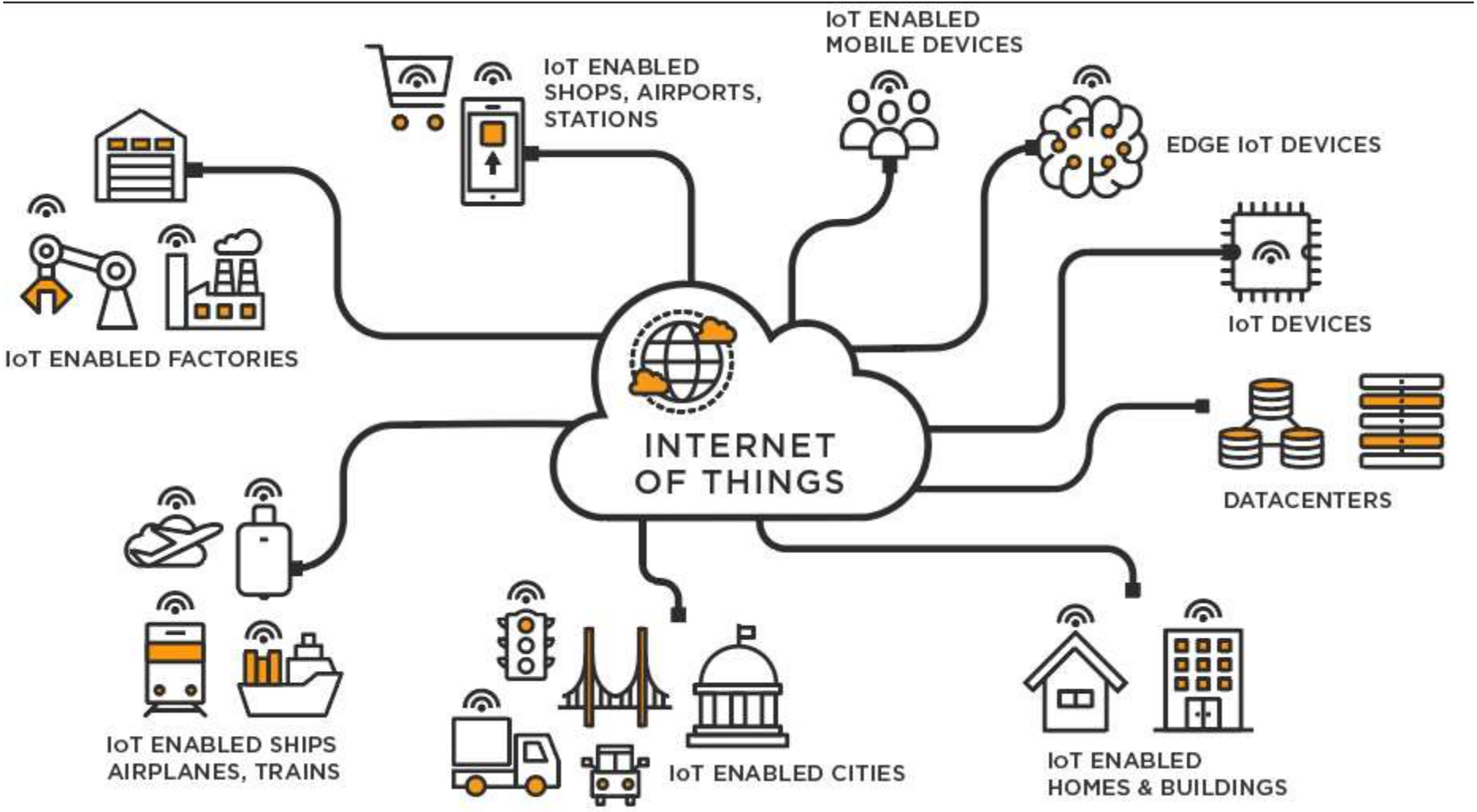
Security, Privacy, and Ethical Challenges in IoT

L10 20th June 2025

Presented by:
Parul Kukrety



Attack zones in IoT

- Devices
- Channels of communication
- Softwares and applications

Security Challenges in IoT

- Weak Authentication
- Lack of Encryption
- Unsigned Firmware
- Scalability Issues
- Complex Supply Chains

Privacy Concerns in IoT

- Data Collection
- Unauthorized Access
- Lack of Transparency
- Private data sharing

Real world examples of IoT Vulnerabilities

Mirai Botnet 2016

DDoS attacks that Exploited default credentials in cameras and DVRs

Chrysler Jeep Hack 2015

Remotely controlled vehicles via loopholes wireless connectivity

Medical Devices 2017

Cardiac device vulnerabilities

Baby monitor hacking

Compromised devices

Top IoT Device Vulnerabilities

Attacks on IoT

- Physical attacks
- DoS and DDoS attacks
- Botnet attacks
- MitM Attacks
- Malware attacks
- Credential attacks
- Firmware attacks
- Side-channel attacks
- Encryption attacks

