusnani Mohd Yussoff, Habibah Hashim, “Trusted Wireless Sensor Node Platform”, In Proceedings of the World Congress on Engineering, Vol I,WCE 2010, June 30 - July 2, London, U.K., 2010.

145. Sindre, G.; Opdahl, A.L. Eliciting security requirements with misuse cases. Requir. Eng. 2005, 10, 34–44. [CrossRef]

146. Zafar, N.; Arnautovic, E.; Diabat, A.; Svetinovic, D. System security requirements analysis: A smart grid case study. Syst. Eng. 2014, 17, 77–88. [CrossRef]

147. Rehman, S.; Gruhn, V. Recommended Architecture for Car Parking Management System based on Cyber-Physical System. In Proceedings of the International Conference on Engineering & MIS, Monastir, Tunisia, 8–10 May 2017.

148. Yan, Q.; Yu, F.R.; Gong, Q.; Li, J. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. 2016, 18, 602–622. [CrossRef]

149. J. Shi, J. Wan, H. Yan, and H. Suo, “A survey of cyber-physical systems,” in Proc. IEEE Int. Conf. Wireless Commun. Signal Process., 2011, pp. 1–6

150. S. Khaitan and J. McCalley, “Design techniques and applications of cyberphysical systems: A survey,” IEEE Syst. J., vol. 9, no. 2, pp. 350–365, Jun. 2015

151. G. P. Hancke, “A practical relay attack on ISO 14443 proximity cards,” Univ. Cambridge Comput. Laboratory, Cambridge, U.K., Tech. Rep., vol. 59, pp. 382–385, 2005

152. M. M. Tehranipoor, U. Guin, and D. Forte, Counterfeit Integrated Circuits: Detection Avoidance. Berlin, Germany: Springer, 2015

153. G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in Proc. 44th Annu. Design Autom. Conf., 2007, pp. 9–14

154. O. Al Ibrahim and S. Nair, “Cyber-physical security using systemlevel pufs,” in Proc. 7th Int. Wireless Commun. Mobile Comput. Conf., 2011, pp. 1672–1676

155. L. Wei, C. Song, Y. Liu, J. Zhang, F. Yuan, and Q. Xu, “Boardpuf: Physical unclonable functions for printed circuit board authentication,” in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, 2015, pp. 152–158

156. B. Sunar, W. J. Martin, and D. R. Stinson, “A provably secure true random number generator with built-in tolerance to active attacks,” IEEE Trans. Comput., vol. 56, no. 1, pp. 109–119, Jan. 2007.

157. M. Stipcevi————————————c and C¸ . K. Koc¸, “True random number generators,” in Open Problems in Mathematics and Computational Science. Berlin, Germany: Springer, 2014, pp. 275–315

158. S. P. Skorobogatov, “Semi-invasive attacks: A new approach to hardware security analysis,” Ph.D. dissertation, Citeseer, NEC Res. Inst. (now NEC Labs), Princeton, NJ, USA, 2005.

159. A.Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, “Challenges for securing cyber-physical systems,” in Proc. Workshop Future Directions Cyber-Phys. Syst. Security, 2009

160. M. M. Tehranipoor, U. Guin, and D. Forte, Counterfeit Integrated Circuits: Detection Avoidance. Berlin, Germany: Springer, 2015

161. P. P. Tsang and S. W. Smith, “YASIR: A low-latency, highintegrity security retrofit for legacy SCADA systems,” in Proc. IFIP TC 11 23rd Int. Inform. Security Conf., 2008, pp. 445–459

162. Mujeeb Ahmed, C., & Zhou, J. (2020). Challenges and Opportunities in CPS Security: A Physics-based Perspective. arXiv e-prints, arXiv-2004.

