Volume 27 Number 1

ACTA CYBERNETICA

Szeged, 2025

ACTA CYBERNETICA

Information for authors. Acta Cybernetica publishes only original papers in the field of Computer Science. Manuscripts must be written in good English. Contributions are accepted for review with the understanding that the same work has not been published elsewhere. Papers previously published in conference proceedings, digests, preprints are eligible for consideration provided that the author informs the Editor at the time of submission and that the papers have undergone substantial revision. If authors have used their own previously published material as a basis for a new submission, they are required to cite the previous work(s) and very clearly indicate how the new submission offers substantively novel or different contributions beyond those of the previously published work(s). There are no page charges. An electronic version of the published paper is provided for the authors in PDF format.

Manuscript Formatting Requirements. All submissions must include a title page with the following elements: title of the paper; author name(s) and affiliation; name, address and email of the corresponding author; an abstract clearly stating the nature and significance of the paper. Abstracts must not include mathematical expressions or bibliographic references.

References should appear in a separate bibliography at the end of the paper, with items in alphabetical order referred to by numerals in square brackets. Please prepare your submission as one single PostScript or PDF file including all elements of the manuscript (title page, main text, illustrations, bibliography, etc.).

When your paper is accepted for publication, you will be asked to upload the complete electronic version of your manuscript. For technical reasons we can only accept files in LaTeX format. It is advisable to prepare the manuscript following the guidelines described in the author kit available at https://cyber.bibl.u-szeged.hu/index.php/actcybern/about/submissions even at an early stage.

Submission and Review. Manuscripts must be submitted online using the editorial management system at https://cyber.bibl.u-szeged.hu/index.php/actcybern/submission/wizard. Each submission is peer-reviewed by at least two referees. The length of the review process depends on many factors such as the availability of an Editor and the time it takes to locate qualified reviewers. Usually, a review process takes 6 months to be completed.

Subscription Information. Acta Cybernetica is published by the Institute of Informatics, University of Szeged, Hungary. Each volume consists of four issues, two issues are published in a calendar year. From 2024, issues are published online only, and articles are made available as soon as they are accepted and copyedited. The content is available free of charge.

Contact information. Acta Cybernetica, Institute of Informatics, University of Szeged. P.O. Box 652, H-6701 Szeged, Hungary. Tel: +36 62 546 396, Fax: +36 62 546 397, Email: acta@inf.u-szeged.hu.

Web access. The above information along with the contents of past and current issues are available at the Acta Cybernetica homepage https://cyber.bibl.u-szeged.hu/.

Zoltan Kato
Department of Image Processing and Computer Graphics, University of Szeged, Hungary
kato@inf.u-szeged.hu

Dragan Kukolj
RT-RK Institute of Computer Based Systems, Novi Sad, Serbia
dragan.kukolj@rt-rk.com

László Lovász
Department of Computer Science, Eötvös Loránd University, Budapest, Hungary
lovasz@cs.elte.hu

Kálmán Palágyi
Department of Image Processing and Computer Graphics, University of Szeged, Hungary
palagyi@inf.u-szeged.hu

Dana Petcu
Department of Computer Science, West University of Timisoara, Romania
petcu@info.uvt.ro

Andreas Rauh
School II – Department of Computing Science, Group Distributed Control in Interconnected Systems, Carl von Ossietzky Universität Oldenburg, Germany
andreas.rauh@uni-oldenburg.de

György Vaszil
Department of Computer Science, Faculty of Informatics, University of Debrecen, Hungary
vaszil.gyorgy@inf.unideb.hu

Special Issue of the International Conference on Applied Informatics 2023

Guest Editors

Imre Varga
University of Debrecen, Hungary
varga.imre@inf.unideb.hu

Gergely Kovásznai
Eszterhazy Karoly Catholic University, Eger, Hungary
kovasznai.gergely@uni-eszterhazy.hu

Acta Cybernetica **27** (2025) 3–3.

Preface

In today's dynamic and ever-evolving digital landscape, applied informatics plays a pivotal role in shaping the future of technology. From refining algorithms for enhanced data analysis to optimizing communication networks and advancing artificial intelligence, the realm of applied informatics continues to drive innovation and transformation across industries.

This current Special Issue features contributions from the 12th International Conference on Applied Informatics (ICAI 2023), which was held in Eger, Hungary on March 2–4, 2023. These research papers explore novel insights, innovative methodologies, and practical applications within the field of computer science and informatics. Each of them represents a valuable contribution to the applied informatics field and offers insights that bridge the gap between theory and practical application. They are a testament to the diversity and dynamism of our field, showcasing a wide range of research topics and applications.

Imre Varga and Gergely Kovásznai
Guest Editors

Acta Cybernetica **27** (2025) 5–14.

Invariants and String Properties in the Analysis of the Knuth-Morris-Pratt Algorithm*

Tibor Ásványi[a]

Abstract

This paper is about the string-matching problem. We find the occurrences of a pattern inside a text. First, we formally define the problem and summarise its naive solution. Next, we analyse an efficient method, the Knuth-Morris-Pratt (KMP) algorithm.

We prove the correctness of the KMP algorithm. We also analyse its efficiency. Our reasoning is based on the properties of the pattern and the text. It is also based on the invariant properties of KMP. In this way, we could develop an extremely compact and elegant proof. And the method of proving program correctness with the invariant properties of the program is already familiar to our students at our university.

Keywords: string-matching, valid shift, prefix function, suffix, invariant

1 Notations and basic notions

$\mathbb{N} = \{\, i \in \mathbb{Z} \mid i \geq 0 \,\}$ $\qquad$ $i\,..\,k = [i\,..\,k] = \{\, j \in \mathbb{N} \mid i \leq j \leq k \,\}$
$[i\,..\,k) = \{\, j \in \mathbb{N} \mid i \leq j < k \,\}$ $\qquad$ $(i\,..\,k) = \{\, j \in \mathbb{N} \mid i < j < k \,\}$
$\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_d\}$ is the *alphabet* where $d \in \mathbb{N} \wedge d > 0$

$T : \Sigma[n]$ is the *text*, an array of letters. $T = T[0\,..\,n) = T[0\,..\,n{-}1]$.
Provided that $0 \leq i \leq l \leq n$, $T[i\,..\,l)$ is a *string;* $T[i\,..\,i) = \varepsilon$ is the *empty string.*
$T[j\,..\,k)$ is a *substring* of $T[i\,..\,l)$ if $i \leq j \leq k \leq l$.

$P : \Sigma[m]$ is the *pattern* $(0 < m \leq n)$. We search for the occurrences of P in T, i.e. we look for the substrings of T equal to P.
$P_{:j} = P[0\,..\,j) = P[0\,..\,j{-}1]$ and similarly for T. $\quad$ Thus, $P_{:0} = \varepsilon$.

$P[j\,..\,k) \sqsupseteq P[i\,..\,k)$ (string $P[j\,..\,k)$ is a *suffix* of string $P[i\,..\,k)$) if $i \leq j \leq k$.
Consequently, $P_{:0} \sqsupseteq P_{:j}$ because $P_{:0} = \varepsilon = P[j\,..\,j) \sqsupseteq P[0\,..\,j)$.

*Thanks to our faculty leaders for financial support, my colleagues for the encouragement and my students for the questions.

[a]Eötvös Loránd University, Faculty of Informatics, Pázmány Péter sétány 1/C, Budapest, 1117, Hungary, E-mail: asvanyi@inf.elte.hu, ORCID: 0000-0002-1715-9195

DOI: 10.14232/actacyb.308844

$P[j\,..\,k) \sqsupset P[i\,..\,k)$ (string $P[j\,..\,k)$ is a *proper suffix* of string $P[i\,..\,k)$) if $i < j \leq k$.
Thus, $P_{:0} \sqsupset P_{:j}$ if $j > 0$. (Proof: $P_{:0} = P[j\,..\,j) \sqsupset P[0\,..\,j)$ if $j > 0$.)

$P[i\,..\,j)$ is a *proper prefix* of $P[i\,..\,k)$) if $i \leq j < k$.
String x is a *proper prefix-suffix* (*PPS*) of y if x is a proper prefix and suffix of y.
Consequently, $P_{:i}$ is a *PPS* of $P_{:j}$ if $P_{:i} \sqsupset P_{:j}$ (because $P_{:i} \sqsupset P_{:j}$ implies $i < j$).

The following three lemmas on strings will be useful. (Most of them are illustrated in [2]. We present their proof here.) The first one tells us that a proper suffix of a string's suffix is a proper suffix of this string.

Lemma 1.1. *(Transitivity of the suffix relation)*
$x \sqsupset y \wedge y \sqsupseteq z \Rightarrow x \sqsupset z$.

Proof. We can suppose that $z = R[i\,..\,l)$ where $R[i\,..\,l)$ is a string. Thus, $y \sqsupseteq z$ means that $y = R[j\,..\,l)$ where $i \leq j \leq l$; and $x \sqsupset y$ means that $x = R[k\,..\,l)$ where $j < k \leq l$. Consequently, $i < k \leq l$. Therefore, $x = R[k\,..\,l) \sqsupset R[i\,..\,l) = z$. □

The second lemma contains three statements. Given two suffixes of a string, (1) one is a suffix of the other if it is not longer than the other, (2) one is a proper suffix of the other if it is shorter than the other, (3) they are equal if their lengths are equal.

Lemma 1.2. *(Overlapping-suffix lemma)*
Suppose that x, y and z are strings such that $x \sqsupseteq z$ and $y \sqsupseteq z$.
$|x| \leq |y| \Rightarrow x \sqsupseteq y$. $|x| < |y| \Rightarrow x \sqsupset y$. $|x| = |y| \Rightarrow x = y$.

Proof. We can suppose that $z = R[i\,..\,l)$. Thus, $x \sqsupseteq y$ means that $x = R[j\,..\,l)$ where $i \leq j \leq l$; and $y \sqsupseteq z$ means that $y = R[k\,..\,l)$ where $i \leq k \leq l$. Consequently, (1) $|x| \leq |y|$ means that $l - j \leq l - k$; therefore, $k \leq j \leq l$, i.e. $x = R[j\,..\,l) \sqsupseteq R[k\,..\,l) = y$. (2) $|x| < |y|$ means that $l - j < l - k$; therefore, $k < j \leq l$, i.e. $x = R[j\,..\,l) \sqsupset R[k\,..\,l) = y$. (3) $|x| = |y|$ means that $l - j = l - k$; therefore, $k = j \leq l$, i.e. $x = R[j\,..\,l) = R[k\,..\,l) = y$. □

The third lemma tells us that given two nonempty strings (the strings on the right side of the logical equivalences), one is a (proper) suffix of the other if and only if their last letters are the same and the first without its last letter is a (proper) suffix of the second without its last letter.

Lemma 1.3. *(Suffix-extension lemma)*
$P_{:j} \sqsupseteq T_{:i} \wedge P[j] = T[i] \iff P_{:j+1} \sqsupseteq T_{:i+1}$.
$P_{:i} \sqsupset P_{:j} \wedge P[i] = P[j] \iff P_{:i+1} \sqsupset P_{:j+1}$.

Proof.
(1) $P_{:j+1} = P[0\,..\,j] \wedge T_{:i+1} = T[0\,..\,i]$. Thus,
$P_{:j+1} \sqsupseteq T_{:i+1} \iff P[0\,..\,j] = T[i-j\,..\,i] \iff$
$P[0\,..\,j) = T[i-j\,..\,i) \wedge P[j] = T[i] \iff P_{:j} \sqsupseteq T_{:i} \wedge P[j] = T[i]$.

(2) $P_{:i+1} = P[0\,..\,i] \wedge P_{:j+1} = P[0\,..\,j]$. Thus,
$P_{:i+1} \sqsupset P_{:j+1} \iff i < j \wedge P[0\,..\,i] = P[j-i\,..\,j] \iff$
$i < j \wedge P[0\,..\,i) = P[j-i\,..\,j) \wedge P[i] = P[j] \iff P_{:i} \sqsupset P_{:j} \wedge P[i] = P[j]$. □

2 Introduction

In this paper, we suppose that $P : \Sigma[m]$, $T : \Sigma[n]$ and their lengths, i.e. m and n are fixed where $0 < m \leq n$. We search for those shifts s of P on T where $T[s\,..\,s{+}m) = P[0\,..\,m)$. Clearly, s must be in $0\,..\,n{-}m$.

Definition 2.1. *$s \in 0\,..\,n{-}m$ is a* possible shift *of P on T.*
It is a valid shift **if** *$T[s\,..\,s{+}m) = P[0\,..\,m)$. Otherwise, it is an* invalid shift*.*

Problem 2.1 (String-matching)**.** *Compute the set V of the valid shifts of P on T, i.e.*

$$V = \{\, s \in 0\,..\,n{-}m \mid T[s\,..\,s{+}m) = P[0\,..\,m) \,\}$$

The naive string-matching (Brute-Force) algorithm checks each possible shift in order and collects the valid shifts with maximal (i.e. worst-case) time complexity $\Theta((n{-}m{+}1){*}m)$, which is $\Theta(n^2)$ if $m = \lfloor n/2 \rfloor$. [2] (See the details of this Θ-notation in the first chapter of [4].)

More advanced methods – like the different versions of the Boyer-Moore [1, 3], Rabin-Karp, and KMP [2] algorithms – use information gained about the pattern and the text. They do not check each possible shift of P on T but often make a jump in T.

We prefer KMP because it runs in $\Theta(n)$ time on all the possible inputs, and it never backtracks on T, making it easy to implement on sequential files. KMP is traditionally introduced as a highly efficient simulation of *string matching with finite automata* [2]. Here, we avoid these automata and start with analysing P and T, i.e. the strings. To introduce KMP, let us see Example 2.1.

Example 2.1. In this example, we suppose there is a longer text T, but we consider only $T[i{-}5\,..\,i{+}2] = BABABABB$ here. The pattern is $P = P_{:6} = BABABB$. The actual shift is $i - 5$. The successfully matched characters are underlined. The unsuccessfully matched character is crossed out.

