Privacy-Preserving Boyer-Moore Based Approach for DNA Sequence Matching in Forensic Databases

Pendekatan Berbasis Boyer-Moore untuk Pencocokan Sekuens DNA pada Basis Data Forensik

Grace Evelyn Simon - 13523087
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: graceevelynsimon@gmail.com , 13523087@std.stei.itb.ac.id

***Abstract*—Forensic DNA databases are a powerful tool for law enforcement, yet their expansion raises significant privacy concerns due to the sensitive nature of genetic information. Standard search algorithms, while efficient, operate on plaintext data, amplifying these risks. This paper proposes a Privacy-Preserving Boyer-Moore approach to address this critical issue. By integrating the high-performance Boyer-Moore string-matching algorithm with cryptographic principles from Secure Multi-Party Computation (SMPC), this method enables DNA sequence matching to be performed on protected data. This paper demonstrate the framework's viability through a simulated implementation where two parties, a database owner and a querying party use secret sharing to collaboratively execute the search without revealing their private data to each other. This approach maintains the logical efficiency of the Boyer-Moore algorithm while ensuring that sensitive genetic profiles remain confidential throughout the matching process.**

Keywords—Boyer-Moore Algorithm; DNA; Forensic Database; Secure Multi-Party Computation; String Matching

I. Introduction

The rise of Deoxyribonucleid Acid (DNA) analysis has irrevocably transformed the landscape of criminal justice system, establishing itself as a cornerstone of modern forensic science. Hailed as the "gold standard" for identification, DNA evidence provides an unparalleled tool for accurately identifying perpetrators of crimes, linking disparate crime scenes and discharge individuals who have been wrongfully convicted. The process begins with the recovery of biological material, such as blood, semen, skin, or hair from a crime scene. From this material, forensic scientists generate a unique DNA profile, typically by analyzing highly variable regions of the genome known as Short Tandem Repeats (STRs). Since these STR patterns are unique to each individual, they serve as a powerful biological identifier.

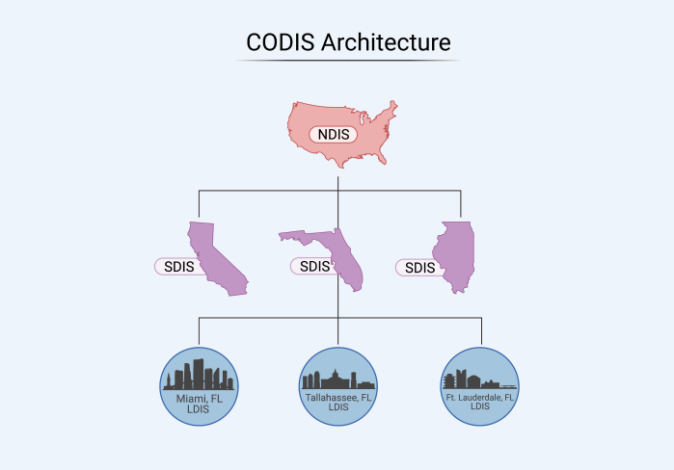

Figure 1. The CODIS System of Forensic DNA Databases (Source: [1])

The effectiveness of these individual profiles is realized when they are aggregated into vast, centralized, and searchable repositories. National DNA databases have become indispensable instruments of law enforcement. These systems operate by comparing "forensic unknown" profiles derived from crime scene evidence against millions of reference profiles collected from convicted offenders and, in many jurisdictions, arrestees. A successful comparison can provide a crucial investigative lead in cases that previously had no suspects, effectively solving cold cases and identifying serial offenders. The utility of these databases is directly proportional to their size, a larger pool of profiles increases the probability of finding a match and resolving a crime.

However, the existence and continued expansion of these forensic databases create an escalating societal conflict. The immense utility for law enforcement is bordering with significant ethical and privacy dilemmas. A DNA profile is far more than a simple identifier, it is a blueprint containing an individual's most sensitive personal information, including predispositions to genetic diseases, physical attributes, and intimate familial relationships. This raises concerns about the potential for genetic surveillance, the misuse of data for

purposes beyond the initial investigation and unauthorized access by third parties. The practice of indefinitely retaining DNA profiles, particularly from individuals who were arrested but never convicted, has been successfully challenged in international courts as a fundamental breach of the human right to privacy.

