



Ovidius University of Constanța  
Faculty of Mathematics and Informatics  
Cybersecurity and Machine Learning

## Digital signature techniques as an authentication method

Students: Memedula Edna

Nicolaev Andrei

# Cuprins

Chapter 1 – Introduction.....	3
1.1 Computer Science .....	3
1.2 Cybersecurity .....	4
1.3 Cryptography.....	5
Chapter 2 – Digital Signature .....	6
2.1 – Terminology.....	6
2.2 – Definition.....	7
2.3 – Who Can Sign Digitally .....	8
2.4 – Digital vs. Handwritten Signature.....	9
2.5 – Purpose of Document Signing.....	9
Chapter 3 – Mathematical Algorithms Used in Electronic Signatures .....	10
3.1 – RSA Algorithm .....	12
Chapter 4 – How Electronic Signatures Work .....	14
4.1 – Role of Digital Signatures .....	17
4.2 – Advantages of Using Digital Signatures.....	18
Conclusion.....	20
Bibliography .....	21

# Chapter 1 – Introduction

## 1.1 Computer Science

**Computer science** is a comprehensive discipline concerned with the design, analysis, and application of computational systems. It encompasses a broad spectrum of theoretical and practical domains, from abstract algorithmic reasoning to concrete hardware-software interaction. Its goal is to solve complex problems, automate processes, and process large-scale data efficiently and securely.

Core areas of computer science include:

- **Algorithms and Data Structures** – methods for organizing and manipulating data efficiently.
- **Computer Architecture and Systems Design** – focusing on how hardware components interact with software layers.
- **Operating Systems and Software Engineering** – ensuring reliability, scalability, and maintainability in complex applications.
- **Networking and Distributed Systems** – enabling communication and resource sharing between remote systems.
- **Databases and Information Systems** – managing, querying, and safeguarding large datasets.
- **Artificial Intelligence and Machine Learning** – systems that simulate intelligent behavior and adapt based on data.
- **Human-Computer Interaction (HCI)** – optimizing user interfaces and usability.

Computer science draws heavily from **mathematics** (logic, combinatorics, graph theory) and **engineering** (digital electronics, control theory). Additionally, it employs **statistical methods** for performance evaluation, machine learning, and stochastic modeling.

A key aspect of the discipline is its focus on **empirical validation**. Researchers and practitioners rely on:

- **Experimental simulations,**

- **Benchmarking** of algorithms and systems,
- **Formal verification** to ensure correctness.

Increasingly, **security and privacy** have become central concerns, influencing the development of secure protocols, access control mechanisms, cryptographic primitives, and data protection strategies.

Computer science is the engine behind today's digital transformation, enabling everything from cloud computing and robotics to mobile applications and quantum computing.

## 1.2 Cybersecurity

**Cybersecurity** is a specialized subfield of computer science dedicated to defending systems, networks, and data against unauthorized access, cyberattacks, and digital threats. As digital devices and online services become integral to modern life, securing these systems is more critical than ever.

Key areas of cybersecurity include:

- **Network Security** – protecting communication protocols and preventing intrusion in networked environments.
- **Application Security** – ensuring software is free of vulnerabilities during design and deployment.
- **Information Security (InfoSec)** – guarding the confidentiality, integrity, and availability of data.
- **Identity and Access Management (IAM)** – controlling who has access to what information and systems.
- **Cryptographic Protocols** – encrypting data to prevent eavesdropping and forgery.
- **Incident Response and Forensics** – identifying, analyzing, and responding to breaches or malicious behavior.

Cybersecurity threats come in various forms:

- **Malware** (e.g., viruses, ransomware),

- **Phishing attacks** (fraudulent communication to extract sensitive data),
- **Denial-of-Service (DoS/DDoS)** attacks,
- **Insider threats**, and
- **Advanced Persistent Threats (APTs)**—often involving **state-sponsored** actors.

