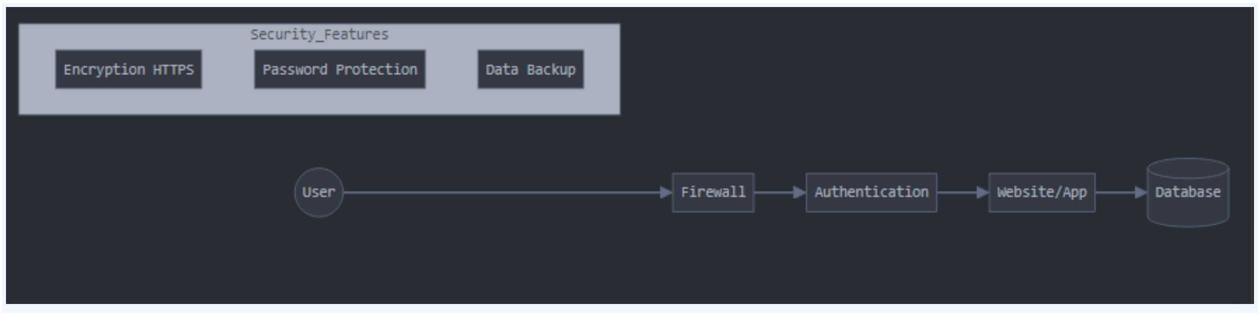
The Web Security



Web Security refers to the protective measures and protocols implemented to safeguard websites, web applications, and web services from cyber threats, unauthorized access, and data breaches. It ensures the confidentiality, integrity, and availability of data transmitted over the internet. Key aspects of web security include:

1. **Authentication and Authorization**:
   - Ensures only legitimate users can access the system (authentication) and restricts their access based on permissions (authorization).
   - Example: Using passwords, multi-factor authentication (MFA), and role-based access control (RBAC).
2. **Data Encryption**:
   - Protects sensitive data (e.g., passwords, credit card details) by encrypting it during transmission.
   - Example: Using HTTPS (SSL/TLS) to secure communication between the client and server.
3. **Input Validation and Sanitization**:
   - Prevents malicious input (e.g., SQL injection, XSS) by validating and sanitizing user inputs.
   - Example: Using parameterized queries and escaping special characters.
4. **Secure Coding Practices**:
   - Writing code that minimizes vulnerabilities and follows security best practices.
   - Example: Regularly updating software, avoiding hardcoded credentials, and using secure libraries.
5. **Firewalls and Intrusion Detection Systems (IDS)**:
   - Monitors and controls incoming and outgoing network traffic to block malicious activities.
   - Example: Web Application Firewalls (WAF) to filter out harmful requests.
6. **Regular Security Audits and Testing**:

    - Identifies vulnerabilities through penetration testing, code reviews, and vulnerability scanning.
    - Example: Using tools like OWASP ZAP or Burp Suite.
7. **Session Management**:
    - Ensures secure handling of user sessions to prevent session hijacking or fixation.
    - Example: Using secure cookies, session timeouts, and regenerating session IDs after login.
8. **Cross-Site Request Forgery (CSRF) Protection**:
    - Prevents unauthorized actions performed on behalf of an authenticated user.
    - Example: Using anti-CSRF tokens in forms.
9. **Content Security Policy (CSP)**:
    - Mitigates risks of cross-site scripting (XSS) by controlling which resources can be loaded.
    - Example: Restricting the execution of inline scripts.
10. **Backup and Recovery**:
    - Ensures data can be restored in case of a breach or data loss.
    - Example: Regularly backing up databases and files to secure locations.

Importance of Web Security:

- Protects sensitive user data (e.g., personal information, financial details).
- Maintains user trust and credibility.
- Prevents financial losses and legal consequences due to breaches.
- Ensures compliance with regulations like GDPR, HIPAA, etc.

The Web Security Problem

The **Web Security Problem** refers to the challenges and vulnerabilities associated with securing websites, web applications, and web services from cyber threats, attacks, and unauthorized access. As the internet has become an integral part of modern life, the risks associated with web-based systems have grown significantly. These risks stem from the complexity of web technologies, the increasing sophistication of attackers, and the vast amount of sensitive data transmitted and stored online.

