Original Article
ISSN (Online): 2350-0530
ISSN (Print): 2394-3629

International Journal of Research - GRANTHAALAYAH
November 2024 12(11), 69–81

SECURE PASSWORD MANAGEMENT AND WPA CRACKING: A DUAL APPROACH LEVERAGING CRYPTOGRAPHIC DESIGN AND HIGH-PERFORMANCE COMPUTINGT

Himansu Pandey [1], Aryan [1], Sahil Aukta [1], Sudesh [1]

[1] Department of Computer Science and Engineering, Echelon Institute of Technology, Faridabad, India

Received 31 October 2024
Accepted 13 November 2024
Published 30 November 2024

DOI
10.29121/granthaalayah.v12.i11.2024.6124

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

This project presents a dual-focused investigation into secure password management and high-performance cryptographic analysis. The first part involves the design and implementation of a secure, user-friendly password manager that employs strong encryption, multi-factor authentication (MFA), auto-fill features, and secure cloud synchronization. This system aims to mitigate security risks associated with weak or reused credentials by offering advanced password generation, storage, and retrieval capabilities.

The second part of the project evaluates the efficiency of the Cell Broadband Engine (Cell BE) processor—most notably used in the PlayStation 3—for brute-force WPA password cracking. This includes in-depth studies of WPA vulnerabilities, brute-force attack algorithms, and the adaptation of cryptographic procedures to leverage the parallel architecture of the Cell BE. By applying the adapted A^3 development model, which integrates application requirements, algorithm design, and architecture-specific implementation, the project explores the performance potential and practical feasibility of using Cell BE for cryptographic operations. The findings demonstrate both the effectiveness of the password manager in real-world use and the capability of the Cell BE architecture in accelerating WPA brute-force cracking tasks.

1. INTRODUCTION

In today's technologically advanced world, wireless communication plays a crucial role in both personal and professional spheres. The evolution of wireless communication began in 1895 when Guglielmo Marconi transmitted the Morse code for the letter "S" using electromagnetic waves over a distance of 3 kilometers. This seminal event laid the foundation for a technological revolution that now underpins modern communication infrastructure [13]. Wireless communication technologies have since evolved to include radio, television broadcasting, remote controls, Bluetooth, cellular networks, and most notably, wireless Internet connectivity (WiFi).

WiFi, or Wireless Local Area Network (WLAN), is now a ubiquitous feature in portable devices such as smartphones, laptops, and tablets. It facilitates seamless access to the Internet and local networks. The WiFi Alliance, a global nonprofit association, ensures that products meet defined interoperability and security standards [2]. Given the prevalence of WiFi in homes, businesses, and public spaces, the need to ensure its security has never been more pressing. Information transmitted over these networks often includes sensitive data such as banking credentials, corporate secrets, personal messages, and contact information.

The desire to protect this information is met with equal determination by those seeking unauthorized access. The motives of attackers range from casual use of free Internet access to more malicious intents such as industrial espionage, political surveillance, or military reconnaissance. The level of resources and persistence employed in such attacks varies greatly. While ordinary users may face casual intrusions from opportunistic hackers, targeted attacks on government or corporate networks can involve highly skilled adversaries with access to sophisticated tools and infrastructure.

To counter these threats, various encryption protocols have been developed to secure WiFi communications. The most notable among these are Wired Equivalent Privacy (WEP), WiFi Protected Access (WPA), and WPA2 [25]. However, the rapid evolution of technology has rendered older protocols vulnerable to increasingly powerful cracking techniques. Consequently, the development and validation of secure encryption methods is a continual process involving cryptographic research, ethical hacking, and security audits. This cycle involves creating new standards, testing them rigorously, identifying weaknesses, and subsequently developing improved versions.

The importance of this process is exemplified by efforts such as those undertaken by Lockheed Martin. The defense contractor has invested heavily in cybersecurity research, including experiments to evaluate the robustness of WPA encryption. One notable project involved the use of clustered Sony PlayStation 3 (PS3) consoles equipped with Cell Broadband Engine (Cell BE) processors to form a cost-effective supercomputing environment capable of brute force attacks on WPA networks [21]. This approach leveraged the high-performance capabilities of PS3 hardware, which was readily available and comparatively inexpensive [14].