163. E. Leverett and R. Wightman, "Vulnerability inheritance in programmable logic controllers," US-CERT Report, 2013. [Online]. Available: https://ics-cert.us cert.gov/content/cyber-threat-source-descriptions

164. R. Santamarta, "Here be backdoors: A journey into the secrets of industrial firmware." CoRR, 2012. [Online]. Available: https://media.blackhat.com/bh-us-12/Briefings/Santamarta/BHUS12Santamarta\BackdoorsWP.pdf

165. I.N. Fovino, A. Carcano, M. Masera, and A. Trombetta, "An experimental investigation of malware attacks on SCADA systems," International Journal of Critical Infrastructure Protection, vol. 2, no. 4, pp. 139 – 145, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874548209000419

166. T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," IEEE Transactions on Dependable and Secure Computing, vol. 2, no. 2, pp. 93–108, 4 2005.

167. S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "Accelprint: Imperfections of accelerometers make smartphones trackable," in Network and Distributed System Security Symposium (NDSS), 2014

168. C. Mujeeb Ahmed, A. Mathur, and M. Ochoa, "NoiSense: Detecting Data Integrity Attacks on Sensor Measurements using Hardware based Fingerprints," ArXiv e-prints, 12 2017

169. D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. Beyah, "Who's in control of your control system? device fingerprinting for cyberphysical systems," in NDSS, 4 2016.

170. P. Mathur and N. O. Tippenhauer, "Swat: a water treatment testbed for research and training on ics security," in 2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater), 4 2016, pp. 31–36

171. Saia, R. (2018). Internet of entities (IoE): a blockchain-based distributed paradigm to security. arXiv preprint arXiv:1808.08809.

172. Hussain, F.: Internet of everything. In: Internet of Things, pp. 1–11. Springer (2017)

173. Miraz, M. H., Ali, M., Excell, P. S., & Picking, R. (2015, September). A review on Internet of Things (IoT), Internet of everything (IoE) and Internet of nano things (IoNT). In 2015 Internet Technologies and Applications (ITA) (pp. 219-224). IEEE.

174. "Dave Evans, ""How the Internet of Everything Will Change the World,"" Cisco Blog, November 2012. [Online]. http://blogs.cisco.cominews/how-the-internet-of-everything-willchange-the-worldfor-the-better-infographicl"

175. Karthiban, M. K., & Raj, J. S. (2019). Big data analytics for developing secure internet of everything. Journal of ISMAC, 1(02), 129-136.

176. Bradley, Joseph, Christopher Reberger, Amitabh Dixit, Vishal Gupta, and James Macaulay. "Internet of Everything (IoE): Top 10 Insights from Cisco's IoE Value at Stake Analysis for the Public Sector." Economic Analysis (2013)

177. Sun, Yunchuan, Houbing Song, Antonio J. Jara, and Rongfang Bie. "Internet of things and big data analytics for smart and connected communities." IEEE access 4 (2016): 766-773.

178. "O'Leary, Daniel E. ""BIG DATA', THE 'INTERNET OF THINGS'AND THE 'INTERNET OF SIGNS."" Intelligent Systems in Accounting, Finance and Management 20, no. 1 (2013): 53-65"

179. Cárdenas, Alvaro A., Pratyusa K. Manadhata, and Sreeranga P. Rajan. "Big data analytics for security." IEEE Security & Privacy 11, no. 6 (2013): 74-76.

180. Gahi, Youssef, Mouhcine Guennoun, and Hussein T. Mouftah. "Big data analytics: Security and privacy challenges." In 2016 IEEE Symposium on Computers and Communication (ISCC), pp. 952-957. IEEE, 2016.

181. Riggins, Frederick J., and Samuel Fosso Wamba. "Research directions on the adoption, usage, and impact of the internet of things through the use of big data analytics." In 2015 48th Hawaii International Conference on System Sciences, pp. 1531-1540. IEEE, 2015.

182. "Laney, Doug. ""3D data management: Controlling data volume, velocity and variety."" META group Laney, Doug. ""3D data management: Controlling data volume, velocity and variety."" META group research notes 6, no. 70 (2001): 1. "

183. "O'Leary, Daniel E. ""BIG DATA', THE 'INTERNET OF THINGS'AND THE 'INTERNET OF SIGNS."" Intelligent Systems in Accounting, Finance and Management 20, no. 1 (2013): 53-65."