Key Aspects of the Web Security Problem

1. **Increasing Complexity of Web Applications**:
    - Modern web applications are complex, often built using multiple frameworks, libraries, and third-party components.
    - This complexity increases the likelihood of vulnerabilities, such as insecure APIs, misconfigurations, or outdated dependencies.
2. **Sophistication of Cyber Attacks**:
    - Attackers are constantly evolving their techniques to exploit vulnerabilities.
    - Common attacks include:
        - **SQL Injection**: Exploiting database vulnerabilities to access or manipulate data.
        - **Cross-Site Scripting (XSS)**: Injecting malicious scripts into web pages viewed by users.
        - **Cross-Site Request Forgery (CSRF)**: Forcing users to perform unwanted actions on a web application.
        - **Distributed Denial of Service (DDoS)**: Overwhelming a website with traffic to make it unavailable.
3. **Human Factor**:
    - Many security breaches occur due to human error, such as weak passwords, misconfigured servers, or falling for phishing attacks.
    - Developers may also inadvertently introduce vulnerabilities through insecure coding practices.
4. **Data Sensitivity and Privacy Concerns**:
    - Websites and applications often handle sensitive data, such as personal information, financial details, and health records.
    - Breaches can lead to identity theft, financial losses, and reputational damage.
    - Compliance with data protection regulations (e.g., GDPR, CCPA) adds another layer of complexity.
5. **Third-Party Risks**:
    - Many websites rely on third-party services (e.g., payment gateways, analytics tools) that may introduce vulnerabilities.
    - A breach in a third-party service can compromise the entire system.
6. **Lack of Security Awareness**:
    - Many organizations and individuals lack awareness of web security best practices.

    - This leads to poor implementation of security measures, such as failing to use HTTPS or not updating software regularly.
7. **Rapidly Changing Threat Landscape**:
    - New vulnerabilities and attack vectors are discovered regularly, making it challenging to stay ahead of threats.
    - Zero-day exploits (unknown vulnerabilities) pose a significant risk.

Consequences of the Web Security Problem

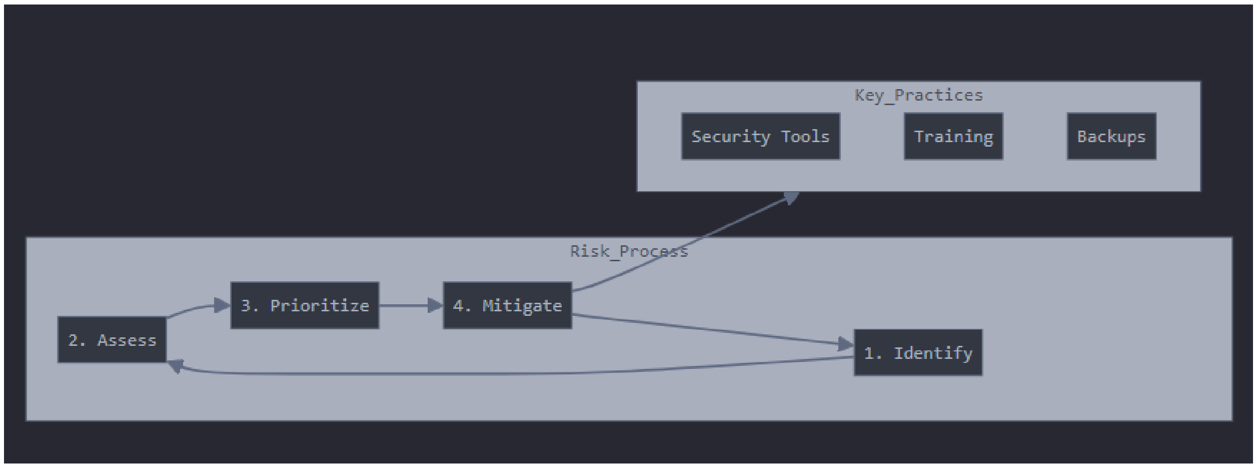
1. **Data Breaches**:
    - Unauthorized access to sensitive data can result in financial losses, legal penalties, and loss of customer trust.
2. **Financial Losses**:
    - Organizations may face direct financial losses due to theft, fraud, or downtime caused by attacks.
3. **Reputational Damage**:
    - A security breach can damage an organization's reputation, leading to loss of customers and business opportunities.
4. **Legal and Regulatory Consequences**:
    - Non-compliance with data protection laws can result in hefty fines and legal action.
5. **Disruption of Services**:
    - Attacks like DDoS can render websites and services unavailable, causing operational disruptions.