$\ldots$	T_{i-5}	T_{i-4}	T_{i-3}	T_{i-2}	T_{i-1}	T_i	T_{i+1}	T_{i+2}
$\ldots$	B	A	B	A	B	A	B	B
$P =$	$\underline{B}$	$\underline{A}$	$\underline{B}$	$\underline{A}$	$\underline{B}$	$\cancel{B}$		
			B	A	B	$\underline{A}$	$\underline{B}$	$\underline{B}$

In the third line of the table, we successfully matched $P_{:5}$ to $T[i{-}5\,..\,i)$ but $P[5] \neq T[i]$. Consequently, $i{-}5$ is not a valid shift.

Thus we make a *minimal additional shift* of P on T so that the $P_{:k}$ $(0 \leq k < 5)$ which is still against $T[i{-}k\,..\,i)$ matches it, i.e. $P_{:k} \sqsupset T_{:i}$. (See the last line of the table.) *This shift* is ≤ 5, because $P_{:0} \sqsupset T_{:i}$. And with *this shift*, we do not jump over any possibly valid shift. Actually $k = 3$. Then we successfully match $P[3]$ to $T[i]$, $P[4]$ to $T[i{+}1]$ and $P[5]$ to $T[i{+}2]$. Thus, $i{-}3$ is a valid shift. The bigger possible shifts would jump over the valid shift $i{-}3$.

Understanding the previous example, the question remains: How do we efficiently determine the value of k above? In the previous example, $j = 5$, but the following argument can be applied to any $j \in 1\,..\,m$ where $i - j$ is the actual shift, $P_{:j} \sqsupseteq T_{:i}$ and $(P[j] \neq T[i] \vee j = m)$.

Unquestionably, a greater additional shift corresponds to a smaller k, and a smaller additional shift corresponds to a greater k. And k corresponds to the *minimal additional shift* of P on T so that $P_{:k} \sqsupset T_{:i}$. Thus, k is the greatest h so that $P_{:h} \sqsupset T_{:i}$ and $0 \leq h < j$. Moreover, $P_{:h} \sqsupset T_{:i}$ is equivalent to $P_{:h} \sqsupset P_{:j}$ because $P_{:j} \sqsupseteq T_{:i}$ and $0 \leq h < j$. Consequently, k is the greatest h so that $P_{:h} \sqsupset P_{:j}$.

As a result, k depends only on $P_{:j}$. After all, we need the longest *PPS* of $P_{:j}$. Its length is defined by the prefix function π. Because P is fixed, this length depends only on j.

Definition 2.2. $\pi(j) = \max\{\, h \in 0\,..\,j-1 \mid P_{:h} \sqsupset P_{:j} \,\} \qquad (j \in 1\,..\,m)$

An efficient calculation of this function is given in Section 4, but before it, in Section 3, we analyse the main procedure of the KMP algorithm.

3 The KMP algorithm

In this section, first, we transform the intuitive approach of the KMP algorithm of the Introduction into **Algorithm 1**. Next, we analyse it.

Algorithm 1 the Knuth-Morris-Pratt algorithm

```
procedure KMP(T : Σ[n] ; P : Σ[m] ; S : ℕ{})
 1: π : ℕ[1..m] ; INIT(π, P) ; S := {} ; i := j := 0
 2: while i < n do
 3:    if P[j] = T[i] then
 4:       i++ ; j++
 5:       if j = m then
 6:          S := S ∪ {i − m} ; j := π[j]
 7:       end if
 8:    else if j = 0 then
 9:       i++
10:    else
11:       j := π[j]
12:    end if
13: end while
```

We will prove in Section 4 that INIT(π, P) collects the values of the π prefix function into the $\pi[1\,..\,m]$ prefix array in $\Theta(m)$ time. And the postcondition of the INIT(π, P) call is $\forall j \in 1\,..\,m : \pi[j] = \pi(j)$. (See **Algorithm 2**.) Our analysis of the KMP algorithm is based on the invariant (Inv) of Theorem 3.1 where $i - j$ is the actual shift. (See Section 2.4 of [4] on an exact program correctness proof with loop invariant.)

3.1 The partial correctness of the procedure KMP(T, P, S)

The following lemma will be appropriate where $i-j$ is the actual shift. (It tells us the following. When we search for a valid shift of P on T, $i-j$ is P's actual shift and $\varepsilon \neq P[0\,..\,j) = T[i-j\,..\,i)$, then the shift values between $i-j$ and $i-\pi(j)$ are invalid: We find no solution there; and this does not depend on the validity of the actual shift, i.e. $i-j$.)

Lemma 3.1. $j \in 1\,..\,m \wedge P_{:j} \sqsupseteq T_{:i} \Rightarrow$ *there is no valid shift in* $(i-j\,..\,i-\pi(j))$.

Proof. Assume indirectly that $k \in (\pi(j)\,..\,j)$ and $i-k$ is a valid shift. This means $T[i-k\,..\,i-k+m) = P[0\,..\,m)$. Clearly, $k < j \leq m$, therefore $k < m$. Thus $T[i-k\,..\,i) = P[0\,..\,k)$, i.e. $P_{:k} \sqsupseteq T_{:i}$. And $P_{:j} \sqsupseteq T_{:i} \wedge k < j$. As a result, $P_{:k} \sqsupset P_{:j}$ because of the Overlapping-suffix lemma (1.2). But $k > \pi(j)$. For this reason, $P_{:k} \not\sqsupset P_{:j}$ follows from Definition 2.2 of the π function. □

The following theorem is the key to the KMP algorithm. It formulates an invariant property of its main loop. Again, $i-j$ is P's actual shift and $P[0\,..\,j) = T[i-j\,..\,i)$, but $P[0\,..\,j)$ is not the the whole pattern. Thus, we do not know whether $i-j$ is a valid shift, but we know that the S set already contains all the valid shifts before the actual shift.

Theorem 3.1.
Statement (Inv) *is an invariant of the loop of the procedure KMP*(T, P, S).
(Inv) $\quad P_{:j} \sqsupseteq T_{:i} \wedge 0 \leq j \leq i \leq n \wedge j < m \;\wedge S = V \cap [0\,..\,i-j)$.

Proof. Immediately before the first loop iteration, we perform the $S := \{\}$; $i := j := 0$ initialisations. Thus, (Inv) holds because $i = j = 0 \wedge P_{:0} \sqsupseteq T_{:0} \wedge 0 \leq 0 \leq 0 \leq n \wedge 0 < m \wedge S = \{\} = V \cap [0\,..\,0)$.

We prove that each iteration of the loop keeps (Inv). The postcondition of the init(π, P) call, i.e. $(\forall j \in 1\,..\,m : \pi[j] = \pi(j))$ is implicitly added to each statement.

Supposing that $i < n$, we enter the loop and
(Inv1) $\quad P_{:j} \sqsupseteq T_{:i} \wedge 0 \leq j \leq i < n \wedge j < m \;\wedge S = V \cap [0\,..\,i-j)$ $\quad$ stands.

- If $P[j] = T[i]$, then
  $P_{:j+1} \sqsupseteq T_{:i+1}$ according to the Suffix-extension lemma (1.3). After increasing i and j we have
  (Inv2) $\quad P_{:j} \sqsupseteq T_{:i} \wedge 0 < j \leq i \leq n \wedge j \leq m \;\wedge S = V \cap [0\,..\,i-j)$.

  1. If $j = m$, then $P_{:m} \sqsupseteq T_{:i}$, i.e. $P[0\,..\,m) = T[i-m\,..\,i)$. This means $i-m$ is a valid shift. Thus, we add it to S. Then, we have the following statement.
     (Inv3) $\quad P_{:j} \sqsupseteq T_{:i} \wedge 0 < j \leq i \leq n \wedge j \leq m \;\wedge S = V \cap [0\,..\,i-j]$.
     Because $j \in 1\,..\,m \wedge P_{:j} \sqsupseteq T_{:i}$, considering Lemma 3.1, we receive that there is no valid shift in the interval $(i-j\,..\,i-\pi(j))$. Thus
     $P_{:j} \sqsupseteq T_{:i} \wedge 0 < j \leq i \leq n \wedge j \leq m \;\wedge S = V \cap [0\,..\,i-\pi(j))$.
     Consider that $P_{:\pi(j)} \sqsupset P_{:j} \sqsupseteq T_{:i}$. Based on the transitivity of the suffix

relation (Lemma 1.1) and $\pi[j] = \pi(j)$:
$P_{:\pi[j]} \sqsupset T_{:i} \wedge 0 < j \leq i \leq n \wedge j \leq m \ \wedge S = V \cap [0\,..\,i{-}\pi[j])$.
Because $\pi[j] = \pi(j) \in [0\,..\,j)$, after the $j := \pi[j]$ assignment already $0 \leq j < i \wedge j < m$ stands. Consequently,
$P_{:j} \sqsupset T_{:i} \wedge 0 \leq j < i \leq n \wedge j < m \ \wedge S = V \cap [0\,..\,i{-}j)$ holds. At the end of the first program branch, this directly implies (Inv).

2. Provided that $j \neq m$, (Inv2) implies $j < m$. Thus, (Inv) holds at the end of the second program branch.

- In case of $P[j] \neq T[i]$, the Suffix-extension lemma (1.3) implies $P_{:j+1} \not\sqsupseteq T_{:i+1}$, i.e. $P[0\,..\,j] \neq T[i-j\,..\,i]$. Based on (Inv1), $j < m$. Thus, $P[0\,..\,m) \neq T[i-j\,..\,i-j+m)$ if $i-j+m < n$. Consequently, $i{-}j$ is not a valid shift. Comparing this to (Inv1), i.e.
$P_{:j} \sqsupseteq T_{:i} \wedge 0 \leq j \leq i < n \wedge j < m \ \wedge S = V \cap [0\,..\,i{-}j)$, we have
(Inv4) $P_{:j} \sqsupseteq T_{:i} \wedge 0 \leq j \leq i < n \wedge j < m \ \wedge S = V \cap [0\,..\,i{-}j]$.

3. If $j = 0$, then considering (Inv4) we receive
$P_{:0} \sqsupseteq T_{:i} \wedge 0 = j \leq i < n \wedge j < m \ \wedge S = V \cap [0\,..\,i{-}j]$.
After performing $i++$,
$P_{:0} \sqsupseteq T_{:i} \wedge 0 = j \leq i \leq n \wedge j < m \ \wedge S = V \cap [0\,..\,i{-}j)$ stands.
Therefore, (Inv) holds at the end of the third program branch:
(Inv) $P_{:j} \sqsupseteq T_{:i} \wedge 0 \leq j \leq i \leq n \wedge j < m \ \wedge S = V \cap [0\,..\,i{-}j)$.
4. Provided that $j \neq 0$, then taking (Inv4) into account, we obtain
$P_{:j} \sqsupseteq T_{:i} \wedge 0 < j \leq i < n \wedge j < m \ \wedge S = V \cap [0\,..\,i{-}j]$
This has the direct consequence
(Inv3) $P_{:j} \sqsupseteq T_{:i} \wedge 0 < j \leq i \leq n \wedge j \leq m \ \wedge S = V \cap [0\,..\,i{-}j]$.
We have already seen in the examination of the first program branch that in the case of (Inv3), after performing assignment $j := \pi[j]$, the invariant (Inv) holds. Finally, (Inv) also stands at the end of the last program branch.

□

Theorem 3.2. *If the KMP algorithm terminates, it solves Problem 2.1 of string-matching, i.e. $S = V$ holds when it returns.*

Proof. Let us consider Theorem 3.1. The (Inv) invariant of KMP's loop with the loop's termination condition, i.e. $P_{:j} \sqsupseteq T_{:i} \wedge 0 \leq j \leq i \leq n \wedge j < m \wedge S = V \cap [0\,..\,i{-}j)$ with $i \geq n$ implies that $i = n \wedge j < m \wedge S = V \cap [0\,..\,n{-}j)$ holds when the loop of KMP becomes completed. Furthermore, $j < m \Rightarrow [0\,..\,n{-}j) \supset [0\,..\,n{-}m) \supseteq \{\, s \in 0\,..\,n{-}m \mid T[s\,..\,s{+}m) = P[0\,..\,m) \,\} = V \Rightarrow [0\,..\,n{-}j) \supset V$. Thus $S = V \cap [0\,..\,n{-}j) = V$. Consequently, $S = V$ holds when the procedure KMP returns. □

3.2 The termination of the procedure KMP(T, P, S)

First, we prove that the loop iterates at least n times. Before the first iteration, $i = 0$. Each iteration increases i by 1 or 0. And the loop terminates with $i = n$

according to the $i < n$ condition and the $0 \leq i \leq n$ invariant. Thus, there are at least n iterations before the loop terminates.

Second, we prove that the loop iterates at most $2n$ times. Let the termination function be $2i - j$ where $0 \leq j \leq i \leq n$ [see the (Inv) invariant in Theorem 3.1]. Thus $2i-j \in 0\,..\,2n$. Before the loop, $2i-j = 0$, and each iteration increases $2i-j$. Consequently, there are at most $2n$ iterations before the loop terminates.

The loop of KMP runs in $\Theta(n)$ time because n is at least the number of the iterations of the KMP loop, which is at most $2n$.

Remember that we will prove in Section 4 that the INIT(π, P) call terminates in $\Theta(m)$ time.

As a result, the time complexity of the KMP(T, P, S) procedure is $\Theta(n) + \Theta(m) = \Theta(n)$ because $n \geq m \geq 0$.

4 Initializing the prefix array

In this section, we will compute the values of the $\pi : [1\,..\,m] \to [0\,..\,m)$ prefix function (see Definition 2.2) and store them in the $\pi[1\,..\,m]$ array. Remember that $\pi(j)$ is the length of the longest PPS of $P_{:j}$. Thus, $\pi(1) = 0$, and we can perform $\pi[1] := 0$. Subsequently, provided that we have filled $\pi[1\,..\,j]$ where $1 \leq j < m$, we want to calculate $\pi(j+1)$, store it in $\pi[j+1]$ and so on.

$\pi(j+1)$ is the length of the longest PPS of $P_{:j+1}$. If this PPS is nonempty, let us denote it with $P_{:k+1}$. Thus $0 \leq k < j$. According to the 1.3 Suffix-extension lemma, $P_{:i+1} \sqsupset P_{:j+1} \iff P_{:i} \sqsupset P_{:j} \wedge P[i] = P[j]$. This means that $P_{:k}$ is the longest $P_{:i} \sqsupset P_{:j}$ where $P[i] = P[j]$. To determine $P_{:k}$ (and hence $P_{:k+1}$), we check the $P_{:i}$ PPSs of $P_{:j}$ in decreasing order according to i and find the first one where $P[i] = P[j]$. If we do not find such a $P_{:i}$, then $\pi(j+1) = 0$.

