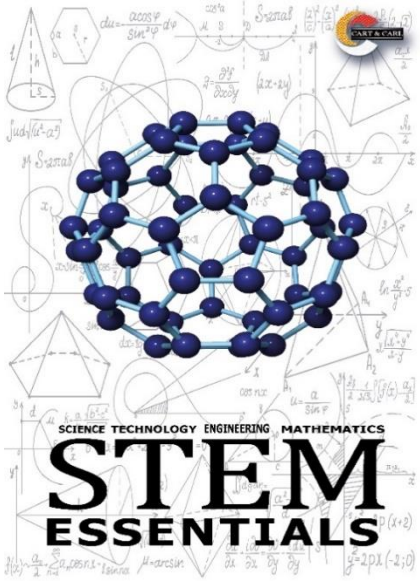
Original Research

STEM ESSENTIALS
https://cartcarl.com/journal/stem-essentials

E-ISSN: 3121-956X



An Improved Data Warehouse Security System Using Hybrid Authentication Techniques

Authors
[a] Zeteh, K. G. [a] Alabi, A. O.

[a]Centre for Information and Telecommunication Engineering, Faculty of Engineering, University of Port Harcourt, Nigeria

Corresponding Author
Zeteh, K. G
(kadilobari.zeteh@uniport.edu.ng)

Received: 15 January 2026
Accepted: 03 February 2026
Published: 05 February 2026

Citation
Zeteh, K. G. and Alabi, A. O. (2026). An Improved Data Warehouse Security System Using Hybrid Authentication Techniques. *STEM Essentials, 2*(1), 06 - 13. *https://doi.org/10.70726/STEM-E.2026.956X006*

Abstract

Data warehouses play a critical role in modern organizations by supporting data-driven decision-making and business intelligence. However, the sensitive and high-value nature of data stored in these systems makes them attractive targets for security breaches, unauthorized access, and insider threats. Conventional data warehouse security approaches, which largely rely on single-factor authentication mechanisms such as usernames and passwords, are increasingly inadequate in addressing contemporary cyber threats. This study focuses on the design and implementation of an improved data warehouse security system using hybrid authentication techniques. The proposed system integrates multiple authentication factors, including knowledge-based, possession-based, and cryptographic authentication mechanisms, to enhance identity verification and access control. By combining these complementary techniques, the hybrid authentication model provides layered security that significantly reduces vulnerabilities associated with single-factor authentication. The study adopts a system design and analytical approach to demonstrate how hybrid authentication can improve data confidentiality, integrity, and availability within data warehouse environments. The findings indicate that the improved hybrid authentication-based security system offers stronger protection against unauthorized access, impersonation attacks, and credential compromise, while also supporting accountability and secure data governance. The study concludes that hybrid authentication techniques constitute an effective and scalable solution for strengthening data warehouse security in modern enterprise systems. The implementation of such systems is therefore recommended for organizations seeking to enhance the reliability and trustworthiness of their data warehousing infrastructures.

Keywords: Data warehouse, Data Integrity, Authentication System, Cyber Security

Introduction

The rapid growth of digital technologies and enterprise information systems has led organizations to rely heavily on data warehouses for strategic decision-making, business intelligence, and analytics. A data warehouse is a centralized repository designed to store integrated, historical, and subject-oriented data that supports management reporting and analytical processing (Inmon, 2005; Kimball & Ross, 2013). Because data warehouses often contain highly sensitive and mission-critical information such as financial records, customer data, and operational intelligence they have become attractive targets for cyberattacks, insider misuse, and unauthorized access. Despite their importance, many existing data warehouse systems still rely on traditional single-factor authentication mechanisms, typically based on usernames and passwords. Such mechanisms are increasingly vulnerable to modern security threats, including password theft, brute-force attacks, phishing, and credential reuse (Stallings, 2018). These vulnerabilities pose serious risks to data confidentiality, integrity, and availability, thereby

support systems. Consequently, ensuring robust authentication and access control has become a critical requirement in data warehouse security architectures.

Hybrid authentication techniques have emerged as an effective approach to addressing these challenges. Hybrid authentication involves the integration of multiple authentication factors, such as knowledge-based factors (passwords or PINs), possession-based factors (smart cards, tokens, or mobile devices), and inherence-based factors (biometrics), to strengthen identity verification (O'Gorman, 2003; Dasgupta et al., 2017). By combining two or more complementary authentication methods, hybrid systems significantly reduce the likelihood of unauthorized access, even if one authentication factor is compromised.

