Table of contents

Introduction

Fundamental Security Services

Security services are the objectives aimed at protecting a system.

- **Confidentiality**: Protection against unauthorized disclosure.
- **Integrity**: Protection against unauthorized modification.
- **Availability**: Guarantee of access for legitimate users.
- **Authentication**:
  - *Entity authentication*: Certifying the identity of an actor.
  - *Data origin authentication*: Certifying the source of data.
- **Non-repudiation**: Inability to deny a transaction.
- **Non-duplication**: Protection against illicit copying.
- **Anonymity**: Preservation of identity or source.

Original version

- **Confidentiality**: Protection of information from unauthorized disclosure.
- **Integrity**: Protection against unauthorized modification of information.
- **Availability**: Ensuring that resources are accessible to legitimate users.
- **Authentication**:
  - **Entity authentication** *(entity authentication)*: Process allowing one entity to be certain of the identity of a second entity, supported by corroborating evidence (e.g., physical presence, cryptographic, biometric, etc.). The term identification is sometimes also used for this service.
  - **Data origin authentication** *(data origin authentication)*: Process allowing one entity to be certain that a second entity is the original source of a set of data. By definition, this service also ensures the integrity of the data.

- **Non-repudiation**: Guarantees that an entity cannot deny being involved in a transaction.
- **Non-duplication**: Protection against illicit copying.
- **Anonymity (entity or data origin)**: Preserves the identity of an entity, the source of information, or a transaction.

Summary: Services, Threats, and Protection Mechanisms

Security Services	Threats and Attacks (*Italic*)	Classic Mechanisms	Digital Mechanisms
Confidentiality	Information leakage, *eavesdropping*, traffic analysis	Seals, safes, padlocks	Encryption, logical authorization
Integrity	Modification, *tampering*, illicit creation or destruction	Special ink, holograms	One-way functions + encryption
Availability	*Denial of Service (DoS)*, viruses, illicit use	Physical access control, video surveillance	Logical access control, audit, antivirus
Entity Authentication	Unauthorized access, password theft, protocol flaw	Presence, voice, ID card, biometrics	Secret + protocol, network address + userid, smart card + PIN
Data Authentication	Falsification of information or signature	Seals, signature, fingerprint	One-way functions + encryption
Non-repudiation	Denying a transaction (*repudiation*), claiming key theft	Seals, notary signature, registered mail	One-way functions + encryption + digital signature
Non-duplication	Duplication, falsification, imitation	Special ink, holograms, tagging	Digital watermarking, cryptographic locking
Anonymity	Identification, transaction analysis, tracing	Voice scramblers, disguise, cash	*Mixers*, *remailers*, e-money, *deep web*

Original version

Threats and Attacks: Summary

Services	Threats	Attacks
Confidentiality	Information leakage	Unauthorized eavesdropping, traffic analysis
Integrity	Information modification	Illicit creation, alteration, or destruction
Availability	Denial of service, illicit use	Viruses, repeated access attempts to disable a system
Entity Authentication	Unauthorized access	Password theft, authentication protocol flaw
Data Authentication	Information falsification	Signature forgery, protocol flaw
Non-repudiation	Denying involvement in a transaction	Claiming key theft or signature protocol flaw
Non-duplication	Duplication	Falsification, imitation
Anonymity	Identification	Transaction analysis, unauthorized access enabling identification

Protection Mechanisms

Services	Classic Mechanisms	Digital Mechanisms
Confidentiality	Seals, safes, padlocks	Encryption, logical authorization
Integrity	Special ink, holograms	One-way functions + encryption
Availability	Physical access control, video surveillance	Logical access control, audit, antivirus
Entity Authentication	Presence, voice, ID card, biometric recognition	Secret + authentication protocol, network address + userid, smart card + PIN
Data Authentication	Seals, signature, fingerprint	One-way functions + encryption
Non-repudiation	Seals, signature, notary signature, registered mail	One-way functions + encryption + digital signature
Non-duplication	Special ink, holograms, tagging	Digital watermarking, cryptographic locking
Anonymity	Voice scrambler, disguise, cash	Mixers, remailers, e-money, deep web

Internet-Related Risks

Malware Delivered via E-Mail

- Also called **malware**.
- Emails designed to **trigger an action** (open attachment or click a link).
- Often **personalized** using **social engineering**.
- **Main consequences:**
  - Malware installation (*ransomware*, *keyloggers*, etc.).
  - **Loss or theft of personal data**.
  - **System hijacking** and **malware propagation**.

Ultra-summary

- Malware spread by email
- Prompting clicks or opens
- Social engineering
- Data theft, loss, hijacking

Original version

Malware Delivered via E-Mail

- Also called **malware**.
- Emails designed to **incite the recipient** to **open an attachment** or **follow a link** containing ads, offensive info, risky programs, etc.
- Often targeted based on victim interests (preliminary social engineering).
- **Consequences:**
  - Malware installation (*ransomware*, *keyloggers*, etc.) on victim systems (*computer*, *tablet*, *smartphone*, *smartwatch*, etc.).
  - Data destruction.
  - Theft of personal information or data.
  - System hijacking for malicious purposes (e.g., illegal bitcoin mining).
  - Malware propagation (potentially to other users).