Addressing the Web Security Problem

To mitigate the web security problem, organizations and developers must adopt a proactive approach, including:

- Implementing secure coding practices.
- Regularly updating and patching software.
- Conducting security audits and penetration testing.
- Educating users and employees about security best practices.
- Using encryption (e.g., HTTPS) to protect data in transit.
- Deploying firewalls, intrusion detection systems, and web application firewalls (WAFs).

Risk Analysis and Best Practices



1. Risk Analysis

Risk Analysis is a systematic process of identifying, assessing, and prioritizing risks to an organization's assets, such as data, systems, and infrastructure. The goal is to understand potential threats, their impact, and the likelihood of occurrence, enabling organizations to implement effective mitigation strategies.

Steps in Risk Analysis:

1. **Identify Assets**:
   - Determine what needs to be protected (e.g., databases, servers, user data, intellectual property).
2. **Identify Threats**:
   - Recognize potential threats, such as hackers, malware, phishing, DDoS attacks, or insider threats.
3. **Identify Vulnerabilities**:
   - Find weaknesses in the system that could be exploited (e.g., unpatched software, weak passwords, misconfigurations).
4. **Assess Impact**:
   - Evaluate the potential consequences of a security breach (e.g., financial loss, reputational damage, legal penalties).
5. **Assess Likelihood**:
   - Determine the probability of a threat exploiting a vulnerability.
6. **Prioritize Risks**:
   - Rank risks based on their impact and likelihood to focus on the most critical ones.

7. **Develop Mitigation Strategies**:
    - Implement controls to reduce the likelihood or impact of risks (e.g., firewalls, encryption, access controls).

Tools for Risk Analysis:

- **Qualitative Analysis**: Uses descriptive scales (e.g., low, medium, high) to assess risks.
- **Quantitative Analysis**: Uses numerical values to calculate risk (e.g., Annualized Loss Expectancy).
- **Risk Matrices**: Visual tools to prioritize risks based on impact and likelihood.

---

2. Best Practices in Risk Management

To mitigate risks effectively, organizations should follow these **best practices**:

---

1. Implement a Risk Management Framework:

- Use established frameworks like **NIST Cybersecurity Framework** or **ISO 27001** to guide risk management processes.

2. Regular Risk Assessments:

- Conduct periodic risk assessments to identify new threats and vulnerabilities.

3. Secure Configuration:

- Harden systems by disabling unnecessary services, applying patches, and using secure configurations.

4. Access Control:

- Implement role-based access control (RBAC) and follow the principle of least privilege (PoLP).

5. Data Encryption:

- Encrypt sensitive data at rest and in transit to protect it from unauthorized access.

6. Employee Training:

- Educate employees about security risks, such as phishing, social engineering, and password hygiene.

7. Incident Response Plan:

- Develop and test an incident response plan to quickly address security breaches.

8. Backup and Recovery:

- Regularly back up data and test recovery procedures to ensure business continuity.

9. Monitor and Audit:

- Implement logging and monitoring to detect suspicious activities.
- Conduct regular security audits to identify and fix vulnerabilities.

10. Use Security Tools:

- Deploy firewalls, intrusion detection systems (IDS), and antivirus software to protect against threats.