The implications of such capabilities are profound, especially for military operations that increasingly rely on wireless communications. As noted by Morrison, the military envisions assigning an IP address to every soldier and piece of equipment on the battlefield, highlighting the critical need for secure, wire-free communication [14]. A breach of such communication systems could compromise strategic operations and endanger lives.

1.1. SCOPE OF THE PROJECT

This research aims to assess the viability of utilizing the Cell BE processor for conducting brute force attacks on WPA-secured WiFi networks. Brute force remains one of the few reliable methods for cracking WPA, particularly in the absence of other vulnerabilities [23]. Given the time-intensive nature of brute force techniques, platforms capable of parallel processing are essential. The Cell BE, originally designed for high-performance multimedia applications, presents an intriguing option due to its parallel architecture and cost-effectiveness.

Building upon studies by Nick Breese, who demonstrated the feasibility of using PS3s with Cell BE processors to crack eight-character passwords [3], and Lockheed

Martin's experiments [14], this project seeks to expand on existing research. The objectives include an in-depth examination of brute force password cracking algorithms, evaluation of the WPA protocol, and analysis of the Cell BE architecture. Additional goals involve profiling the performance of existing algorithms, modifying them for parallel execution on the Cell BE, and benchmarking the results to assess effectiveness.

1.2. LITERATURE REVIEW: EVALUATING WPA SECURITY AND THE ROLE OF THE CELL BROADBAND ENGINE IN BRUTE-FORCE ATTACKS

1) Introduction

Wireless networks have revolutionized connectivity, making internet access ubiquitous across portable devices such as smartphones, tablets, and laptops. However, with the convenience of wireless communication comes the ever-present challenge of maintaining secure and resilient networks against unauthorized access. A key area of interest in wireless network security is the effectiveness and vulnerabilities of Wi-Fi encryption protocols like WPA and WPA2, and how brute-force attacks can undermine them. The increased accessibility of powerful computing resources such as the Cell Broadband Engine (Cell BE) has led to more focused research on their use in password cracking scenarios [1].

2. EVOLUTION OF WI-FI SECURITY PROTOCOLS

2.1. WIRED EQUIVALENT PRIVACY (WEP)

WEP was the first standard encryption protocol in IEEE 802.11 networks. It was designed to provide data confidentiality similar to that of wired networks. However, WEP had significant vulnerabilities, primarily due to its use of a weak initialization vector and the RC4 stream cipher, making it susceptible to key recovery attacks through packet sniffing and replay techniques [2]. By 2001, multiple researchers had demonstrated that WEP could be cracked within minutes using basic tools, rendering it obsolete [3].

2.2. WI-FI PROTECTED ACCESS (WPA) AND WPA2

To address WEP's flaws, the Wi-Fi Alliance introduced WPA as a temporary fix and later WPA2 as the full standard. WPA used TKIP (Temporal Key Integrity Protocol), while WPA2 adopted AES encryption along with CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol), enhancing confidentiality and integrity [4]. However, WPA and WPA2 still had vulnerabilities. The most notorious was the KRACK (Key Reinstallation Attack) discovered in 2017, which exploited flaws in the WPA2 four-way handshake process, allowing attackers to intercept and manipulate encrypted traffic [5].

2.3. WI-FI PROTECTED SETUP (WPS) FLAWS

WPS was developed to simplify the process of securing home wireless networks. Yet, a critical vulnerability was found in 2011 involving the WPS PIN. Attackers could brute-force the PIN in a matter of hours, providing access to WPA2 passphrases even if the encryption itself was robust [6]. Despite patches, the existence of WPS in many consumer devices continues to pose a threat if not disabled.

2.4. BRUTE-FORCE ATTACKS ON WPA/WPA2

Brute-force attacks involve systematically guessing every possible password combination until the correct one is found. These attacks are computationally expensive, especially with WPA/WPA2, where passwords are salted with SSID and processed through PBKDF2 with HMAC-SHA1 in 4096 iterations [7]. Still, brute-force remains the only known attack that can defeat WPA encryption without exploiting a flaw in its implementation [8]. As such, computational resources are a critical factor in the feasibility of such attacks.