Malware Delivered via Web

- Method called ***drive-by download***: **automatic infection when visiting a website**.
- Sources can be:
  - A **malicious website**
  - A **compromised legitimate site** (e.g., *cross-site scripting*).
- **User caution** greatly reduces this propagation method.
- **Impacts are similar** to email-borne infections.
- **Script restriction** (*Java/JavaScript*) reduces risk but can **affect browsing**.

Ultra-summary

- *Drive-by download* = infection without user action
- Malicious or compromised sites
- Awareness + restricted scripts = protection

Original version

Malware Delivered via Web

- Often called ***drive-by download***, allows **infecting the system** (*computer*, *tablet*, *smartphone*, *smartwatch*, etc.) **simply by visiting a website**.
- Sources may be:
  - Malicious site containing malware
  - Legitimate website previously infected (e.g., *cross-site scripting*). Infection may only affect certain pages.
- User awareness (avoiding suspicious sites) reduces the effectiveness.
- Consequences are similar to email infections.
- Restricted script execution (*Java/JavaScript*) in browsers can limit infection but may constrain navigation.

Phishing

- Technique to **collect private information** through **indiscriminate fishing**.
- ***Phishing*** can be:

  - **General** (broad targeting)
  - **Targeted** (*spear phishing*) for a specific person or organization.

- Main vector: **forged email address**, hard to detect.
- Goal: obtain **sensitive data** (credentials, passwords, personal or banking info).
- Attacks use **credible or threatening pretexts** to prompt victim cooperation.

Ultra-summary

- Information theft by deception
- Forged emails
- *Spear phishing* = targeted attack
- Urgent or threatening pretexts

Original version

Phishing

- The word ***phishing*** comes from English ***"password"***, ***"harvesting"***, and ***"fishing"***.
- Shows the technique's main goal: **collect as much private info** via indiscriminate fishing.
- Targeted attacks are called ***spear phishing*** (from *spear fishing*).
- Transmission vector: email with **forged sender address** requesting private info: emails, social media credentials, passwords, ID numbers, bank accounts, etc.
- Pretexts vary (system update, service stoppage, delivery withdrawal) and may threaten the user if ignored.

Spam

- **Unwanted emails**, often ads, or **unsolicited pop-ups** during web browsing.
- Represent about **60% of global emails**.
- Main consequences:
  - **Resource consumption** and time loss.
  - Some can **transmit malware**.
- Often target short addresses or come from **sold/exchanged address lists**.
- **Anti-spam filters** incur **significant costs** for organizations.

Ultra-summary

- Unwanted emails/ads
- Risks: time/resource loss, malware
- Targeting: short addresses or lists
- Filtering costly for organizations

Original version

Spam

- Includes all **unwanted emails** (often ads) received by people and organizations.
- Also applies to **pages/pop-up windows shown without user consent** during web browsing.
- Around **60%** of global emails belong to this category.
- Consequences: resource consumption and wasted time, but some spam can also **transmit malware**.
- Often target short email addresses or list-based addresses (sold/exchanged).
- **Anti-spam filtering** incurs high organizational costs.

Ransomware

- Trojan-type malware that **encrypts data** to make it inaccessible.
- Demands a **ransom** (often in bitcoins) to recover files.
- Can remain **dormant**, triggered by an event or date.
- Main vector: **malicious emails**.
- Other effects: **DoS attacks, extortion**.

Ultra-summary

- Data encrypted by Trojan
- Ransom to restore access
- Possible programmed dormancy
- Infection via malicious emails

Original version

Ransomware

- Trojan malware family.

- Typically **encrypts victim's data** to make it completely inaccessible.
- Then displays a message requesting **ransom payment** (often in **bitcoins**).
- May stay in **dormant state** triggered by event or date.
- Infection vectors vary, but **malicious email attachments** often responsible.
- Variants exist and continue to evolve.
- Other behaviors: **DoS, targeted extortion, threats**, etc.

Attacks on *Internet of Things (IoT)* Devices

- Target **connected objects** (cameras, TVs, sensors, alarms, etc.).
- Devices are **easy to compromise** due to:
  - **Known vulnerabilities**
  - **Default passwords**
  - **User unawareness**
- **Remote control** enables:
  - **Entry point** to the network
  - **Device abuse** for illicit activities (DDoS, hacking, mining)
- A **precise inventory** of connected devices is essential.

Ultra-summary

- Targets connected objects
- Weak security (vulnerabilities, default passwords)
- Risk of network access and abuse
- IoT inventory needed

Original version

Attacks on IoT Devices

- Target connected objects (cameras, TVs, fridges, sensors, alarm systems, etc.).
- Often **easier to hack** than traditional systems due to:
  - Many vulnerabilities known to attackers
  - Default passwords
  - User negligence
- Remote takeover allows:

  - Entry point to home/corporate network
  - Device use for illicit activities (hacking, DDoS, bitcoin mining)

- Maintaining a detailed directory of all connected devices is necessary.