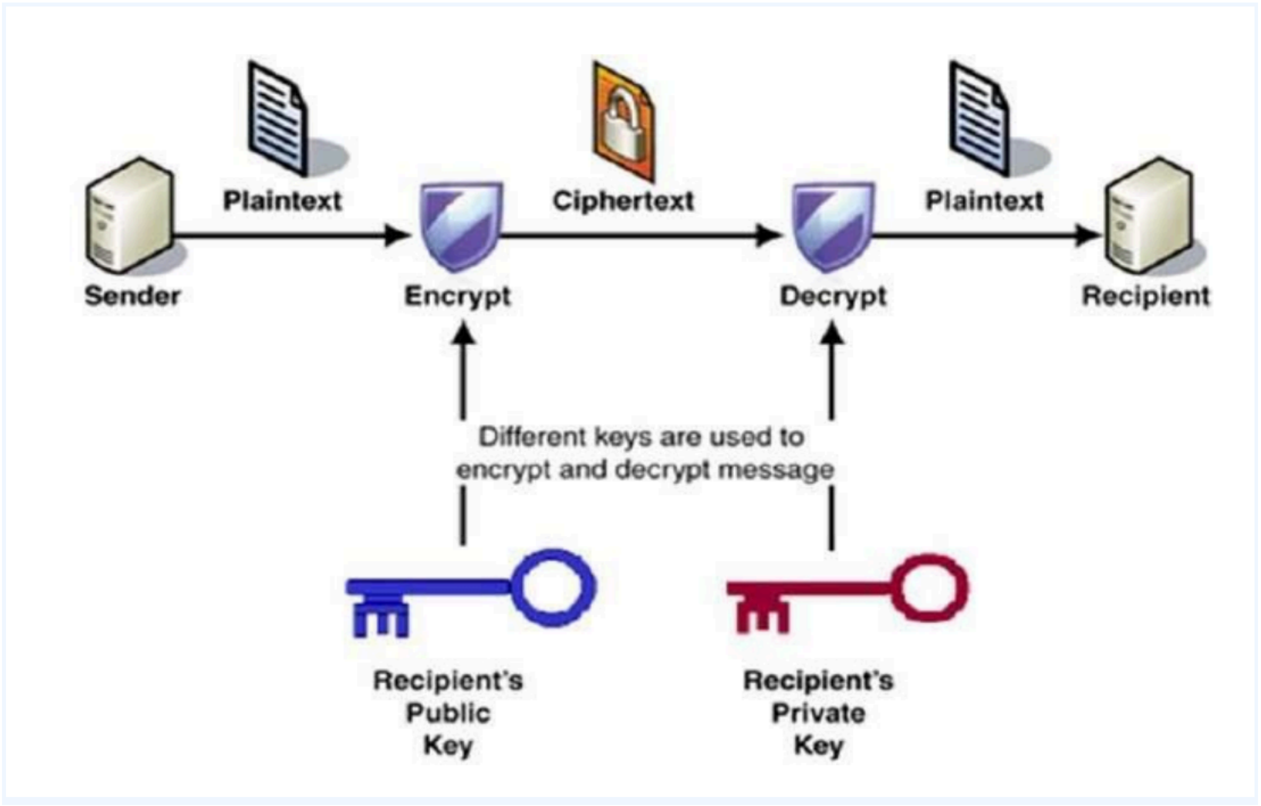
---

3. Importance of Risk Analysis and Best Practices

- **Proactive Approach**: Helps identify and address vulnerabilities before they are exploited.
- **Cost-Effective**: Reduces the financial impact of security breaches.
- **Compliance**: Ensures adherence to regulatory requirements (e.g., GDPR, HIPAA).
- **Reputation Management**: Protects the organization's reputation by preventing data breaches.
- **Business Continuity**: Ensures the availability and reliability of systems and services.

Cryptography and the Web: Cryptography and Web Security

Cryptography plays a vital role in ensuring the security of data transmitted and stored on the web. It provides the foundation for confidentiality, integrity, authentication, and non-repudiation in web-based systems. By leveraging cryptographic techniques, web applications can protect sensitive information from unauthorized access, tampering, and other cyber threats.



---

1. What is Cryptography?

Cryptography is the science of securing communication and data through the use of mathematical algorithms and protocols. It transforms plaintext (readable data) into ciphertext (unreadable data) using encryption and reverses the process using decryption. The primary goals of cryptography in web security are:

1. **Confidentiality**: Ensuring that data is accessible only to authorized parties.
2. **Integrity**: Ensuring that data has not been altered or tampered with during transmission or storage.
3. **Authentication**: Verifying the identity of users or systems.
4. **Non-Repudiation**: Ensuring that a party cannot deny the authenticity of their actions or communications.

---

2. Cryptographic Techniques Used in Web Security

a. Symmetric Key Cryptography:

- Uses a single shared key for both encryption and decryption.

- **Example Algorithms**: AES (Advanced Encryption Standard), DES (Data Encryption Standard).
- **Use Case**: Encrypting large amounts of data efficiently (e.g., HTTPS for secure communication).

b. Asymmetric Key Cryptography (Public Key Cryptography):

- Uses a pair of keys: a public key for encryption and a private key for decryption.
- **Example Algorithms**: RSA (Rivest-Shamir-Adleman), ECC (Elliptic Curve Cryptography).
- **Use Case**: Secure key exchange (e.g., SSL/TLS handshake), digital signatures, and encryption of small data.

c. Hash Functions:

- Converts data into a fixed-size hash value, which is irreversible.
- **Example Algorithms**: SHA-256 (Secure Hash Algorithm), MD5 (Message Digest Algorithm).
- **Use Case**: Ensuring data integrity (e.g., verifying file integrity, storing passwords securely).

d. Digital Signatures:

- Combines hashing and asymmetric cryptography to verify the authenticity and integrity of data.
- **Use Case**: Authenticating software updates, ensuring non-repudiation in transactions.

e. Certificates and Public Key Infrastructure (PKI):

- Uses digital certificates issued by Certificate Authorities (CAs) to verify the identity of entities.
- **Use Case**: Establishing trust in SSL/TLS for secure web communication.