In the context of data warehouse environments, hybrid authentication techniques offer enhanced protection against both external and internal threats. Research has shown that multi-factor and hybrid authentication frameworks improve resistance to impersonation attacks, replay attacks, and insider abuse while supporting accountability and secure access control (Alassaf et al., 2021). Furthermore, integrating cryptographic techniques such as secure key exchange and mutual authentication within hybrid authentication frameworks further strengthens data warehouse security by ensuring secure communication and trusted user verification.

Therefore, an improved data warehouse security system based on hybrid authentication techniques is essential for safeguarding sensitive data assets in modern organizations. Such a system not only enhances security resilience but also aligns with best practices in information assurance and access management. This study focuses on designing and evaluating an improved hybrid authentication-based security model that enhances the protection, reliability, and trustworthiness of data warehouse systems.

Literature Review

In the era of data-driven decision-making, data warehouses (DWs) play a central role by providing integrated and structured data storage systems to support analytics and business intelligence. However, with increasing reliance on digital infrastructures, the security of data warehouses has become a major concern due to the sensitivity of the stored data and the evolving nature of cybersecurity threats. Traditional security models, which focus on perimeter-based defense, have proven inadequate for the complex and distributed architectures typical of modern data ecosystems. In response, researchers and practitioners have increasingly adopted hybrid security techniques, which combine encryption, access control, anonymization, blockchain, and AI-driven analytics to provide multi-layered protection.

A critical innovation in recent years is the use of purpose-based access control. Tran, Elnikety, and Zou (2025) introduced Data Guard, a secure system that allows granular access control policies based on user intent. This model enforces compliance through SQL views and supports sub-cell-level data masking, enabling tailored data visibility without compromising privacy. This aligns with the growing regulatory demand for data minimization and accountability in data access.

Another breakthrough is the integration of Dynamic Data Masking (DDM) into cloud-native platforms. For instance, Azure SQL Data Warehouse provides real-time data obfuscation without altering the underlying storage, reducing the risk of data leaks during query operations (Microsoft Azure, 2020). These advances reflect a shift toward runtime data protection mechanisms that are transparent to users but enforce strict visibility controls.

In parallel, the rise of confidential computing has introduced a hardware-based solution to secure data in use. By leveraging Trusted Execution Environments (TEEs), confidential computing ensures that data remains encrypted even during processing, only becoming accessible within a protected enclave (Wikipedia, 2025). This mitigates risks from insider threats and compromised operating systems, representing a foundational enhancement in computational privacy.

Furthermore, blockchain technology has found application in data warehousing through its inherent immutability and transparency. Ramahlosi, Ditsa, and Ntoimo (2024) proposed a blockchain-based framework for heterogeneous information systems, which assures data provenance and prevents tampering across distributed data pipelines. Blockchain's decentralized verification model provides an additional layer of integrity assurance, particularly useful in multi-stakeholder environments.

Encryption remains a core element of hybrid security strategies. End-to-end encryption, including AES-256 for data at rest and TLS for data in transit, has become a standard requirement. These protocols are complemented by anonymization methods such as k-anonymity and l-diversity, especially in non-production environments, to protect personally identifiable information (Data Science Dojo, 2023). These approaches allow realistic data usage in testing and training environments while upholding privacy standards.

Recent literature also highlights the growing role of Artificial Intelligence (AI) and Machine Learning (ML) in security monitoring. These technologies enable systems to learn usage patterns and identify anomalies in real time. Datafinz (2024) reported that ML-driven analytics have become crucial in predictive security, allowing organizations to detect and respond to threats proactively rather than reactively.

Materials and Methods

The materials used in this project are the three-authentication mechanism which are stated as follows: username and passwords, token generations and figure print.

1. Username and Password Authentication

Username and password authentication is the first layer of security (Figure 1) (something the user *knows*).

- The username identifies the user.
- The password verifies identity.
- Passwords should never be stored in plain text; instead, they are hashed using secure algorithms (e.g., SHA-256, bcrypt).

In a data warehouse security system, this layer ensures only registered users can initiate access.