Historical cases such as **Marcus Hess**, a hacker who infiltrated US defense systems for the KGB, illustrate how vulnerabilities—if unpatched—can be exploited by attackers for political, financial, or ideological motives.

Hacktivists, cybercriminals, and nation-state actors each have different agendas, but all exploit weaknesses in systems, human behavior, or supply chains. Cybersecurity professionals must anticipate, detect, and mitigate these threats in real time.

As more systems shift online, from healthcare to critical infrastructure, cybersecurity becomes not just a technical concern but a national and global imperative.

### 1.3 Cryptography

**Cryptography** is the mathematical science of secure communication. It ensures that data remains **confidential**, **authentic**, and **tamper-proof**—whether stored or in transit. It plays a foundational role in cybersecurity, banking, blockchain technology, and secure messaging.

The term comes from the Greek *kryptós* (“hidden”) and *gráfein* (“to write”). While early cryptographic techniques such as the **Caesar cipher** were used by historical figures like Julius Caesar to secure military messages, modern cryptography involves highly complex mathematical transformations.

Cryptography is divided into:

- **Symmetric Cryptography** – the same key is used for encryption and decryption (e.g., AES).
- **Asymmetric Cryptography** – uses a **public-private key pair**, enabling secure key exchange, digital signatures, and zero-knowledge proofs (e.g., RSA, ECC).
- **Cryptanalysis** – the study of methods for breaking cryptographic systems.

- **Post-Quantum Cryptography** – developing algorithms resistant to quantum computer attacks.
- **Quantum Cryptography** – using quantum physics to achieve unbreakable encryption, like in Quantum Key Distribution (QKD).

A message in its readable form is called **plaintext**, and once encrypted, becomes **ciphertext**.

This transformation is done via a combination of:

- A **cryptographic algorithm** (e.g., AES, RSA),
- A **key** (a number or bit string that drives the algorithm's operation).

Key concepts include:

- **Encryption** – the process of transforming plaintext into ciphertext to prevent unauthorized access.
- **Decryption** – converting ciphertext back into readable plaintext using the correct key.
- **Hash Functions** – irreversible one-way functions that map data to a fixed-length output, essential in digital signatures and blockchain.

A secure cryptographic system allows the encryption method to be public, as long as the **key** remains secret. Without the correct key, decrypting the message should be computationally infeasible, even with full knowledge of the algorithm.

Cryptography has evolved from simple codes into an advanced scientific domain embedded in protocols like HTTPS, VPNs, digital currencies, and secure multi-party computation.

## Chapter 2 – Digital Signature

### 2.1 – Terminology

A **signer** is a person who initiates a digital signature using a signature creation device. This individual may act in a personal capacity or represent another party—such as a company, government body, or other legal entity. In the context of secure digital communications, the signer is responsible for ensuring that the signature is generated using trusted means and in a secure environment.

**Signature creation data** refers to any unique digital credentials—typically cryptographic in nature—used to generate a digital signature. These may include private keys, secure tokens, or specialized credentials embedded in hardware modules like smart cards or USB tokens. This data must remain confidential and securely stored, as unauthorized access could compromise the integrity of the signature process.

A **digital signature creation device** can be software-based (e.g., cryptographic libraries, mobile applications) or hardware-based (e.g., hardware security modules or smart cards). These devices are responsible for securely applying the private key to the message digest or content, producing a signature that is mathematically tied to the signer and the data being signed.

## 2.2 – Definition

A **digital signature scheme** is a cryptographic framework designed to provide authentication, data integrity, and non-repudiation in digital communications. It typically involves the following three core algorithms:

- **Key Generation Algorithm:** This algorithm generates a key pair—comprising a private key (kept secret) and a corresponding public key (shared openly). These keys are mathematically linked through a one-way function, ensuring the private key cannot be derived from the public key.
- **Signing Algorithm:** This algorithm takes two inputs—a message (or its hash) and a private key—and outputs a digital signature. The result is a unique fingerprint that is cryptographically bound to the content and the identity of the signer.
- **Verification Algorithm:** This algorithm takes the message, the digital signature, and the signer's public key. It checks whether the signature is valid by comparing the decrypted signature against a freshly computed hash of the message. If they match, authenticity and data integrity are confirmed.