---

3. Cryptography in Web Security

Cryptography is integral to various aspects of web security, including:

a. Secure Communication (HTTPS):

- HTTPS uses SSL/TLS protocols to encrypt data transmitted between a client (browser) and a server.
- **How it works**:
    1. The server presents its digital certificate to the client.

2. The client verifies the certificate with a trusted CA.
3. A secure session is established using symmetric encryption for data transfer.

b. Password Storage:

- Passwords are hashed (not encrypted) before being stored in databases.
- **Best Practice**: Use salted hashes (e.g., bcrypt, PBKDF2) to protect against rainbow table attacks.

c. Data Integrity:

- Hash functions are used to verify that data has not been tampered with during transmission.
- **Example**: Checksums for file downloads.

d. Authentication:

- Cryptographic protocols like OAuth and OpenID Connect use tokens and digital signatures to authenticate users.

e. Secure Key Exchange:

- Protocols like Diffie-Hellman enable secure key exchange over insecure channels, ensuring confidentiality.

---

4. Challenges in Cryptographic Web Security

1. **Key Management**: Safely generating, storing, and distributing cryptographic keys is complex.
2. **Performance Overhead**: Encryption and decryption can introduce latency, especially for large datasets.
3. **Quantum Computing Threat**: Quantum computers could potentially break current cryptographic algorithms (e.g., RSA, ECC).
4. **Implementation Flaws**: Poor implementation of cryptographic algorithms can lead to vulnerabilities.

---

5. Best Practices for Cryptographic Web Security

1. **Use Strong Algorithms**: Prefer AES-256, SHA-256, and RSA-2048 or higher.
2. **Keep Software Updated**: Regularly update cryptographic libraries and frameworks.

3. **Implement HTTPS**: Use SSL/TLS to encrypt all web traffic.
4. **Secure Key Storage**: Use hardware security modules (HSMs) or secure key management systems.
5. **Follow Standards**: Adhere to cryptographic standards like NIST, FIPS, and OWASP.
6. **Regular Audits**: Conduct security audits to identify and fix cryptographic vulnerabilities.

---

6. Importance of Cryptography in Web Security

- **Protects Sensitive Data**: Ensures confidentiality of personal, financial, and business data.
- **Builds Trust**: Enhances user confidence in web applications and services.
- **Ensures Compliance**: Helps meet regulatory requirements (e.g., GDPR, HIPAA).
- **Prevents Attacks**: Mitigates risks of data breaches, man-in-the-middle attacks, and tampering.

—--

Working Cryptographic Systems and Protocols

Cryptographic systems and protocols are essential for ensuring secure communication, data integrity, and authentication in modern web and network environments. They provide the foundation for protecting sensitive information from unauthorized access, tampering, and other cyber threats.

---

1. Cryptographic Systems

Cryptographic systems are frameworks that use mathematical algorithms and protocols to secure data. They can be broadly categorized into two types:

a. Symmetric Key Cryptography:

- Uses a single shared key for both encryption and decryption.
- **Advantages**: Fast and efficient for encrypting large amounts of data.
- **Disadvantages**: Key distribution and management can be challenging.
- **Examples**:
    - **AES (Advanced Encryption Standard)**: Widely used for securing data.

    - **DES (Data Encryption Standard)**: Older standard, now considered insecure.

b. Asymmetric Key Cryptography (Public Key Cryptography):

- Uses a pair of keys: a public key for encryption and a private key for decryption.
- **Advantages**: Solves the key distribution problem; enables digital signatures.
- **Disadvantages**: Slower than symmetric cryptography.
- **Examples**:
    - **RSA (Rivest-Shamir-Adleman)**: Used for encryption and digital signatures.
    - **ECC (Elliptic Curve Cryptography)**: Provides strong security with smaller key sizes.

---

2. Cryptographic Protocols

Cryptographic protocols are sets of rules and procedures that use cryptographic systems to achieve specific security goals. They are widely used in web security, network communication, and data protection.