Sample Code (Python – Password Hashing & Verification)

```
import hashlib
def hash_password(password):
  return hashlib.sha256(password.encode()).hexdigest()
# Store password (during registration)
stored_password = hash_password("mypassword123")
# Verify password (during login)
def verify_password(input_password, stored_hash):
  return hash_password(input_password) == stored_hash
# Test
if verify_password("mypassword123", stored_password):
  print("Username and Password Authenticated")
else:
  print("Authentication Failed")
```

Hashing protects user credentials even if the database is compromised.

Figure 1: Username and Password Authentication

2. Token Generation Authentication (One-Time Password – OTP)

Token generation is the **second layer** of security (Figure 2) (something the user *has*).

- A temporary token (OTP) is generated after successful username/password login.
- The token is valid for a short time and used only once.
- This prevents replay attacks and reduces risks from stolen passwords.

Common token types: SMS OTP, email OTP, authenticator apps.

Sample Code (Python – Random OTP Generation)

```
import random
import time
def generate_otp():
  return random.randint(100000, 999999)
otp = generate_otp()
print("Your OTP is:", otp)
# Simulate OTP expiration
start_time = time.time()
user_input = int(input("Enter OTP: "))
if user_input == otp and (time.time() - start_time) <= 60:
 print("Token Authentication Successful")
else:
  print("Invalid or Expired Token")
```

Even if an attacker knows the password, they still need the temporary token.

Figure 2: Token Generation Authentication

3. Fingerprint Authentication (Biometric)

Fingerprint authentication is the third layer of security (Figure 3) (something the user *is*).

- It uses unique biometric features of a user.
- Fingerprint data is captured, converted into a template, and stored securely.
- During login, a new fingerprint scan is compared with the stored template.

This is especially useful for high-security data warehouses.

Conceptual Code (Simulation Example)

⚠ Note: Real fingerprint systems require hardware and SDKs.
This example **simulates fingerprint matching** for academic purposes.

```
# Simulated fingerprint templates
stored_fingerprint = "FP_USER_001"
def verify_fingerprint(scanned_fingerprint):
    return scanned_fingerprint == stored_fingerprint
# Simulated scan
scan = input("Scan Fingerprint (enter fingerprint ID): ")
if verify_fingerprint(scan):
    print("Fingerprint Authentication Successful")
else:
    print("Fingerprint Authentication Failed")
```

Figure 3: Fingerprint Authentication

System Development Methodology

The Incremental Development Model is adopted for system development. This method enables the system to be built in phases, where each authentication module (username and password, token generation, and fingerprint authentication) is developed, tested, and integrated sequentially. The incremental model ensures flexibility, early error detection, and improved system reliability.

Authentication Techniques Used

The proposed system integrates three authentication methods:

- Username and Password Authentication to verify registered users.
- Token Generation (One-Time Password – OTP) to provide an additional layer of security through temporary access codes.
- Fingerprint Authentication to ensure biometric verification of the user.

The combination of these techniques forms a hybrid authentication framework that enhances confidentiality, integrity, and access control within the data warehouse

Algorithm for the Proposed Hybrid Authentication Process
Algorithm Name
Hybrid Data Warehouse Authentication Algorithm (HDWAA)
Input

- Username
- Password

- One-Time Password (OTP)
- Fingerprint Data

Output

- Access Granted or Access Denied

Step-by-Step Algorithm Description

Step 1:

Start the authentication process.

Step 2:

Prompt user to enter Username and Password.

Step 3:

Hash the entered password and compare it with the stored password hash in the database.

- If the username or password is incorrect, deny access and terminate the process.
- If correct, proceed to Step 4.

Step 4:

Generate a One-Time Password (OTP) and send it to the user's registered device (email/SMS/app).

Step 5:

Prompt the user to enter the OTP within a specified time limit.

- If the OTP is invalid or expired, deny access and terminate the process.
- If valid, proceed to Step 6.

Step 6:

Capture the user's fingerprint biometric data.

Step 7:

Compare the captured fingerprint with the stored fingerprint template.

- If the fingerprint does not match, deny access and terminate the process.
- If it matches, proceed to Step 8.

Step 8:

Grant access to the data warehouse.

Step 9:

End the authentication process.

