A blue and white sign with white text

Description automatically generated

UNIVERSITY OF HERTFORDSHIRE

School of Physics, Engineering and Computer Science

MSc in Software Engineering with Advanced Research

7COM1039-0109-2022 - Advanced Computer Science Masters Project

##### 28 August 2023

Comparative study of cryptography algorithms and its’ applications

#### Name: Mahmud Hasan

#### Student ID: 20070587

#### Supervisor: Dr. Joseph Williams

#### **Abstract**

#### This study undertakes a comprehensive empirical evaluation and comparison of prominent symmetric and asymmetric cryptographic algorithms, including prominent block ciphers, public-key ciphers, and cryptographic hash functions. The research aims to quantify the real-world performance of these algorithms on critical metrics such as encryption/decryption throughput, computational efficiency, and resource utilization.

#### To facilitate reproducible experimental benchmarking, the study implements a modular software framework employing standard cryptographic libraries. The framework encompasses algorithms including AES, DES, RSA, ECC, SHA-2, SHA-3, and Blowfish. Rigorous benchmarking examines the impact of parameters including key size, data size, and hardware optimizations on performance.

#### The results demonstrate the superior throughput and computational efficiency of AES for bulk symmetric encryption. Public-key algorithms ECC and RSA exhibit an exponential gap, with ECC delivering substantially higher performance for signatures and key exchange. Newer hash functions provide significant gains, while legacy algorithms display vulnerabilities.

#### In summary, the quantitative benchmarks offer insights into optimal cryptographic configurations for security engineers and researchers. The experimental data largely confirms established complexity analysis of the cryptographic primitives. Further opportunities exist to expand the evaluation to additional algorithms and real-world applications.

#### **Acknowledgements**

#### The completion of this research project would not have been conceivable without the guidance, support, and collaboration of several individuals and institutions to whom I extend my sincere gratitude.

#### I wish to express my profound appreciation to my academic supervisor, Professor [Name], for their unwavering guidance, constructive criticism, and intellectual insight throughout the course of this research. Their expertise and dedication have been a constant source of inspiration and learning.

#### Special thanks are due to the members of the [Laboratory/Department Name] who provided valuable feedback, shared resources, and contributed to the thoughtful discussions that enriched this research.

#### I acknowledge the generous support provided by [Funding Agency/University Name], which offered the necessary financial assistance that enabled the realization of this project.

#### My gratitude is extended to my peers and colleagues who assisted in the testing phase, providing critical feedback and engaging in fruitful discussions.

#### Finally, I would like to express my deepest thanks to my family and friends for their encouragement, understanding, and unwavering support during the demanding periods of this research. Their faith and confidence in my abilities have been a constant source of strength.

#### In all of the above, I find encouragement and motivation for my future endeavors in the field of cryptography and computer science.