The question is, given a $P_{:i} \sqsupset P_{:j}$ where $i > 0$ and $P[i] \neq P[j]$, how to determine the next longest PPS of $P_{:j}$. Let it be $P_{:l}$. As a result, $P_{:i} \sqsupset P_{:j} \wedge P_{:l} \sqsupset P_{:j} \wedge l < i$. Thus, $P_{:l} \sqsupset P_{:i}$ (see Lemma 1.2) and it is the longest one. Consequently, $l = \pi(i)$. $\pi(i) = \pi[i]$ because $i < j$ and $\pi[1\,..\,j]$ is already calculated. As a result, we can apply $i := \pi[i]$ and have the following intuitive loop invariant:

$\pi[1\,..\,j]$ have been calculated where $1 \leq j \leq m$, and if $j < m$, then $P_{:i}$ is the longest PPS of $P_{:j}$ for which still there is a chance that $P[i] = P[j]$.

Based on this invariant, we can write the INIT(π, P) procedure, i.e. Algorithm 2.

The following lemmas will be appropriate to prove the correctness of Algorithm 2. The first one says that the π-values can only increase at most one by one (when they grow). When $\pi(j)$ has been calculated, it provides an upper limit for $\pi(j+1)$.

Lemma 4.1. $j \in [1\,..\,m) \Rightarrow \pi(j+1) \leq \pi(j) + 1$

Proof. If $\pi(j+1) = 0 \Rightarrow \pi(j+1) = 0 \leq 0 + 1 \leq \pi(j) + 1$ because $\pi(j) \geq 0$ by definition. If $\pi(j+1) > 0 \Rightarrow$ with Definition 2.2, $P_{:(\pi(j+1)-1)+1} = P_{:\pi(j+1)} \sqsupset P_{:j+1} \Rightarrow$ with Lemma 1.3, $P_{:\pi(j+1)-1} \sqsupset P_{:j} \Rightarrow$ again with Definition 2.2, $\pi(j+1) - 1 \leq \pi(j) \Rightarrow \pi(j+1) \leq \pi(j) + 1$. □

Algorithm 2 Knuth-Morris-Pratt initialization

procedure INIT($\pi : \mathbb{N}[1\,..\,m]$; $P : \Sigma[m]$)
1: $\pi[1] := i := 0$; $j := 1$
2: **while** $j < m$ **do**
3: **if** $P[i] = P[j]$ **then**
4: $i++$; $j++$; $\pi[j] := i$
5: **else if** $i = 0$ **then**
6: $j++$; $\pi[j] := 0$
7: **else**
8: $i := \pi[i]$
9: **end if**
10: **end while**

The next lemma also helps us when we are going to calculate $\pi(j+1)$, i.e. the length of the longest PPS of $P_{:j+1}$: In this situation, because of the following invariants of the INIT() procedure, the conditions of the lemma are almost satisfied. If $P[i] \neq P[j]$, then $\pi(j+1) \leq i$ will be found. In this case, if $i = 0$, then $\pi(j+1) = 0$ follows; and if $i > 0$, then the lemma reduces the upper bound of $\pi(j+1)$.

Lemma 4.2. $P_{:i} \sqsupset P_{:j} \wedge 0 < i \wedge j < m \wedge \pi(j+1) \leq i \Rightarrow \pi(j+1) \leq \pi(i) + 1$

Proof. If $\pi(j+1) = 0$ then $\pi(j+1) < 0 + 1 \leq \pi(i) + 1$ because $\pi(i) \geq 0$ by definition. Provided that $\pi(j+1) > 0$, $k := \pi(j+1) - 1$. Thus $i > k \geq 0$ and $k + 1 = \pi(j+1)$. By the definition of the π function, $P_{:k+1} \sqsupset P_{:j+1}$. Consequently $P_{:k} \sqsupset P_{:j}$. Considering $P_{:i} \sqsupset P_{:j}$ and $k < i$, we have $P_{:k} \sqsupset P_{:i}$, thus $k \leq \pi(i)$. Therefore $\pi(j+1) = k + 1 \leq \pi(i) + 1$. □

4.1 The partial correctness of the procedure INIT(π, P)

The following invariant of the INIT(π, P) procedure's loop is the formalised version of the intuitive loop invariant above. It is the key to the partial correctness of this subroutine, which means that $\forall k \in 1\,..\,m : \pi[k] = \pi(k)$ stands when it returns.

Theorem 4.1.
Statement (inv) *is an invariant of the loop of the procedure* INIT(π, P).
(inv) $P_{:i} \sqsupset P_{:j} \wedge (\,\forall k \in 1\,..\,j : \pi[k] = \pi(k)\,) \wedge$
$0 \leq i < j \leq m \wedge (\,j < m \rightarrow \pi(j+1) \leq i + 1\,)$.

Proof. Because of the initialisations $\pi[1] := i := 0$; $j := 1$, immediately before the first iteration of the loop, (inv) corresponds to the following statement: $P_{:0} \sqsupset P_{:1} \wedge (\,\forall k \in 1\,..\,1 : \pi[k] = \pi(k) = \pi(1) = 0\,) \wedge 0 \leq 0 < 1 \leq m \wedge (\,1 < m \rightarrow \pi(2) \leq 1\,)$. To prove the elements of this formula, the $P_{:0}$ empty string is a proper suffix of any nonempty string; according to Definition 2.2, $\pi(1) = 0$; the size m of the pattern P is not zero; and finally, because of Lemma 4.1, $\pi(1+1) \leq \pi(1) + 1 = 1$, provided that $m > 1$.

Still, we have to prove that the iterations of the loop keep the (inv) invariant, i.e. provided that (inv) holds before an iteration of the loop, it will also stand at the end of any branch of the loop's body. When we enter into the body of the loop, (inv) and the loop's condition ($j < m$) implies

(inv1) $P_{:i} \sqsupset P_{:j} \wedge (\,\forall k \in 1\,..\,j : \pi[k] = \pi(k)\,) \wedge$
$0 \leq i < j < m \wedge \pi(j+1) \leq i+1.$

1. If $P[i] = P[j]$, according to Lemma 1.3 we have $P_{:i+1} \sqsupset P_{:j+1}$ because $P_{:i} \sqsupset P_{:j}$ [see (inv1)]. Based on the definition of the π prefix function (2.2), $P_{:i+1} \sqsupset P_{:j+1}$ implies $\pi(j+1) \geq i+1$. But $\pi(j+1) \leq i+1$ is found in (inv1). Consequently, $\pi(j+1) = i+1$.
Performing the assignments $i++; j++; \pi[j] := i$,

   $P_{:i} \sqsupset P_{:j} \wedge (\,\forall k \in 1\,..\,j : \pi[k] = \pi(k)\,) \wedge 0 < i < j \leq m \wedge \pi(j) = i.$

   Provided that $j < m$, $\pi(j) = i$ and Lemma 4.1 implies $\pi(j+1) \leq \pi(j)+1 = i+1$. Therefore, at the end of the first branch of the loop's body, (inv) holds.

2-3. If $P[i] \neq P[j]$, then $P_{:i+1} \not\sqsupset P_{:j+1}$ because of Lemma 1.3. Thus, $\pi(j+1) \neq i+1$, according to Definition 2.2. In (inv1) we have $\pi(j+1) \leq i+1$. Consequently, $\pi(j+1) \leq i$.

   Comparing this to (inv1), we receive that (inv2) stands in line 5 before the if-statement:

   (inv2) $P_{:i} \sqsupset P_{:j} \wedge (\,\forall k \in 1\,..\,j : \pi[k] = \pi(k)\,) \wedge$
   $0 \leq i < j < m \wedge \pi(j+1) \leq i.$

   2. Provided that $i = 0$, consider $\pi(j+1) \leq i$ from (inv2).
   We have $\pi(j+1) = 0$ because the π function is non-negative.

      Comparing this to (inv2), after the assignments $j++;\ \pi[j] := 0,$ we receive

      $P_{:i} \sqsupset P_{:j} \wedge (\,\forall k \in 1\,..\,j : \pi[k] = \pi(k)\,) \wedge 0 = i < j \leq m \wedge \pi(j) = i.$

      If $j < m$, Lemma 4.1 and $\pi(j) = i$ implies $\pi(j+1) \leq \pi(j)+1 = i+1$. Therefore, at the end of the second branch of the loop's body, (inv) also holds.

   3. Provided that $i \neq 0$, then (inv2) implies (inv3):

      (inv3) $P_{:i} \sqsupset P_{:j} \wedge (\,\forall k \in 1\,..\,j : \pi[k] = \pi(k)\,) \wedge$
      $0 < i < j < m \wedge \pi(j+1) \leq i.$

      Thus, we can apply Lemma 4.2, and we have $\pi(j+1) \leq \pi(i)+1$.

      On the other hand, from (inv3) we receive $\pi[i] = \pi(i)$. Comparing this to Definition 2.2, we receive $P_{:\pi[i]} \sqsupset P_{:i}$. In (inv3), $P_{:i} \sqsupset P_{:j}$ is found. With Lemma 1.1, $P_{:\pi[i]} \sqsupset P_{:j}$ follows.

      Consider (inv3) and $\pi(j+1) \leq \pi(i)+1$. After the assignment $i := \pi[i]$, we receive

$P_{:i} \sqsupset P_{:j} \wedge (\,\forall k \in 1\,..\,j : \pi[k] = \pi(k)\,) \wedge$
$0 \leq i < j < m \wedge \pi(j+1) \leq i+1.$

Therefore, at the end of the last branch of the loop's body, (inv) also holds. □

Corollary 4.1. *The loop invariant* (inv) *and the negation of the loop's condition, i.e.* $j \geq m$*, implies* $j = m$*. As a result, procedure* INIT$(\pi : \mathbb{N}[1\,..\,m]\ ;\ P : \Sigma[m])$ *has the post-condition* $\forall k \in 1\,..\,m : \pi[k] = \pi(k)$.

4.2 The termination of the procedure INIT(π, P)

First, we prove that the loop iterates at least $m-1$ times. Before the first iteration, $j = 1$. Each iteration increases j by 1 or 0. And the loop terminates with $j = m$ according to the $j < m$ condition and the $j \leq m$ invariant. Thus, there are at least $m-1$ iterations before the loop terminates.

Second, we prove that the loop iterates at most $2m-2$ times. Let the termination function be $2j - i$ where $0 \leq i < j \leq m$. Thus $2j - i \in 2\,..\,2m$. Before the loop, $2j - i = 2$, and each iteration increases $2j - i$. Consequently, the loop iterates not more than $2m-2$ times.

As a result, the time complexity of the INIT(π, P) procedure is $\Theta(m)$.

5 Summary

In this paper, we found relatively simple and short proof of the correctness and efficiency of the KMP algorithm. (Compare it to 32.3-4 in [2].) It is based on
(1) the properties of strings,
(2) the appropriate invariant properties of the loops of the algorithm and
(3) the suitable termination functions of these loops.

References

[1] Boyer, R. S. and Moore, J. S. A fast string searching algorithm. *Communications of the ACM*, 20(10):762–772, 1977. DOI: 10.1145/359842.359859.

[2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *String Matching.* In *Introduction to Algorithms (4th Edition)*, pages 957–985. The MIT Press, 2022. ISBN: 978-0-262-04630-5.

[3] Crochemore, M. and Lecroq, T. Tight bounds on the complexity of the Apostolico-Giancarlo algorithm. *Information Processing Letters*, 63(4):195–203, 1997. DOI: 10.1016/S0020-0190(97)00107-5.

[4] Stinson, D. R. *Techniques for Designing and Analyzing Algorithms (1st Edition).* CRC Press (Taylor & Francis Group), Boca Raton, London, New York, 2022. DOI: 10.1201/9780429277412.

Acta Cybernetica **27** (2025) 15–41.

How Egocentric Distance Estimation Changes in Virtual Environments by Using a Desktop Display or the Gear VR*

Tibor Guzsvinecz[ab], Judit Szűcs[ac], and Erika Perge[de]

Abstract

Due to the importance of depth perception in virtual spaces, the combined effects of display devices and human factors on egocentric distance estimation were investigated. We developed a virtual environment that can assess distance estimation skills of users at 10 various distances, starting from 25 cm and ending at 160 cm. Our results show that people are either accurate or overestimate distances on a desktop display, while underestimation occurs with the Gear VR in most cases. Combined with display devices, human factors also had effects on distance estimates. With the Gear VR, 35.73% − 57.14% faster estimation times were obtained, and these can also be influenced by human factors and distances.

Keywords: desktop display, distance estimation, Gear VR, human-computer interaction, immersion, virtual reality

1 Introduction

The definition of *egocentric distance* is the distance between the observer and the object. The perception of egocentric distances is crucial as it is required for reaching, grasping, and interception tasks [4, 25]. Since this is a cognitive skill, it can be

*The first author was supported by the ÚNKP-22-4 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. This work has been implemented by the TKP2021-NVA-10 project with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the 2021 Thematic Excellence Programme funding scheme.

[a]Department of Information Technology and its Applications, Faculty of Information Technology, University of Pannonia, Gasparich M. utca 18/A, 8900 Zalaegerszeg, Hungary

[b]E-mail: guzsvinecz.tibor@zek.uni-pannon.hu, ORCID: 0000-0003-3273-313X

[c]E-mail: szucs.judit@zek.uni-pannon.hu, ORCID: 0000-0002-9828-3322

[d]Department of Basic Technical Studies, Faculty of Engineering, University of Debrecen, Ótemető utca 2, 4028 Debrecen, Hungary

[e]E-mail: perge@eng.unideb.hu, ORCID: 0000-0003-1285-1374

DOI: 10.14232/actacyb.307740

trained over time [26]. As virtual spaces can stimulate cognitive functions [19, 22], this skill can be improved using virtual reality (VR) technologies [1]. Since VR also redefines human-computer interfaces, interaction with VR applications can differ between each of them [20]. When designing and creating virtual environments (VEs), developers should also focus on humans [9, 33]. They are equally important as other, non-human parts of a VR system [32, 21, 15], and they also play a crucial role in it [5]. Because humans are important in such a system, cognition is also of crucial importance [2, 17]. Therefore, VEs should be carefully designed, since they can affect spatial skills [29, 14, 11].