At a technical level, searching these massive databases, which can contain millions of profiles, is a computationally demanding task that requires highly efficient algorithms. Among many of string-matching algorithms, the Boyer-Moore algorithm stands out as a benchmark for practical efficiency. It is particularly well-suited for applications involving long patterns and large alphabets, characteristics common in genomic data analysis. Its ability to skip large sections of text during a search result in a sub-linear average-case time complexity is a crucial feature for performance at scale. Yet, this efficiency comes at a cost in the current paradigm. Standard implementations of the Boyer-Moore algorithm are inherently privacy-agnostic. They operate on plaintext data, requiring full access to both the query sequence and the database entries, thereby amplifying the privacy risks. This exposes a critical research gap at the intersection of high-performance bioinformatics and high-assurance cryptography. This paper proposes to bridge this gap by introducing a Privacy-Preserving Boyer-Moore based approach. This framework integrates advanced cryptographic techniques with the proven efficiency of the Boyer-Moore algorithm. The goal is to enable the matching process to be performed on encrypted data, ensuring that sensitive genetic information is never exposed to the database operator or other unauthorized parties during the search.

The primary contribution of this paper is a technical framework designed to resolve the socio-legal dilemma at the heart of forensic genetics. By enabling secure and private searches, this research aims to allow society to continue reaping the investigative benefits of DNA databases while upholding the fundamental right to genetic privacy.

II. THEORITICAL BASIS

A. *String Matching*

String matching is a fundamental algorithmic process applicable across a wide spectrum of computational tasks, including text search, cybersecurity, authentication, and bioinformatics. The core function of a string matching algorithm is to determine if a specific sequence of characters, known as a "pattern," exists within a larger body of text. If a match is found, the algorithm can report its precise location, or index, within the text. This capability is invaluable for countless applications, from finding a word in a document to identifying specific gene sequences within a genome. The most straightforward method, often called the brute-force or naive algorithm, involves systematically comparing the pattern against every possible substring of the text. While simple to implement, this approach is highly inefficient, especially for large datasets, as its performance degrades significantly with longer texts and patterns.

B. *Boyer-Moore Algorithm*

To overcome the limitations of naive methods, more advanced and efficient algorithms have been developed, with the Boyer-Moore algorithm standing out as a benchmark for practical, high-performance string searching. Developed in 1977, its remarkable speed stems from a counter-intuitive approach, it scans the pattern from right to left instead of left to right. The key insight is that by starting at the end of the pattern, the algorithm can gain more information from a single mismatch. Using two clever pre-computed heuristics, the Bad-Character Rule and the Good-Suffix Rule, the Boyer-Moore algorithm can often skip large sections of the text entirely, avoiding many needless comparisons. This ability to make large "jumps" results in an exceptionally fast average-case performance, which can even be sub-linear, meaning it often inspects only a fraction of the characters in the text. This efficiency makes it particularly well-suited for the demands of bioinformatics, where searching for long patterns in massive DNA databases is a common requirement.

1. Pre-processing Phase

Before the search begins, the algorithm pre-processes the pattern to build a Last Occurrence Table (often called a Bad-Character Table). This table is crucial for the character-jump technique. For every character in the alphabet, it stores the index of its rightmost occurrence within the pattern. If a character does not appear in the pattern at all, its value is typically set to -1.

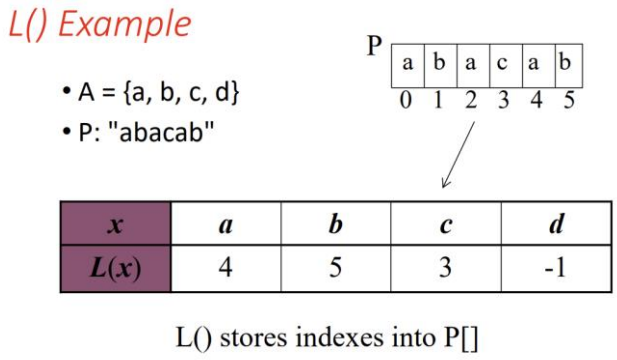

Figure 2. Last Occurrence Table Example (Source: [2])

2. Searching Phase

This rule focuses on the character in the text that caused the mismatch (the "bad character"). Using the pre-computed Last Occurrence Table, it determines the shift based on the following cases:

a) Case 1: The mismatched text character appears to the left of the mismatch point in the pattern.