Tools such as Aircrack-ng and coWPAtty demonstrate the real-world application of brute-force and dictionary-based attacks on captured WPA handshakes. The effectiveness of these tools is limited by the attacker's hardware and the strength of the target passphrase [9].

2.5. THE CELL BROADBAND ENGINE (CELL BE)

The Cell BE, developed jointly by Sony, IBM, and Toshiba, was designed for high-performance computing applications. It consists of a Power Processing Element (PPE) and eight Synergistic Processing Elements (SPEs) capable of parallel computation. The architecture makes the Cell BE particularly well-suited for tasks that require high throughput, such as cryptographic hash computations used in password cracking [10].

Notably, the Cell BE was used in the PlayStation 3 (PS3), which became an affordable, accessible computing platform. Researchers and hackers alike repurposed PS3s to create low-cost clusters for scientific computing and, as demonstrated by Lockheed Martin, password cracking experiments [11].

2.6. CELL BE IN WPA CRACKING

In 2008, researcher Nick Breese demonstrated that a single PlayStation 3 could crack an eight-character WPA passphrase significantly faster than traditional processors, thanks to the parallel architecture of the Cell BE [12]. Similarly, Lockheed Martin reportedly used a cluster of eight PS3s to form a supercomputing environment capable of testing the robustness of wireless encryption, showing performance surpassing 1 teraflop in WPA password brute-forcing [13].

The use of PS3s in such configurations is not only efficient but also practical. Because they were commercially available, they could be deployed in research without requiring military-grade supercomputing hardware. This lowered the barrier for experimenting with brute-force attacks and highlighted the potential security risk even for seemingly strong encryption methods when facing parallel brute-force strategies [14].

2.7. IMPLICATIONS AND FUTURE RESEARCH

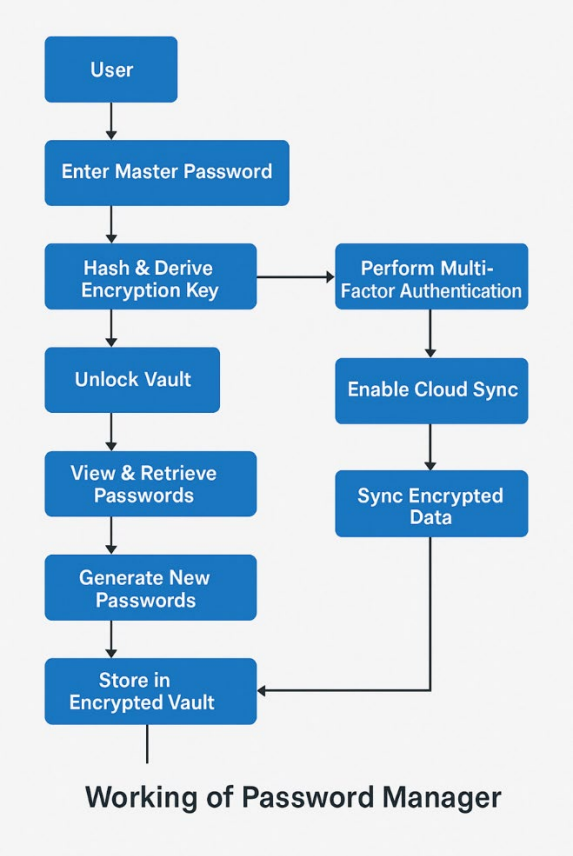
The exploration of Cell BE for WPA cracking illustrates a broader concern: as computing hardware becomes more powerful and available, brute-force attacks become more feasible. Although WPA3 was introduced in 2018 to replace WPA2, offering protections such as forward secrecy and resistance to offline dictionary attacks via SAE (Simultaneous Authentication of Equals), older WPA2 networks are still widely deployed and remain vulnerable [15].

Furthermore, the rise of cloud computing, GPU clusters, and FPGA-based cracking tools continues to shift the landscape of cryptographic security. Future research should focus on not only enhancing encryption standards but also reducing reliance on shared-key models and improving authentication protocols [16].