Information Spoofing and Website Defacement

- Attacks aiming to **alter information** on websites and social media.
- Impact: **compromised reputation** and **economic damage**.
- Websites: secure host system, restrictive configuration, **regular audits**.
- Social media: strong passwords, **multi-factor authentication**, session closure, cookie deletion.

Ultra-summary

- Altered info on websites and social media
- Risks: reputation, economic losses
- Websites: security + audits
- Social media: strong passwords, MFA, closed sessions, cookies cleared

Original version

Information Spoofing and Website Defacement

- Target **integrity** of published info on websites and social media.
- Affects **reputation** and can cause **economic damage**.
- **Websites**: secure host system, as restrictive configuration as possible, recurring security audits recommended.
- **Social media**: depends on authentication process:
  - Avoid simple passwords
  - Prefer strong, possibly multi-factor authentication
  - Close sessions properly
  - Clear cookies

Denial of Service (DoS / DDoS) Attacks

- Aim to **make IT systems inaccessible**, especially for organizations.

- **DDoS**: distributed attack using thousands of devices, generating massive traffic.
- Classic protections (*firewalls*, IDS/IPS sensors) often **insufficient**.
- Consequences:
  - **Affected reputation**
  - **Financial losses** (sometimes ransom)
  - **High risk for critical infrastructure** (hospitals, power plants, Internet backbone)

Ultra-summary

- DDoS = inaccessible systems via massive attacks
- Limited protections
- Risks: reputation, finances, critical infrastructures

Original version

Denial of Service (DoS / DDoS) Attacks

- Aim to **render IT systems inaccessible**, mainly targeting private or governmental organizations.
- **DDoS** (*Distributed Denial of Service*): multiple devices (**often tens of thousands**) simultaneously target victim system(s).
- Traffic can reach several hundred Gbps.
- Traditional protection (*firewalls*, *intrusion detection/prevention sensors*) has limited effectiveness.
- Service unavailability can cause:
  - **Reputational issues**
  - Significant **financial losses** (ransom demands)
  - **High security risks (even physical)** for **critical infrastructures** (hospitals, power plants, Internet backbone).

Digital Security Methods

Problem: Protecting digital information

- in a distributed environment
- globally accessible
- without physical boundaries

Solution:

- Cryptography
  - Symmetric
  - Asymmetric
  - + One-way functions
  - + (Pseudo) random generators

Ultra-summary

- **Problem:** Security in a distributed/global environment.
- **Solutions:**
  - Crypto (symmetric/asymmetric).
  - One-way functions (hashing).
  - Random generators (physical/pseudo).

Original version

Problem: Protecting digital information

- in a distributed environment
- globally accessible
- without physical boundaries

Solution:

- Cryptography
  - Symmetric
  - Asymmetric
  - + One-way functions
  - + (Pseudo) random generators

Cryptographic Hash Functions

- **Functions easy to compute in one direction but virtually impossible to reverse**.
- Any modification of the source document radically changes the **digest** (avalanche effect).
- **Key properties**:
  - **One-way**: impossible to retrieve the input from the hash.

- **Collision-free**: impossible to find two inputs with the same hash.

- Digest size: 160 to 512 bits.
- Algorithms (very **performant**): SHA-1, SHA-256, SHA-3.

Ultra-summary

- **One-way + collision-free**.
- Size: 160-512 bits.
- Algos: SHA-1/256/3.
- Usage: integrity, signatures.

Original version

- **Functions easy to compute in one direction but virtually impossible to compute in the reverse direction**.
- Any modification (even insignificant) of the source document results in a fundamentally different **digest**.
- It is virtually impossible to retrieve the source document using only the digest (**one-way**).
- It is virtually impossible to find a second source document producing the same digest (**collision-free**).
- Usual digest length: 160 to 512 bits.
- One-way algorithms are very performant.
- Examples: SHA-1, SHA-256, SHA-3, etc.

(Pseudo) Random Generators

- **Characteristics**
  - **random**
  - **unpredictable**
  - **non-reproducible**
- **Critical** for security (keys, IV, secrets).
- **Types**:
  - **True random**: based on physical phenomena (radioactivity, quantum).
  - **Pseudo-random**: deterministic (based on a *seed*: initial random sequence).
- **Risk**: "Pseudo-security" if the *seed* is predictable (Pitkin quote).

- Applications: session keys, IV (DES-CBC), signatures (ElGamal).

Ultra-summary

- **True random**: physical (quantum).
- **Pseudo-random**: deterministic (*seed*).
- **Risk**: predictable *seed* = vulnerability.
- Uses: keys, IV, signatures.