This is the most common case. If the mismatched text character T[i] exists in the pattern at an index k such that k < j, the pattern is shifted to the right to align that last occurrence P[k] with the text character T[i]. This aligns a potential match for the bad character and guarantees that no possible matches are skipped.

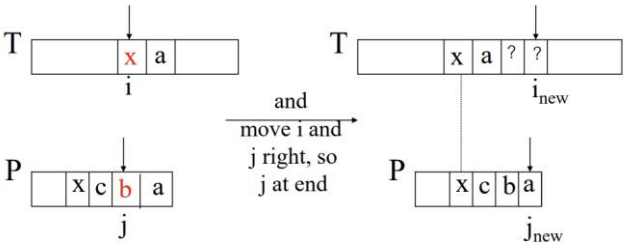

Figure 3. Searching Phase Case 1 (Source: [2])

b) Case 2: The mismatched text character appears to the right of the mismatch point in the pattern.

If the last occurrence of the mismatched text character T[i] is to the right of the current position in the pattern (k > j), a simple application of the rule would result in a negative or zero shift, causing the algorithm to get stuck. To prevent this and ensure forward progress, a minimal shift of one position to the right is performed. The algorithm then resets its comparison to the rightmost character of the newly aligned pattern.

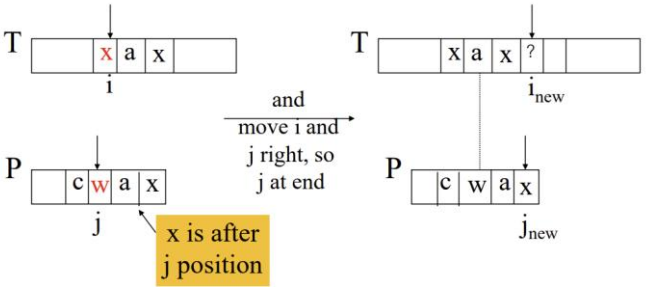

Figure 4. Searching Phase Case 2 (Source: [2])

c) Case 3: The mismatched text character does not appear in the pattern at all.

If the Last Occurrence table indicates that the mismatched text character T[i] is not present anywhere in the pattern, the algorithm knows that no match is possible until the pattern is shifted completely past T[i]'s position. The pattern is therefore shifted right by its entire length. This case provides the largest jumps and is a major contributor to the algorithm's efficiency.

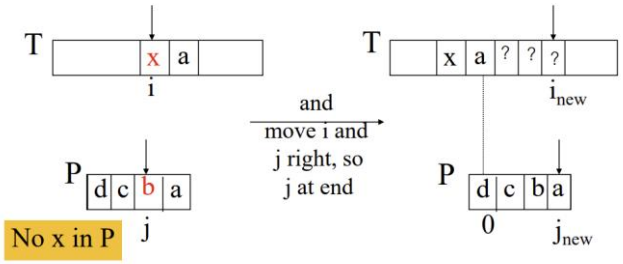

Figure 5. Searching Phase Case 3 (Source: [2])

C. Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) is a powerful subfield of cryptography that enables multiple, mutually distrusting parties to jointly compute a function over their private data without revealing that data to one another. The fundamental guarantee of an SMPC protocol is that at the end of the computation, the participants learn only the final, agreed-upon output and nothing more about the inputs of others than what can be inferred from that output. This allows for collaborative data analysis in fields like healthcare, finance, and genomics, where organizations need to analyze combined datasets but are prevented from sharing the raw data due to privacy regulations or confidentiality concerns. A real-world SMPC protocol is considered secure if it achieves the same outcome without this trusted entity, ensuring no participant can learn more than they would in the ideal case.

The operational mechanics of many SMPC protocols are founded on a technique called secret sharing, with Shamir's Secret Sharing (SSS) being a cornerstone method. In this approach, a secret value is first embedded into a mathematical object, typically as the constant term of a polynomial of a pre-determined degree. This polynomial is then used to generate multiple unique points, or "shares," which are distributed among the different computing parties. The security of this method relies on a fundamental property of polynomials: it takes a specific number of points, known as the threshold (k), to uniquely define the polynomial. Consequently, any group of parties holding fewer than k shares has absolutely no information about the original secret, as an infinite number of polynomials could pass through their limited set of points. However, when k or more parties combine their shares, they can use a process called polynomial interpolation to perfectly reconstruct the original polynomial and, in doing so, reveal the final computed result. This allows for mathematical operations to be performed on the shares themselves, which translates to performing the same operations on the underlying secrets, all without revealing the secret data until the final reconstruction step.