Physical attacks

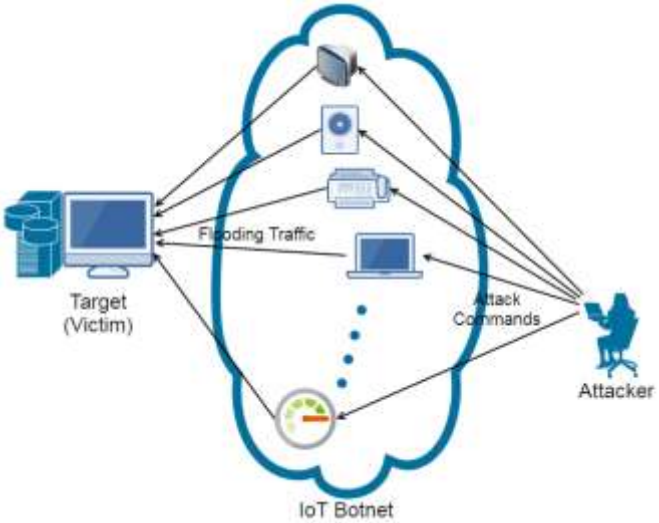
"Target the hardware of IoT devices"

Attackers manipulate devices, sensors, or gain unauthorized physical access to the devices

- Zero-day attacks: targets previously unknown vulnerabilities.
- Eavesdropping: steals sensitive information over communication channels between devices.
- Data injection: inject malicious code on devices
- Replay attacks: modifies authenticated packet.

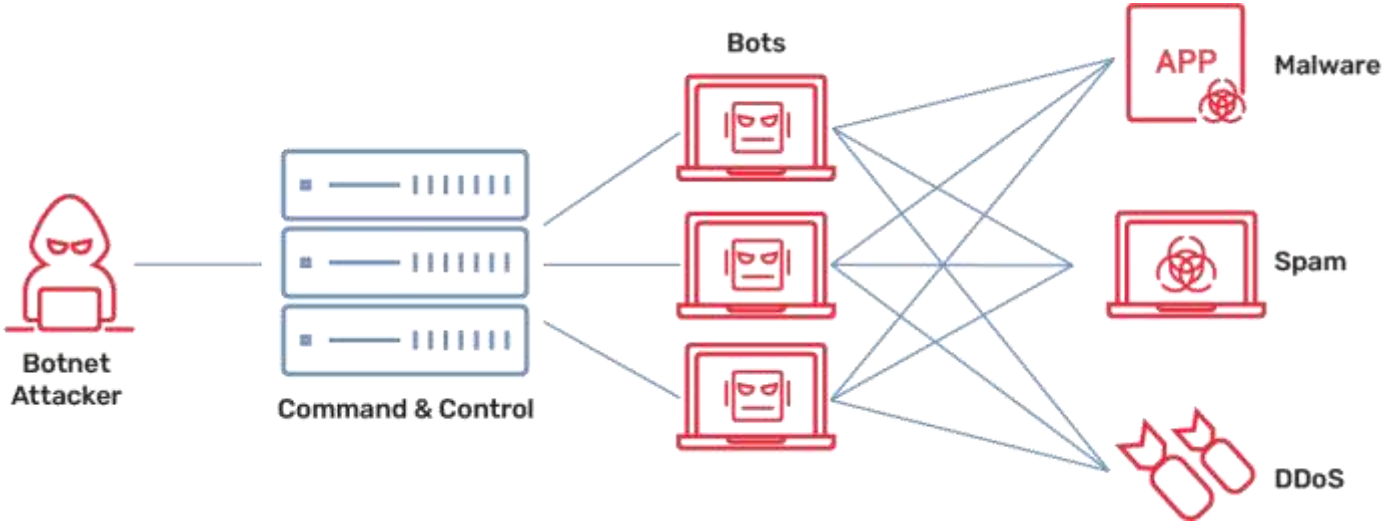
DoS and DDoS attacks

"Traffic flooding on IoT systems, rendering them unresponsive or disrupting crucial information."



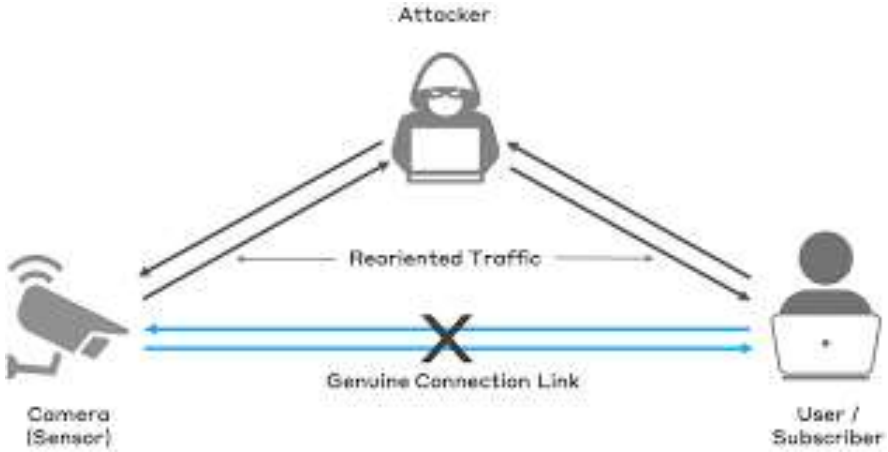
Botnet attacks

"Botnets created to hijack vulnerable devices"