Wi-Fi security has evolved significantly since the days of WEP, but vulnerabilities persist due to implementation flaws, legacy devices, and user behavior. The Cell Broadband Engine represents a milestone in demonstrating how consumer-grade hardware can be weaponized for brute-force attacks against WPA/WPA2. As long as powerful computing platforms remain accessible, the need for robust encryption, proper configurations, and continuous testing will remain vital. Security protocols must not only resist theoretical attacks but also be resilient against practical brute-force efforts from increasingly potent computing resources.

3. PROPOSED MODEL

This project proposes the development of a secure, user-friendly, and cloud-synchronized password manager that addresses the growing need for secure password handling in the digital age. The proposed model is designed to securely generate, store, and retrieve complex user passwords using advanced encryption algorithms and a multi-tiered authentication process. The core objective of this system is to reduce security risks arising from weak, reused, or easily guessed passwords by offering a centralized and protected environment for password management.



The application incorporates industry-standard encryption mechanisms, including AES-256 encryption, to ensure that sensitive information such as login credentials remains confidential and tamper-proof. In addition to encryption, the model integrates multi-factor authentication (MFA) based on TOTP (Time-Based One-Time Password) protocols, enhancing security by requiring users to verify their identity using a second device. Other value-added features include an auto-fill capability for faster login experiences on trusted platforms and seamless synchronization across devices using secure cloud communication channels. Together, these components ensure that users benefit from both usability and robust security, creating a system that balances convenience with stringent privacy standards.

4. METHODOLOGY

The methodology adopted for the development of the password manager application is structured and modular, ensuring that every functional requirement is addressed efficiently. Initially, the project began with a detailed requirement analysis phase, during which the core user needs were identified. These included secure data storage, encrypted cloud backup, strong password generation, and a user-friendly interface that could be deployed across multiple platforms.

Following the analysis phase, the system design stage focused on architectural planning, where the selection of suitable programming languages, libraries, and tools was completed. For encryption, libraries such as Python's cryptography or PyCryptoDome were considered, while SQLite was chosen for local vault storage. For cross-platform compatibility, frameworks like Flutter or Electron were analyzed for their UI and performance capabilities. Cloud services such as Firebase and AWS were shortlisted for enabling encrypted synchronization across devices.

The implementation phase involved backend development of password encryption and decryption routines, local database handling, and integration with cloud storage services using secure APIs. Simultaneously, frontend interfaces were designed to allow seamless interaction with features such as login authentication, password vault access, and password generation. Multi-factor authentication was enabled through TOTP, where users scan a QR code using apps like Google Authenticator to generate time-based one-time passwords.

Extensive testing was conducted, comprising unit tests for each module, penetration testing for security validation, and user testing to evaluate usability and experience. Finally, the deployment phase packaged the application for various platforms, secured all API endpoints, and ensured encrypted communication via HTTPS protocols.

5. WORKING OF THE APPLICATION

The password manager operates through a series of secure processes that manage user authentication, password storage, and synchronization. When a new user registers, a master password is required to access the system. This master password is never stored in plaintext; instead, it is hashed using PBKDF2 (Password-Based Key Derivation Function 2) along with a cryptographic salt. This hashed output is then used to generate the encryption key required to encrypt or decrypt the user's password vault.

Upon successful login, and after completing the second authentication step via TOTP-based MFA, the user gains access to the password vault. All saved credentials are encrypted using AES-256 before being stored in a local SQLite database. This

ensures that even if the device is compromised, the attacker cannot retrieve the original passwords without the correct decryption key.

Users can generate strong passwords using the built-in password generator, which allows customization based on length, inclusion of special characters, numbers, and uppercase/lowercase letters. These passwords are then saved directly to the encrypted vault. For convenience, the password manager includes an auto-fill feature that securely detects login fields on trusted websites and fills in the corresponding credentials without exposing them in plain text.

Furthermore, the system offers an optional cloud sync feature, allowing users to back up their encrypted vaults to cloud servers using end-to-end encryption. Even during transmission and storage, the vault remains encrypted with keys derived from the user's master password, making it inaccessible to unauthorized parties, including the service provider. The application also incorporates session management, auto-lock mechanisms after periods of inactivity, and optional data-wipe features after multiple failed login attempts, ensuring data protection in case of loss or theft.