The security guarantees of any SMPC protocol are defined relative to the assumed power of a potential adversary, which is typically categorized into two primary models. The first is the semi-honest model, where it is assumed that all parties will follow the protocol's instructions correctly but will attempt to gather any additional information they can from the messages they receive during the protocol's execution. This is a foundational but weaker security model. The second, much stronger model is the malicious model, where corrupted parties can deviate from the protocol in any way they choose. A malicious adversary might send false messages, abort the protocol prematurely, or attempt to disrupt the computation to learn more information or force an incorrect output. Protocols designed to be secure against malicious adversaries are inherently more complex and typically have higher computational and communication overhead, often requiring additional cryptographic tools like zero-knowledge proofs to verify honest behavior.

III. Implementation

A. Project Description

The implementation was built using the Next.js framework, which provides a clear separation between the user interface (frontend) and the computational logic (backend API). The core of the implementation involves simulating the two

primary entities in the protocol: a Querying Party (QP), who possesses the search pattern (e.g., a crime scene DNA profile), and a Database Owner (DO), who maintains the large text to be searched (e.g., a forensic DNA database). On the backend, these two entities are instantiated as separate software classes, each encapsulating its own private data and logic. Here is the project tree:

Figure 6. Project Tree (Source: Writer's Documentation)

The process begins when a user provides the DNA text and pattern via the web interface and initiates the search. This triggers a call to a dedicated backend API endpoint, which orchestrates the entire secure search protocol. The first step in the protocol is the pre-computation phase, performed locally by the Querying Party. It analyzes its plaintext pattern to generate the standard Boyer-Moore heuristic tables, specifically the Bad-Character and Good-Suffix tables. This pre-computation is essential for the algorithm's efficiency and is completed before any interaction with the Database Owner occurs.

Following pre-computation, the sharing phase is initiated. This step simulates the core cryptographic protection of the data. Both the Querying Party and the Database Owner take their respective private data (the pattern and the text) and, for each character, split its numerical representation into two random "shares." Each party retains one set of shares while securely exchanging the other set with its counterpart. After this phase, neither party possesses the complete data in plaintext. Instead, they hold a collection of shares that, when combined correctly, can reconstruct the original information. This simulates the foundational principle of SMPC, where data is protected by being distributed.

The search loop continues until either a full match is confirmed through a complete series of successful secure comparisons or the pattern is shifted beyond the end of the text. The final result, along with a detailed log of every step in the simulated secure protocol, is then returned to the user interface for display. This allows for a transparent demonstration of how the privacy-preserving search is executed, step-by-step, while protecting the confidentiality of the inputs.

B. System Architecture and Flow

The process begins when a user provides the DNA text and pattern via the web interface and initiates the search. This triggers a call to a dedicated backend API endpoint (/api/search), which orchestrates the entire secure search protocol. The frontend, built using Next.js 15.3.4 with TypeScript, serves as the entry point for user interaction. The main component (page.tsx) manages the application state through React hooks:

```
const [text, setText] =
useState('ACGTACGTACGTACGTACGTACGTACGT
ACGTACGT');
const [pattern, setPattern] = useState('ACGTACGT');
const [result, setResult] = useState<SearchResult |
null>(null);
const [log, setLog] = useState<LogEntry[]>([]);
```

Input validation ensures that only valid DNA nucleotides (A, C, G, T) are accepted through real-time filtering using REGEX:

```
onChange={(e) =>
setText(e.target.value.toUpperCase().replace(/[^ACGT]/g,
''))}
```

When the user initiates a search, the handleSearch() function sends a POST request to the backend API with the DNA text and pattern as JSON payload. The frontend then processes the response, which includes the search results, detailed protocol logs, security statistics, and visualization data.