**Two essential security properties** of digital signature schemes are:

1. **Authenticity** – A digital signature created using a specific private key and message must be verifiable by the corresponding public key. This ensures the signature was indeed created by the legitimate signer.

2. **Unforgeability** – It must be **computationally infeasible** for an attacker to generate a valid signature without knowing the private key, even if they have access to multiple message-signature pairs.

A **digital signature** functions as an electronic seal of authenticity. The signer "locks" their approval onto a digital document using a cryptographic method, and this seal can be verified by others using the corresponding public key. A well-known implementation is the **Digital Signature Algorithm (DSA)**, standardized by NIST for government and commercial applications.

An **advanced electronic signature (AES)** must meet additional legal and technical criteria:

- **Uniquely linked** to the signer through a secure private key;
- **Capable of identifying** the signer without ambiguity;
- **Created using means** under the sole control of the signer;
- **Bound to the signed data** so that any tampering can be detected immediately.

## 2.3 – Who Can Sign Digitally

Any individual—whether a private citizen, employee, or government official—can apply for a **digital certificate** from a recognized Certificate Authority (CA) and use it to sign documents electronically. The scope of use can be categorized as:

- **Optional Use** – For non-binding purposes, such as confirming information in casual digital correspondence.
- **Mandatory Use** – For submitting legally binding documents, especially in regulated sectors (e.g., taxation, public procurement, or financial reporting).

Unlike handwritten signatures, **electronic signatures can be delegated**. Authorized individuals can sign:

- **On their own behalf**, or
- **On behalf of an organization**, with appropriate legal authorization and role-based permissions.

However, it's important to note that a **basic electronic signature** (e.g., a scanned image or typed name) **does not encrypt or secure** the document. It offers **no guarantee of confidentiality**, and the content remains readable to any viewer. By contrast, **advanced or qualified electronic signatures**, backed by valid digital certificates, **are legally equivalent to handwritten signatures** under EU regulations like eIDAS and are enforceable in courts.

## 2.4 – Digital vs. Handwritten Signature

**Handwritten signatures** are visually recognizable marks that can be forged or copied through manual or digital reproduction. Although they carry legal weight, their verification often requires forensic expertise, especially in dispute resolution scenarios.

**Digital signatures**, on the other hand, are **mathematically generated codes** that bind the signer's identity to the content of the document. They are tamper-evident and **non-transferable**—a signature created for one document cannot be reused or applied to another without invalidating the cryptographic binding.

In traditional contracts, signatures may appear on the final page, leaving earlier pages vulnerable to tampering. **Digital signatures cover the entire content**, such that **even a single character change** will result in verification failure. This ensures **document integrity** and prevents post-signature modifications.

Furthermore, **digital signatures are verifiable by anyone**, without relying on expert knowledge. Verification tools are widely available in PDF readers, document management systems, and email clients, making this technology both **accessible and transparent**.

## 2.5 – Purpose of Document Signing

Document signing serves multiple essential purposes in legal, organizational, and technical contexts:

- **Proof of Origin:** Confirms the **identity of the signer**. A unique signature can be attributed to an individual based on exclusive access to the private key or personal signing method.
- **Legal Authority:** Indicates that the signer understands the **legal implications** of the content and is willingly assuming responsibility for it. This is crucial in contracts, government filings, and declarations.

- **Explicit Consent or Agreement:** In regulated sectors, signatures represent formal **authorization or approval**, signifying that the signer has agreed to the document's terms or decisions.
- **Operational Efficiency:** Replaces manual processes, shortens approval cycles, reduces printing/scanning costs, and speeds up decision-making—especially important in remote or large-scale operations.