Due to the complexity of a VR system, several factors can affect egocentric distance estimation. These include human, technical, and compositional factors, along with distance itself [31, 34]. Consequently, distances are perceived differently in VEs [35, 18]. Depending on the display device used and distances, accuracy of estimates can change in a VR system. We also examined in another study that display devices can affect egocentric distance estimation [13]. For improved depth perception, users should be provided with binocular disparity [31, 8, 24]. Regarding human factors, gender could have an effect [10, 6], although there are conflicting observations in the literature [7, 16, 28]. Murgia and Sharkey assessed the height of participants, but no significant effects were observed in their research [27]. However, according to another study [23], if the virtual height is varied, an effect could appear. Age can also influence distance estimation [30]. As shown in the study by Bian and Andersen [3], the accuracy of distance estimates can increase with age. In our other study [12], we came to the conclusion that the possibilities of correct estimates can be affected by multiple human factors.

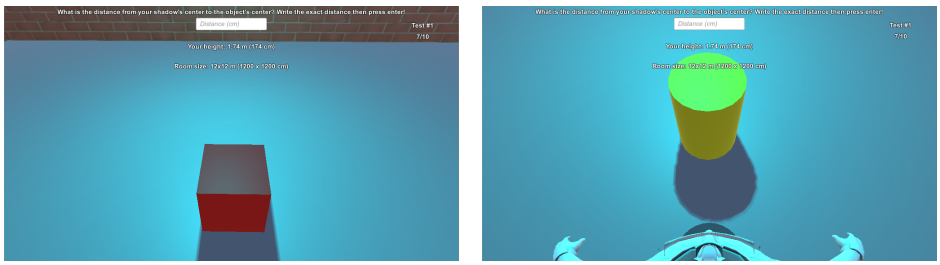
Naturally, there are other human factors beside gender, height, and age. Therefore, to understand the effect of several human factors, we evaluated multiple factors in addition to gender and height. In other words, we investigated the influence of handedness, video game playtime a week, what the participants study, and whether they wore glasses or had previous VR experience. Regarding technical factors, we examined immersion level with two display devices: a desktop display and the Gear VR. We also formed the following research question: *Does immersion level combined with these human factors influence egocentric distance estimation and its time?* To find an answer to this research question, we have developed a VE that can be used with the two display devices mentioned above. We used this VE to assess the egocentric distance estimation skills of several participants. While we have assessed the influence of human factors on the probabilities of correct distance estimates and the effect of display devices on estimates in other studies [13, 12], we have not compared the actual results grouped by human factors.

The structure of the article is as follows. In Section 2, we detail the materials and methods used in this research. Section 3 shows the results, while they are discussed in Section 4. Conclusions can be found in Section 5.

2 Materials and methods

The aforementioned VE can be used with a desktop display on PC, or with the Gear VR on Android. The used desktop display was an LG 20M37A (19.5") device, while the Gear VR had a Samsung Galaxy S6 Edge+ inside it. Overall, the egocentric distance estimation skills of 239 participants were measured. 157 people ($min_{age} = 17, max_{age} = 38, M_{age} = 19.80, SD_{age} = 2.09$) used the desktop display, while 72 ($min_{age} = 18, max_{age} = 42, M_{age} = 22.51, SD_{age} = 6.63$) used the Gear VR. Those participants who used the desktop display were either civil engineering, mechanical engineering, or vehicle engineering students. Contrarily, those who used the Gear VR were IT students. Participants joined the study of their own volition and no names were gathered. Before the measurements commenced, they had to input some parameters such as the age, gender, height, etc. in the VE's menu.

Participants could not move in the VE. Only the virtual camera could be rotated either with a mouse on PC or with their head on Android. This camera was placed at their actual height. Everyone had to estimate the egocentric distances to cubes, spheres, or cylinders between 25 cm and 160 cm at 15 cm intervals. Each of these had to be estimated twice in randomized order. Therefore, they had to estimate distances 20 times, one per round. For the last 10 rounds, a scale appeared on the ground. This scale consisted of 17 cubes and the dimension of each was 10 cm × 10 cm × 10 cm. In the PC version, the estimates had to be entered into an input box, while the participants had to verbally estimate in the Android version and a researcher typed the estimates into the dataset at the same time. Then, the participants had to look up at the ceiling and press enter or the touchpad on the Gear VR to advance to the next round. Figure 1 shows a test on the PC version.



(a) A test with a cube. (b) A test with a cylinder.

Figure 1: Two tests with different objects.

If a round was finished, the collected data would be saved into a CSV file. Therefore, all human, technical, and display factors were written in a line inside the previously mentioned file. Even compositional factors were saved, but they were not focused on in this research. Each participant had 20 lines of data as there were 20 round on the test. It should be noted that before the whole procedure started, participants were informed about the process. We told them how to look around the VE and how to estimate in the respective version of the virtual space. The

room's dimensions and scale's dimensions were also told to them. However, they were not informed about the investigated distances as well as the 15 cm intervals. We only mentioned that the distance was never zero. Still, zero was entered 10 times on PC, and one time in the VR version. For participating in the research, the students were motivated with extra points on certain subjects at the university. However, few students estimated distances very quickly, indicating that they only participated in the study because of the extra points. This could explain the outlier values.

Data distributions were assessed with the Shapiro-Wilk test in both versions. Neither the distribution of estimates in the PC version was Gaussian ($W = .88,\ p < .001$), nor in the Android version ($W = .79,\ p < .001$). The distributions of estimation times were also non-Gaussian in the PC ($W = .71,\ p < .001$), and in the Android version ($W = .68,\ p < .001$). Thus, the Wilcoxon rank sum test was used when either the estimates or their times were compared between platforms, while the test's signed rank variant was used for comparing the estimates to the actual distances. An $\alpha = .05$ was chosen for the analyses. This value represents the probability of committing a Type I error, which occurs when the null hypothesis is incorrectly rejected. This α value also establishes a threshold against which p-values are compared to determine statistical significance. In other words, if $p \leq .05$, we can consider the results statistically significant and we can reject the null hypothesis. If $p > .05$, we cannot consider the results statistically significant, meaning there is insufficient evidence to reject the null hypothesis.

3 Results

This section is divided into multiple subsections. Each of them involve results regarding a human factor. Before presenting them, however, the general results are shown in this section. The results regarding estimates are observable in Figure 2, while those that involve estimation times are presented in Figure 3. Box plots were used to illustrate the results. They allow to see the minimum, maximum values, the median, the quartiles and the outliers in a graphical form. Outlier values are shown with dots, while the median is represented by a black line in the boxes.

The same distances on both platforms were compared. The two smallest differences were at 145 cm ($W = 25514,\ p = .026$), and at 160 cm ($W = 25814,\ p = .014$). The differences were strongly significant for the remaining distances as $p < .001$.

When we compared the estimates with the actual distances in the PC version, not all of them were significant; however, most of them was overestimated. Between 40 cm and 160 cm, these were below 10%. Significant overestimates were found at 130 cm ($V = 20885,\ p = .017$), and at 160 cm ($V = 21559,\ p = .014$).

Contrarily, when comparing the estimates with actual distances in the Android version, most underestimates occurred at all distances except at 25 cm. Therefore, the distances were overestimated for the latter. The results of the comparisons were the following: 25 cm ($V = 2610,\ p = .001$), 40 cm ($V = 1426,\ p < .001$), 55 cm

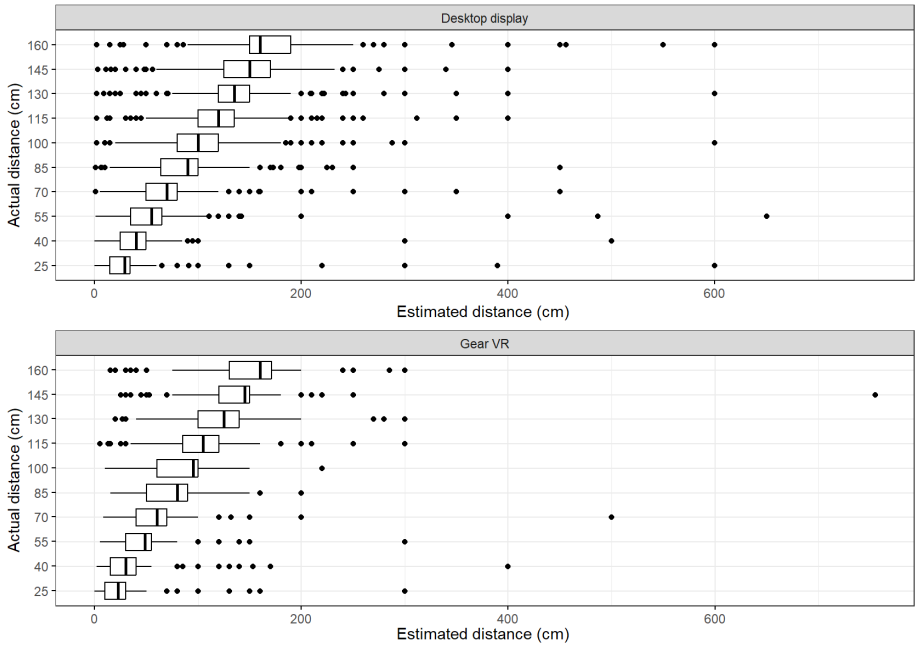


Figure 2: Estimates on both versions at every investigated distance.

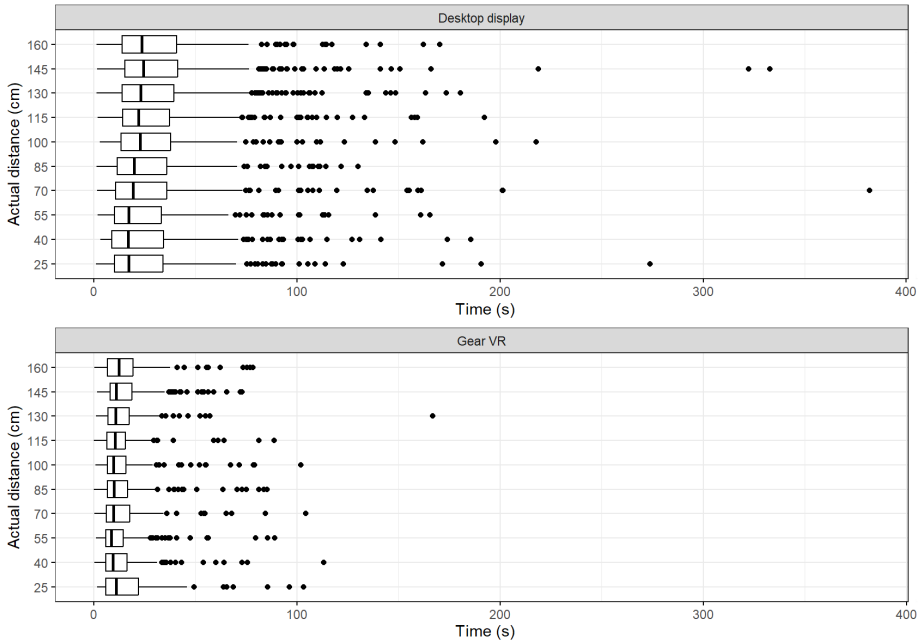


Figure 3: Estimation times on both versions at every investigated distance.

($V = 1718,\ p < .001$), 70 cm ($V = 1301,\ p < .001$), 85 cm ($V = 2224,\ p < .001$), 100 cm ($V = 976,\ p < .001$), 115 cm ($V = 2893,\ p < .001$), and 130 cm ($V = 2334,\ p < .001$). The underestimates at 145 cm and 160 cm were not significant.

As can be seen in Figure 3, estimation times were also compared between platforms. The results show that those who used the Gear VR, were significantly faster in estimating distances ($29260 \leq W \leq 35550,\ p < .001$). Depending on the distances, the distance estimation process of the students became faster by $35.73\% - 57.14\%$.

3.1 Analyses by gender

128 males and 29 females used the PC version, while 49 males and 23 females used the Android version. Their estimates and estimation times can be observed in Figures 4 and 5. When comparing estimates to actual distances in the PC version, the following two significant differences were found in case of males: at 40 cm ($V = 9762,\ p < .032$) and at 70 cm ($V = 9124.5,\ p = .015$). In case of females, only one significant difference was found at 160 cm ($V = 943,\ p = .043$). However, in the Android version, the number of significant differences increased. For males, they were found between 25 cm and 130 cm ($393.5 \leq V \leq 1304,\ p < .015$). Contrarily, in case of females they were observable up to 115 cm ($111.5 \leq V \leq 317.5,\ p < .038$). Overall, males were 31.44% and 34.38% accurate in the PC and Android versions, respectively. The accuracy of females was 34.65% and 40.43%, respectively.

Regarding estimation times, they were significantly different between display devices at all distances in case of males ($17191 \leq W \leq 20103,\ p < .001$). However, in terms of females, these types of differences were only found between 40 cm and 160 cm ($1826 \leq W \leq 2113,\ p < .001$). Thus, the time it took for females to estimate distances at 25 cm was similar between the investigated display devices.

3.2 Analyses by handedness

122 right-handed and 35 left-handed students used the PC version. The Android version was used by 67 right-handed and 5 left-handed students. Their estimates and estimation times can be seen in Figures 6 and 7. Regarding right-handed students in the case of the PC version, significant differences between estimated and actual distances were found at 40 cm ($V = 8597,\ p = .011$), 70 cm ($V = 8292,\ p = .007$), and 160 cm ($V = 13025,\ p = .037$). Four significant differences were found in case of left-handed students in this version. These were at 25 cm ($V = 1559,\ p = .008$), 100 cm ($V = 888,\ p = .006$), 130 cm ($V = 1497,\ p < .001$), and 145 cm ($V = 1501,\ p = .044$). When the results of right-handed students were analyzed in the Android version, significant differences occurred up to 145 cm ($923 \leq V \leq 3289.5,\ p < .044$). Regarding left-handed students, significant differences could be found up to 115 cm ($0 \leq V \leq 7.5,\ p < .044$). Overall, the accuracy of both groups on both platforms was quite similar. 32.21% and 36.34% of the estimates of the right-handed students were accurate on PC and Android, respectively.