184. Manogaran, Gunasekaran, Ramachandran Varatharajan, Daphne Lopez, Priyan Malarvizhi Kumar, Revathi Sundarasekar, and Chandu Thota. "A new architecture of Internet of Things and big data ecosystem for secured smart healthcare monitoring and alerting system." Future Generation Computer Systems 82 (2018): 375-387.

185. Naqishbandi, Tawseef, C. Imthyaz Sheriff, and S. Qazi. "Big data, CEP and IoT: redefining holistic healthcare information systems and analytics." Int J Eng Res and Technol 4, no. 1 (2015): 1-6

186. Khorshed, Md Tanzim, Neeraj Anand Sharma, Kunal Kumar, Mishal Prasad, ABM Shawkat Ali, and Yang Xiang. "Integrating Internet-of-Things with the power of Cloud Computing and the intelligence of Big Data analytics—A three layered approach." In 2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), pp. 1-8. IEEE, 2015.

187. Zimmermann, L., Scholz, A., Tahoori, M. B., Sikora, A., & Aghassi-Hagmann, J. (2020, September). Hardware-intrinsic security with printed electronics for identification of ioe devices. In 2020 European Conference on Circuit Theory and Design (ECCTD) (pp. 1-4). IEEE.

188. "S. V. Vandebroek, “1.2 three pillars enabling the internet of everything: Smart everyday objects, information-centric networks, and automated real-time insights,” in 2016 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2016, pp. 14–20."

189. B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, “Silicon physical random functions,” in Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002, pp. 148–160

190. "H. Ma, Y. Gao, O. Kavehei, and D. C. Ranasinghe, “A puf sensor: Securing physical measurements,” in 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2017, pp. 648–653"

191. "R. Maes, physically unclonable functions: Constructions, properties and applications. Springer Science & Business Media, 2013"

192. Mohanty, S. P., Yanambaka, V. P., Kougianos, E., & Puthal, D. (2020). PUFchain: A hardware-assisted blockchain for sustainable simultaneous device and data security in the internet of everything (IoE). IEEE Consumer Electronics Magazine, 9(2), 8-16.

193. D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, “Everything you wanted to know about the blockchain: Its promise, components, processes, and problems,” IEEE Consum. Electron. Mag., vol. 7, no. 4, pp. 6–14, Jul. 2018

194. S. P. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal, “PUFchain: Hardware-assisted blockchain for sustainable simultaneous device and data security in the internet of everything (IoE),” Sep. 2019. [Online]. Available: https://arxiv.org/abs/1909.06496

195. H. Lu, K. Huang, M. Azimi, and L. Guo, “Blockchain technology in the oil and gas industry: A review of applications, opportunities, challenges, and risks,” IEEE Access, vol. 7, pp. 41 426–41 444, 2019

196. S. P. Mohanty, U. Choppali, and E. Kougianos, “Everything you wanted to know about smart cities,” IEEE Consum. Electron. Mag., vol. 5, no. 3, pp. 60–70, Jul. 2016

197. D. Evans, “The internet of everything: How more relevant and valuable connections will change the world,” Cisco Internet Business Solutions Group, Cisco Systems Inc., San Jose, CA, USA, 2012

198. H. Wu and C. Tsai, “Toward blockchains for healthcare systems: Applying the bilinear pairing technology to ensure privacy protection and accuracy in data sharing,” IEEE Consum. Electron. Mag, vol. 7, no. 4, pp. 65–71, Jul. 2018.

199. Wei, L., Wu, J., Long, C., & Lin, Y. B. (2019). The convergence of ioe and blockchain: Security challenges. IT Professional, 21(5), 26-32.

200. F. A. Alaba et al., “Internet of things security: A survey,” J. Netw. Comput. Appl., vol. 88, pp. 10–28, 2017

201. T. M. Fernandez-Caram es and P. Fraga-Lamas, “A review on the use of blockchain for the internet of things,” IEEE Access, vol. 6, pp. 32979–33001, 2018

202. M. T. Hammi et al., “Bubbles of trust: A decentralized blockchain-based authentication system for IoT,” Comput. Secur., vol. 78, pp. 126–142, 2018

203. R. Pass and E. Shi, “Hybrid consensus: Efficient consensus in the permissionless model,” in Proc. 31st Int. Symp. Distrib. Comput., 2017, pp. 39:1–39:16.