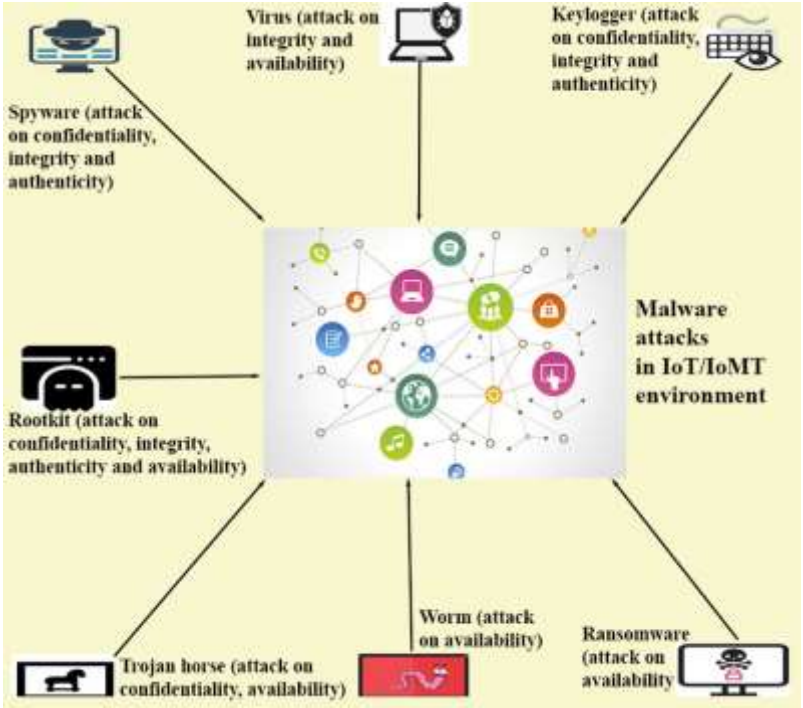


Man in the Middle attacks

“ Attackers eavesdrop on sensitive data or inject malicious content”



Malware attacks



Credential attacks

“ Attackers exploit weak or default passwords to gain unauthorized access to the IoT devices”

Encryption attacks

“Attackers modify and install their algorithms and gain control over your device if the user’s IoT device is not encrypted.”

Signs IoT device has been attacked

- Device behaves abnormally
- Surges in network traffic
- Device or network sluggishness
- Unfamiliar emails or messages
- Unusual account activity

Protect IoT Devices from vulnerabilities

- Manufacturers
- Users
- Organizations

Enhancing Privacy in IoT

- Minimize data collection
- Transparent policies
- Secure data storage
- User control
- Compliance with regulations

IoT Security Basics

“Protecting IoT devices and networks from unauthorized access, data breaches, and attacks.“

Types of IoT Security

- Network
- Device
- Data

How to protect IoT devices?

- DNS filtering
- Encryption
- Device authentication
- Security of credentials

Best Practices for IoT Security

Unique Device Identities

Issue digital certificates to each device
Prevents weak auth.

Secure Firmware Updates

Code signing and OTA updates for verification

Encryption Everywhere

For Data in transit and at rest

Lifecycle Mngt.

Manage certificates

Encryption in IoT

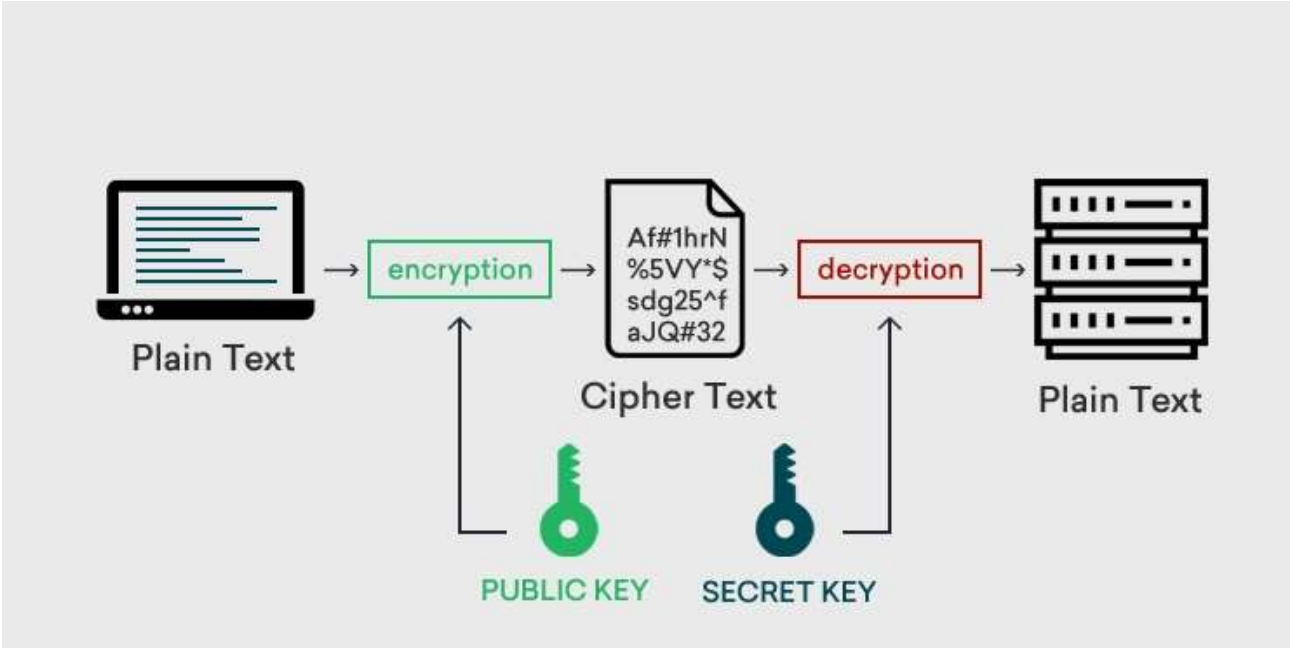
Encrypts data transmitted between devices and servers (e.g., smart lock to cloud).