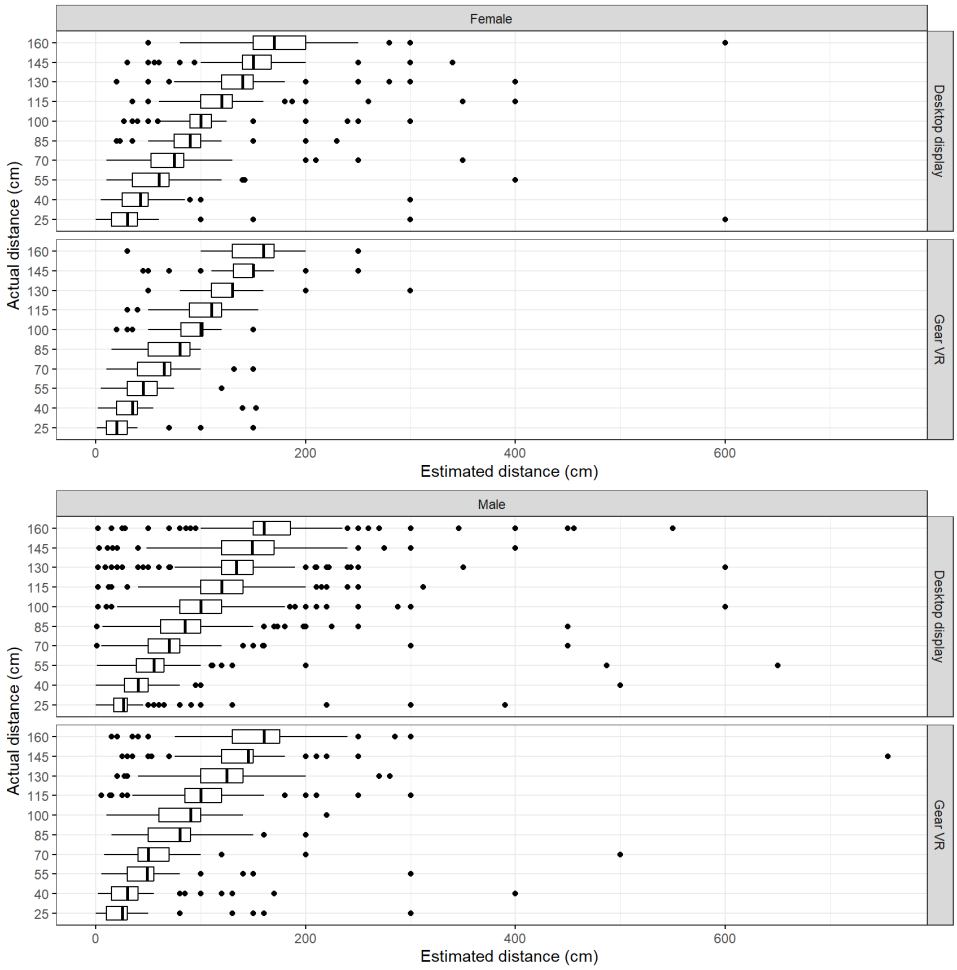


Figure 4: Estimates on both versions at every investigated distance grouped by gender.

31.42% and 36% of the estimates were accurate in the case of left-handed students on the respective two platforms.

When the estimation times were investigated, the following could be observed. The estimation times of right-handed students were significantly different between the two display devices ($21233 \leq W \leq 25401,\ p < .001$). Similarly to females, the estimation times of left-handed students were only significantly different between the display devices from 40 cm to 160 cm ($487 \leq W \leq 634,\ p < .047$).

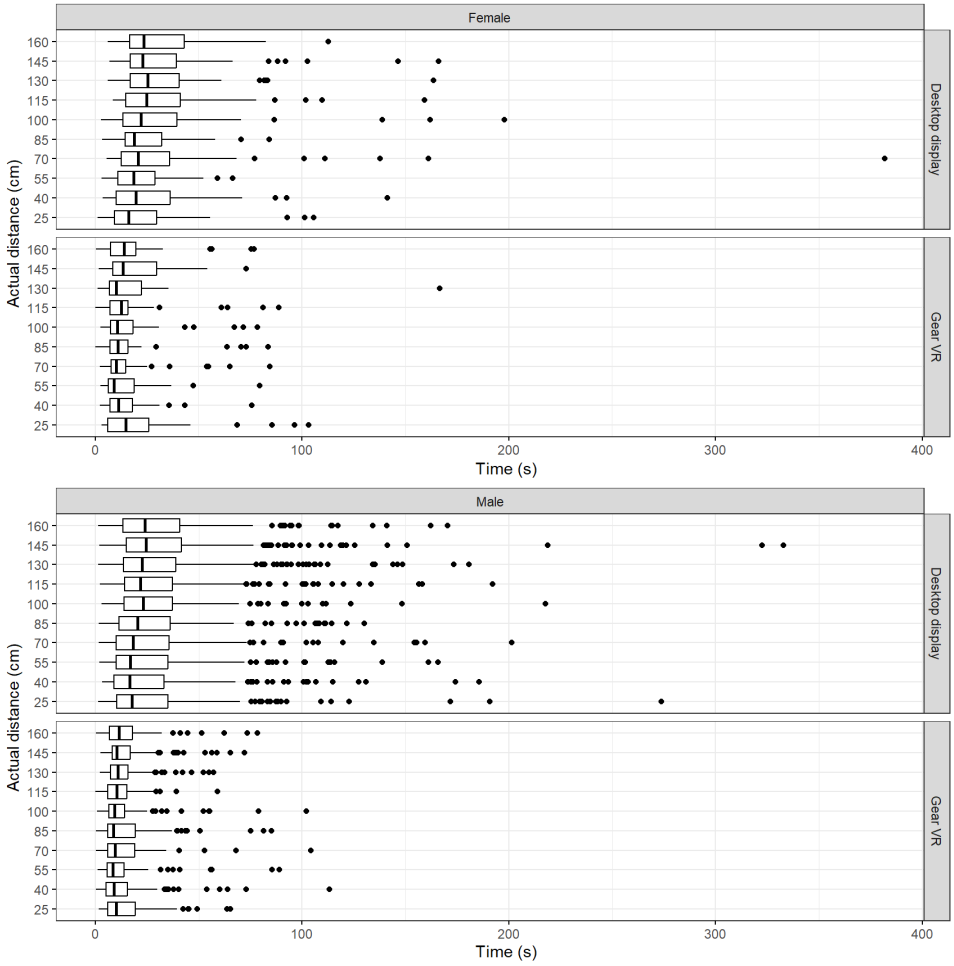


Figure 5: Estimation times on both versions at every investigated distance grouped by gender.

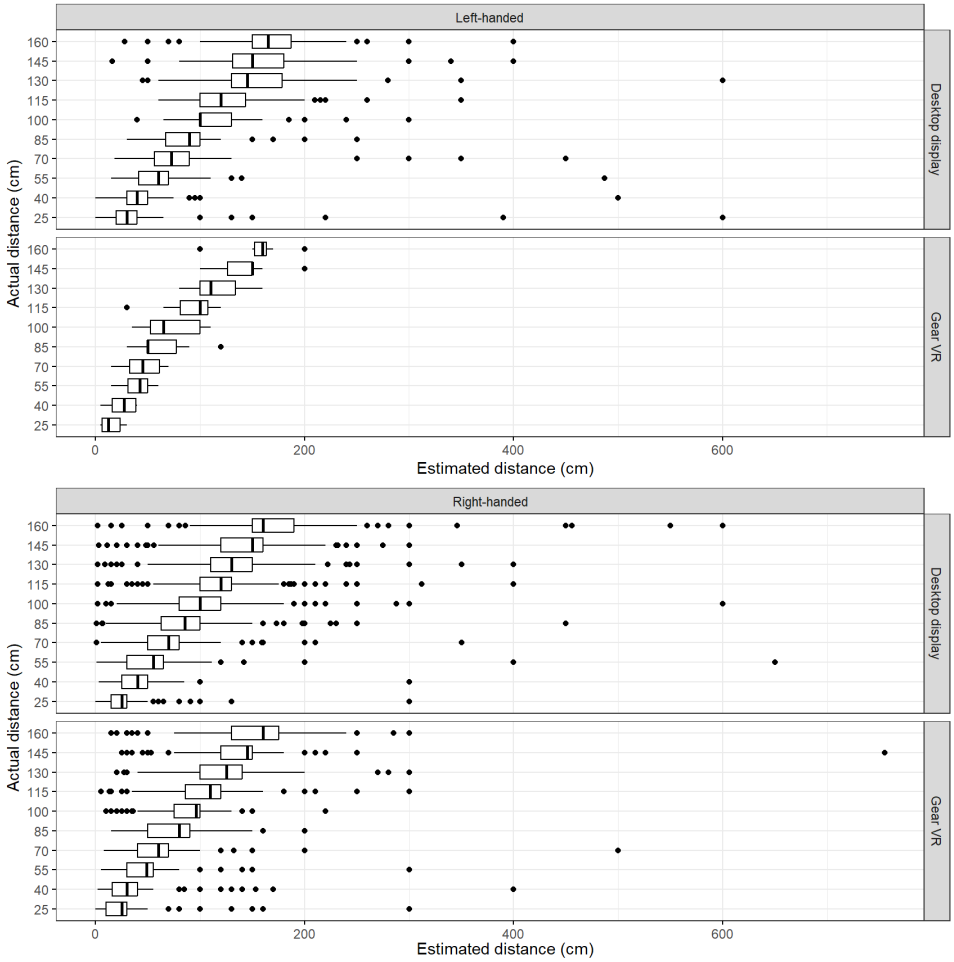


Figure 6: Estimates on both versions at every investigated distance grouped by dominant hand.

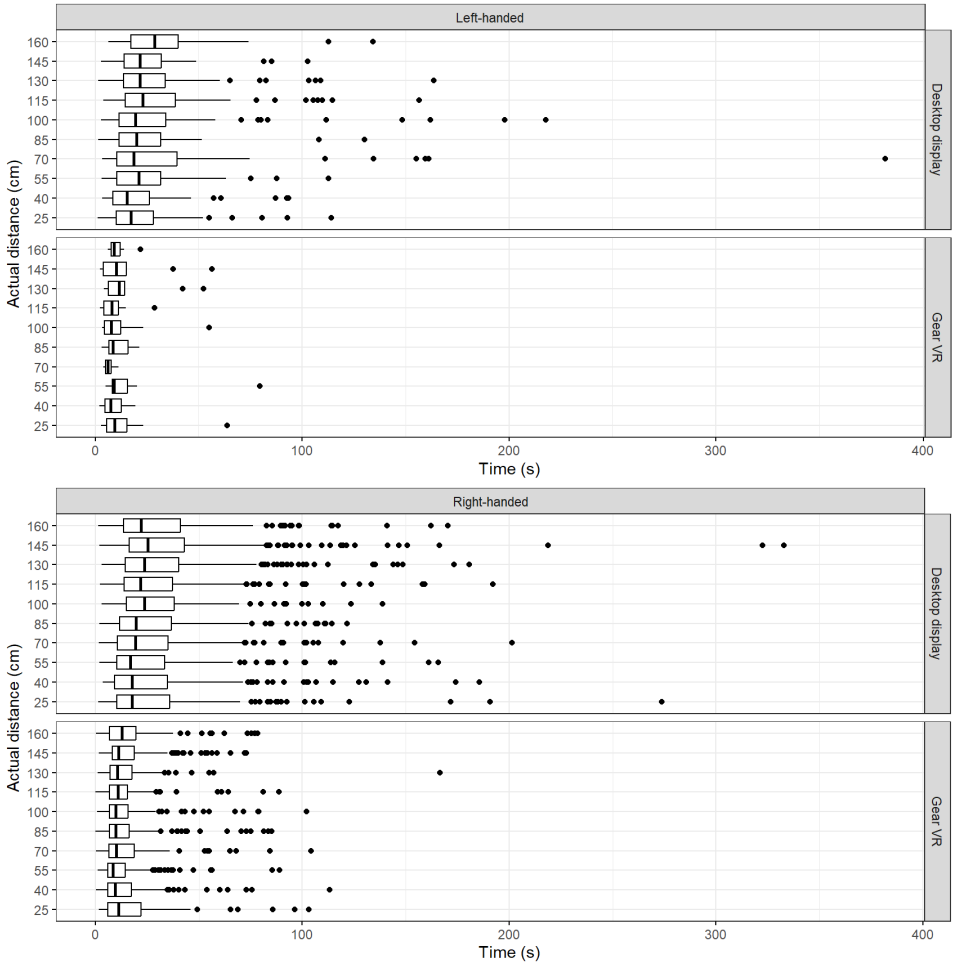


Figure 7: Estimation times on both versions at every investigated distance grouped by dominant hand.

3.3 Analyses by height

The students who took the tests in the PC, had an average height of 178.72 cm, while 175.86 cm in the Android versions. The standard deviations were 9.51 cm and 10.26 cm, respectively. The number of students grouped by display device and height can be seen in Table 1. Their estimates and estimation times can be found in Figures 8 and 9.

Table 1: The number of students in the dataset, grouped by display device and height (cm).

Height (cm)	Desktop display	Gear VR
150–154	2	1
155–159	3	2
160–164	6	7
165–169	16	13
170–174	15	8
175–179	34	13
180–184	30	12
185–189	33	10
190–194	11	3
195–199	5	3
200–204	2	0

The most common height range among students using a desktop display was 175–179 cm (34 students). This was followed closely by students in the 180–184 cm and 185–189 cm ranges. The smallest number of students was in the height range of 200–204 cm. In contrast, students using Gear VR were most frequently found in the 165–169 cm range (13 students). It can also be seen that fewer students were taller than 190 cm, and the number of Gear VR users decreases in the highest height range (200–204 cm, with zero users).