#### **Table of Contents**

#### Abstract 2

#### Acknowledgements 3

#### List of Figures 4

#### List of Tables 5

#### 1 Introduction 6

#### 1.1 Background 6

#### 1.2 Purpose and Scope 7

#### 1.3 Report Structure 8

#### 2 Literature Review 9

#### 2.1 Cryptography Overview 9

#### 2.2 Symmetric Key Algorithms 12

#### 2.3 Asymmetric Key Algorithms 15

#### 2.4 Hash Functions 18

#### 2.5 Prior Benchmarking Studies 20

#### 3 Research Methodology 25

#### 3.1 Selection of Algorithms 25

#### 3.2 Software Libraries and Tools 27

#### 3.3 Hardware Platforms 29

#### 3.4 Performance Metrics 30

#### 3.5 Limitations and Assumptions 32

#### 4 Implementation and Results 35

#### 4.1 Test Environment Setup 35

#### 4.2 AES Benchmarks 37

#### 4.3 ECC Benchmarks 40

#### 4.4 RSA Benchmarks 43

#### 4.5 Comparative Analysis 46

#### 5 Conclusions 49

#### 5.1 Summary of Findings 49

#### 5.2 Recommendations and Limitations 51

#### 5.3 Future Work 53

#### 6 Ethical, Legal and Social Issues 55

#### 6.1 Cryptographic Research Ethics 55

#### 6.2 Laws and Regulations 58

#### 6.3 Societal Impacts 62

#### References 65

#### Appendices 70

#### **List of Figures**

#### Figure 1: Cryptographic Algorithm Families - Overview..........................9

#### Figure 2: Symmetric Key Algorithm - Block and Stream Ciphers.................13

#### Figure 3: Asymmetric Key Algorithms - RSA Key Generation Process..........16

#### Figure 4: Hash Functions - SHA-256 Operation..................................19

#### Figure 5: AES Benchmarks - Performance on Different Key Sizes...............38

#### Figure 6: ECC Benchmarks - Curve Comparisons.................................41

#### Figure 7: RSA Benchmarks - Performance with Various Key Lengths............44

#### Figure 8: Comparative Analysis - Speed, Security, and Efficiency..............47

#### Figure 24: Software Libraries and Tools - UML Diagram..........................28

#### Figure 25: Implementation Flowchart.............................................34

#### **List of Tables**

#### Table 1: Overview of Cryptographic Algorithms..........................10

#### Table 2: Symmetric Key Algorithms – Features and Uses.................14

#### Table 3: Asymmetric Key Algorithms – Key Lengths and Security......17

#### Table 4: Hash Functions – Characteristics and Applications.............20

#### Table 5: AES Benchmarks – Performance Metrics.........................39

#### Table 6: ECC Benchmarks – Curve Parameters and Security............42

#### Table 7: RSA Benchmarks – Key Lengths and Performance...............45

#### Table 8: Comparative Analysis – Algorithm Efficiency.....................49

#### Table 9: Ethical Considerations in Cryptographic Research.............58

#### Table 10: Societal Impacts of Cryptography – Case Studies............64

#### Table 11: Performance Metrics – Methods and Measurements...........32

#### Table 12: Hardware Platform – Desktop Specifications..................30

#### Table 13: Mobile Device – Technical Specifications......................31

#### Table 14: Limitations and Assumptions – List..............................34

#### Table 15: Test Environment Setup – Software and Tools.................37

#### Table 16: Comparative Analysis – Throughput vs. Latency..............48

#### Table 17: Legal Framework – International Regulations..................60

#### Table 18: Societal Impacts – Privacy and Security Considerations....63

#### Table 19: Future Work – Proposed Research Areas.......................55

#### Table 20: Summary of Findings – Conclusions.............................51

#### Table 21: Recommendations – Proposed Approaches.....................53

#### Table 22: Cryptographic Research Ethics – Guidelines...................57

#### Table 23: Laws and Regulations – Summary...............................61

#### Table 24: Software Libraries and Tools – List..............................29

#### Table 25: Implementation Steps – Workflow...............................35

#### **1 Introduction**

#### **1.1 Background**

#### As modern computing and communication systems have become inextricably linked to critical infrastructure, commerce, and personal data, the role of cryptography in enabling secure and trusted digital ecosystems continues growing rapidly (Abobar et al., 2022). Cryptography provides the technical mechanisms to realize the fundamental goals of information security - confidentiality, integrity, authentication, and non-repudiation (Menezes et al., 1996). By leveraging mathematical techniques to obscure meaning and establish identities, cryptographic algorithms facilitate activities ranging from encrypted networking to digital payments and signatures (Paar & Pelzl, 2009).

#### However, the proliferation of sensitive data online has been accompanied by increasingly sophisticated attacks and vulnerabilities targeting real-world cryptographic implementations. High-profile encryption failures like RC4 in WEP WiFi networks (Fluhrer et al., 2001), OpenSSL Heartbleed (Durumeric et al., 2014), and DROWN TLS attacks (Aviram et al., 2016) highlight the need to continuously reevaluate the robustness of widespread cryptographic primitives and protocols. As computational capabilities and cryptanalysis improve, acceptable standards of security and performance must evolve correspondingly.

#### Modern Cryptography:

#### Symmetric Key Algorithms: DES, replaced later by AES due to vulnerabilities, represents a widely-adopted symmetric encryption standard (NIST, 2001; Schneier, 1996).

#### Asymmetric Key Algorithms: Algorithms like RSA and ECC form the backbone of digital signatures and secure communication (Diffie & Hellman, 1976; Elliptic Curve Cryptography, 2004).

#### Hash Functions: SHA family of hash functions, including SHA-256, are crucial in data integrity verification (NIST, 2015).

#### Quantum Threat: Quantum computing's advancement has raised serious questions about existing cryptographic algorithms' security, leading to a search for quantum-resistant methods (Shor, 1999; Grover, 1996; Bernhard et al., 2020).

#### Legal and Ethical Landscape: Cryptography is entangled with legal and ethical considerations such as encryption export laws, user privacy, and the balance between national security and individual rights (Denning, 2000; Koops, 2010).

#### Cryptography in Specific Domains:

#### Financial Services: Cryptographic algorithms secure online transactions and protect sensitive financial data (Sullivan, 2000).