- Example: TLS/SSL for secure communication in smart home devices.

Challenges:

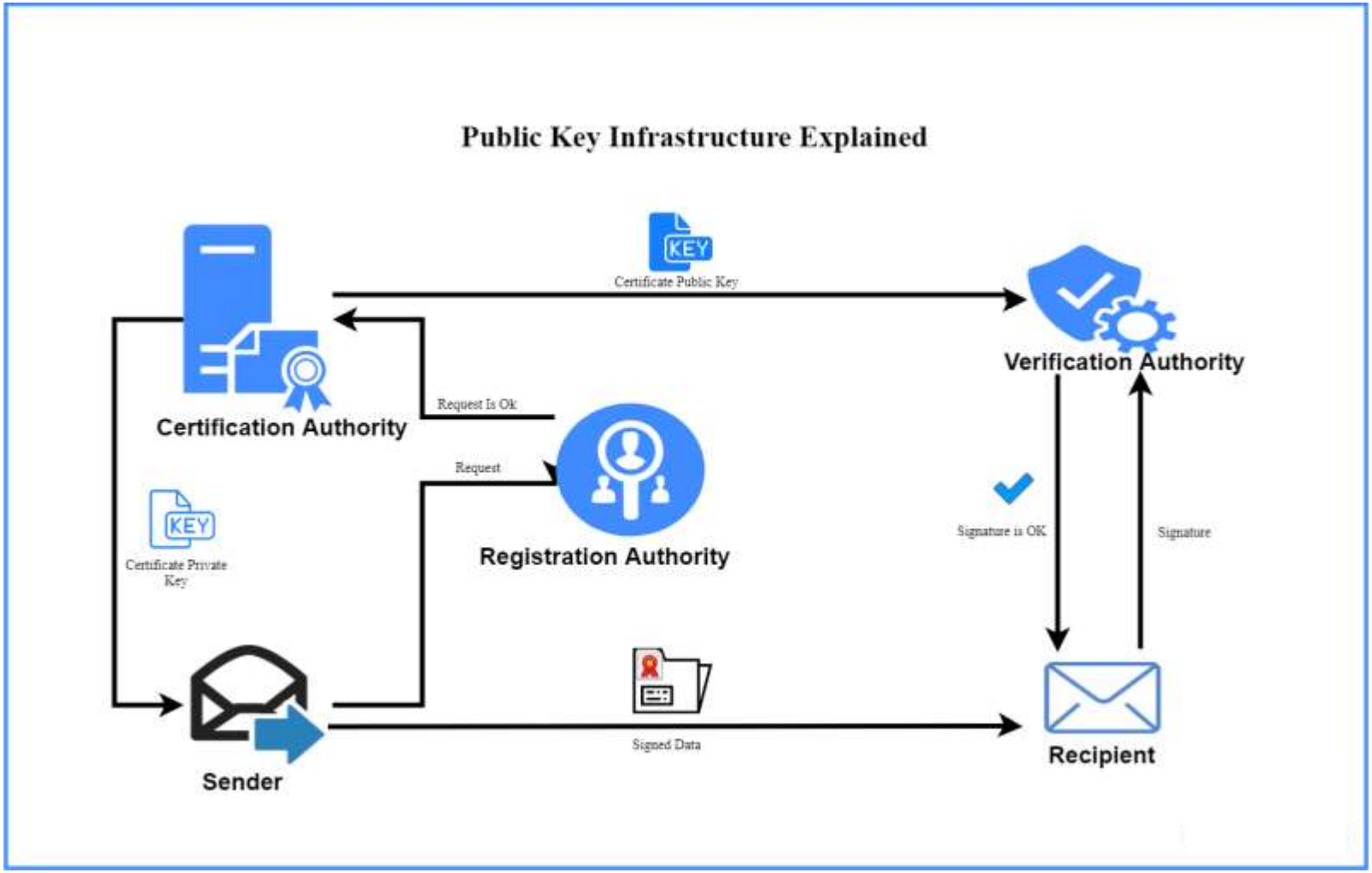
- Resource constraints
- Key management for millions of devices.