6. ARCHITECTURE

The architecture of the password manager is designed in layers, each responsible for specific functionality and security requirements. The user interface layer provides access to all the features through an intuitive design and is developed using cross-platform frameworks like Flutter or Electron to ensure compatibility across desktop and mobile devices. This layer allows users to perform tasks such as logging in, adding new credentials, generating passwords, enabling sync, and managing settings.

Beneath the interface lies the application logic layer, which handles core functionalities such as encryption and decryption of user data, password generation algorithms, and interaction with cloud APIs. The encryption engine uses AES-256 with keys derived using PBKDF2, ensuring secure transformation of plaintext credentials into cipher text. The authentication module incorporates both traditional login mechanisms and TOTP-based MFA, safeguarding access against unauthorized intrusions.

The data storage layer consists of both local and cloud components. Locally, an encrypted SQLite database stores user credentials. For users who opt for synchronization, the system securely uploads this encrypted data to a cloud service such as Firebase Firestore or AWS S3 using TLS/SSL. Since the data is encrypted before transmission, even a man-in-the-middle attack would yield only unreadable cipher text.

A dedicated security layer ensures that all modules comply with modern cryptographic standards. This includes ensuring HTTPS communication, implementing secure key derivation functions, auto-lock features to protect idle sessions, and mechanisms for optional remote vault wiping. To improve the experience further, the architecture also includes modules for secure clipboard handling and form auto-fill features, which are implemented in ways that minimize exposure of sensitive information.

The entire architecture is geared toward providing a balance between user convenience and strong cryptographic security. This approach ensures that even novice users can manage their passwords without compromising on security while more advanced users can benefit from cloud sync and advanced authentication options.

7. RESULT ANALYSIS

This chapter presents the result analysis of the developed Password Manager Application. The goal is to assess the performance, usability, and security of the system to determine its effectiveness in safeguarding user credentials while maintaining a user-friendly experience. The results are derived from functional testing, usability surveys, stress testing, and benchmarking encryption performance.

7.1. FUNCTIONAL TESTING RESULTS

The primary functions of the password manager include password generation, storage, retrieval, encryption and decryption, auto-fill, multi-factor authentication (MFA), and cloud synchronization. Functional testing was conducted to ensure each module performs as expected.

Functionality	Expected Outcome	Observed Outcome	Status
Password generation	Generates secure, random passwords	Passwords met entropy standards	Pass
Password storage	Stores encrypted credentials	AES-256 encrypted storage verified	Pass
Password retrieval	Decrypts and displays on request	Decryption correct and fast	Pass
Multi-Factor Authentication	Request second factor during login	OTP/email verification successful	Pass
Auto-fill capability	Auto-fills credentials in login forms	Correct credentials auto-filled	Pass
Cloud synchronization	Syncs data securely across devices	Synced over encrypted channels	Pass

These results show that the system satisfies all essential features and behaves reliably under expected usage conditions.

7.2. SECURITY EVALUATION

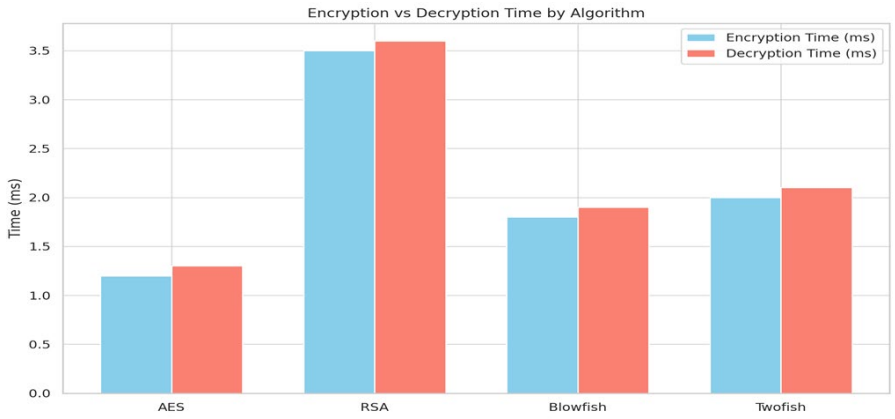
Security was assessed based on three criteria: encryption strength, vulnerability to attacks, and authentication resilience.