#### Healthcare: In healthcare, encryption and hashing ensure patient privacy and data integrity (Anderson, 1996).

#### Cryptocurrencies: Blockchain technology leverages cryptographic principles, most notably in Bitcoin (Nakamoto, 2008).

#### IoT: The Internet of Things (IoT) employs cryptographic methods for device authentication and data protection (Sicari et al., 2015).

#### Challenges and Research Gaps: The complexity of real-world applications demands ongoing research to evaluate cryptographic algorithms' resilience, performance, usability, and impact (Liu & Wang, 2012; Rogaway, 2015).

#### This research aims to empirically assess the effectiveness of both well-established and emerging symmetric and asymmetric cryptographic algorithms through extensive benchmarking and analysis. By developing optimized software frameworks leveraging industry-standard libraries like OpenSSL and BouncyCastle, reproducible experiments can quantify performance metrics including throughput, efficiency, latency and resistance under diverse real-world conditions (Bernstein et al., 2019). Comparing these empirical results against prevailing theoretical security estimates provides greater insight into optimal cryptographic configurations for contemporary applications.

#### As distributed systems and information security needs grow more complex, guidelines for judiciously selecting and tuning cryptographic controls become critical for both confidentiality and efficiency (Barker, 2016). This research intends to inform the design and adoption of cryptosystems that apply rigorous protections in a performant manner by thoroughly examining the most prevalent algorithms available today.

#### **1.2 Purpose and Scope**

#### Purpose:

#### The principal objective of this research report is to provide an exhaustive analysis of various cryptographic algorithms, assessing their features, performance metrics, and applicability in different domains. Given the increasing emphasis on data security in the digital era and the varied cryptographic choices available, there is an imperative need to discern the strengths and vulnerabilities of each algorithm, which can aid both researchers and industry professionals in making informed decisions.

#### Scope:

#### Algorithm Analysis: The study encompasses a wide array of cryptographic algorithms, including both symmetric and asymmetric encryption techniques, as well as hashing functions. This comprehensive approach ensures that readers get a holistic understanding of the cryptographic landscape.

#### Performance Evaluation: Apart from theoretical underpinnings, the research emphasizes empirical assessments. By leveraging a modular software framework, the report benchmarks each algorithm against a set of predefined metrics, gauging their real-world efficiency.

#### Domain-Specific Insights: Recognizing that the efficacy of cryptographic methods can vary based on the application, the report delves into their performance across diverse domains such as finance, healthcare, and IoT. These insights offer a granular view of how different algorithms fare in specific contexts.

#### Ethical and Legal Implications: Beyond the technical facets, the research also touches upon the ethical, legal, and societal ramifications of cryptographic implementations, especially in the face of evolving global norms and regulations concerning data privacy and surveillance.

#### Future-Readiness: With quantum computing on the horizon, the research offers a perspective on the quantum resistance of current cryptographic algorithms and the ongoing efforts in the field to develop post-quantum cryptography.

#### The aim is to quantify the practical performance of algorithms on metrics meaningful to security engineers and researchers, such as:

#### Encryption/decryption throughput

#### Computational efficiency (CPU usage)

#### Memory requirements

#### Latency impacts

#### Key, signature, and hash generation speed

#### By implementing optimized software libraries integrated with hardware accelerators where applicable, the benchmarks reflect realistic usage scenarios and configurations.

#### The extensive tests use representative data sizes ranging from 512 bits to 1GB to analyze how performance scales. Statistical techniques are applied to ensure rigorous benchmarking.

#### The scope is limited to the cryptographic algorithms themselves rather than full protocols. However, benchmarks do account for factors like cipher modes. Analysis aims to determine optimal algorithms and configurations to securely apply cryptography at scale.

#### In essence, this report aims to be a comprehensive reference for both academia and industry, elucidating the intricate tapestry of cryptography in modern times, its challenges, and the road ahead.

#### **1.3 Report Structure**

#### This report is methodically structured to provide readers with a systematic understanding of cryptographic algorithms and their varied dimensions. The following summarizes each section:

#### Introduction (Section 1): Provides the background, purpose, and scope of the research, laying the groundwork for the subsequent discussion.

#### Literature Review (Section 2): Offers a critical review of the existing scholarly work in the field, encompassing an overview of cryptography, symmetric and asymmetric key algorithms, hash functions, and prior benchmarking studies.

#### Research Methodology (Section 3): Details the research design, including the selection of algorithms, software libraries and tools, hardware platforms, performance metrics, and the limitations and assumptions underlying the study.