204. V. P. Yanambaka, S. P. Mohanty, and E. Kougianos, “Making use of manufacturing process variations: A dopingless transistor based-PUF for hardware-assisted security,” IEEE Trans. Semicond. Manuf., vol. 31, no. 2, pp. 285–294, May 2018

205. V. P. Yanambaka, S. P. Mohanty, E. Kougianos, and D. Puthal, “PMsec: Physical unclonable functionbased robust and lightweight authentication in the internet of medical things,” IEEE Trans. Consum. Electron., vol. 65, no. 3, pp. 388–397, Aug. 2019

206. Desai, S., Alhadad, R., Chilamkurti, N., & Mahmood, A. (2019). A survey of privacy preserving schemes in IoE enabled smart grid advanced metering infrastructure. Cluster Computing, 22(1), 43-69.

207. Buchmann, E., Bohm, K., Burghardt, T., Kessler, S.: Re-identification of smart meter data. Pers. Ubiquitous Comput. 17(4), 653–662 (2013)

208. Tan, S., De, D., Song, W., Yang, J., Das, S.: Survey of security advances in smart grid: a data driven approach. IEEE Commun. Surv. Tutor. 19(1), 397–422 (2017

209. Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid-the new and improved power grid: a survey. IEEE Commun. Surv. Tutor. 14(4), 944–980 (2012)

210. Gharavi, H., Ghafurian, R.: Smart grid: the electric energy system of the future [scanning the issue]. Proc. IEEE 99(6), 917–921 (2011)

211. UCA.: Security Profile For Advanced Metering Infrastructure. Utility Communications Architecture International Users Group (2010)

212. FIPS.: Standards for Security Categorization of Federal Information and Information Systems (2004)

213. Kang, J.: Information privacy in cyberspace transactions. Stanf. Law Rev. 50, 1193–1294 (1998)

214. Smith, H.J.: Managing privacy: information technology and corporate America. UNC Press Books, Chapel Hill (1994

215. Saputro, N., Akkaya, K.: On preserving user privacy in smart grid advanced metering infrastructure applications. Secur. Commun. Netw. 7(1), 206–220 (2014)

216. Kalogridis, G., Efthymiou, C., Denic, S.Z., Lewis, T.A., Cepeda, R.: Privacy for smart meters: towards undetectable appliance load signatures. In: 2010 First IEEE International Conference on Smart Grid Communications, pp. 232–237 (2010)

217. "sharma, A., Ojha, V.: Implementation of cryptography for privacy preserving data mining. Int. J. Database Manag. Syst. 2(3), 57–65 (2010)"

218. Sun, X., Wang, H., Li, J., Zhang, Y.: Satisfying privacy requirements before data anonymization. Comput. J. 55(4), 422–437 (2012). https://doi.org/10.1093/comjnl/bxr028

219. Simmons, G.J.: Symmetric and asymmetric encryption. ACM Computing Surveys (CSUR) 11(4), 305–330 (1979)

220. "Kranz, M.: Industrial applications are the juicy part of the internet of things. LSE Business Review (2017)"

221. Dong, C., He, G., Liu, X., Yang, Y., & Guo, W. (2019). A multi-layer hardware trojan protection framework for IoT chips. IEEE Access, 7, 23628-23639.

222. Lesjak, C., Hein, D., & Winter, J. (2015, November). Hardware-security technologies for industrial IoT: TrustZone and security controller. In IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society (pp. 002589-002595). IEEE.

223. "Bo Yu and Bin Xiao. 2006. Detecting selective forwarding attacks in wireless sensor networks. In Parallel and distributed processing symposium, 2006. IPDPS 2006. 20th international. IEEE, 8–pp."

224. Mathias Claes, Vincent van der Leest, and An Braeken. 2011. Comparison of SRAM and FF PUF in 65nm technology. In Nordic Conference on Secure IT Systems. Springer, 47–64

225. Babaei, A., & Schiele, G. (2019). Physical unclonable functions in the internet of things: State of the art and open challenges. Sensors, 19(14), 3208.

226. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical one-way functions. Science 2002, 297, 2026–2030. [CrossRef]

227. Lichen Zhang, Zhipeng Cai, and Xiaoming Wang. 2016. Fakemask: A novel privacy preserving approach for smartphones. IEEE Transactions on Network and Service Management 13, 2 (2016), 335–348.

228. Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. 2002. Physical one-way functions. Science 297, 5589 (2002)

229. Pim Tuyls and Lejla Batina. 2006. RFID-tags for anti-counterfeiting. In CryptographersâĂŹ Track at the RSA Conference. Springer, 115–131

230. Bertrand Cambou and Marius Orlowski. 2016. PUF designed with Resistive RAM and Ternary States. In Proceedings of the 11th Annual Cyber and Information Security Research Conference. ACM, 1.

231. Clemens Helfmeier, Christian Boit, Dmitry Nedospasov, Shahin Tajik, and Jean-Pierre Seifert. 2014. Physical vulnerabilities of physically unclonable functions. In Proceedings of the conference on Design, Automation & Test in Europe. European Design and Automation Association, 350

232. Yuan-Hao Chang, Jen-Wei Hsieh, and Tei-Wei Kuo. 2007. Endurance enhancement of flash-memory storage systems: an efficient static wear leveling design. In Proceedings of the 44th annual Design Automation Conference. ACM, 212–217.

233. Hiroyuki Akinaga and Hisashi Shima. 2010. Resistive random access memory (ReRAM) based on metal oxides. Proc. IEEE 98, 12 (2010), 2237–2251

234. Saied Tehrani. 2006. Status and outlook of MRAM memory technology. In Electron Devices Meeting, 2006. IEDM'06. International. IEEE, 1–4.

235. Hyunho Kang, Yohei Hori, Toshihiro Katashita, Manabu Hagiwara, and Keiichi Iwamura. 2014. Performance analysis for puf data using fuzzy extractor. In Ubiquitous Information Technologies and Applications. Springer, 277–284

236. Bin Chen, Tanya Ignatenko, Frans MJ Willems, Roel Maes, Erik van der Sluis, and Georgios Selimis. 2017. High-Rate Error Correction Schemes for SRAM-PUFs based on Polar Codes. arXiv preprint arXiv:1701.07320 (2017).

237. Ashwija Korenda, Fatemeh Afghah, Bertrand Cambou, and Christopher Philabaum. 2019. A Proof-of-Concept SRAM-based Physically Unclonable Function (PUF) Key Generation Mechanism for IoT Devices. In IEEE SECON Workshop on Security, Trust, and Privacy in Emerging Cyber-Physical Systems.

238. Halak, B., Zwolinski, M., & Mispan, M. S. (2016, October). Overview of PUF-based hardware security solutions for the Internet of Things. In 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS) (pp. 1-4). IEEE.

239. "C. Herder, M. D. Yu, F. Koushanfar, and S.Devadas, ""Physical Unclonable Functions and Applications: A Tutorial,"" Proceedings of the IEEE, vol. 102, pp. 1126-1141, 2014"

240. M. A. Alam, H. Kufluoglu, D. Varghese, and S. Mahapatra, "A comprehensive model for PMOS NBTI degradation: Recent progress," Microelectronics Reliability, vol. 47, 2007

241. U. Rührmair, and J. Sölter, "PUF modeling attacks: An introduction and overview," in 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2014, pp. 1-6.

242. Aman, M.N.; Chua, K.C.; Sikdar, B. Mutual authentication in iot systems using physical unclonable functions. IEEE Internet Things J. 2017, 4, 1327–1340. [CrossRef]

243. Standaert, F.-X. Introduction to side-channel attacks. In Secure Integrated Circuits and Systems; Springer: Boston, MA, USA, 2010; pp. 27–42.

244. Aman, M.N.; Chua, K.C.; Sikdar, B. Position paper: Physical unclonable functions for iot security. In Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security, Xi'an, China, 30 May–June 2016; pp. 10–13.