Original version

- Random number generation is a very important process that can compromise the security of many encryption systems.
- Applications: session key generation, initialization vectors (DES - CBC mode), secrets for signatures (ElGamal), etc.
- A **random generator** is a device capable of generating numbers in a **random**, **unpredictable** and **non-reproducible** way. (e.g. based on physical phenomena: radioactive or quantum source).
- **Pseudo-random generators** are deterministic processes developed from an initial random sequence (**seed**) (e.g. user keystrokes, disk access).
- *Quote:* R. Pitkin in [Kau95]: "The use of pseudo-random processes to generate secret quantities can result in pseudo-security"

Symmetric Cryptography

- **History**: Used since Julius Caesar (1st century BC).
- **Principle**: A single key for encryption/decryption.
- **Scheme**: Plaintext → Encryption (Key) → Ciphertext → Decryption (Key) → Plaintext.
- **Characteristics**:
  - Algorithms: AES, DES, IDEA, RC4.
  - Services: Confidentiality, Authentication, Integrity.
  - **Limit**: No signatures (shared key).
  - **Problem**: Secure key exchange required.

Ultra-summary

- **1 key** to encrypt/decrypt.
- **Fast** (AES, DES).

- **Problem**: key exchange.
- Uses: personal documents, closed groups.

Original version

- Also called conventional cryptography or secret key cryptography (1st century BC, Julius Caesar).
- **Idea:** Based on a single secret key, perform a transformation capable of respectively making information unreadable and restoring it.
- **Scheme:** Plaintext → Encryption (Key) → Ciphertext → Decryption (Key) → Plaintext.
- **Characteristics:**
  - Algorithms: AES, DES, IDEA, RC4, RC5, etc. (some are free and openly available)
  - Services: Confidentiality, Authentication, Integrity.
  - No direct support for digital signatures (because key known by both).
  - Requires a confidential channel to exchange the key.
  - Ideal for protection of personal documents or closed groups.

Asymmetric Cryptography

- Also called **public cryptography** (1976, Diffie & Hellman).
- **Principle**
  - Key pair (public/private) for encryption and signatures.
- **Two main uses**:
  1. **Confidentiality**:
     - Encryption: recipient's public key
     - Decryption: recipient's private key
  2. **Digital signature**:
     - Signature: sender's private key
     - Verification: sender's public key
     - *Optimization*: Generally sign the **hash** of the document
     - **Fundamental properties**:
       * **Integrity**: Any modification invalidates the signature
       * **Non-collision**: Impossible to have 2 documents with the same signature

        * **Non-repudiation**: Only the holder of the private key can sign

- **Technical aspects**:
    - **Algorithms**: RSA, ElGamal
    - **Services**: Integrity, Authentication, Non-Repudiation
    - **Performance**: much slower than symmetric (100x slower)
    - **Advantage**: No need for a confidential channel for key exchange

Ultra-summary

- **2 keys**: public (encrypt/verify) + private (decrypt/sign)
- **2 uses**:
    - Confidentiality: encrypt for a recipient
    - Signature: prove authenticity
- **Signatures**:
    - Integrity + non-repudiation
- **Algorithms**: RSA/ElGamal
- **Advantage**: No need for secure channel to exchange keys
- **Disadvantage**: Slow

Original version

Asymmetric Cryptography

- Also called public cryptography or public key cryptography (1976, W. Diffie & M. Hellman).
- **Idea:** Use two different keys - one **secret** and one **public** - respectively for encryption and decryption operations.
- Each user has a **keyring**.

Confidentiality: * Sender encrypts with the **recipient's public key**. * Recipient decrypts with their **private key**. * Only the recipient's key is used!
Digital Signature: * Sender signs with their **private key**. * Recipient verifies with the **sender's public key**. * Only the sender's key is used! * *Note:* Generally sign the **digest** of the document (hash) for performance reasons.
Signature characteristics: * The signature changes if the document changes, while the private key remains the same. * If the document or signature is modified, verification fails (**integrity guaranteed**). * It is virtually impossible, even for the holder of the private key, to generate a second document producing the same signature (one-way function **without collisions**). * Only the holder of the private key can generate a signature

verifiable using the corresponding public key (**non-repudiation**). * **Algorithms**: RSA, ElGamal. * **Services**: Integrity, Authentication, Non-Repudiation. * **Slowness**: Up to 50 times slower than symmetric cryptography. * **Advantage**: No need for a confidential channel to exchange keys (unlike symmetric).

Asymmetric + Symmetric Cryptography (Hybrid)

- **Principle**: Use asymmetric to exchange a symmetric key (session key).
- **Steps**:
  1. A generates a random symmetric key K_s.
  2. A encrypts K_s with B's public key.
  3. A and B then communicate using K_s (symmetric).

Ultra-summary

- **Asymmetric**: exchange of symmetric key.
- **Symmetric**: data encryption.
- **Advantage**: combines security + performance.

Original version

- **Idea:** Use public cryptography only to exchange symmetric keys (Session keys).
- A generates a random key K_s and transmits it to B by encrypting it with B's public key.
- A & B then communicate using K_s (symmetric).