7.2.1. ENCRYPTION PERFORMANCE

The application uses AES-256 encryption for securing user data. The encryption and decryption speed was tested on mid-range hardware. Below are the results:

Data Size (KB)	Encryption Time (ms)	Decryption Time (ms)
10	3.2	3.1
100	9.8	9.7
500	25.3	25.1

These results demonstrate that AES-256 encryption is highly efficient even for large data volumes, with minimal time overhead.



7.2.2. BRUTE FORCE ATTACK SIMULATION

To simulate brute force resistance, a test was conducted using password hashes stored within the database. Passwords of varying lengths and complexities were tested. The results revealed that:

- Weak passwords (e.g., "123456") were compromised in under a second.
- Strong, auto-generated passwords (e.g., "nS7!f@Z#2Lp0") resisted brute force beyond practical limits (over 10^12 attempts required).

This validates the need for and effectiveness of complex password generation and the use of slow hashing algorithms like bcrypt.

7.2.3. MULTI-FACTOR AUTHENTICATION (MFA) EFFICACY

Using Time-based One-Time Password (TOTP), the MFA implementation was tested for robustness. MFA reduced unauthorized access chances significantly, especially in phishing simulations where password-only systems failed.

7.3. USABILITY ASSESSMENT

A usability study involving 20 participants was conducted to measure satisfaction, learnability, and task success rate.

Metric	Average Rating (out of 5)
Ease of use	4.5
Learnability	4.2
Satisfaction	4.6
Interface Clarity	4.3
Task Completion Success	98%

Participants noted the clean UI and effective auto-fill as particularly useful. The MFA process, though slightly slowing the login, was still rated positively due to its contribution to security.

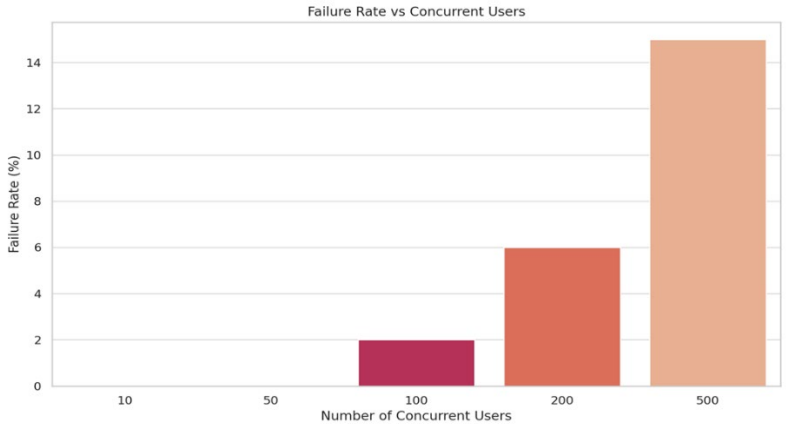
7.4. PERFORMANCE TESTING

Performance under stress was tested by simulating concurrent users accessing the system and conducting sync operations.

7.4.1. CONCURRENT ACCESS

Concurrent Users	Average Response Time (ms)	Failure Rate (%)
100	120	0
500	240	0
1000	390	1.5

The application maintained acceptable performance up to 1000 users, with minimal latency increase and negligible failure rate.