**Digital signature technology surpasses traditional handwritten signing in every dimension**—from automation and traceability to security and legal defensibility. In a world increasingly reliant on digital workflows and cross-border transactions, the digital signature has become a **cornerstone of trust** in the digital economy.

## Chapter 3 – Mathematical Algorithms Used in Electronic Signatures

In practical applications, **public-key cryptographic algorithms**—while highly secure—can be computationally expensive, especially when applied to large datasets or high-frequency operations like digital signatures. To address performance limitations, cryptographic systems typically **combine public-key encryption with cryptographic hash functions**, allowing only a small, fixed-length digest of the message to be signed rather than the full message.

This hybrid approach provides strong security guarantees while optimizing performance.

### Core Steps in Secure Digital Signing and Verification:

1. **Hashing:** A **cryptographic hash function** (e.g., SHA-256) is applied to the input message or document, producing a fixed-size string called a **digest** or hash value. This digest uniquely represents the content, such that even a one-bit change in the document alters the hash completely.
2. **Signing:** The hash is then **encrypted with the sender's private key**, producing the digital signature. This step links the signature to both the content and the identity of the signer.
3. **Transmission:** The original document and its digital signature are **sent together** to the recipient.

4. **Verification:** The recipient performs three tasks:

- Recomputes the hash of the received document.
- Decrypts the received signature using the sender's public key to recover the original hash.
- Compares both hashes. If they match, the signature is deemed valid, and the document is verified as **authentic, unaltered, and signed by the rightful party**.

This mechanism provides **integrity, authentication, and non-repudiation**, which are fundamental pillars of digital security.

## **Widely Used Algorithms**

### **Hash Functions:**

Hash functions condense data into a compact representation while ensuring:

- **Determinism:** The same input always produces the same hash.
- **Preimage resistance:** It's computationally infeasible to reverse a hash back into its original input.
- **Collision resistance:** Two different inputs shouldn't produce the same hash.

Commonly used hash algorithms include:

- **MD2, MD4, MD5** – Designed by Ronald Rivest; now largely deprecated due to vulnerabilities.
- **SHA-1** – Once standard, now discouraged for cryptographic use due to collision attacks.
- **SHA-2 family (SHA-224, SHA-256, SHA-384, SHA-512)** – Currently recommended by NIST for digital signatures.
- **SHA-3** – A newer, Keccak-based algorithm, designed as a secure alternative to SHA-2.

### **Signature Algorithms:**

- **RSA** – Based on the hardness of factoring large integers.
- **ElGamal** – A probabilistic encryption algorithm used in digital signature schemes.

- **DSA (Digital Signature Algorithm)** – Standardized by NIST; uses discrete logarithms over finite fields.

Each of these schemes offers unique performance and security trade-offs. RSA, for example, is widely adopted due to its simplicity and dual-use in encryption and signing.

### 3.1 – RSA Algorithm

**RSA (Rivest–Shamir–Adleman)** is one of the most important public-key cryptosystems and a foundational technology for secure digital communications. Developed at MIT in 1977, it is considered a benchmark for cryptographic security and is used in countless systems including TLS/SSL, email encryption, software licensing, and digital signatures.

#### Security Principle:

RSA relies on the computational difficulty of the **integer factorization problem**: given a large composite number  $n$ , it is practically impossible to determine its prime factors  $p$  and  $q$  in a reasonable amount of time.

#### Mathematical Foundations:

Let's define the key parameters:

1. **p and q** – Two large prime numbers (e.g., 1024-bit or 2048-bit), randomly chosen and kept secret.
2.  **$n = p \times q$**  – The modulus used for both encryption and decryption; made public.
3.  **$\phi(n) = (p-1)(q-1)$**  – Euler's totient function, used in key generation; not publicly revealed.
4. **PRIV (Private Key)** – A number chosen such that it is relatively prime to  $\phi(n)$ .
5. **PUB (Public Key)** – The modular inverse of the private key with respect to  $\phi(n)$ , computed using the Extended Euclidean Algorithm.
6. **M** – The message or document to be signed.
7.  **$H(M)$**  – The message digest, obtained by applying a secure hash function to  $M$ .