Results

From all indications of the resulting output, the results from the implementation of the software is précised on the security of the warehouse Data presented and variables were studied, a well predictable set of data were used and graphics representing each state of instances. The table 4.1 shows the parameters for experiment 1 in order to discover the beauty of the proposed system. If a user logs in successfully, he/she will have to search and read the document searched for if it exists in the Database.

Experiment 1: From the experiment in Table 1, the application is designed to help or enable users enter the keyword that will aid the searching of document(s) in the proposed system database and if the document is available, the user can download, read and view the full thesis or Pdf document while the application still running. But if the document is unavailable for any reason, it means that the users cannot view it. Then the user has to view related works attached to the database to avoid searching for non-existing document. The value 1 as shown in the table above indicates that the user cannot use keywords simultaneously on the search box.

Table 1: the parameters for experiment 1.

Parameter	**Value**
Keyword	1
Search document	1
Available	1
Unavailable	0

Experiment 2: Table 2 shows experiment 2 where we investigate our program speed and efficiency in searching for document like (improved data warehouse authentication system using hybrid methods that ensure that only authenticated user would be accessing the critical warehouse, thereby curbing malicious intrusions and guaranteeing information confidentiality and security in the warehouse) other result are shown in Figure 4 to Figure 9.

Table 2: Search Document Efficiency

Keywords	**Existing system**	**Proposed system**
Secure information in the Data warehouse system	0.94sec	0.64sec
Private information	0.80sec	0.6sec