Asymmetric Cryptography: Operation (RSA)

Key Construction

1. **Choice of prime numbers**:
   - p and q: two large prime numbers (> 1024 bits)
   - $n = pq$: RSA modulus (size = 2048+ bits)

2. **Calculation of Euler's totient function**:

   - $\phi(n) = (p-1)(q-1)$
   - *Property*: For any a coprime with n, $a^{\phi(n)} \equiv 1 \pmod{n}$

3. **Selection of exponents**:

   - e: integer coprime with $\phi(n)$ (public exponent)
   - d: modular inverse of e (private exponent), such that $ed \equiv 1 \pmod{\phi(n)}$

Encryption/decryption process

- **Public key**: (n, e)
- **Private key**: (d)
- **Encryption**: $C = P^e \pmod{n}$
- **Decryption**: $P = C^d \pmod{n}$

Mathematical proof

1. **Fundamental congruence**:

   - $ed = 1 + k\phi(n)$ (by definition of d)

2. **Application of Euler's theorem**:

   - $P^{\phi(n)} \equiv 1 \pmod{n}$ (if P coprime with n)

3. **Demonstration**:

$$
\begin{aligned}
(P^e)^d &\equiv P^{ed} \pmod{n} \\
&\equiv P^{1+k\phi(n)} \pmod{n} \\
&\equiv P \cdot (P^{\phi(n)})^k \pmod{n} \\
&\equiv P \cdot 1^k \pmod{n} \\
&\equiv P \pmod{n}
\end{aligned}
$$

System security

- **Hard problem**: Factorization of n into p and q
- **Recommended size**:
  - n: 2048 bits (minimum for current security)
  - p and q: 1024+ bits each
- **Known vulnerabilities**:
  - Side-channel attacks (timing, power analysis)
  - Inappropriate parameter choices (e too small, p and q too close)

Ultra-summary

- **Keys**:
  - Public: (n, e) where $n = pq$
  - Private: (d) with $ed \equiv 1 \pmod{\phi(n)}$
- **Operations**:
  - Encryption: $P^e \mod n$
  - Decryption: $C^d \mod n$
- **Security**: Factorization of n difficult
- **Size**: 2048+ bits for n

Original version

Asymmetric Cryptography: Operation (RSA)

- Let $n := pq$ with p and q two large prime numbers (> 1024 bits).
- Let $\phi(n) = (p-1)(q-1)$.
- Let e and d such that $ed \equiv 1 \pmod{\phi(n)}$.
- By definition of congruences: $ed = 1 + k\phi(n)$
- Euler's theorem: $a^{\phi(n)} \equiv 1 \pmod{n}$.
- **Encryption:** $C = P^e \pmod{n}$. **Public key:** (n, e).
- **Decryption:** $P = C^d \pmod{n}$. **Private key:** (d).
- *Proof:* $(P^e)^d \equiv P^{ed} \equiv P^{1+k\phi(n)} \equiv (P \pmod{n})(P^{\phi(n)} \pmod{n})^k \equiv P \pmod{n}$.

Asymmetric Cryptography: Conclusions

- **Dominant algorithms**: RSA (most used), Rabin, ElGamal
- **Complete services**:
  - Confidentiality
  - Authentication
  - Integrity
  - Digital signature & Non-repudiation
  - Non-duplication
- **Performance**:
  - 50x slower than symmetric
  - **Optimal solution**: Combination of asymmetric (key exchange) + symmetric (encryption)
- **Key management**:
  - **Advantage**: Public key exchange without confidential channel
  - **Risk**: Need to verify authenticity of public keys
    - Authenticated acquisition channel **or**
    - Certification by trusted third party

Ultra-summary

- **Algos**: RSA (dominant), Rabin, ElGamal
- **Services**: Confidentiality + Authentication + Integrity + Signatures
- **Slowness**: 50x vs symmetric → **hybrid recommended**
- **Keys**: Simple public exchange but **authentication crucial**

Original version

Asymmetric Cryptography: Conclusions

- There are a few asymmetric encryption systems (**Rabin**, **ElGamal**, etc.) but the most used is **RSA**.
- **Supported services**: **Confidentiality**, **Authentication**, **Integrity**, **Digital Signature & Non-Repudiation**, (**Non-Duplication**).
- Operations related to **asymmetric cryptography** are up to **50 times (!) slower** than those of **symmetric cryptography**. A **combination of the two methods** is often desirable.
- **Key distribution** is simplified by the fact that only **public keys** need to be

exchanged between participants (no need for an alternative confidential channel) but…

- … it is necessary to **verify that the public key actually belongs to the recipient**:
  - Either the **acquisition channel** of the public key is protected against any modification (**authenticated**)
  - Or the key is **certified accurate by a third party**

Symmetric vs Asymmetric Comparison

Comparative advantages

- **Symmetric**:
  - **Performance**: 100x faster
  - **Implementation**: Easy in hardware
  - **Keys**: Short (128 bits = 16 memorable characters)
- **Asymmetric**:
  - **Key exchange**: Authenticated channel sufficient (no need for confidentiality)
  - **Management**: 1 key pair for n correspondents (vs n keys in symmetric)

Common issues

- **Weak link**: Key management by users
- **Security basis**: Empirical rather than theoretical
- **Legal constraints**: Usage and export restrictions