#### Signing and Verification:

- To sign: Encrypt  $H(M)$  with **PRIV**, producing the signature  $S$ .
- To verify: Decrypt  $S$  using **PUB** and compare the result with a newly computed  $H(M)$ .

#### **Dual Use:**

Unlike DSA or ElGamal (which are dedicated to signatures), RSA can also be used for **encryption and decryption**, enabling it to support secure message exchange and key distribution.

#### **Practical Security:**

RSA's strength depends on the key size:

- A 1024-bit key is considered **minimum secure**.
- 2048-bit and 3072-bit keys are standard for high-security environments.
- Keys smaller than 1024 bits are no longer considered safe due to advances in factoring techniques and computational power.

#### **Secure File Transmission with RSA (SSFTP-style workflow)**

Though RSA itself is not a file transfer protocol, it is often used **alongside symmetric encryption** in secure systems such as **SSFTP** (Secure Signed File Transfer Protocol):

1. Generate a message digest  $H(M)$  using a secure hash function.
2. Encrypt the digest with the **sender's private key** (digital signature).
3. Encrypt the signed digest with the **recipient's public key** (confidentiality).
4. Encrypt the original document with the recipient's public key.
5. Send both the **encrypted document and encrypted signature**.
6. The **recipient decrypts** the document with their **private key**.
7. Recalculates the hash of the decrypted document.
8. Decrypts the received signature with their **private key**.
9. Further decrypts the result using the **sender's public key**.

10. Compares the hash from step 7 with the one obtained in step 9. If they match, the signature is **authentic and the document is untampered**.

The RSA algorithm remains one of the most trusted methods for digital signatures and public-key encryption. Its rigorous mathematical foundations, proven resistance to cryptanalysis, and widespread support across platforms and protocols make it a cornerstone of modern cryptographic infrastructure.

In systems like **SSFTP**, RSA demonstrates how **asymmetric cryptography**, when combined with hashing and secure key exchange, ensures end-to-end confidentiality, authenticity, and integrity in the transmission of sensitive digital content.

## Chapter 4 – How Electronic Signatures Work

Electronic signatures, while not inherently encrypting the content of the document, are built upon **cryptographic algorithms** that ensure **integrity, authenticity, and non-repudiation**. These signatures rely on mathematical principles from public-key cryptography and are governed by standardized protocols that define how the signing and verification processes occur.

### Understanding Cryptographic Algorithms

To understand how digital signatures work, let's first break down what a cryptographic algorithm does.

Imagine the message: “**Ana has apples**”. If we wanted to obscure it, a basic method might be to shift each letter by one in the alphabet, resulting in: “**Bob bsf bqqmft**”. This is known as **Caesar cipher**—a very simple form of encryption. However, such basic ciphers are trivially easy to crack.

In modern systems, we use **encryption keys** in combination with algorithms. A **key** is a string of bits (characters, numbers, symbols) that, when applied through a cryptographic algorithm, transforms readable data (plaintext) into an unreadable format (ciphertext). Only someone with the appropriate key can decrypt the message.

### Symmetric Encryption

In **symmetric cryptography**, the same key is used to both encrypt and decrypt the data. This method is fast and suitable for large data volumes, but it has a critical flaw: **key distribution**. If an unauthorized party gains access to the key, they can both decrypt and forge messages.

Common symmetric encryption algorithms include:

- **DES (Data Encryption Standard)** – now obsolete due to its limited key size.
- **AES (Advanced Encryption Standard)** – the modern standard, with 128/192/256-bit key options.