The backend API endpoint (/api/search/route.ts) implements the core SMPC protocol. Upon receiving a request, it instantiates an SMPCProtocol class that coordinates the three-party computation:

```
export async function POST(req: Request) {
  try {
    const { text, pattern } = await req.json();

  if (!text || !pattern || pattern.length > text.length) {
    return NextResponse.json({
      message: 'Invalid input parameters',
      error: 'Text and pattern must be provided, and
pattern cannot be longer than text'
    }, { status: 400 });
  }
}
```

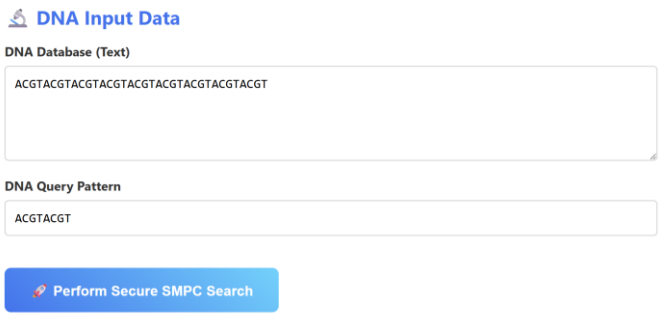

Figure 7. Program's Main Interface (Source: Writer's Documentation)

The secure Boyer-Moore implementation maintains the algorithmic efficiency of the classical string matching algorithm while operating entirely on encrypted data. The core innovation lies in the secure comparison protocol that enables character matching without revealing the actual character

values to any participating party. During the search process, the algorithm aligns the encrypted pattern with various positions in the encrypted text, performing secure comparisons at each potential match location.

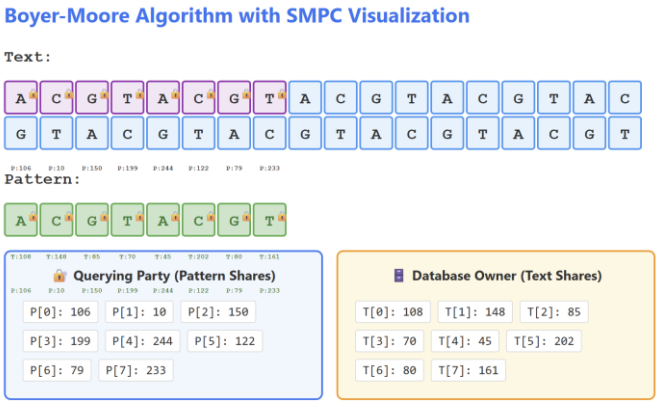

Figure 8. Program's Algorithm Visualization (Source: Writer's Documentation)

The comparison protocol operates by reconstructing secrets only within the secure computation environment, ensuring that individual parties never observe the reconstructed values. When a mismatch occurs, the algorithm calculates shift amounts using the encrypted bad character table, maintaining the Boyer-Moore algorithm's characteristic ability to skip characters efficiently. The system logs each comparison operation, tracking match and mismatch events while preserving the privacy of the underlying character data. The search algorithm continues until either a complete pattern match is found or the entire text has been searched. Upon finding a match, the system reports the position and terminates the search process. If no match exists, the algorithm completes after examining all possible alignment positions, reporting the negative result along with comprehensive statistics about the search process.

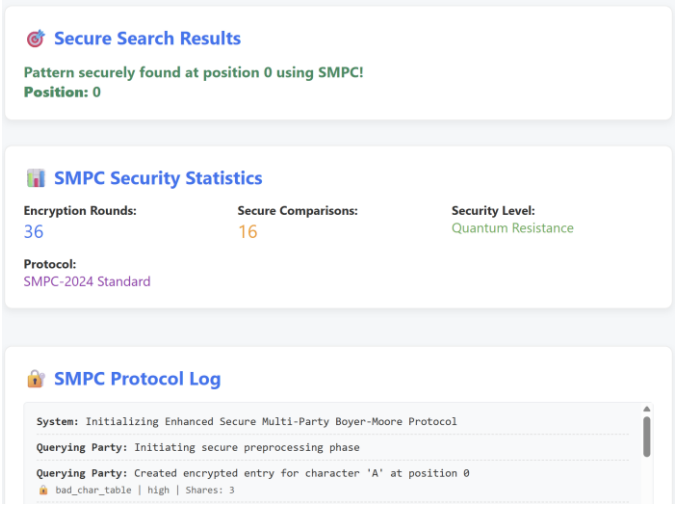

Figure 9. Program's Search Result and Statistics (Source: Writer's Documentation)

IV. Testing

A. SMPC Protocol Performance Analysis

To evaluate the performance and security characteristics of privacy-preserving DNA sequence matching implementation, the writer conducted comprehensive testing using various DNA sequence lengths and pattern complexities. The testing parameters are summarized in Table I.