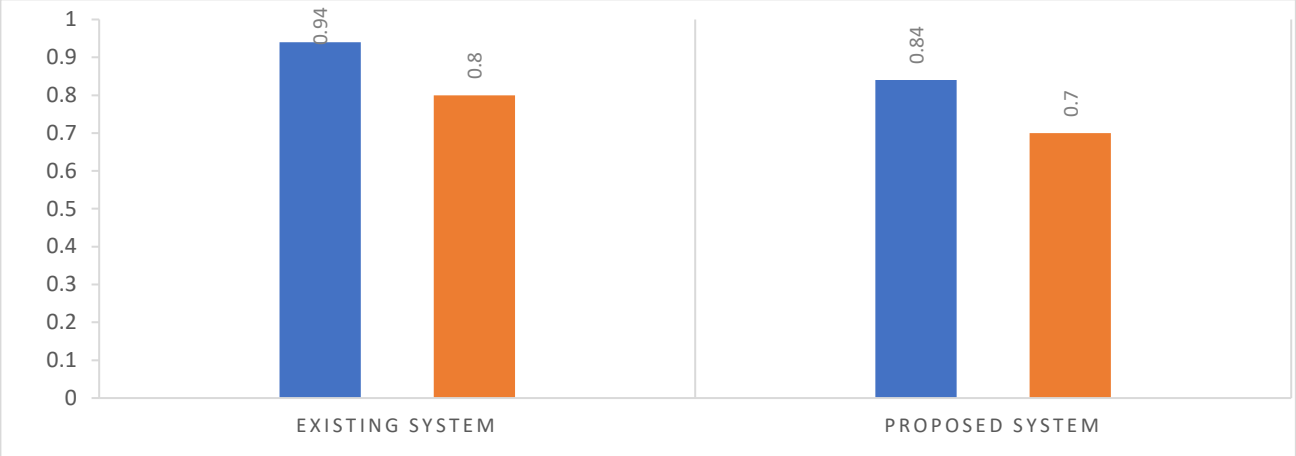


Figure 4: Graph comparing the Existing and Proposed System

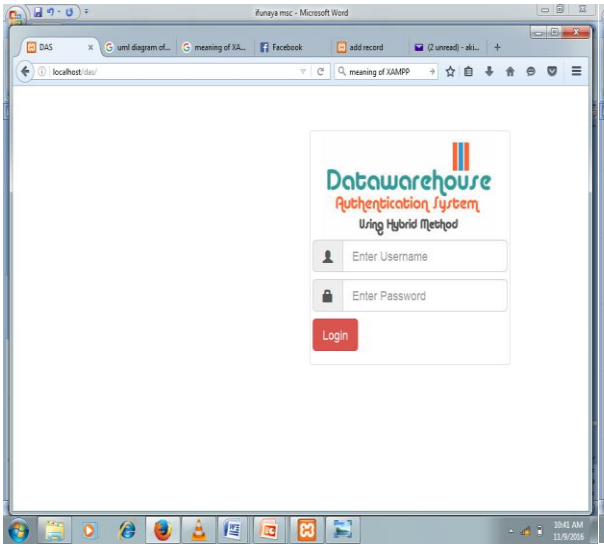


Figure 5: Interface of the New System

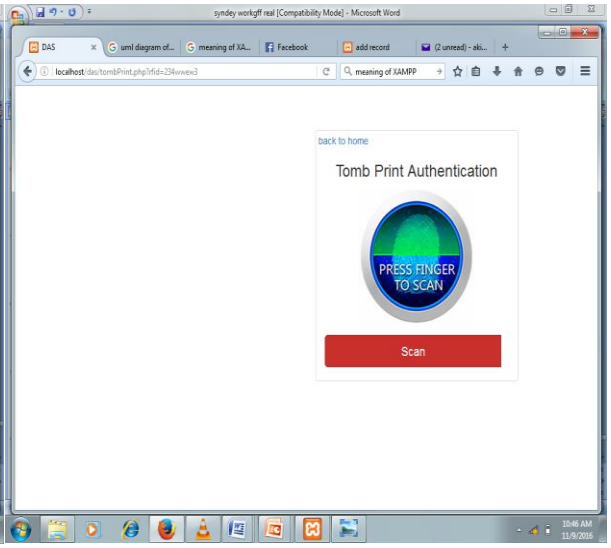


Figure 6: Finger Biometrics Interface I

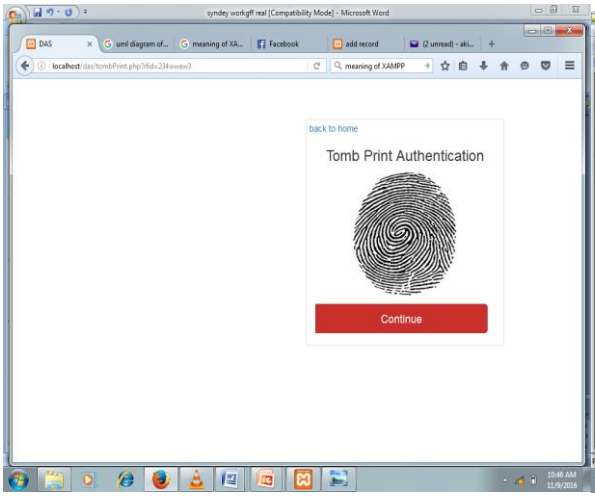


Figure 7: Finger Biometrics Interface II

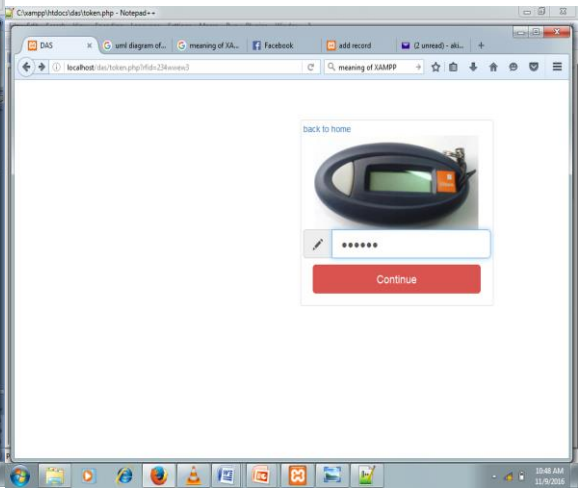


Figure 8: Token Generation

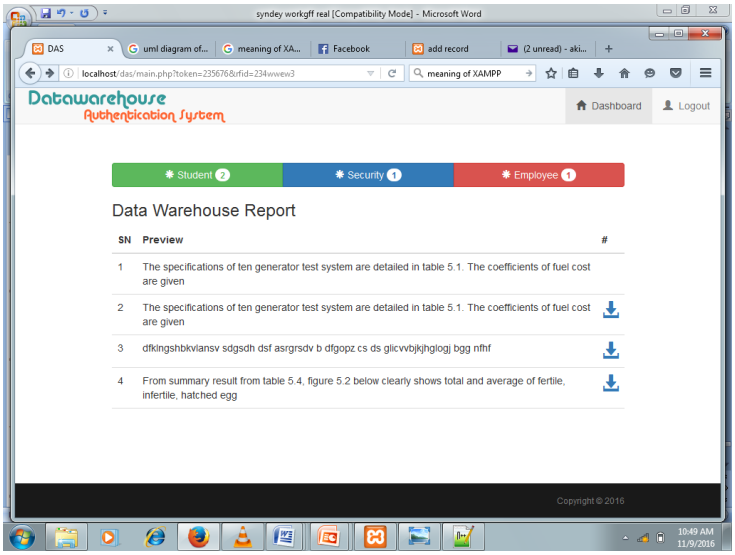


Figure 9: Data Warehouse Storage

Conclusion

This study concludes that an improved data warehouse security system based on hybrid authentication techniques provides a more robust and reliable security framework. By integrating multiple authentication factors such as passwords, tokens, biometrics, and cryptographic verification—hybrid authentication significantly enhances user identity validation and reduces the likelihood of successful attacks. The layered security approach ensures that the compromise of a single authentication factor does not automatically lead to unauthorized system access.

Furthermore, the adoption of hybrid authentication techniques strengthens accountability and access traceability within data warehouse systems, thereby supporting secure data governance and compliance with information security best practices. The improved authentication model also demonstrates scalability and adaptability, making it suitable for modern enterprise environments where data volumes, users, and access points continue to grow.

In conclusion, implementing hybrid authentication techniques in data warehouse security architectures is an effective and sustainable solution for mitigating authentication-related vulnerabilities. Organizations that adopt such systems are better positioned to safeguard critical data assets, maintain stakeholder trust, and support secure, data-driven decision-making in an increasingly complex threat landscape.

Based on the findings and conclusions of this study on an improved data warehouse security system using hybrid authentication techniques, the following recommendations are made:

1. Adoption of Hybrid Authentication Models: Organizations should adopt hybrid authentication mechanisms that combine at least two or more authentication factors—such as passwords, security tokens, biometrics, or one-time passwords (OTPs)—to strengthen access control in data warehouse environments. This layered approach significantly reduces the risk of unauthorized access.