## **Asymmetric Encryption (Public-Key Cryptography)**

Used in digital signatures, **asymmetric encryption** involves two keys:

- A **private key**, known only to the signer, used to create the digital signature.
- A **public key**, distributed freely, used by others to verify the signature.

This model ensures that:

- Only the holder of the private key can sign a message.
- Anyone can verify the authenticity of the signature using the public key.  
This forms the foundation of public-key algorithms like **RSA** and **DSA**, which are secure due to the mathematical difficulty of problems such as **factoring large numbers** or **solving discrete logarithms**.

Asymmetric cryptography is computationally intensive, so it's generally used **only to sign a hash** of a message, not the full message itself.

## **Creating and Verifying a Digital Signature**

The digital signature process combines hashing and asymmetric encryption.

### **Steps to Sign a Document:**

1. **Hashing the document:** A **cryptographic hash function** (like SHA-256) converts the document into a fixed-length digest (e.g., a 64-character alphanumeric string). This **digest is unique** to the content—any change in the document results in a completely different digest.

2. **Encrypting the hash with the private key:** The signer encrypts the digest using their **private key**, creating the digital signature. This binds the signature to both the document content and the identity of the signer.
3. **Transmitting the document:** The **original document and the digital signature** are sent together to the recipient.

#### **Steps to Verify the Signature:**

1. The recipient **decrypts the signature** using the sender's **public key**, extracting the original hash.
2. They **recalculate the hash** of the received document.
3. The two hashes are **compared**:
  - If they **match**, the document is authentic, has not been tampered with, and was signed by the private key holder.
  - If they **don't match**, the document has been altered or the signature is forged.

This process guarantees:

- **Integrity** – The data hasn't changed.
- **Authentication** – The signer's identity is confirmed.
- **Non-repudiation** – The signer cannot later deny their involvement.

#### **Key Management in Practice**

Every user in a digital signature system maintains a **key pair**:

- The **public key** can be freely shared (via digital certificates).
- The **private key** must be kept **secret and secure**.

Key properties:

- It should be **computationally infeasible** to derive the private key from the public one.
- The system must implement **key generation**, **digital signing**, and **signature verification** algorithms securely.

## Smart Cards and Hardware Security

For higher assurance, private keys should **not be stored on standard hard drives**. Instead:

- **Smart cards or USB tokens** can securely store private keys.
- These devices often require a **PIN code** for activation, enabling **two-factor authentication**.
- During signing, the document's hash is sent to the card, which performs the encryption internally and returns the signature—preventing direct access to the key.

## 4.1 – Role of Digital Signatures

In an increasingly digital economy, digital signatures play a **central role in establishing trust online**. Their importance extends beyond technical validation to **legal and operational assurance**.

### Why Digital Signatures Matter:

- **Verify Identity:** Recipients can confirm the signer's identity via their public key and certificate.
- **Ensure Integrity:** Any unauthorized modification of the document invalidates the signature.
- **Prevent Denial:** Since only the private key can create the signature, the signer **cannot deny** having signed it.

They are now foundational in:

- **Online banking and financial transactions**
- **E-commerce contracts and customer agreements**
- **Secure email communication (S/MIME, PGP)**
- **Government e-services**
- **Blockchain and cryptocurrency transactions**

In Romania, **digital certificates** issued by accredited providers enable individuals and businesses to sign documents with **full legal equivalence to handwritten signatures**, under the eIDAS regulation.

#### **Core Security Objectives Satisfied:**

- **Confidentiality** – Only intended parties can access sensitive parts of a transaction.
- **Authentication** – The origin of the message can be reliably identified.
- **Integrity** – Any tampering is immediately detectable.
- **Non-repudiation** – The signer cannot later deny the action.
- **Selective Disclosure** – Only specific parts of a document (e.g., credit card number) can be encrypted or signed.