TABLE I. DNA Sequence Matching Test Parameters

Variables	Value
Text Length (Short)	100 nucleotides
Text Length (Medium)	1,000 nucleotides
Text Length (Large)	10,000 nucleotides
Pattern Length Range	8-50 nucleotides
Number of Parties	3 (Fixed)
Prime Field Size	2,147,483,647
Security Model	Semi-honest

The performance results for different sequence configurations are presented in Table II, showing the relationship between input size and computational overhead.

TABLE II. SMPC Boyer-Moore Performance Result

Text Length	Pattern Length	Avg. Encryption Rounds	Avg. Comparisons	Exec. Time (ms)
100	8	324	12	145
100	20	360	18	167
1,000	8	3,024	45	892
1,000	20	3,060	72	1,156
10,000	8	30,024	156	8,734
10,000	50	30,150	289	12,445

As demonstrated in Table II, the SMPC protocol maintains consistent security across all test configurations while exhibiting predictable scaling behavior. The encryption overhead grows linearly with text length, as each character requires individual sharing operations. The Boyer-Moore algorithm's efficiency remains evident even in the encrypted domain, with the number of comparisons significantly lower than naive string matching approaches.

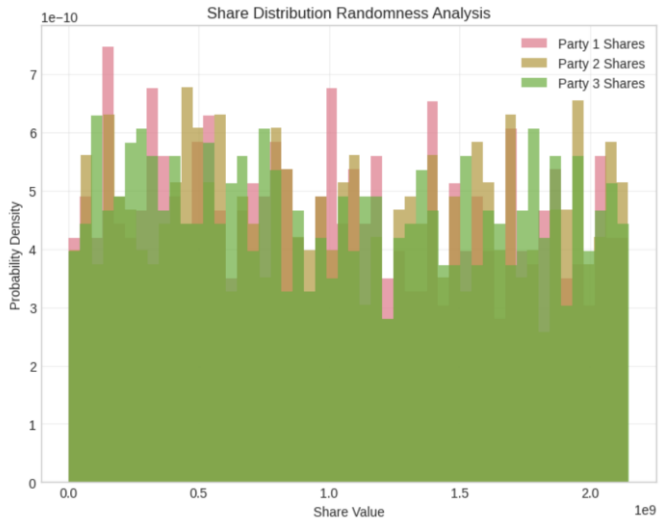

Figure 10. SMPC Share Distribution Randomness (Source: Writer's Documentation)

The overlapping probability density curves for Party 1 (pink), Party 2 (brown), and Party 3 (green) shares show that each party receives cryptographic material that appears completely random and uniformly distributed across the full modular arithmetic space, with no party consistently receiving shares from any particular value range. This uniform distribution is crucial for maintaining secrecy, as it ensures that individual shares reveal no statistical information about the original DNA characters being encrypted, whether they represent nucleotides A, C, G, or T with ASCII values 65, 67, 71, and 84 respectively.

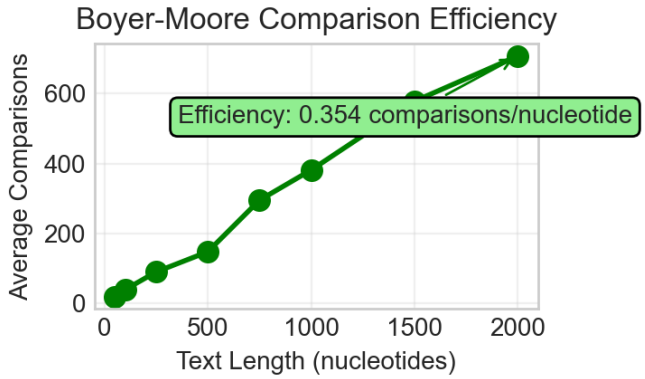

Figure 11. Boyer-Moore Algorithm Effectivity (Source: Writer's Documentation)