---

a. SSL/TLS (Secure Sockets Layer / Transport Layer Security):

- **Purpose**: Secures communication over the internet by encrypting data between a client (e.g., browser) and a server.
- **How it works**:
    1. The client and server perform a handshake to establish a secure connection.
    2. The server presents its digital certificate, which is verified by the client.
    3. A symmetric session key is generated for encrypting data during the session.
- **Use Case**: HTTPS for secure web browsing.

b. IPsec (Internet Protocol Security):

- **Purpose**: Secures IP communication by encrypting and authenticating data packets.
- **Components**:
    - **AH (Authentication Header)**: Ensures data integrity and authenticity.
    - **ESP (Encapsulating Security Payload)**: Provides encryption and integrity.
- **Use Case**: VPNs (Virtual Private Networks) for secure remote access.

c. SSH (Secure Shell):

- **Purpose**: Provides secure remote access to systems and secure file transfers.
- **How it works**: Uses asymmetric cryptography for authentication and symmetric cryptography for encrypting data.
- **Use Case**: Remote server administration, secure file transfer (SFTP).

d. PGP/GPG (Pretty Good Privacy / GNU Privacy Guard):

- **Purpose**: Encrypts and signs emails and files to ensure confidentiality and authenticity.
- **How it works**: Combines symmetric and asymmetric cryptography.
- **Use Case**: Secure email communication, file encryption.

e. Kerberos:

- **Purpose**: Provides authentication in a network environment using tickets.
- **How it works**: Uses symmetric cryptography and a trusted third party (Key Distribution Center).
- **Use Case**: Authentication in Windows domains and enterprise networks.

f. OAuth and OpenID Connect:

- **Purpose**: Enables secure authorization and authentication for web applications.
- **How it works**: Uses tokens (e.g., JWT) to grant access to resources without sharing credentials.
- **Use Case**: Single sign-on (SSO) for web applications.

---

3. Components of Cryptographic Systems and Protocols

1. **Encryption Algorithms**:
   - Transform plaintext into ciphertext (e.g., AES, RSA).
2. **Hash Functions**:
   - Generate fixed-size hash values for data integrity (e.g., SHA-256).
3. **Digital Signatures**:
   - Verify the authenticity and integrity of data using asymmetric cryptography.
4. **Certificates and PKI (Public Key Infrastructure)**:
   - Use digital certificates issued by Certificate Authorities (CAs) to establish trust.
5. **Key Management**:
   - Safely generate, store, distribute, and revoke cryptographic keys.

---

4. Importance of Cryptographic Systems and Protocols

- **Confidentiality**: Ensures that data is accessible only to authorized parties.
- **Integrity**: Protects data from tampering or unauthorized modification.
- **Authentication**: Verifies the identity of users or systems.
- **Non-Repudiation**: Prevents parties from denying their actions or communications.
- **Secure Communication**: Enables safe transmission of data over insecure networks (e.g., the internet).

---

5. Challenges in Cryptographic Systems and Protocols

1. **Key Management**: Safely generating, storing, and distributing keys is complex.
2. **Performance Overhead**: Encryption and decryption can introduce latency.
3. **Quantum Computing Threat**: Quantum computers could potentially break current cryptographic algorithms (e.g., RSA, ECC).
4. **Implementation Flaws**: Poor implementation can lead to vulnerabilities (e.g., Heartbleed in OpenSSL).

---

6. Best Practices for Using Cryptographic Systems and Protocols

1. **Use Strong Algorithms**: Prefer AES-256, SHA-256, and RSA-2048 or higher.
2. **Keep Software Updated**: Regularly update cryptographic libraries and frameworks.
3. **Implement HTTPS**: Use SSL/TLS to encrypt all web traffic.
4. **Secure Key Storage**: Use hardware security modules (HSMs) or secure key management systems.
5. **Follow Standards**: Adhere to cryptographic standards like NIST, FIPS, and OWASP.
6. **Regular Audits**: Conduct security audits to identify and fix cryptographic vulnerabilities.

—---

Legal Restrictions on Cryptography

Cryptography is a powerful tool for ensuring data security, privacy, and integrity. However, its use is subject to various legal restrictions and regulations worldwide. These restrictions are often imposed by governments to balance the benefits of cryptography with concerns related to national security, law enforcement, and public safety.

---

1. Reasons for Legal Restrictions

Governments impose legal restrictions on cryptography for several reasons:

1. **National Security**:
    - Strong encryption can be used by malicious actors (e.g., terrorists, criminals) to hide their activities, making it difficult for law enforcement and intelligence agencies to monitor and prevent threats.
2. **Law Enforcement**:
    - Encryption can hinder criminal investigations by preventing access to encrypted communications or data.
3. **Export Control**:
    - Cryptographic technologies are considered dual-use goods (i.e., they can be used for both civilian and military purposes). Governments regulate their export to prevent them from falling into the wrong hands.
4. **Public Safety**:
    - Restrictions may be imposed to ensure that encryption does not compromise public safety (e.g., by enabling illegal activities).