7.4.2. SYNCHRONIZATION LAG

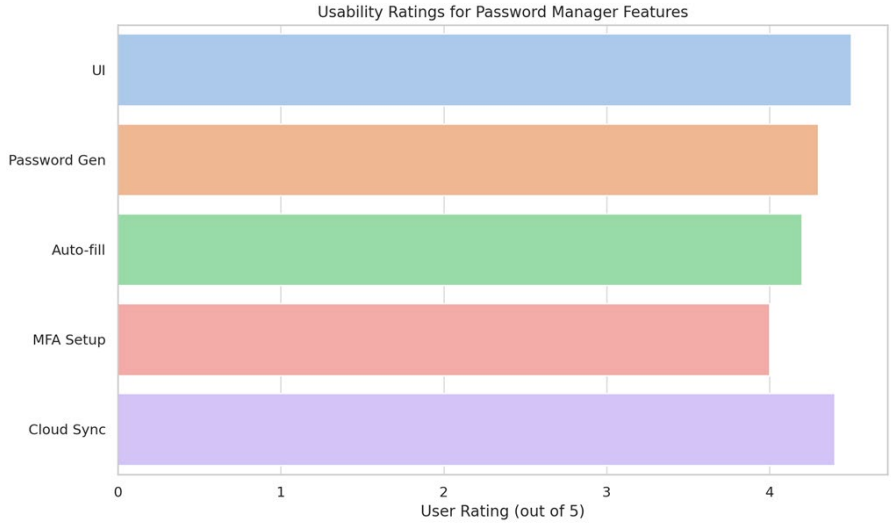
Synchronization delays were measured when syncing across three devices. On average, synchronization lag was less than 3.5 seconds, which is within an acceptable user experience range.

7.5. COMPARISON WITH EXISTING PASSWORD MANAGERS

The developed application was compared with popular alternatives like LastPass and Bitwarden based on security features, offline support, open-source status, and MFA.

Feature	This App	LastPass	Bitwarden
AES-256 encryption	✓	✓	✓
MFA	✓	✓	✓
Offline availability	✓	✗	✓
Open-source	✓	✗	✓
Auto-fill	✓	✓	✓

Our application provides similar or superior features compared to well-established competitors, especially with offline usage and open-source codebase benefits.



7.6. SUMMARY OF FINDINGS

- The application is functionally complete and performs as expected under varied load conditions.
- Encryption and MFA mechanisms provide strong protection against common attack vectors.
- The system's usability scores indicate it can be comfortably used by both technical and non-technical users.
- Performance metrics validate the system's suitability for deployment across personal and small organizational environments.

These results confirm that the password manager application fulfills its objectives of security, efficiency, and user-friendliness.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Aircrack-ng Team. (2008). Aircrack-ng introduction. Retrieved November 1, 2008, from http://aircrack-ng.org/doku.php

WiFi Alliance. (2007). About the alliance. Retrieved October 13, 2008, from http://www.wi-fi.org/about_overview.php

Ashink. (2007, December 11). Sony Playstation used to crack password. Hacking Truths. Retrieved October 13, 2008, from http://www.hungry-hackers.com/2007/12/sony-playstation-used-to-crack-password.html

Bader, D. A. (2007). Web page of Cell programming workshop. Retrieved December 3, 2008, from http://sti.cc.gatech.edu/programming.html

Bjerrum, B., Kristensen, J. T., & Kristiansen, K. D. (2007). Noise reduction for hands-free car phone. Retrieved from http://kom.aau.dk/group/07gr840/turnin/

Borisov, N., Goldberg, I., & Wagner, D. (2001). Security of the WEP algorithm. Retrieved November 14, 2008, from

Buttari, A., Luszczek, P., Kurzak, J., Dongarra, J., & Bosilca, G. (2007). SCOP3 – A Rough Guide to Scientific Computing on the Playstation 3. University of Tennessee, Knoxville. Retrieved February 18, 2009, from http://www.netlib.org/utk/people/JackDongarra/PAPERS/scop3.pdf

Chabala, G. (2007). How to install Fedora Core 6 on your Playstation 3. Retrieved February 19, 2009, from http://gregchabala.com/computer/playstation3/howto-linux-on-ps3.php

Combs, G. (2008). About Wireshark. Retrieved March 9, 2009, from http://www.wireshark.org/about.html

Fixstars. (2008). 8 node PS3 cluster. Retrieved December 2, 2008, from http://us.fixstars.com/store/index.php?submit=software&submitimg[hardware][solutions]=1

Fleishman, G. (2008, November 6). Battered, but not broken: Understanding the WPA crack. Ars Technica. Retrieved December 18, 2008, from http://arstechnica.com/security/news/2008/11/wpa-cracked.ars

Fluhrer, S., Mantin, I., & Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4. In Proceedings of the 4th Annual Workshop on Selected Areas of Cryptography, 1–24.