Usage recommendations

Use case	Recommended solution	Justification
Personal documents	Symmetric	Speed + memorable keys
Groups of close users	Symmetric	Speed + easy confidential exchange
Distant/unknown users	Asymmetric	No need for confidential channel

Use case	Recommended solution	Justification
Remote transactions	Hybrid (Asymmetric + Symmetric)	Asymmetric for key exchange, symmetric for data
Software protection (distribution)	Hybrid	Unique symmetric key per version, encrypted with asymmetric
Network segments	Symmetric	Speed + controlled environment (easy key exchange between administrators)

Ultra-summary

Symmetric:

```
Fast (100x)
Short keys (128 bits)
Confidential key exchange required
```

Asymmetric:

```
Simplified key exchange
1 key pair for n correspondents
Slow (50x)
Long keys (1024+ bits)
```

Hybrid: Best of both worlds **Common problems**: Key management, empirical basis, legal restrictions

Original version

Symmetric vs. Asymmetric Cryptography

- There are **hundreds of symmetric and asymmetric algorithms** capable of providing a sufficient level of **confidentiality**.
- **Symmetric solutions** offer the following advantages:
  - **Speed** (up to **100 times faster** than asymmetric solutions)
  - **Ease of hardware implementation**
  - **Reduced key length**: **128 bits** (= 16 characters   memorable!) instead of **1024 bits** for asymmetric equivalents.
- **Asymmetric solutions** have as main arguments:

- **Simplified key exchange**: keys must be exchanged through an **authenticated but non-confidential channel**
- **Simplified key management**: a single **public/private key pair** is sufficient for a user to receive confidential messages from **n users** (instead of **n different keys** in the symmetric case).

- **Problems common to both techniques**:
  - **Key management by the user** remains the **weakest link**
  - Security (normally) based on **empirical arguments** rather than **theoretical** ones
  - **Legal restrictions** on usage and export

Symmetric vs. Asymmetric Cryptography (II)

Activity	Recommendation	Remarks
Protection of personal documents	**Symmetric crypto**	**Speed**, easily memorable keys
Protection of documents in a group of close users	**Symmetric crypto**	**Speed**, ease of exchanging confidential keys
Establishment of confidential channels between distant users (unknown)	**Asymmetric crypto**	No need to have a confidential channel: **authenticity suffices**
Transactions between two distant users, Software protection (multicast distribution)	**Asymmetric crypto for symmetric key protection + Symmetric crypto for data protection**	**Speed**, Only the symmetric key needs to be re-encrypted for each correspondent, Encrypted copy of software can be made public
Protection of network segments	**Symmetric crypto**	**Speed**, Stable environment $\rightarrow$ easy confidential key exchange between sysadmins

Dissection of an Attack: Ransomware

Definition and Impact

- **Definition**: Malicious software that encrypts data and demands a ransom for its restoration.

- **Limitations of the classic definition**:
  - Does not cover the impact on **critical infrastructure** (e.g., Colonial Pipeline, May 2021)
  - Underestimates the **systemic scope** of attacks
- **Alarming statistics**:
  - Billions of annual attacks
  - Considered the **most dangerous cyber threat** in 2021 ("Ransomware Everywhere")

Ultra-summary

- **Malware**: Encrypts data → demands ransom
- **Impact**: Critical infrastructure (e.g., Colonial Pipeline)
- **Threat #1** in cybersecurity (2021)
- **Targets**: Individuals + businesses + states

Original version

"Ransomware (from English **ransomware**), ransom software, extortion software, is malicious software that holds personal data hostage. To do this, ransomware **encrypts personal data** then asks their owner to send money in exchange for the **decryption key**" (Wikipedia September 21, 2021).

- **Incomplete definition** because **ransomware** affects a **vast spectrum of IT infrastructure**
- For example, in May 2021, a **ransomware attack** against the **Colonial Pipeline** company caused a **fuel supply disruption** for a large part of the US coast
- With a **total number of attacks** counted in **billions per year**, "**Ransomware Everywhere**" is globally considered the **most direct, visible and dangerous threat** for users and companies in 2021!

Ransomware Attack Lifecycle

Prevention and Response

Phase	Measures
Prevention	- Regular patching- Detection solutions (Firewalls, WAFs, IDS/IPS)- Anti-malware scans (emails, files)
Protection	- **Offline backups** (essential!)- Strict security policies- User training
Response	- **Do not pay** (official recommendation)- Forensic analysis- Restoration from backups

Technical Dissection

1. **Infection**:
    - Vectors: Phishing, exploits, vulnerable RDP
    - Propagation: Lateral (network) or vertical (system)
2. **Execution**:
    - Encryption of targeted files
    - Deletion of shadow copies
    - Persistence (registry, scheduled tasks)
3. **Extortion**:
    - Display of ransom demand
    - Payment in cryptocurrencies (Bitcoin, Monero)
    - Payment deadlines with penalties
4. **Obfuscation**:
    - Code obfuscation
    - Communication via TOR/Deep Web
    - Log erasure