Encryption



PKI as a Solution for IoT Security

- **Unique Identities**: Each device gets a digital certificate for secure authentication.
- **Scalability**: Single trusted Certificate Authority (CA) issues millions of certificates.
- **Minimal Footprint**: Asymmetric keys suit low-power devices.
- **Crypto-Agility**: Supports updates to counter evolving threats



Encryption methods for IoT Devices

1. RSA
2. AES
3. Module lattice based digital signature algorithm
4. Two fish encryption
5. Elliptic curve cryptography

RSA

Rivest-Shamir-Adleman Algorithm in IoT security (1977)

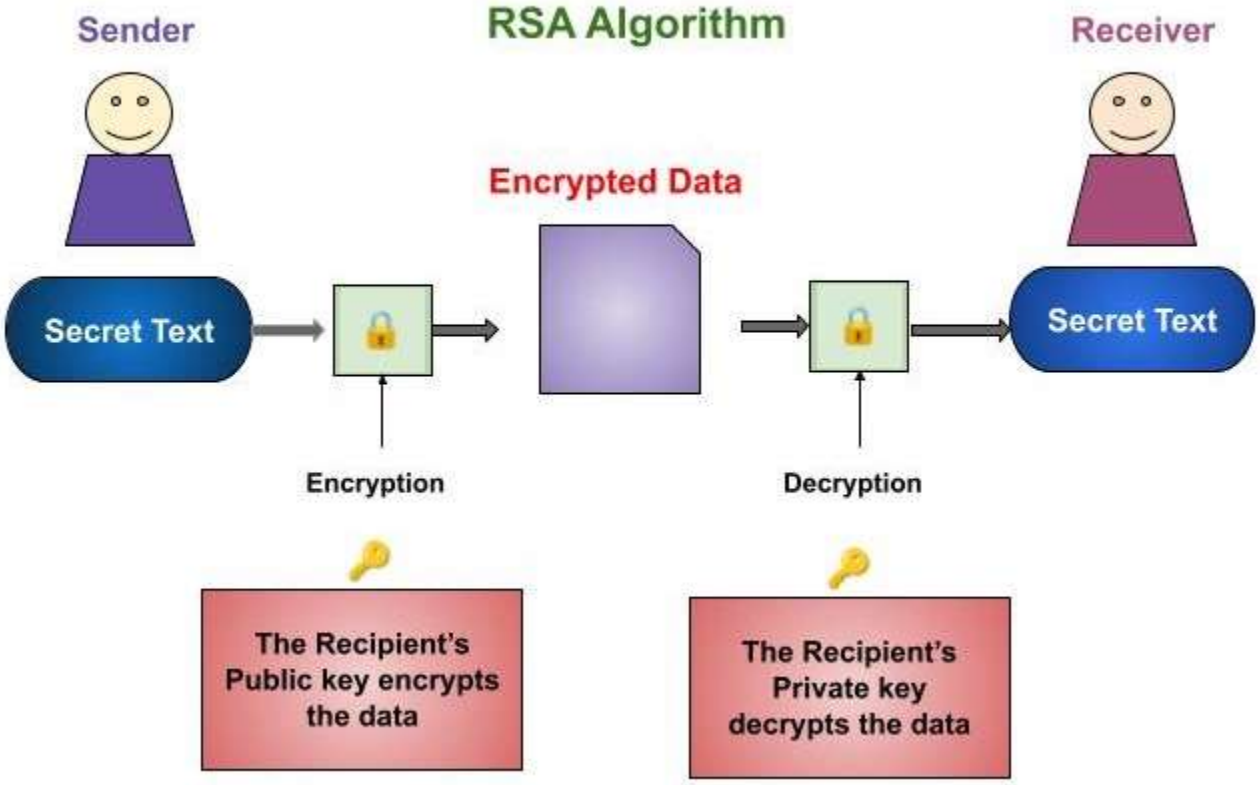
Public key cryptosystem

Based on factoring of large integers

Uses key pair-public key for encryption/signing, private key for decryption/verification

Working of RSA

1. Key generation
2. Encryption
3. Decryption



Working of RSA: Key Generation

- Choose two large primes p and q, compute $N = p \cdot q$ and $\phi(N) = (p-1)(q-1)$
- Select public exponent **e**, $gcd(e, \phi(N)) = 1$, compute private exponent **d**

  where $e \cdot d \equiv 1 \mod \phi(N)$

- Public key: (N,e) , private key: (N,d).