---

2. Types of Legal Restrictions

Legal restrictions on cryptography vary by country and can include:

a. Export Controls:

- Many countries regulate the export of cryptographic software and hardware to prevent their use by hostile entities.
- **Examples**:
    - **United States**: Historically, cryptographic software was classified as a munition under the International Traffic in Arms Regulations (ITAR). Today, restrictions are more relaxed but still exist under the Export Administration Regulations (EAR).
    - **Wassenaar Arrangement**: An international agreement that controls the export of dual-use technologies, including cryptography.

b. Key Escrow and Backdoors:

- Some governments require the use of key escrow systems or backdoors, where a copy of encryption keys is held by a trusted third party (e.g., government agency).
- **Purpose**: To allow law enforcement to access encrypted data when necessary.
- **Controversy**: Critics argue that such systems weaken security and create vulnerabilities that can be exploited by malicious actors.

c. Restrictions on Encryption Strength:

- Some countries impose limits on the strength of encryption that can be used domestically.
- **Example**:
  - **France (Historically)**: Required authorization for the use of strong encryption.

d. Mandatory Disclosure of Encryption Keys:

- In some jurisdictions, individuals or organizations can be compelled to disclose encryption keys to law enforcement.
- **Example**:
  - **United Kingdom**: Under the Regulation of Investigatory Powers Act (RIPA), failure to disclose encryption keys can result in legal penalties.

e. Bans on Cryptography:

- A few countries have outright bans or severe restrictions on the use of cryptography.
- **Example**:
  - **China**: Requires companies to provide decryption capabilities to the government upon request.

---

3. Impact of Legal Restrictions

1. **On Individuals and Businesses**:
   - Restrictions can limit the ability to use strong encryption, potentially compromising privacy and security.
   - Compliance with regulations can increase costs and complexity for businesses.
2. **On Innovation**:
   - Strict regulations can stifle innovation in cryptographic technologies and cybersecurity.
3. **On Global Trade**:

    - Export controls can create barriers to international trade in cryptographic products and services.
4. **On Privacy and Human Rights**:
    - Restrictions that weaken encryption can infringe on individuals' rights to privacy and freedom of expression.

---

4. Balancing Security and Privacy

The debate over legal restrictions on cryptography revolves around finding a balance between:

- **Security**: Enabling law enforcement and intelligence agencies to protect national security.
- **Privacy**: Protecting individuals' rights to privacy and secure communication.

Organizations like the **Electronic Frontier Foundation (EFF)** and **Internet Society** advocate for strong encryption and oppose backdoors, arguing that they undermine security and privacy.

---

5. Examples of Legal Frameworks

1. **United States**:
    - **Export Administration Regulations (EAR)**: Govern the export of cryptographic software and hardware.
    - **Communications Assistance for Law Enforcement Act (CALEA)**: Requires telecommunications providers to ensure their systems can be accessed by law enforcement.
2. **European Union**:
    - **General Data Protection Regulation (GDPR)**: Encourages the use of encryption to protect personal data but does not impose specific restrictions.
3. **India**:
    - **Information Technology Act, 2000**: Allows the government to intercept and monitor encrypted communications under certain conditions.
4. **China**:
    - **Cybersecurity Law**: Requires companies to provide decryption capabilities to the government upon request.

---

6. Best Practices for Compliance

1. **Understand Local Laws**:
   - Stay informed about cryptographic regulations in the countries where you operate.
2. **Use Compliant Solutions**:
   - Ensure that cryptographic products and services comply with local laws and international standards.
3. **Implement Strong Security**:
   - Use strong encryption where legally permitted to protect data and communications.
4. **Engage with Policymakers**:
   - Advocate for balanced policies that protect both security and privacy.

Digital Identification in Web Security

Digital Identification (Digital ID) is a crucial aspect of web security, ensuring that users, devices, and systems can be securely authenticated and authorized. It involves the use of cryptographic techniques to verify identities in digital environments.