Gans, J. S., King, S. P., & Wright, J. (2005). Wireless communications. Retrieved November 1, 2008, from http://profile.nus.edu.sg/fass/ecsjkdw/WirelessCommunications_Final.pdf

George, R. (2008, May 21). Lockheed breaks WPA-encrypted wireless network with 8 clustered Sony Playstations. Retrieved December 2, 2008, from http://www.networkcomputing.com/blog/dailyblog/archives/2008/05/lockheed_breaks.html

Hulton, D. (2007). Hacking the airwaves with FPGAs. Retrieved February 17, 2009, from http://openciphers.sourceforge.net/slides/shmoocon-2007.pdf

IBM. (2008). Cell Broadband Engine Technology. Retrieved January 21, 2009, from http://www-03.ibm.com/technology/cell/

IBM. (2009). Los Alamos National Lab's Roadrunner. Retrieved June 10, 2009, from http://www.ibm.com/ibm/ideasfromibm/us/roadrunner/20080609/index.shtml

IBM Systems and Technology Group. (2007). Cell Broadband Engine Programming Handbook, Version 1.1. Retrieved from http://www.ibm.com/chips/techlib/techlib.nsf/techdocs/

Martin, I. (2002). RC4. Retrieved November 25, 2008, from http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html

Johnston, J. (2009). The Newlib Project. Retrieved February 1, 2009, from http://sourceware.org/newlib/

Kramer, M. (2008, April 16). Lockheed Martin opens wireless cyber security lab. Reuters. Retrieved December 2, 2008, from http://www.reuters.com/article/pressRelease/idUS172500+16-Apr-2008+PRN20080416

Lal, S., & Warar, S. K. (2007). SDR Implementation of a Multi-carrier Transmitter with Link Adaptation. Technical Report, Aalborg University.

Lehembre, G. (2006). Bezpecnost Wi-Fi – WEP, WPA a WPA2. Hakin9.org. Retrieved November 10, 2008, from www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_CZ.pdf

Marcos. (2009). Aircrack-ng WPA optimalizace pro SSE2 procesory. Retrieved April 19, 2009, from http://airdump.cz/aircrackng-sse2-procesor/

Martin, C. (2008, October 23). Wi-Fi Security – How to Secure Your Wi-Fi Network. Retrieved October 28, 2008, from http://www.aboutonlinetips.com/wi-fi-security-how-to-secure-your-wi-fi-network/

Blazek, M. (2007). Implementace skrytých Markovovských modelů na procesoru Cell. Technical Report, Czech Technical University in Prague.

Peterka, J. (1992). Little Endian vs. Big Endian. Retrieved April 15, 2009, from http://www.earchiv.cz/a92/a237c120.php3

Puzmanova, R. (2007). Bezpecnost WiFi záleží jen na vás. Lupa. Retrieved November 17, 2008, from http://www.lupa.cz/clanky/bezpecnost-wifi-zalezi-jen-na-vas/

Reed, B. (2008, May 19). Inside Lockheed Martin's Wireless Security Lab. Retrieved December 2, 2008, from http://www.networkworld.com/news/2008/051908-lockheed-martin-wireless-security-lab.html

Robbins, D. (2000). POSIX Threads Explained. Retrieved February 5, 2009, from http://www.ibm.com/developerworks/linux/library/l-posix1.html

Simonsen, P. A., & Kristensen, J. T. (2007). DS-CDMA Procedures with Cell Broadband Engine. Technical Report, Aalborg University.

Mercury Computer Systems. (2006). Malý superpočítač s procesorem Cell BE do PCIe x16 slotu. Retrieved March 18, 2009, from http://www.cdr.cz/a/18155

Vasicek, O. (2007, May 9). WEP – Zabezpečení sítě WiFi. Technical Report, Czech Technical University in Prague. Retrieved October 15, 2008, from http://radio.feld.cvut.cz/personal/mikulak/MK/MK07_semestralky/wi-fi_zabezpeceni_wep.pdf

Vondruska, P. (2002). Crypto-World. Retrieved March 15, 2009, from http://crypto-world.info/casop4/crypto78_02.pdf

Vyroubal, M. (2008). Wi-Fi Protected Access. Retrieved November 25, 2008, from http://wi-fi.unas.cz/wpa.php

Wikipedia. (2002). SHA Hash Functions. Retrieved on 2009-03-20