Abstract. In this paper we consider the problem of secure pattern matching that allows single character wildcards and substring matching in the malicious (stand-alone) setting. Our protocol, called 5PM, is executed between two parties: Server, holding a text of length $n$, and Client, holding a pattern of length $m$ to be matched against the text, where our notion of matching is more general and includes non-binary alphabets, non-binary Hamming distance and non-binary substring matching.

5PM is the first protocol with communication complexity sub-linear in circuit size to compute non-binary substring matching in the malicious model (general MPC has communication complexity which is at least linear in the circuit size). 5PM is also the first sublinear protocol to compute non-binary Hamming distance in the malicious model. Additionally, in the honest-but-curious (semi-honest) model, 5PM is asymptotically more efficient than the best known scheme when amortized for applications that require single character wildcards or substring pattern matching. 5PM in the malicious model requires $O((m+n)k^2)$ bandwidth and $O(m + n)$ encryptions, where $m$ is the pattern length and $n$ is the text length. Further, 5PM can hide pattern size with no asymptotic additional costs in either computation or bandwidth. Finally, 5PM requires only 2 rounds of communication in the honest-but-curious model and 8 rounds in the malicious model. Our techniques reduce pattern matching and generalized Hamming distance problem to a novel linear algebra formulation that allows for generic solutions based on any additively homomorphic encryption. We believe our efficient algebraic techniques are of independent interest.

Keywords: Secure pattern matching, wildcard pattern matching, substring pattern matching, non-binary Hamming distance, secure two-party computation, malicious adversary, full simulation, homomorphic encryption, threshold encryption.

1 Introduction

Pattern matching is fundamental to computer science. It is used in many areas, including text processing, database search, networking and security applications.

---

1 The full version of this paper can be found at [http://www.ics.uci.edu/~keldefra/papers/5pm_scn12.pdf](http://www.ics.uci.edu/~keldefra/papers/5pm_scn12.pdf)

and recently in the context of bioinformatics and DNA analysis \cite{3,4,5}. It is a problem that has been extensively studied, resulting in several efficient (although insecure) techniques to solve its many variations, e.g., \cite{6,7,8,9}. The most common interpretation of the pattern matching problem is the following: given a finite alphabet $\Sigma$, a text $T \in \Sigma^n$ and a pattern $p \in \Sigma^m$, the exact pattern matching decision problem requires one to decide whether or not a pattern appears in the text. The exact pattern matching search problem requires finding all indices $i$ of $T$ (if any) where $p$ occurs as a substring starting at position $i$. If we denote by $T_i$ the $i$th character of $T$, the output should be the set of matching positions $MP := \{i \mid p \text{ matches } T \text{ beginning at } T_i \}$. The following generalizations of the exact matching problem are often encountered, where the output in all cases is the set $MP$:

- **Pattern matching with single character wildcard**\cite{2}: There is a special character “$*$” $\not\in \Sigma$ that matches any single character of the alphabet, where $p \in \{\Sigma \cup \{\ast\}\}^m$ and $T \in \Sigma^n$. Using such a “wildcard” character allows one pattern to be specified that could match several sequences of characters. For example the pattern “$T*A$” , would match any of the following character sequence in a text\cite{3}: $TAA$, $TAC$, $TAG$, and $TAT$.

- **Substring pattern matching**: Fix some $l \leq m$; a match for $p$ is found whenever there exists in $T$ an $m$-length string that differs in $l$ characters from $p$ (i.e., has Hamming distance $l$ from $p$). For example, the pattern “$TAC$” has $m = 3$. If $l = 1$, then any of the following words would match: $\ast AC$, $T \ast C$, or $TA\ast$; note that this is an example of non-binary substring matching.

A secure version of pattern matching has many applications. For example, secure pattern matching can help secure databases containing medical information, such as DNA records, while still allowing one to perform pattern matching operations on such data. The need for privacy-preserving DNA matching has been highlighted in recent papers \cite{10,11,12}. In addition to the case of DNA matching, where substring matching may be particularly useful, Hamming distance-based approximate matching has also been demonstrated in the case of secure facial recognition \cite{3}. We note that both of these settings require computation over non-binary alphabets.

1.1 **Our Contributions**

This paper presents a new protocol for arbitrary alphabets, 5ecure Pattern Matching (or 5PM), that addresses, in addition to exact matching, more expressive search queries including single character wildcards and substring pattern matching, in addition to providing the ability to hide pattern length.

5PM is the first protocol with communication complexity sub-linear in circuit size (as opposed to general MPC, which has communication complexity linear in

\footnote{Such wildcards are also called “do not cares” and “mismatches” in the literature.}

\footnote{Here and throughout, we use the DNA alphabet ($\Sigma = \{A,C,G,T\}$) for examples.