When comparing estimates to actual distances in the PC version, six significant differences were found. The first two were observed in the case of students whose height was between 160 cm and 164 cm. These differences were observed at 85 cm ($V = 67,\ p = .029$), and 160 cm ($V = 52,\ p = .013$). The next three were found when the height of the students was between 175 cm and 179 cm. These were observed at 40 cm ($V = 565,\ p = .024$), 55 cm ($V = 759,\ p = .04$), and 70 cm ($V = 595,\ p = .028$). The last one was observed at 160 cm ($V = 1075.5,\ p = .023$), when the students' height was between 185 cm and 189 cm. 24 significant differences were found in the Android version. One was observed between the heights of 160 cm and 164 cm at 40 cm ($V = 13,\ p = .013$). Six were between the heights of 165 cm and 169 cm among the distances of 25 cm and 100 cm ($8 \leq V \leq 75.5,\ p < .039$). Two were observed between the heights of 170 cm and 174 cm, at the distances of 70 cm ($V = 10.5,\ p = .003$) and 100 cm ($V = 12,\ p = .037$). Seven were between the heights at 175 cm and 179 cm. These were observable between 25 cm and 130 cm ($0 \leq V \leq 73.5,\ p < .016$). Four were observable between the heights of 180 cm and

184 cm, at the distances of 40 cm ($V = 38,\ p = .012$), 55 cm ($V = 46.5,\ p = .016$), 70 cm ($V = 27.5,\ p = .006$), and 100 cm ($V = 8,\ p < .001$). Two were between the heights of 190 cm and 194 cm. These were observable at 115 cm ($V = 0,\ p = .035$), and 145 cm ($V = 0,\ p = .035$). The final one was between the heights of 194 cm and 199 cm at the distance of 130 cm ($V = 0,\ p = .031$).

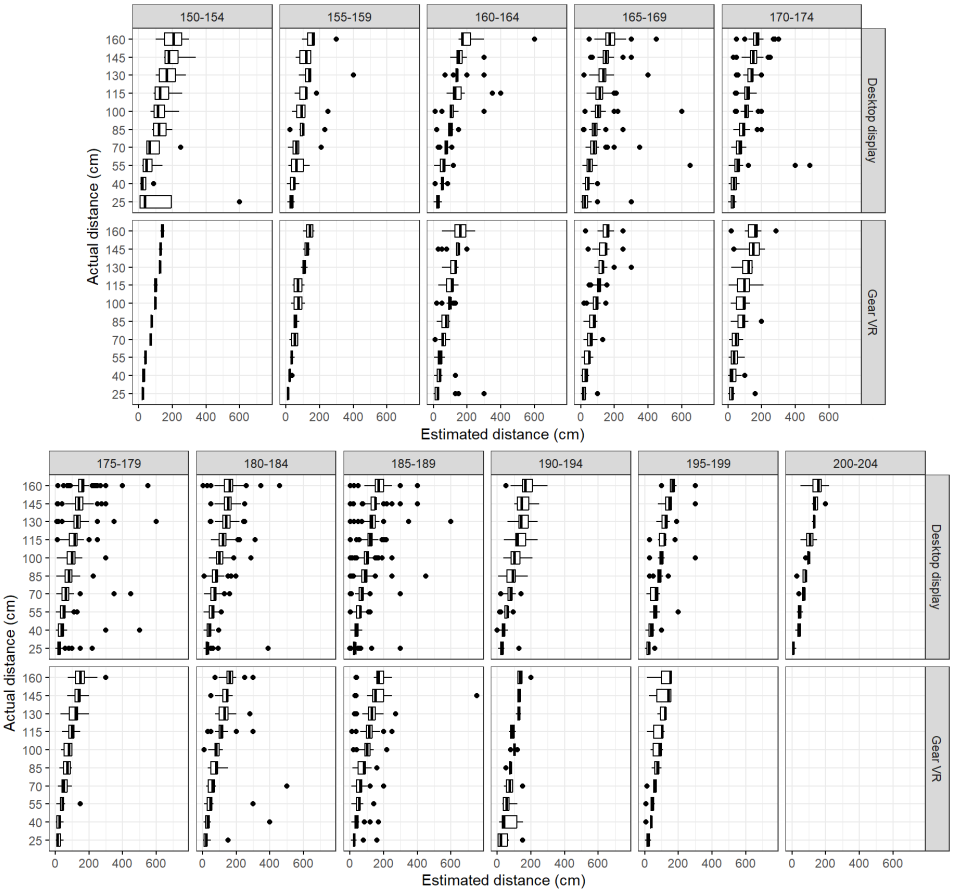


Figure 8: Estimates on both versions at every investigated distance grouped by height.

When the estimation times were examined between the display devices, the following could be observed. There were no significant differences in the height groups of 150–154 cm, 155–159 cm. This was also true for the group of 200–204 cm, although no student was that tall who used the Android version. Starting from the height of 160 cm, significant differences in time appeared from 55 cm. From the height of 170 cm, significant differences started at 25 cm. However, at the height 190 cm or above, the numbers of these differences decreased.

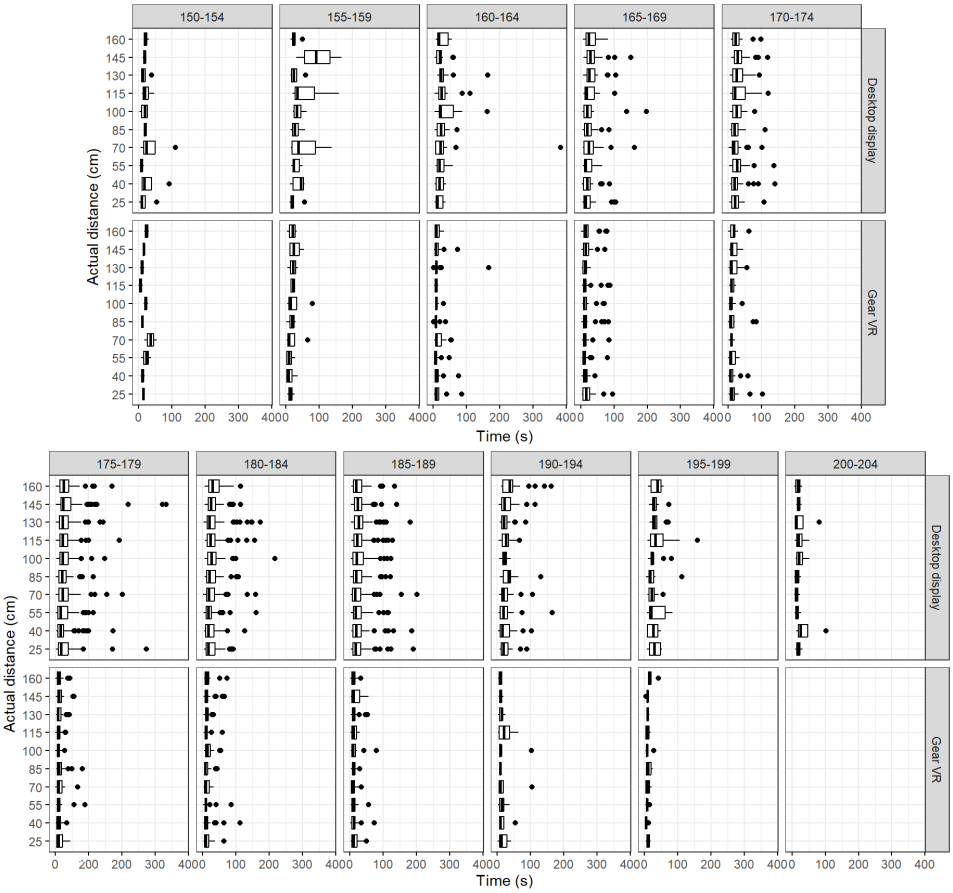


Figure 9: Estimation times on both versions at every investigated distance grouped by height.

3.4 Analyses by whether students wore glasses

In the case of the PC version, 66 students had glasses and 91 did not wear them. These numbers were 32 and 40, respectively, in the Android version. Their estimates and estimation times can be observed in Figures 10 and 11. Regarding students with glasses, three significant differences were found in the PC version when estimates were compared with actual distances. These were found at 130 cm ($V = 3850$, $p = .045$), 145 cm ($V = 4713$, $p = .219$), and 160 cm ($V = 4353$, $p = .022$). One significant difference was observed at 70 cm ($V = 4687$, $p = .037$) in case of students with no glasses. Regarding the Android version, significant differences were found between 40 cm and 115 cm ($194.5 \leq V \leq 620$, $p < .022$) when the results of students with glasses were looked at. In this version, significant differences were found between 25 cm and 130 cm ($295.5 \leq V \leq 849.5$, $p < .001$) in case of students

who wore no glasses. Still, those who had glasses were more accurate as their accuracy was 33.63% in the PC, and 40.46% in the Android version, respectively. These numbers were 30.87% and 33% in case of those without glasses, respectively.

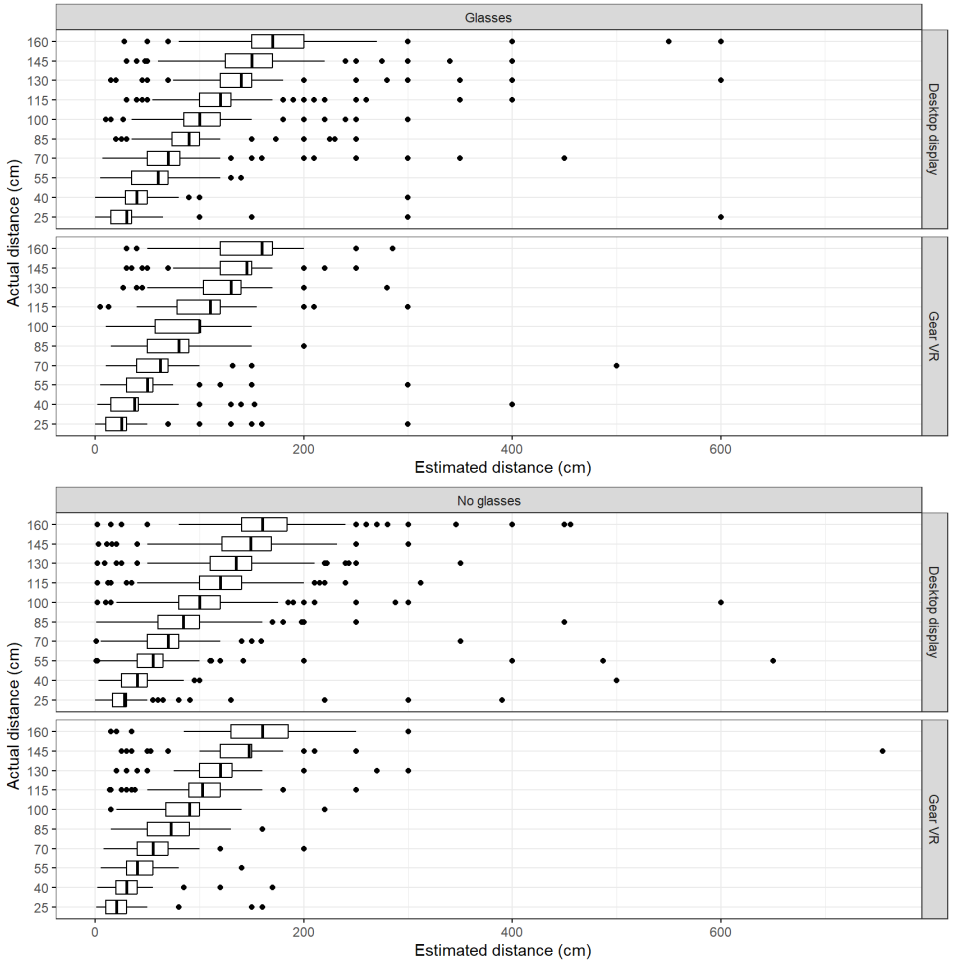


Figure 10: Estimates on both versions at every investigated distance grouped by whether glasses were worn.

Regarding estimation times grouped by whether one wore glasses, the following can be concluded: significant differences in estimation times could be found between the two display devices. These differences were less likely to have occurred by chance in the case of those who had glasses ($5305 \leq W \leq 6506,\ p < .003$), than those who did not wear them ($9582 \leq W \leq 11739,\ p < .001$). Since all were significant, it did not matter whether glasses were worn.

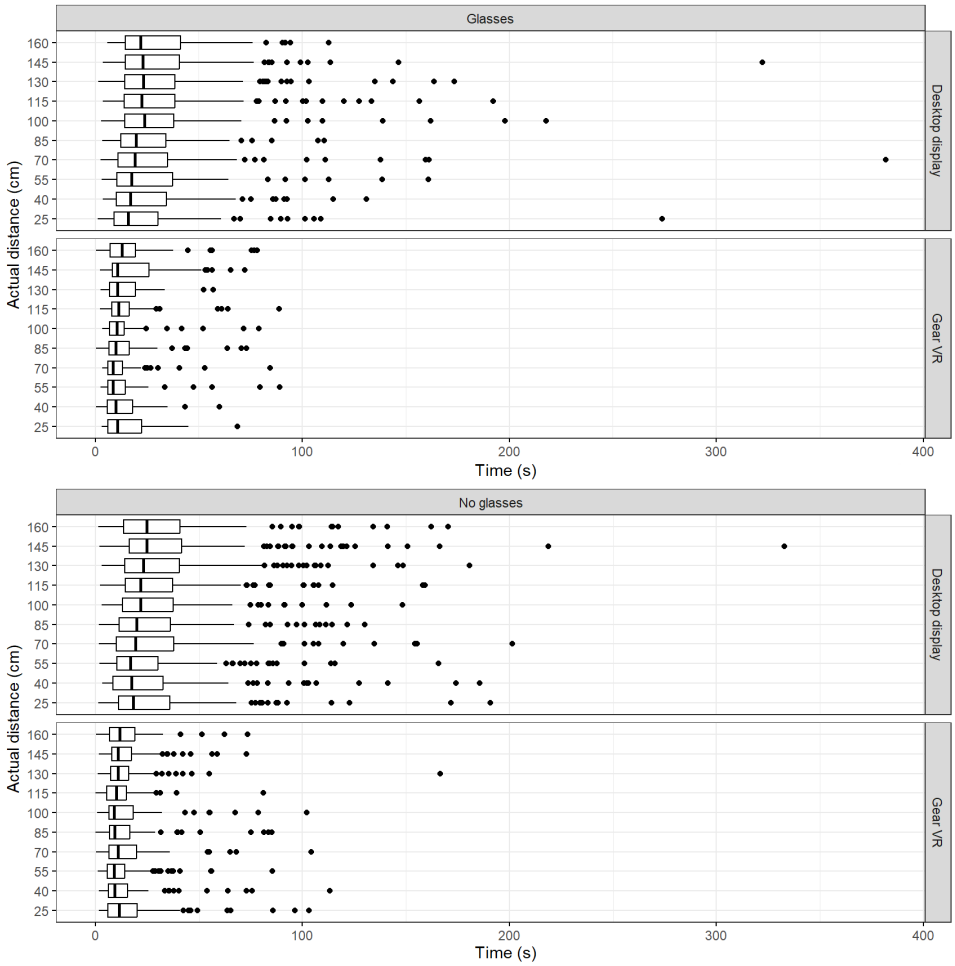


Figure 11: Estimation times on both versions at every investigated distance grouped by whether glasses were worn.