Ultra-summary

Attack cycle:

1. Infection (phishing/exploits)
2. Execution (encryption + persistence)
3. Extortion (ransom in crypto)
4. Obfuscation (TOR + trace erasure)

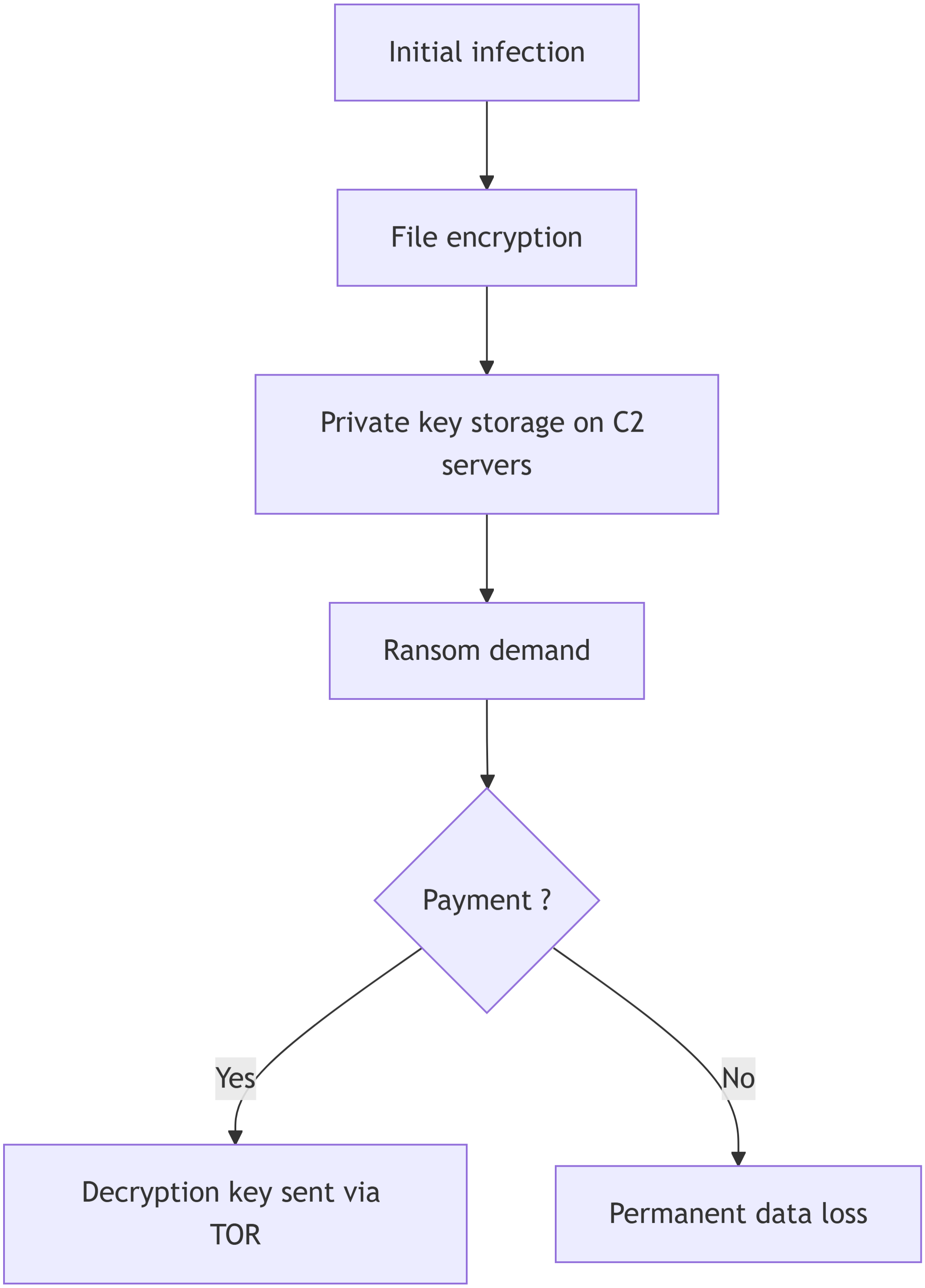
Countermeasures:

```
Offline backups
Patching + detection
```

```
Training
Do not pay
```

Cryptolocker: Technical Analysis

Attack Scheme


Initial infection
File encryption
Private key storage on C2 servers
Ransom demand
Payment ?
Yes
No
Decryption key sent via TOR
Permanent data loss

Preferred Targets

- **Critical extensions** (extract):
  - Documents: `.docx`, `.xlsx`, `.pdf`, `.pptx`
  - Databases: `.mdb`, `.sql`, `.sqlite`
  - Media: `.jpg`, `.png`, `.mp4`, `.avi`
  - Development: `.java`, `.cpp`, `.py`, `.php`
  - Financial: `.qbw`, `.qbb`, `.wallet`
- **Behavior**:
  - **Selective** encryption (recent/modified files)
  - **Double extortion**: Encryption + threat of leakage
  - **RaaS** (Ransomware-as-a-Service): Economic model

Ultra-summary

Mechanism: - Private key stored on C2 servers - Payment → key via TOR - Targets: 100+ extensions (docs, DB, media)
Recent evolutions: - Double extortion (encryption + leakage) - RaaS (ransomware rental)

Original version

Ransomware: Complete View

(Source: 2017 State of Cybersecurity, F-Secure Inc.)