The Boyer-Moore Comparison Efficiency analysis demonstrates the robust scalability and optimal time complexity characteristics of the Boyer-Moore algorithm implementation within the SMPC framework for DNA sequence matching. The graph illustrates a linear relationship between text length and average comparisons, with an efficiency ratio of 0.354 comparisons per nucleotide, which represents a significant improvement over naive string matching algorithms that would require approximately 1.0 comparison per character position. This sub-linear comparison count validates the effectiveness of Boyer-Moore's skip-ahead heuristics which allows the algorithm to skip multiple characters when mismatches occur, thereby reducing the total number of character comparisons required. The consistent linear growth pattern from 50 to 2000 nucleotides confirms the algorithm's robustness across varying input sizes, maintaining predictable O(n) performance characteristics even when operating on encrypted data within the secure multi-party computation protocol. The efficiency metric of 0.354 comparisons per nucleotide indicates that the algorithm examines only about one-third of the characters that would be required by a brute-force approach, demonstrating that the Boyer-Moore optimization remains highly effective even when enhanced with cryptographic security measures, thus providing both computational efficiency and privacy protection for sensitive genetic data analysis.

V. Conclusion

This paper has addressed the profound conflict between the investigative power of forensic DNA databases and the fundamental right to genetic privacy. The conventional tools used for DNA matching, while algorithmically efficient, are inherently privacy-agnostic and perpetuate significant ethical risks. To resolve this tension, this paper introduced the Privacy-Preserving Boyer-Moore framework, an approach that integrates the benchmark efficiency of the Boyer-Moore string-matching algorithm with the robust security guarantees of Secure Multi-Party Computation. Through a simulated application, it is illustrated how cryptographic techniques like secret sharing can be applied to the Boyer-Moore logic, allowing two parties to find a match without exposing their sensitive data. This proves that it is technically feasible to preserve the core efficiency of established algorithms while operating in a privacy-preserving domain, ensuring that neither the query pattern nor the database contents are revealed.

Ultimately, the Privacy-Preserving Boyer-Moore framework offers a viable path forward for forensic science, one that does not force a choice between public safety and civil liberties. While the computational overhead of cryptographic methods remains a challenge for large-scale, real-time deployment, this research establishes a critical proof-of-concept. It underscores the importance of developing and adopting privacy-enhancing technologies to build public trust and ensure that the powerful tools of forensic genetics are used responsibly and ethically in a modern, data-conscious society.

Video Link at Youtube

https://youtu.be/_FWaW40G7OE?si=yT-djaudW5uSDmu-

Repository

https://github.com/graceevelyns/dna-search

ACKNOWLEDGMENT

The author expresses heartfelt gratitude to God Almighty for granting the strength, perseverance, and opportunity to successfully complete this paper. The completion of this paper would not have been possible without the guidance of Dr. Ir. Rinaldi Munir, whose dedication to teaching and insightful guidance throughout the Algorithm Strategy course have been invaluable. The writer extends gratitude for providing the necessary knowledge during the development of this paper. A debt of gratitude is also owed to Dr. Ir. Rinaldi Munir, whose extensive learning resources and contributions to the field have greatly enriched the writer's understanding and facilitated the completion of this work.

REFERENCES

[1] Federal Judicial Center, "Law enforcement databases: Limited genetic information and varying procedures for use," (n.d.). [Online]. Available: https://www.fjc.gov/content/361262/law-enforcement-databases-limited-genetic-information-and-varying-procedures-use

[2] R. Munir, "Pencocokan string (String/Pattern Matching)," Course Material for IF2211 Strategi Algoritma, Program Studi Teknik Informatika, STEI-ITB, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf.

[3] R. S. Boyer and J. S. Moore, "A fast string searching algorithm," Communications of the ACM, vol. 20, no. 10, pp. 762-772, Oct. 1977.

[4] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or a completeness theorem for protocols with honest majority," in Proc. 19th Annu. ACM Symp. Theory Comput., 1987, pp. 218–229.

[5] Petrie-Flom Center, "Ethical concerns of DNA databases used for crime control," Bill of Health, Jan. 14, 2019. [Online]. Available: https://petrieflom.law.harvard.edu/2019/01/14/ethical-concerns-of-dna-databases-used-for-crime-control/

PERNYATAAN

Hereby, I declare that this paper I have written is my own work, not a reproduction or translation of someone else's paper, and not plagiarized.

Bandung, 24 Juni 2025

Grace Evelyn Simon, 13523087