3.5 Analyses by video game playtime a week

According to the students, most of them did not play video games at all. However, many students played video games between 5–10 hours or 11–19 hours a week. The number of students grouped by display device and gaming hours a week can be observed in Table 2. Their estimates and estimation times can be seen in Figures 12 and 13.

Table 2: The number of students grouped by display device and gaming hours a week.

Gaming hours per week	Desktop display	Gear VR
0	42	21
1–2	31	8
3–4	18	8
5–10	32	18
11–19	20	10
20 or more	14	7

As can be seen in Table 2, the majority of students using a desktop display reported either 0 gaming hours (42 students), 1–2 hours (31 students) or 5–10 hours (32 students) per week. Similarly, most Gear VR users also fell into the zero gaming hours category (21 students). In the case of both devices, fewer students engaged in more than 20 hours of gaming per week: only 14 desktop display users and 7 Gear VR users were in this category.

When comparing estimates with actual distances, three significant differences were found between them in the PC version. One was at 25 cm in case of those who do not play video games ($V = 1962.5,\ p = .034$). Another at 70 cm in the case of those who play 1–2 hours a week ($V = 491.5,\ p = .047$), and the last one at 160 cm in the case of those who play 5–10 hours a week ($V = 968,\ p = .011$). Similarly to previous factors, the number of significant differences arose in the Android version. They were found between 40 cm and 85 cm in case of those who play zero hours a week ($71 \leq V \leq 185,\ p < .028$). Significant differences started at 55 cm and ended at 160 cm in case of those who play 1–2 hours a week ($1 \leq V \leq 19.5,\ p < .023$). Three such differences were observed in the case of those who play 3–4 hours a week: at 40 cm ($V = 10,\ p = .043$), 70 cm ($V = 7.5$, $p = .014$), and 100 cm ($V = 12.5,\ p = .04$). Those who play 5–10 hours a week had significant differences between 55 cm and 100 cm ($78.5 \leq V \leq 157.5,\ p < .035$). Two significant differences were observed in case of those tho play 11–19 hours a week: at 55 cm ($V = 26,\ p = .017$), and at 100 cm ($V = 4.5,\ p = .001$). In those who played 20 or more hours, four significant differences were found: at 40 cm ($V = 3,\ p = 051$), 55 cm ($V = 6,\ p = .006$), 70 cm ($V = 9,\ p = .019$), and 100 cm ($V = 2,\ p = .029$). In the PC version, those who did not play video games at all were the most accurate (33.80%). They were followed by students who play 5–10 hours a week (33.12%), 3–4 hours a week (33.05%), 1–2 hours a week (30.64%), 11–19 hours a week (30%), and lastly, 20 or more hours a week (28.92%). In the

Android version, the order of accuracy was the following: those who play 3–4 hours a week (41.25%), zero hours a week (41.11%), 11-19 hours a week (36.5%), 20 or more hours a week (36.42%), 5–10 hours a week (33.33%), and finally, 1–2 hours a week (25%).

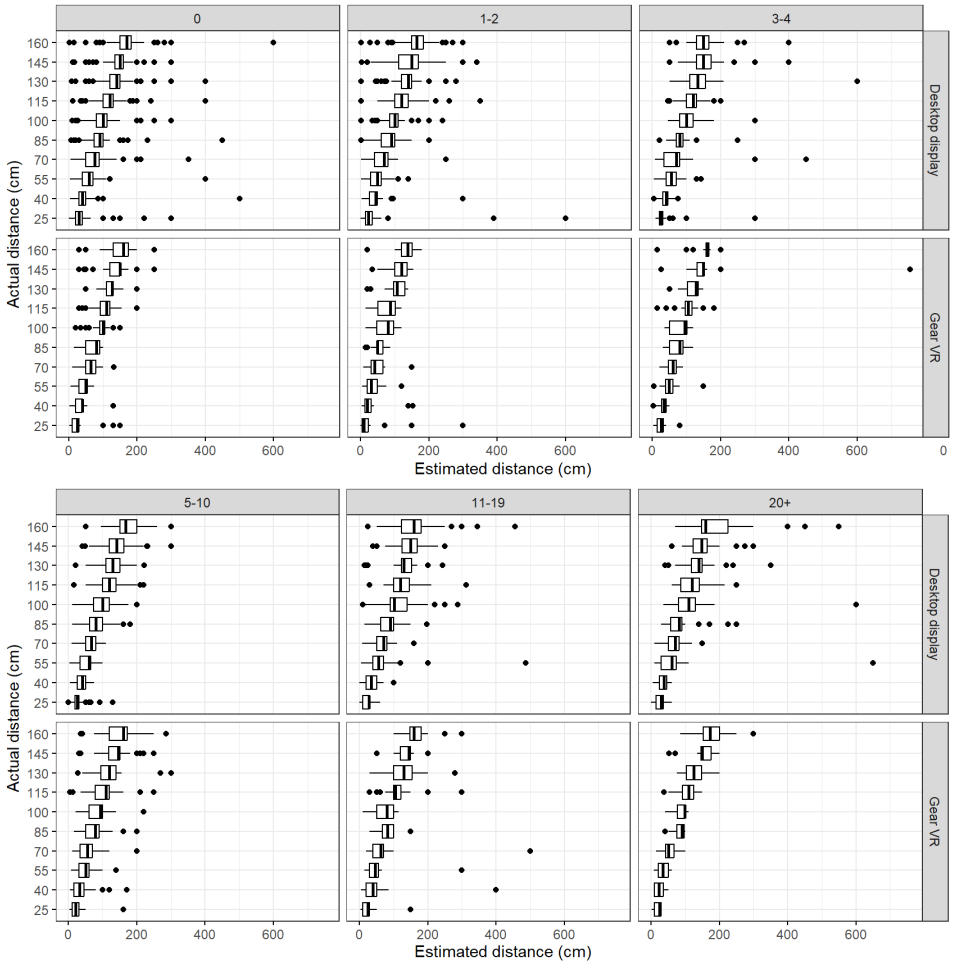


Figure 12: Estimates on both versions at every investigated distance grouped by video game playtime.

Regarding estimation times between the two display devices, the number of non-significant differences increased as the playtime a week increased. The number of such differences was as follows. Zero in the case of those did not play at all, one in the case of those who play 1–2 hours a week (at 25 cm), zero in the case of those who play 3–4 hours a week, one in the case of those who play 5–10 hours a week (at 40 cm), three in the case of those who play 11–19 hours a week (at 25 cm, 40 cm, and 145 cm), and five in the case of those who play 20 hours or more a week (at 25 cm, 40 cm, 55 cm, 70 cm, 85 cm, and 160 cm).

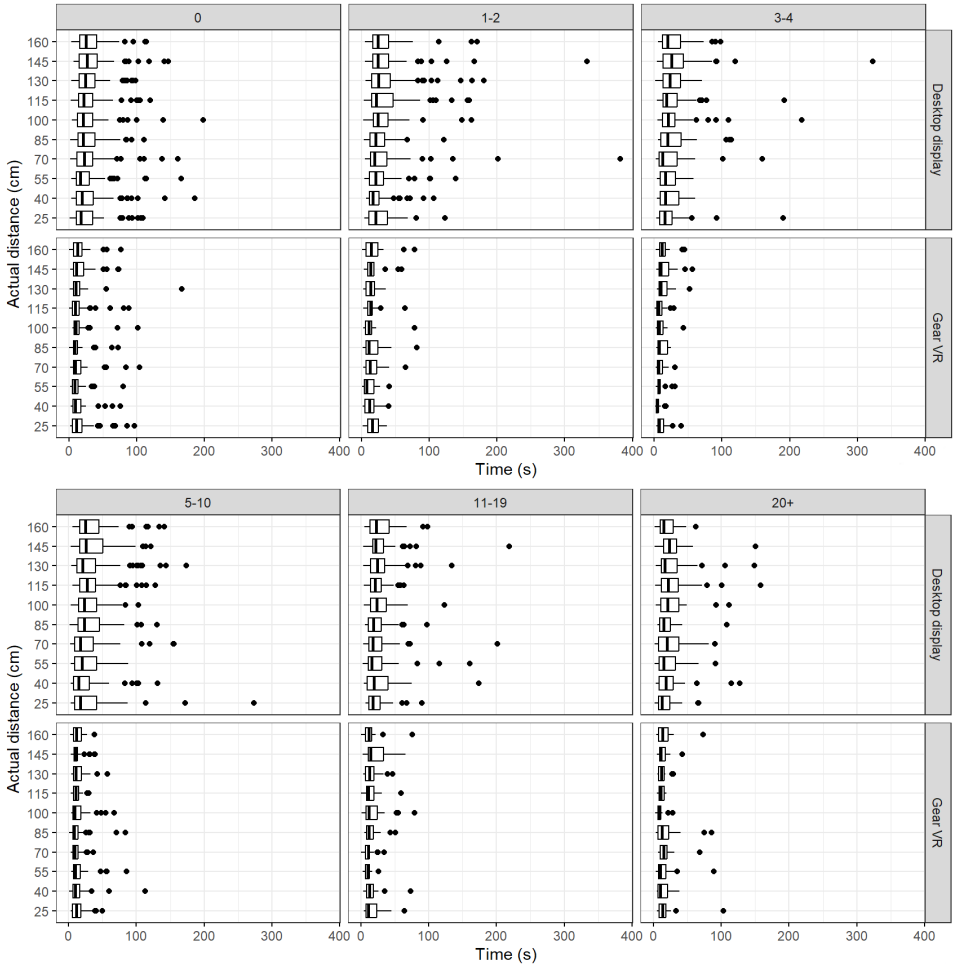


Figure 13: Estimation times on both versions at every investigated distance grouped by video game playtime.

3.6 Analyses by whether the participants had previous VR experience

In the PC version, 57 students indicated that they had previous VR experience, while 100 indicated that they did not. These numbers were 29, and 43 in the Android version, respectively. As before, the estimates were first compared to actual distances. Their estimates and estimation times can be found in Figures 14 and 15. For those who had previous VR experience, only one was found at 145 cm ($V = 3986.5,\ p = .009$) in the PC version. No significant differences were found for those who had no previous VR experience. In the Android version, significant differences could be found from 40 cm to 145 cm in case of people who had previous VR experience ($150.5 \leq V \leq 461.5,\ p < .038$). If those who did not have this kind of experience were investigated, significant differences were observable up to 130 cm ($367 \leq V \leq 1159,\ p < .007$). Regarding the estimates of those people who had previous VR experience, 33.68% of them were accurate in the PC version. This number was quite similar in the Android version: 33.27%. On the contrary, those who did not have this kind of experience were less accurate on PC (31.1%). However, these participants were more accurate in the Android version (38.37%).

When the estimation times were analyzed by whether the participants had previous VR experience, the following could be concluded: Significant differences could be found between the two display devices in both groups at all distances. The significance of the estimation times for those who had previous VR experience was less likely to have occurred by chance ($4444 \leq W \leq 5121,\ p < .001$) than in the case of those who did not have such experience ($10822 \leq W \leq 13602,\ p < .001$). Still, it did not matter whether one had previous VR experience.

3.7 Analyses by what the participants study

From those who used the PC version, 81 were civil engineering, 27 were mechanical engineering, and 49 were vehicle engineering students. All those who used the Android version were IT students (72). As this fact reduced the number of possible comparisons, only the estimates of engineering students were compared to each other. Their estimates can be observed in Figure 16. According to the results, four significant differences were found. One regarding civil engineering students at 40 cm ($V = 3789,\ p = .031$). Two in the case of mechanical engineering students, one at 130 cm ($V = 831.5,\ p = .004$), and another one at 160 cm ($V = 721,\ p = .048$). The last one occurred when the estimates of vehicle engineering students were assessed. It was found at 160 cm ($V = 2236.5,\ p = .043$). Still, possibly due to the Gear VR, IT students were the most accurate with 36.31%. They were followed by vehicle engineering students (34.48%), then civil engineering students (32.40%), and lastly, mechanical engineering students (26.48%).