245. Tajik, S.; Dietz, E.; Frohmann, S.; Seifert, J.-P.; Nedospasov, D.; Helfmeier, C.; Boit, C.; Dittrich, H. Physical characterization of arbiter pufs. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Busan, Korea, 23–26 September 2014; pp. 493–509.

246. Becker, G.T.; Kumar, R. Active and passive side-channel attacks on delay based puf designs. IACR Cryptol. ePrint Arch. 2014, 2014, 287

247. Mahmoud, A.; Rührmair, U.; Majzoobi, M.; Koushanfar, F. Combined modeling and side channel attacks on strong pufs. IACR Cryptol. ePrint Arch. 2013, 2013, 632.

248. Delvaux, J.; Verbauwhede, I. Fault injection modeling attacks on 65 nm arbiter and ro sum pufs via environmental changes. IEEE Trans. Circuits Syst. I Regul. Pap. 2014, 61, 1701–1713. [CrossRef]

249. Rührmair, U.; Sehnke, F.; Sölter, J.; Dror, G.; Devadas, S.; Schmidhuber, J. Modeling attacks on physical unclonable functions. In Proceedings of the 17th ACM conference on Computer and Communications Security, Chicago, USA, 4–8 October 2010; pp. 237–249.

250. Merli, D. Attacking and Protecting Ring Oscillator Physical Unclonable Functions and Code-Offset Fuzzy Extractors. Ph.D. Thesis, Technische Universität München, München, Germany, 2014

251. Ganji, F. On the Learnability of Physically Unclonable Functions. Springer: Berlin/Heidelberg, Germany, 2018

252. Gassend, B.; Clarke, D.; van Dijk, M.; Devadas, S. Silicon physical random functions. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 18–22 November 2002; pp. 148–160.

253. Idriss, T.; Idriss, H.; Bayoumi, M. A puf-based paradigm for iot security. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; pp. 700–705.

254. Babaei, A.; Schiele, G. Spatial reconfigurable physical unclonable functions for the internet of things. In Proceedings of the International Conference on Security, Privacy and

Anonymity in Computation, Communication and Storage, Guangzhou, China, 12–15 December 2017; pp. 312–321

255. Bhargava, M.; Cakir, C.; Mai, K. Reliability enhancement of bi-stable pufs in 65nm bulk cmos. In Proceedings of the 2012 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), San Francisco, CA, USA, 3–4 Jun 2012; pp. 25–30

256. Intrinsic ID BV. Available online: https://www.intrinsic-id.com/ (accessed on 21 July 2019)

257. Barbareschi, M.; Bagnasco, P.; Mazzeo, A. Authenticating iot devices with physically unclonable functions models. In Proceedings of the 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, Poland, 4–6 November 2015; pp. 563–567.

258. "Cherkaoui, A.; Bossuet, L.; Seitz, L.; Selander, G.; Borgaonkar, R. New paradigms for access control in constrained environments. In Proceedings of the 2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), Montpellier, France, 26–28 May 2014; pp. 1–4."

259. Bayon, P.; Bossuet, L.; Aubert, A.; Fischer, V. Electromagnetic analysis on ring oscillator-based true random number generators. In Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, 19–23 May 2013; pp. 1954–1957.

260. Marchand, C.; Bossuet, L.; Mureddu, U.; Bochard, N.; Cherkaoui, A.; Fischer, V. Implementation and characterization of a physical unclonable function for iot: A case study with the tero-puf. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 2018, 37, 97–109. [CrossRef]

261. Xu, T.; Wendt, J.B.; Potkonjak, M. Security of iot systems: Design challenges and opportunities. In Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 2–6 November 2014; pp. 417–42

262. Yanambaka, V.P.; Mohanty, S.P.; Kougianos, E. Novel finfet based physical unclonable functions for efficient security integration in the iot. In Proceedings of the 2016 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS), Gwalior, India, 19–21 December 2016; pp. 172–177.