2. Integration with Cryptographic Security Techniques: Hybrid authentication systems should be integrated with strong cryptographic protocols, including secure key exchange, encryption, and mutual authentication mechanisms, to protect data transmission and prevent interception or replay attacks during authentication processes.
3. Role-Based and Least-Privilege Access Control: Data warehouse administrators should implement role-based access control (RBAC) in conjunction with hybrid authentication to ensure that users can only access data and functionalities relevant to their roles. Enforcing the principle of least privilege minimizes the impact of insider threats and accidental data exposure.
4. Regular Security Audits and Updates: Organizations should conduct periodic security audits and vulnerability assessments of their data warehouse authentication systems. Authentication methods, cryptographic algorithms, and system configurations should be updated regularly to address emerging threats and evolving attack techniques.

5. User Awareness and Training: End users and system administrators should receive continuous training on secure authentication practices, including password hygiene, token handling, and biometric usage. Improved user awareness reduces the likelihood of social engineering and credential compromise.
6. Scalability and Performance Considerations: While implementing hybrid authentication, organizations should ensure that the system is scalable and does not significantly degrade system performance. Efficient authentication workflows and optimized security checks should be designed to accommodate increasing data volumes and user access without compromising usability.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Reference

Alassaf, N., Gutub, A., Parah, S. A., & AlGhamdi, M. (2021). Enhancing data security of cloud-based data warehouses using multi-factor authentication and encryption techniques. *Egyptian Informatics Journal, 22*(2), 141–152. https://doi.org/10.1016/j.eij.2020.07.002

Dasgupta, D., Roy, A., & Nag, A. (2017). Advances in user authentication. *IEEE Computer, 50*(5), 91–95. https://doi.org/10.1109/MC.2017.133

Inmon, W. H. (2005). *Building the data warehouse* (4th ed.). John Wiley & Sons.

Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling* (3rd ed.). John Wiley & Sons.

O'Gorman, L. (2003). Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE, 91*(12), 2021–2040. https://doi.org/10.1109/JPROC.2003.819611

Stallings, W. (2018). *Effective cybersecurity: A guide to using best practices and standards*. Addison-Wesley.