## 4.2 – Advantages of Using Digital Signatures

Digital signatures offer major **technical, operational, and legal benefits** that make them an essential component of digital transformation:

#### **Key Benefits:**

- **Paperless operations:** Streamline workflows, eliminate printing and scanning.
- **Cost efficiency:** Reduce logistics, storage, and administrative costs.
- **Speed:** Enable instant approval processes and remote validation.
- **Enhanced cybersecurity:** Ensure secure communication across all digital platforms.

#### **Protocol Guarantees:**

- **Authenticity** – Each signature is verifiable via a publicly available key.
- **Forgery resistance** – Only the signer's private key can create a valid signature.
- **One-time use** – A signature is **bound to a specific document hash**, making it **non-reusable**.
- **Tamper detection** – Even minor changes to the document invalidate the signature.

- **Independent verification** – The recipient can validate the signature **without further input from the sender**.

Digital signatures provide a robust, legally recognized, and technologically advanced solution for verifying identity and ensuring the integrity of digital communications. As societies move toward paperless, secure, and efficient digital ecosystems, digital signatures stand out as an indispensable tool for trust, compliance, and security.

## Conclusion

**Cryptographic hash functions** and **public-key cryptography** form the foundational pillars of modern digital signature systems. Together, they provide the mathematical and security mechanisms that allow for reliable identity verification, data integrity, and non-repudiation in the digital realm. When correctly implemented, digital signatures become a powerful tool for securing communications, protecting sensitive information, and verifying the authenticity of digital content across virtually all industries.

In **blockchain technology**, digital signatures play an especially vital role. Every transaction on a blockchain—whether in Bitcoin, Ethereum, or other decentralized systems—is signed using a private key and verified with a corresponding public key. This ensures that **only the rightful owner of the private key can authorize the transfer of digital assets**, thereby preventing fraud and ensuring trust in a trustless environment.

Despite the increasing usage of electronic and digital signatures in areas such as **e-government, e-commerce, online banking, and healthcare**, many institutions still rely on traditional, paper-based workflows. This reliance limits efficiency and introduces vulnerabilities in document handling and verification.

However, as the world continues to shift toward **digital-first infrastructure**, we can expect to see broader adoption of **advanced and qualified electronic signature schemes**. These technologies not only streamline administrative processes but also offer stronger legal protection, automation potential, and compliance with international regulations such as **eIDAS** in the EU or **ESIGN** and **UETA** in the United States.

Looking ahead, the integration of digital signatures with technologies like **smart contracts**, **IoT**, and **cloud-native platforms** will further expand their application, driving the transition to **secure, paperless, and decentralized systems**. Ultimately, digital signatures are not just a convenience—they are a cornerstone of trust in the modern digital society.

## Bibliography

[https://ro.wikipedia.org/wiki/Securitate\\_\(informatică\)](https://ro.wikipedia.org/wiki/Securitate_(informatică))

[https://ro.wikipedia.org/wiki/Semnătură\\_digitală](https://ro.wikipedia.org/wiki/Semnătură_digitală)

<https://startco.ro/blog/ce-este-semnatura-electronica-digitala/#h-cine-poate-sa-semnize-electronic>

<https://revistaie.ase.ro/content/23/ivan.pdf>

[DENN82] Denning, Dorothy E. *Cryptography and Data Security*. Addison-Wesley, New York, 1982.

Patriciu, Victor Valeriu și colectiv. *Semnături Electronice și Securitatea Informatică*. Editura Bic All, 2006.

Stallings, William. *Cryptography and Network Security: Principles and Practice*. Pearson Education, 7th Edition, 2017.

Schneier, Bruce. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley, 2nd Edition, 1996.

National Institute of Standards and Technology (NIST). *FIPS PUB 186-4: Digital Signature Standard (DSS)*, July 2013.

Rivest, R., Shamir, A., Adleman, L. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM*, 1978.

European Union. *Regulation (EU) No 910/2014 (eIDAS Regulation)* – On electronic identification and trust services for electronic transactions in the internal market.