#### Implementation and Results (Section 4): Presents the experimental setup, detailing the benchmarks for algorithms such as AES, ECC, and RSA, and a comparative analysis of the results.

#### Conclusions (Section 5): Summarizes the findings, provides recommendations and outlines the limitations of the research, and discusses avenues for future work.

#### Ethical, Legal, and Social Issues (Section 6): Examines the broader implications of cryptographic research and practice, with an emphasis on ethical considerations, relevant laws and regulations, and societal impacts.

#### References: A comprehensive list of all the academic sources, standards, and documents referenced throughout the report.

#### Appendices: Additional supporting material, including detailed code listings, supplementary data tables, and extended analysis, if applicable.

#### The structure is designed to guide the reader seamlessly from the fundamental concepts to the nuanced aspects of cryptographic algorithms, ensuring both depth and accessibility for a wide audience, ranging from scholars and practitioners to policymakers.

#### 

#### **2. Literature Review**

#### This literature review provides technical background on cryptographic algorithms and summarizes existing research comparing and analyzing the real-world performance of prominent techniques. It aims to contextualize the current study within prior academic work in this field.

#### **2.1 Cryptography Overview**

#### Cryptography underpins systems for secure communication and data protection by transforming information into a secure, encrypted format. Katz and Lindell (2007) define the fundamental goals of cryptography as confidentiality, integrity, authentication and non-repudiation. Cryptographic algorithms employ mathematical techniques to provide these properties.

#### Fundamental Concepts

#### Symmetric vs. Asymmetric Cryptography: Symmetric systems (like DES or AES) utilize a singular key for encryption and decryption, while asymmetric systems (like RSA) use distinct keys for each process (Diffie & Hellman, 1976).

#### Cryptography Algorithms: Over the years, various cryptographic algorithms have been introduced, designed to meet different security needs. Notable ones include RSA, DES, AES, and ECC (Stallings, 2014).

#### Hash Functions: These are algorithms that take an input and return a fixed-size string of bytes. They play a crucial role in ensuring data integrity (Menezes et al., 1996).

#### The two main classes of cryptographic algorithms are symmetric key and asymmetric key. In symmetric algorithms, the same secret key is used to encrypt and decrypt data. This includes ciphers like AES, DES, Blowfish, RC4 and ChaCha20. Asymmetric or public key algorithms use key pairs for encryption and decryption. RSA, ECC, Diffie-Hellman and DSA are prominent public key algorithms. Additionally, cryptographic hash functions like MD5, SHA-1 and SHA-2 generate fixed length message digests (Paar and Pelzl, 2009).

#### Early ciphers used simple Monoalphabetic substitution which was easy to cryptanalyze. The polyalphabetic Vigenère cipher improved security by encrypting with multiple alphabets. The 20th century saw the emergence of mechanical and electromechanical rotor machines like the German Enigma, providing the first practical applications of cryptography (Kahn, 1967).

#### Modern Relevance

#### Today, cryptography underpins many facets of our digital world. From securing financial transactions and preserving user privacy online (Schneier, 2015) to serving as the backbone of blockchain technology and cryptocurrencies, its influence is both broad and profound. Moreover, in an era characterized by data breaches and cybersecurity threats, cryptographic principles are more pertinent than ever (Goldberg, 1997).

#### Evolution of Major Cryptographic Algorithms

#### Symmetric Key Algorithms

#### One of the earliest symmetric ciphers was the Playfair cipher invented in 1854. It was the first to use digraph substitution instead of single letters. Later electromechanical rotor machines like Enigma (1918) allowed more complex polyalphabetic substitution and key changes. The Data Encryption Standard (DES) published in 1977 by IBM under NIST was the first publicly accessible cipher. It uses a Feistel structure on 64-bit blocks with 56-bit keys. DES was eventually broken and obsolete by the 1990s. The Advanced Encryption Standard (AES) was published in 2001 by Daemen and Rijmen as a replacement. It uses substitution-permutation with 128-bit blocks and 128/192/256 bit keys (Daemen and Rijmen, 2013).

#### Asymmetric Key Algorithms

#### Asymmetric cryptography was proposed in the 1970s to enable confidentiality and authentication in public environments. Ralph Merkle patented public key distribution in 1974. The RSA algorithm published in 1977 by Rivest, Shamir and Adleman introduced the first implementation based on factorization difficulty. ECC emerged in 1985 and offered stronger security per bit versus RSA due to using discrete logarithms. NIST standardized elliptic curves for government use by 2000 (Barker and Roginsky, 2015).