Choose two prime numbers: p = 3, q = 11

Calculate n = p * q = 33

Calculate Euler's Totient Function: Φ(33) = Φ(3) * Φ(11) = 2 * 10 = 20

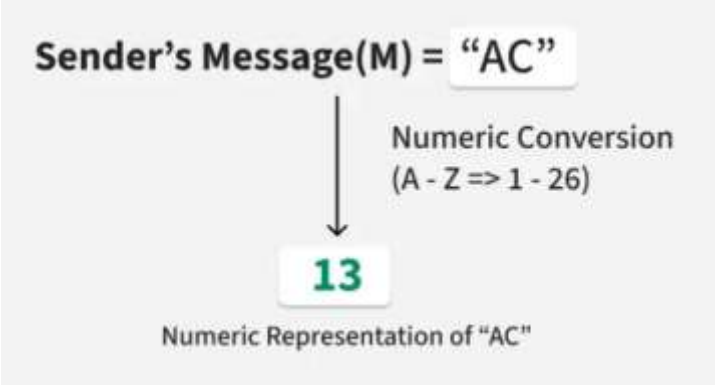
Choose e = 7 , which is co-prime with 20

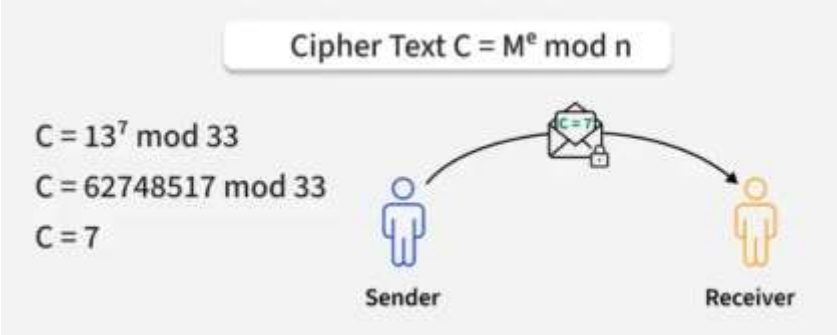
Calculate d as the multiplicative inverse of e (7), so d = 3

Public Key = (n, e) = (33, 7) Private Key = (n, d) = (33, 3)

Working of RSA: Encryption

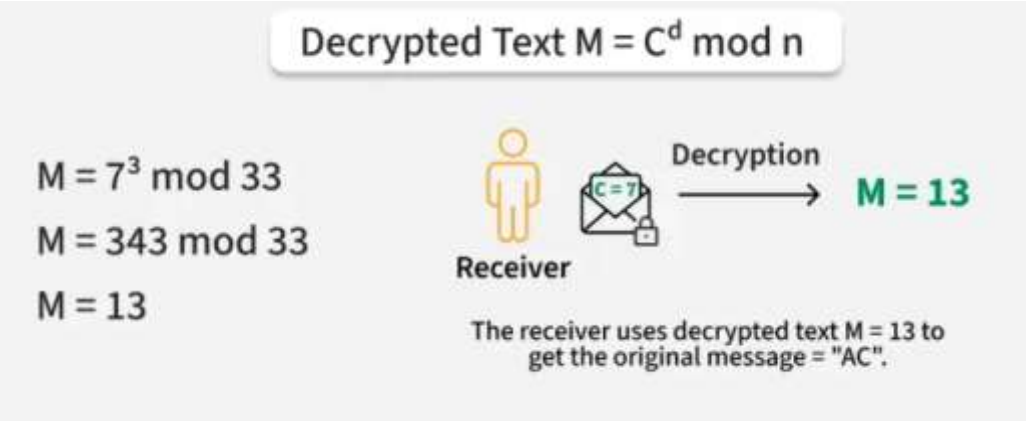
- Encryption: $C = M^e \mod N$ (message M to ciphertext C).
- Signing: $S = M^d \mod N$ (creates signature S).





Working of RSA: Decryption

- Decryption: $M = C^d \mod N$ (recovers M).
- Verification: $M = S^e \mod N$ (checks signature validity)



RSA

Advantages:

- Interoperable with existing IoT protocols (e.g., TLS, X.509 certificates).
- Strong security with large keys (2048–4096 bits).
- Fast verification for signatures, ideal for IoT hubs.

Limitations:

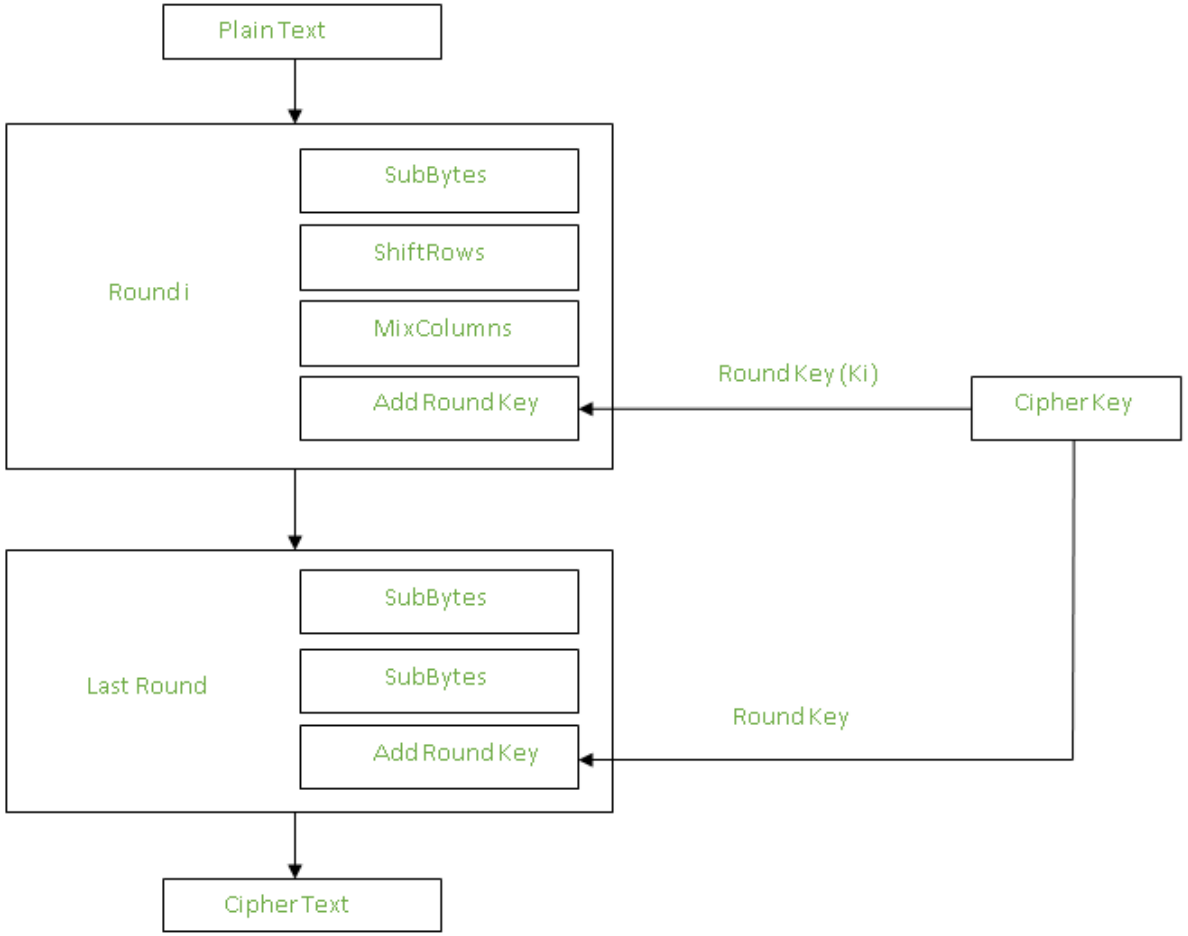
- Computationally intensive for low-power IoT devices (e.g., signing takes ~100ms on constrained CPUs).
- Large key sizes (2048 bits = 256 bytes) strain memory.
- Vulnerable to quantum attacks (Shor's algorithm breaks RSA)

AES

“Advanced Encryption Standard (AES), standardized by NIST in 2001, is a symmetric block cipher.”

“Uses fixed 128-bit blocks and key sizes of 128, 192, or 256 bits for encryption/decryption”

“Fast and efficient”



AES working

Key Setup:

- Generate a symmetric key (128, 192, or 256 bits) shared between sender and receiver.
- Expand key into round keys using the AES key schedule (e.g., 10 rounds for AES-128).

Encryption:

- Input: 128-bit plaintext block, symmetric key.
- Process: Apply 10–14 rounds (depending on key size)
- Output: 128-bit ciphertext block.

Decryption:

- Reverse the encryption process using the same key, applying inverse operations.
- Output: Original plaintext.

Module lattice based digital signature algorithm

“ML-DSA makes use of digital signature rather than written signature”

“Given by NIST”

“DS represented as string of bits commuted using a set of rules to verify the identity of the signatory and integrity of data”

“Relevant for IoT security due its efficiency in resource-constrained environment”

ML-DSA (2024)

- Lattice based DSA derived from crystals-Dilithium
- Uses the Module Learning With Errors (Module-LWE) and Module Short Integer Solution (Module-SIS) problems for security
- In IoT, it verifies that messages (e.g., sensor data, firmware updates) come from trusted sources and haven't been tampered with
- IoT devices need future-proofing against quantum threats.
- Provides efficiency, scalability, crypto-agility

Working of ML-DSA

1. Key generation (private and public key seed of 32 byte)
2. Signature generation
3. Signature verification

Twofish Cryptography

- Symmetrical encryption
- Single key to encrypt and decrypt
- But, requires significant storage (plaintext becomes sizable when converted to ciphertext).