263. Delvaux, J.; Gu, D.; Schellekens, D.; Verbauwhede, I. Helper data algorithms for puf-based key generation: Overview and analysis. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 2015, 34, 889–902. [CrossRef]

264. Aman, M.N.; Chua, K.C.; Sikdar, B. Mutual authentication in iot systems using physical unclonable functions. IEEE Internet Things J. 2017, 4, 1327–1340. [CrossRef]

265. "Gao, Y.; Li, G.; Ma, H.; Al-Sarawi, S.F.; Kavehei, O.; Abbott, D.; Ranasinghe, D.C. Obfuscated challenge-response: A secure lightweight authentication mechanism for puf-based pervasive devices. In Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), Sydney, NSW, Australia, 14–18 March 2016; pp. 1–6"

266. Yu, M.-D.; Hiller, M.; Delvaux, J.; Sowell, R.; Devadas, S.; Verbauwhede, I. A lockdown technique to prevent machine learning on pufs for lightweight authentication. IEEE Trans. Multi-Scale Comput. Syst. 2016, 2, 146–159. [CrossRef]

267. Babaei, A.; Schiele, G. Spatial reconfigurable physical unclonable functions for the internet of things. In Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Guangzhou, China, 12–15 December 2017; pp. 312–321.

268. Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future generation computer systems, 82, 395-411.

269. " A.M. Antonopoulos, Mastering Bitcoin: Unlocking Digital Crypto-Currencies, first ed., O'Reilly Media, Inc., 2014"

270. A.Friese, J. Heuer, N. Kong, Challenges from the Identities of Things: Introduction of the Identities of Things discussion group within Kantara initiative, in: 2014 IEEE World Forum on Internet of Things (WF-IoT), 2014, pp. 1–4. http://dx.doi.org/10.1109/WF-IoT.2014.6803106

271. Sidhu, S., Mohd, B. J., & Hayajneh, T. (2019). Hardware security in IoT devices with emphasis on hardware Trojans. Journal of Sensor and Actuator Networks, 8(3), 42.

272. Allerin. Available online: https://www.allerin.com/blog/authentication-and-device-identification-in-iotsecurity (accessed on 9 April 2019)

273. Utimaco. Available online: https://hsm.utimaco.com/solutions/applications/key-injection/ (accessed on 15 April 2019).

274. Bajikar, S. Trusted Platform Module (TPM) Based Security on Notebook PCs—White Paper; Mobile Platforms Group Intel Corporation: Santa Clara, CA, USA, 2002.

275. Synopsys. Available online: https://www.synopsys.com/designware-ip/technical-bulletin/understandinghardware-roots-of-trust-2017q4.html (accessed on 9 April 2019)

276. Mattoon, D. DICE: Foundational Trust for IoT. In Proceedings of the Microsoft Flash Memory Summit, Santa Clara, CA, USA, 8–10 August 2017.

277. Frustaci, M., Pace, P., Aloi, G., & Fortino, G. (2017). Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet of things journal, 5(4), 2483-2495.

278. "Y. Zou, J. Zhu, X. Wang, and L. Hanzo, “A survey on wireless security: Technical challenges, recent advances, and future trends,” Proc. IEEE, vol. 104, no. 9, pp. 1727–1765, Sep. 2016"

279. J. Ahamed and A. V. Rajan, “Internet of Things (IoT): Application systems and security vulnerabilities,” in Proc. 5th Int. Conf. Electron. Devices Syst. Appl. (ICEDSA), Ras al-Khaimah, UAE, 2016, pp. 1–5

280. The Internet of Things (IoT): An Overview. Accessed: Mar. 2017. [Online]. Available: https://www.internetsociety.org/doc/iot-overview

281. CVSS. Accessed: Mar. 2017. [Online]. Available: https://en.wikipedia.org/wiki/CVSS

282. X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Gener. Comput. Syst. (2017). http://dx.doi.org/10.1016/j. future.2017.08.020.

283. Xi, X.; Zhuang, H.; Sun, N.; Orshansky, M. Strong subthreshold current array puf with 265 challenge-response pairs resilient to machine learning attacks in 130 nm cmos. In Proceedings of the 2017 Symposium on VLSI Circuits, Kyoto, Japan, 5–8 June 2017; pp. C268–C269