Key Components of Digital Identification:

1. **Authentication:**
   - The process of verifying a user's identity using credentials such as usernames, passwords, biometrics, or multi-factor authentication (MFA).
2. **Authorization:**
   - Once authenticated, the system grants appropriate access permissions based on user roles and privileges.
3. **Digital Certificates & Public Key Infrastructure (PKI):**
   - Digital certificates (issued by Certificate Authorities) help in authenticating websites and encrypting communications using SSL/TLS protocols.
4. **Biometric Authentication:**
   - Uses fingerprint scanning, facial recognition, or iris scans for secure access.
5. **Two-Factor & Multi-Factor Authentication (2FA/MFA):**
   - Enhances security by requiring multiple verification methods (e.g., password + OTP or fingerprint).
6. **Blockchain-Based Identity:**

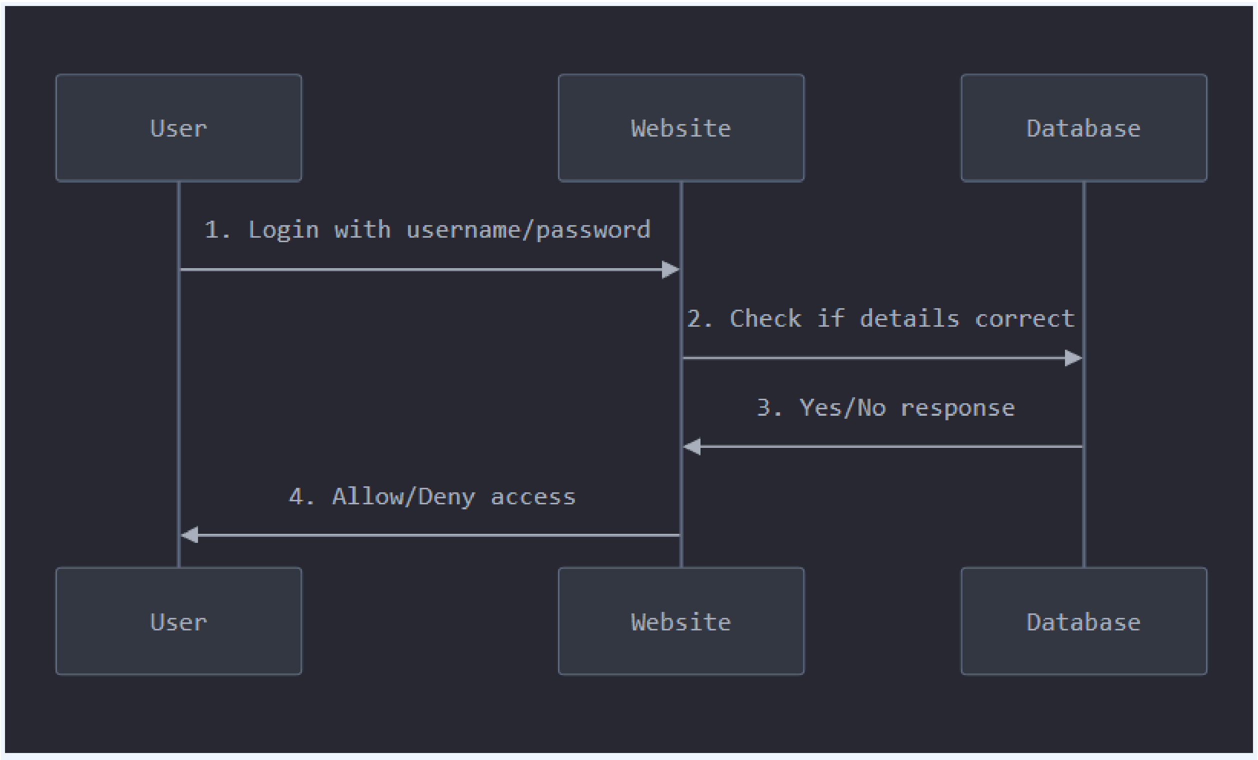
- Uses decentralized ledgers to create secure, tamper-proof digital identities.

Importance of Digital Identification in Web Security:

- **Prevents unauthorized access** to sensitive data.
- **Protects against identity theft** and phishing attacks.
- **Ensures secure online transactions** in banking and e-commerce.
- **Verifies website legitimacy** to prevent fraud.

Challenges in Digital Identification:

- Identity theft and data breaches.
- Weak password security leading to account compromise.
- Privacy concerns regarding biometric data.
- Phishing and social engineering attacks.


User
Website
Database
1. Login with username/password
2. Check if details correct
3. Yes/No response
4. Allow/Deny access
User
Website
Database