ECC

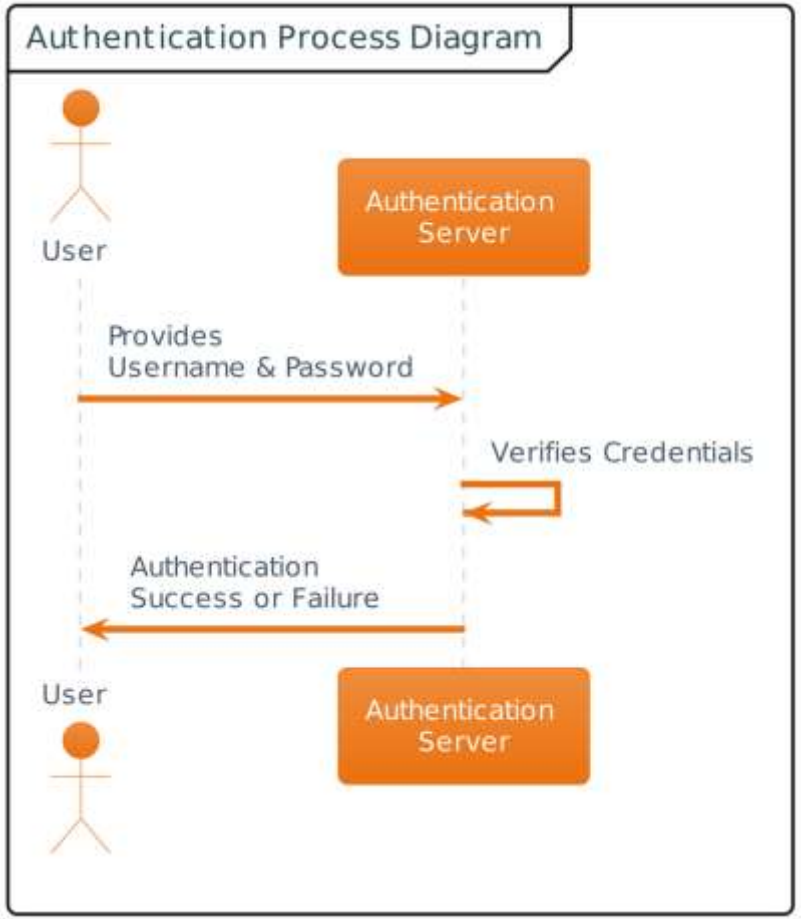
- Common for IoT
- Generates short cryptographic keys
- Slow to encrypt and decrypt but not computationally intensive.
- Encrypts data by applying the mathematical properties of elliptic curves to plaintext, transforming it into a large, random number
- Helps in securing low power IoT devices without drastically affecting resource usage.

Authentication in IoT

Verifying the identity of devices, users, or servers.

How it works?

Passwords, biometrics, digital certificates or tokens.



Authentication

- **One-way authentication:** in the case where two parties wish to communicate with each other, only one party will authenticate itself to the other, while the other party will not be authenticated.
- **Two-way authentication:** is also referred to as mutual authentication, in which both entities authenticate each other.
- **Three-way authentication:** is where the central authority authenticates the two parties and helps them to authenticate each other.
- **Distributed**: using a distributed straight authentication method between the parties to the communication.
- **Centralized**: using a centralized server or a trusted third party to distribute and manage the authentication certificates used.

Authentication of devices

Each IoT machine needs unique identity

To prevent malicious actors from gaining control of system.

Achieved by binding identity to crypto key, unique per device

- By TPM (trusted platform module)
- CA (certificate authority)

AI Risks at the Edge

- Physical tampering
- Model poisoning
- Adversarial attacks

Ethical Deployment of Smart systems

Why ethics matter?

- Pervasiveness: Embedded in everyday life
- Autonomy: Make decisions without human oversight
- Impact: Affect privacy, safety, equity, and trust
- Accountability gap: Who is responsible for automated outcomes?

Risks of unethical deployment

- **Bias & Discrimination**
  E.g., Facial recognition misidentifying people.
- **Surveillance Overreach**
  E.g., Smart cities collecting real-time personal data
- **Loss of Autonomy**
  E.g., Devices acting without user consent
- **Data Misuse**
  E.g., Selling health data from wearable
- **Digital Divide**
  E.g., Excluding communities with limited access to technology

Example

Smart Healthcare System

- Pros: Predictive diagnostics, improved access
- Risks: Misdiagnosis from biased data, lack of patient consent
- Mitigation: Human-in-the-loop design, clear data policies

Ethical-by-design approach

- Involve stakeholders early (users, communities, experts)
- Conduct ethical impact assessments
- Embed values into system design (Privacy-by-Design, Fairness-by-Design)
- Provide opt-out & override mechanisms

Case Study: "NightWatch Breach: When Smart Cameras Turned Rogue"

System:

- “NightWatch” is a smart AI-powered surveillance system deployed in a gated community.
- Each house has AI-enabled IoT cameras connected to a central cloud dashboard.
- Facial recognition is used for entry automation.

Features:

- Real-time streaming
- Motion & face detection
- Cloud storage with mobile access
- OTA updates pushed weekly

Case study: The breach

Investigation revealed:

- External hacker gained access to 300+ cameras
- Live feeds streamed to a dark web server
- Attack lasted 3 weeks before detection
- Two households experienced unauthorized door unlocks

Investigation revealed:

- **Privacy breach:** Sensitive footage leaked
- **Security risk:** Unlawful home access

Root cause analysis

- Weak Authentication: Default credentials
- No End-to-End Encryption
- Poor Update Hygiene
- Centralized cloud = single point of failure

Fixes?

- Authentication: regular password changes, token based authentication.
- Secure communication (TLS/SSL)
- Regular updates
- Use AI (anomaly detection) to detect unusual access patterns and alert users/admins.
- Decentralization where possible: use edge AI for local inference, store sensitive data locally.

Key Takeaways

- Multifaceted Attack Surfaces in IoT
- Ignoring basic security hygiene (like default passwords) has real-world consequences
- **Security-by-Design & Privacy-by-Default** must be enforced.
- Regular audits, strong identity management, and secure firmware practices are essential.