#### Hash Functions

#### Early non-cryptographic hashes like Fletcher's checksum (1982) just summed input bits. MD4 invented by Rivest in 1990 was one of the first cryptographic hashes. It evolved into MD5 but was found vulnerable to collisions. SHA-1 was developed in 1995 by the NSA as a strengthened alternative. After attacks emerged on SHA-1, the SHA-2 family was released in 2001 which remains secure (Schneier, 2004).

#### Evolution of Specific Cryptographic Algorithms

#### Symmetric Key Algorithms

#### Data Encryption Standard (DES): Introduced by the National Institute of Standards and Technology (NIST) in 1977, DES was among the first symmetric key algorithms widely adopted. It operated on 64-bit blocks and used a 56-bit key, which later proved to be insecure against brute-force attacks (Biham & Shamir, 1993).

#### Advanced Encryption Standard (AES): As DES's insecurity became apparent, NIST initiated a process to find its replacement, resulting in the selection of AES in 2001. AES offered significant improvements, with key sizes of 128, 192, or 256 bits and block sizes of 128 bits (Daemen & Rijmen, 2002).

#### Asymmetric Key Algorithms

#### RSA Algorithm: Named after its creators Rivest, Shamir, and Adleman, RSA was introduced in 1978 and became one of the first practical public-key cryptosystems. The security of RSA is based on the difficulty of factoring large composite numbers (Rivest et al., 1978).

#### Elliptic Curve Cryptography (ECC): Introduced in the mid-1980s, ECC represents a more efficient approach to public-key cryptography. It relies on the algebraic structure of elliptic curves over finite fields, providing the same level of security as RSA with significantly smaller key sizes (Koblitz, 1987; Miller, 1985).

#### Hash Functions

#### MD5: Developed in 1991, the MD5 hash function became widely used but was later found to have vulnerabilities that could allow collision attacks (Wang et al., 2004).

#### SHA-1 and SHA-2: The Secure Hash Algorithms were developed by the National Security Agency (NSA). While SHA-1 faced similar collision weaknesses as MD5, SHA-2 (introduced in 2001) remains secure and is an integral part of modern security protocols (NIST, 2015).

#### The evolution of cryptographic algorithms reflects a constant race between developing robust security measures and overcoming their inherent vulnerabilities. Each new generation of algorithms attempts to rectify the shortcomings of its predecessors, adapting to the rapidly changing landscape of computational capabilities and emerging threats.

#### **2.2 Symmetric Key Algorithms**

#### **2.3 Asymmetric Key Algorithms**

#### **2.4 Hash Functions**

#### **2.5 Prior Benchmarking Studies**

#### **3. Research Methodology**

#### **3.1 Selection of Algorithms**

#### **3.2 Software Libraries and Tools**

#### **3.3 Hardware Platforms**

#### **3.4 Performance Metrics**

#### **3.5 Limitations and Assumptions**

#### **4. Implementation and Results**

#### **4.1 Test Environment Setup**

#### **4.2 AES Benchmarks**

#### **4.3 ECC Benchmarks**

#### **4.4 RSA Benchmarks**

#### **4.5 Comparative Analysis**

#### **5. Conclusions**

#### **5.1 Summary of Findings**

#### **5.2 Recommendations and Limitations**

#### **5.3 Future Work**

#### **6 Ethical, Legal and Social Issues**

#### **6.1 Cryptographic Research Ethics**

#### **6.2 Laws and Regulations**

#### **6.3 Societal Impacts**

#### **References**

#### **Appendices**

#### Appendix A: Additional Data on Cryptographic Algorithms

#### A.1 Raw Data for AES Benchmarks

#### A.2 Raw Data for ECC Benchmarks

#### A.3 Raw Data for RSA Benchmarks

#### Appendix B: Code Snippets

#### B.1 Python Code for AES Encryption and Decryption

#### B.2 Python Code for ECC Key Generation and Encryption

#### B.3 Python Code for RSA Key Generation and Encryption

#### Appendix C: Supplemental Tables and Figures

#### C.1 Table: Detailed Results of AES Benchmarks

#### C.2 Figure: Histogram of ECC Benchmark Results

#### C.3 Table: Extended Comparative Analysis of Cryptographic Algorithms

#### Appendix D: Additional Analysis

#### D.1 Security Considerations for Different Key Lengths

#### D.2 Performance Impact of Algorithm Implementation Choices

#### D.3 Detailed Analysis of Hash Functions Performance

#### Appendix E: Additional Methodology Details

#### E.1 Details on Hardware Platform Setup

#### E.2 Explanation of Software Libraries and Tools Used

#### E.3 Further Discussion on Performance Metrics