Ransomware: Complete View

Prevention, Remediation and Response

- **Patching**
- **Active and passive detection** (Firewalls, WAFs, IDS, IPS, email malware scan, etc.)
- **Offline backups** !
- **Security Policy** - Rules for proper email usage
- **Training** !
- **To pay or not to pay…**

Technical Dissection of the Attack

- **Infection and propagation**
- **Execution**

- **Payment** (Crypto-currencies / Bitcoin)
- **Obfuscation** (Obfuscation, TOR Networks/Deep Web)

Generic Scheme of a Cryptolocker Ransomware

- **Decryption private keys** are stored on the **attacker's** servers
- They are sent to the **victim** after **bitcoin payment**
- The **trace** is shredded using **TOR networks**

Ransomware Cryptolocker: Targets

Targeted file extensions: .jin, .xls, .xlsx, .pdf, .doc, .docx, .ppt, .pptx, .txt, .dwg, .bak, .bkf, .pst, .dbx, .zip, .rar, .mdb, .asp, .aspx, .html, .htm, .dbf, .3dm, .3ds, .3fr, .jar, .3g2, .xml, .png, .tif, .3gp, .java, .jpe, .jpeg, .jpg, .jsp, .php, .3pr, .7z, .ab4, .accdb, .accde, .accdr, .accdt, .ach, .kbx, .acr, .act, .adb, .ads, .agdl, .ai, .ait, .al, .apj, .arw, .asf, .asm, .asx, .avi, .awg, .back, .backup, .backupdb, .pbl, .bank, .bay, .bdb, .bgt, .bik, .bkp, .blend, .bpw, .c, .cdf, .cdr, .cdr3, .cdr4, .cdr5, .cdr6, .cdrw, .cdx, .ce1, .ce2, .cer, .cfp, .cgm, .cib, .class, .cls, .cmt, .cpi, .cpp, .cr2, .craw, .crt, .crw, .phtml, .php5, .cs, .csh, .csl, .tib, .csv, .dac, .db, .db3, .dbjournal, .dc2, .dcr, .dcs, .ddd, .ddoc, .ddrw, .dds, .der, .des, .design, .dgc, .djvu, .dng, .dot, .docm, .dotm, .dotx, .drf, .drw, .dtd, .dxb, .dxf, .dxg, .eml, .eps, .erbsql, .erf, .exf, .fdb, .ffd, .fff, .fh, .fmb, .fhd, .fla, .flac, .flv, .fpx, .fxg, .gray, .grey, .gry, .h, .hbk, .hpp, .ibank, .ibd, .ibz, .idx, .iif, .iiq, .incpas, .indd, .kc2, .kdbx, .kdc, .key, .kpdx, .lua, .m, .m4v, .max, .mdc, .mdf, .mef, .mfw, .mmw, .moneywell, .mos, .mov, .mp3, .mp4, .mpg, .mrw, .msg, .myd, .nd, .ndd, .nef, .nk2, .nop, .nrw, .ns2, .ns3, .ns4, .nsd, .nsf, .nsg, .nsh, .nwb, .nx2, .nxl, .nyf, .oab, .obj, .odb, .odc, .odf, .odg, .odm, .odp, .ods, .odt, .oil, .orf, .ost, .otg, .oth, .otp, .ots, .ott, .p12, .p7b, .p7c, .pab, .pages, .pas, .pat, .pcd, .pct, .pdb, .pdd, .pef, .pem, .pfx, .pl, .plc, .pot, .potm, .potx, .ppam, .pps, .ppsm, .ppsx, .pptm, .prf, .ps, .psafe3, .psd, .pspimage, .ptx, .py, .qba, .qbb, .qbm, .qbr, .qbw, .qbx, .qby, .r3d, .raf, .rat, .raw, .rdb, .rm, .rtf, .rw2, .rwl, .rwz, .s3db, .sas7bdat, .say, .sd0, .sda, .sdf, .sldm, .sldx, .sql, .sqlite, .sqlite3, .sqlitedb, .sr2, .srf, .srt, .srw, .st4, .st5, .st6, .st7, .st8, .std, .sti, .stw, .stx, .svg, .swf, .sxc, .sxd, .sxg, .sxi, .sxi, .sxm, .sxw, .tex, .tga, .thm, .tlg, .vob, .war, .wallet, .wav, .wb2, .wmv, .wpd, .wps, .x11, .x3f, .xis, .xla, .xlam, .xlk, .xlm, .xlr, .xlsb, .xlsm, .xlt, .xltm, .xltx, .xlw, .ycbcra, .yuv

Source: Intel Security Advanced Threat Research - http://www.intelsecurity.com