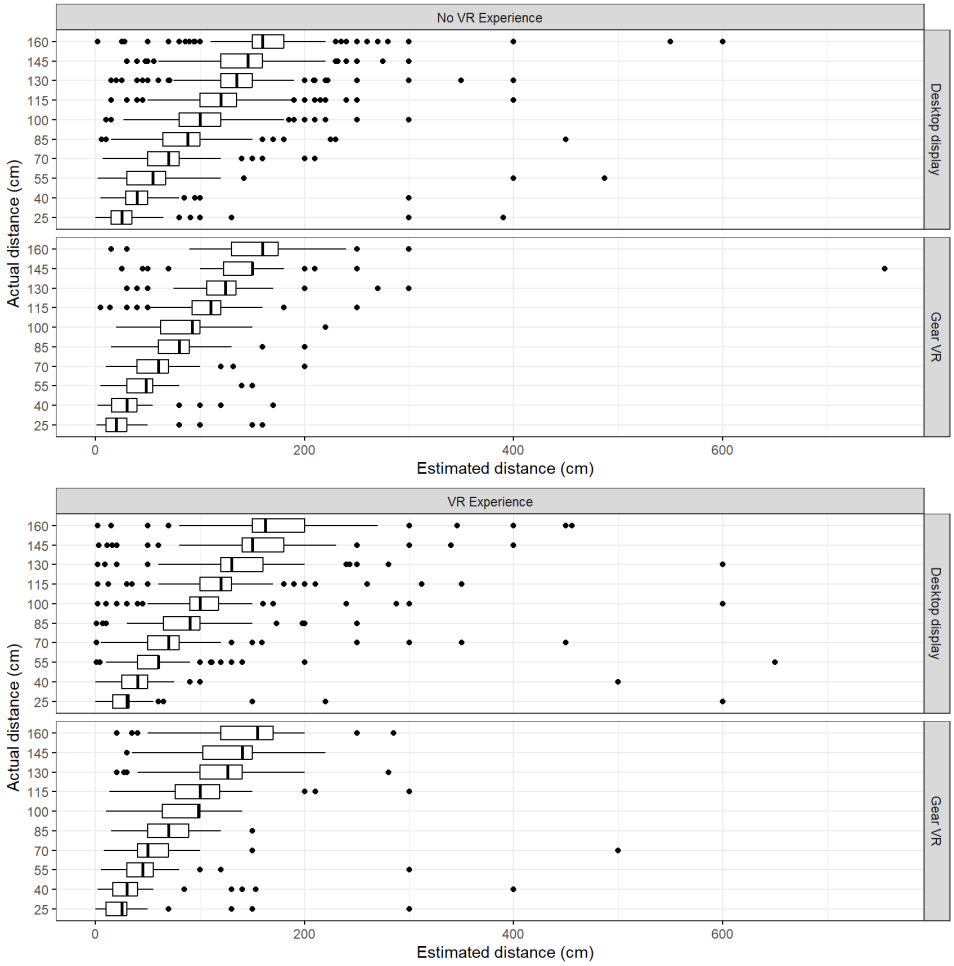


Figure 14: Estimates on both versions at every investigated distance grouped by whether one had previous VR experience.

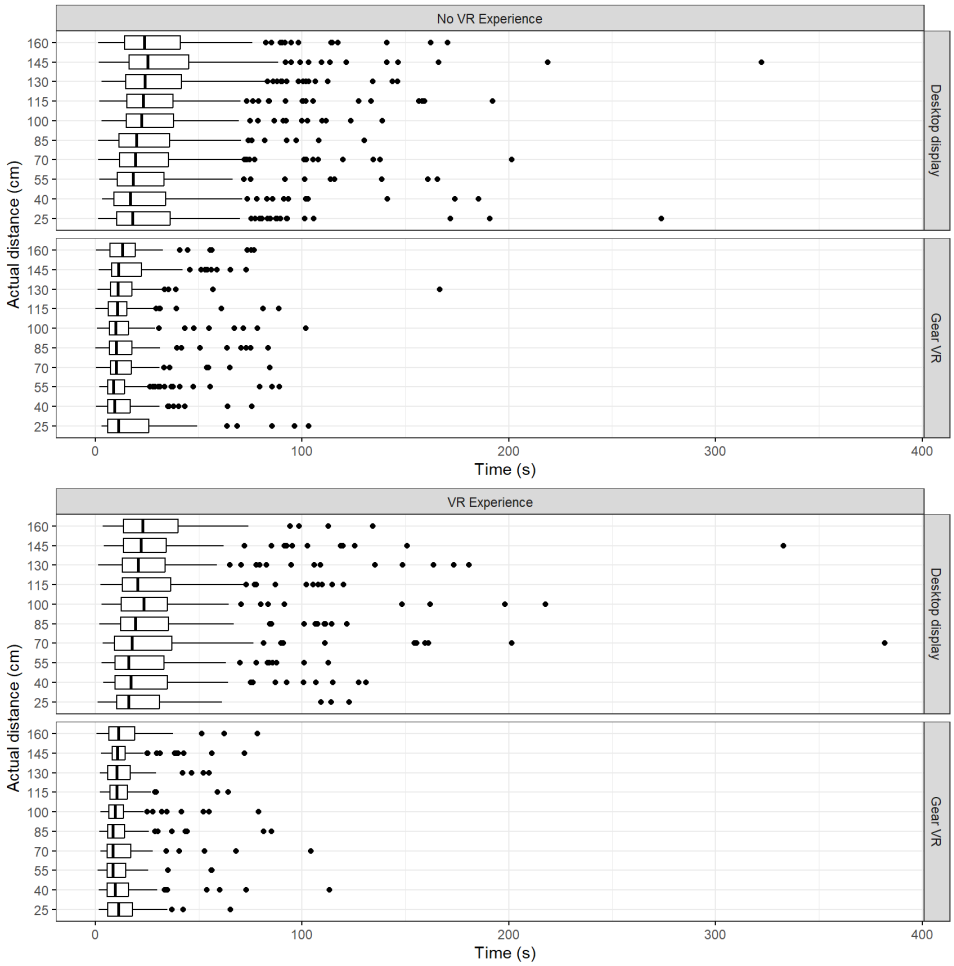


Figure 15: Estimation times on both versions at every investigated distance grouped by whether one had previous VR experience.

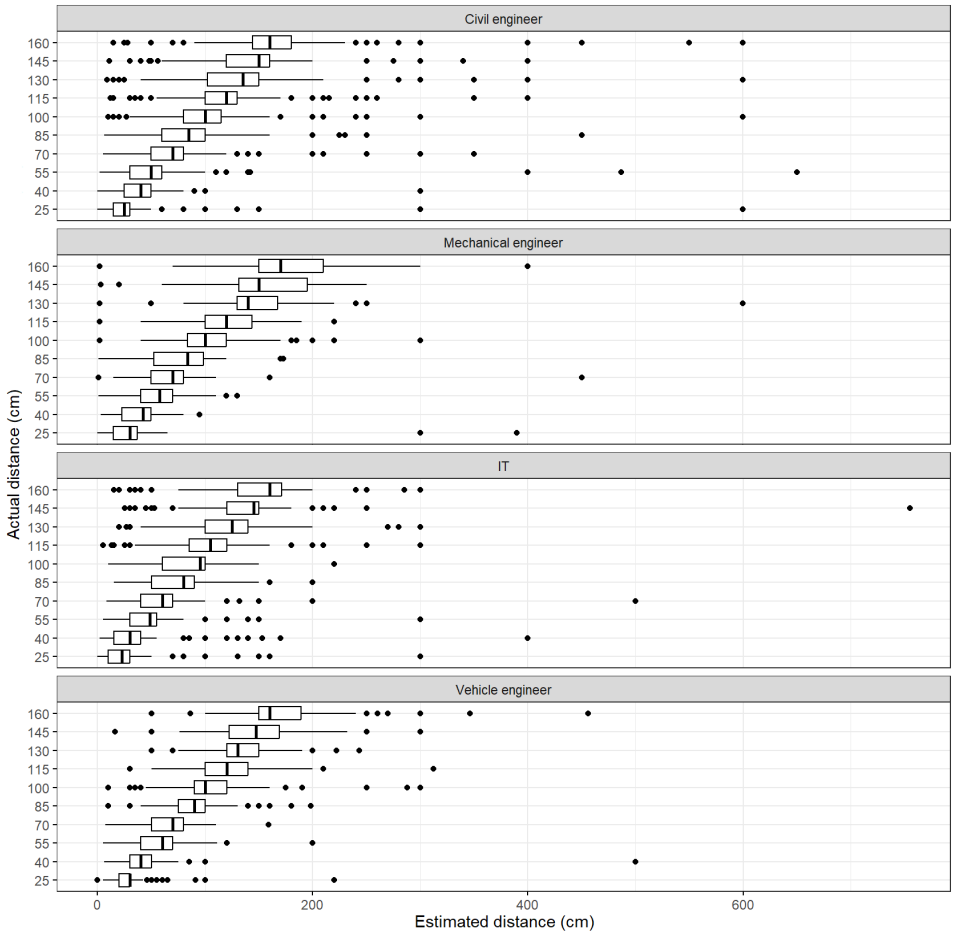


Figure 16: Estimates at every investigated distance grouped by what the participants study.

4 Discussion

The results show that our research question was answered. Naturally, display devices have an effect on distance estimation by themselves [29, 31, 8, 24]. By combining immersion levels with human factors, both the egocentric distance estimation process and estimation time can be influenced. In addition to this fact, each factor had a different effect on them.

Regarding the effects of gender, there is no consensus in the literature. Some say that it does not have a significant effect on distance estimation [7, 16, 28]. However, some studies have mentioned that it actually has an effect [10, 6]. Our results show that females had better accuracy than males in both versions. Gender can also influence estimation times. We also found that the time of females was similar between both versions at 25 cm.

The effect of handedness was smaller, as the difference between the accuracy of both groups was small in both versions. Still, the estimation times of left-handed students were similar between the two versions at 25 cm.

Also, according to multiple studies, distance estimation is not affected by height [6, 27]. However, we found that students in certain groups can estimate distances differently. The estimation times of the students in the height groups of 150–159 cm and 190–199 cm had zero or few significant differences between the display devices. All others were significantly different.

Students with glasses had more significant differences between estimates and actual distances than those who did not wear them in the PC version. The reverse of this statement could be observed in the Android version. Regarding estimation times, whether one wore glasses or not did not had an effect on them.

The number of hours of playing video games a week can also have effects on both estimates and estimation times. No pattern was found with respect to the estimates. However, the more the students played, the more non-significant differences occurred regarding estimation times.

Whether students had previous VR experience provided interesting results. Those who did not have such experience performed the best on the Gear VR. Whether having previous VR experience in the PC version did not influence the estimates. The case was similar with estimation times. This factor did not affect them.

Lastly, it was assessed whether the studies of the participants affected the estimates. When comparing the three types of engineering students, it could be observed that the studies had a small influence. Estimation times were not compared due to having different types of students on each display device.

5 Conclusions

A VE was developed to assess egocentric distance estimation skills of university students. In addition to the components of VEs, display devices and human factors are crucial for the estimation process. Depending on the level of immersion, either

overestimation or underestimation can occur. In general, students were more likely to overestimate distances to objects at 130 cm and 160 cm with a desktop display. With the Gear VR, they underestimated distances to objects that were 40 cm–130 cm away, while overestimation occurred when objects were 25 cm away from them. However, each human factor had different effects on estimates and estimation times. By using the Gear VR, estimation times can also be decreased by 35.73%–57.14% depending on the human factors and distances. The designers of future VEs have to keep these in mind.

References

[1] Backlund, P., Engstrom, H., Hammar, C., Johannesson, M., and Lebram, M. Sidh – a game based firefighter training simulation. In *2007 11th International Conference Information Visualization (IV'07)*, pages 899–907. IEEE, 2007. DOI: 10.1109/IV.2007.100.

[2] Baranyi, P., Csapo, A., and Sallai, G. *Cognitive infocommunications (CogInfoCom)*. Springer, 2015. DOI: 10.1007/978-3-319-19608-4.

[3] Bian, Z. and Andersen, G. J. Aging and the perception of egocentric distance. *Psychology and aging*, 28(3):813, 2013. DOI: 10.1037/a0030991.

[4] Bingham, G. P. Calibration of distance and size does not calibrate shape information: Comparison of dynamic monocular and static and dynamic binocular vision. *Ecological Psychology*, 17(2):55–74, 2005. DOI: 10.1207/s15326969eco1702_1.

[5] Burdea, G. C. and Coiffet, P. *Virtual reality technology*. John Wiley & Sons, 2003.

[6] Coelho, H., Melo, M., Branco, F., Vasconcelos-Raposo, J., and Bessa, M. The impact of gender, avatar and height in distance perception in virtual environments. In *New Knowledge in Information Systems and Technologies: Volume 2*, pages 696–705. Springer, 2019. DOI: 10.1007/978-3-030-16184-2_66.

[7] Creem-Regehr, S. H., Willemsen, P., Gooch, A. A., and Thompson, W. B. The influence of restricted viewing conditions on egocentric distance perception: Implications for real and virtual indoor environments. *Perception*, 34(2):191–204, 2005. DOI: 10.1068/p5144.

[8] Cutting, J. E. and Vishton, P. M. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In *Perception of Space and Motion*, pages 69–117. Elsevier, 1995. DOI: 10.1016/B978-012240530-3/50005-5.

[9] Drettakis, G., Roussou, M., Reche, A., and Tsingos, N. Design and evaluation of a real-world virtual environment for architecture and urban planning. *Presence: Teleoperators and Virtual Environments*, 16(3):318–332, 2007. DOI: 10.1162/pres.16.3.318.

[10] Foreman, N., Sandamas, G., and Newson, D. Distance underestimation in virtual space is sensitive to gender but not activity-passivity or mode of interaction. *CyberPsychology & Behavior*, 7(4):451–457, 2004. DOI: 10.1089/cpb.2004.7.451.

[11] Guzsvinecz, T., Orbán-Mihálykó, É., Sik-Lányi, C., and Perge, E. The effects of display parameters and devices on spatial ability test times. *Applied Sciences*, 12(3):1312, 2022. DOI: 10.3390/app12031312.

[12] Guzsvinecz, T., Perge, E., and Szűcs, J. Analyzing Accurate Egocentric Distance Estimates of University Students in Virtual Environments with a Desktop Display and Gear VR Display. *Electronics*, 12(10):2253, 2023. DOI: 10.3390/electronics12102253.

[13] Guzsvinecz, T., Perge, E., and Szűcs, J. Examining the Results of Virtual Reality-Based Egocentric Distance Estimation Tests Based on Immersion Level. *Sensors*, 23(6):3138, 2023. DOI: 10.3390/s23063138.

[14] Guzsvinecz, T., Sik-Lanyi, C., Orban-Mihalyko, E., and Perge, E. The influence of display parameters and display devices over spatial ability test answers in virtual reality environments. *Applied Sciences*, 10(2):526, 2020. DOI: 10.3390/app10020526.

[15] Horvath, I. Innovative engineering education in the cooperative VR environment. In *2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 000359–000364. IEEE, 2016. DOI: 10.1109/coginfocom.2016.7804576.

[16] Interrante, V., Ries, B., and Anderson, L. Distance perception in immersive virtual environments, revisited. In *IEEE Virtual Reality Conference (VR 2006)*, pages 3–10. IEEE, 2006. DOI: 10.1109/VR.2006.52.

[17] Katona, J. A review of human–computer interaction and virtual reality research fields in Cognitive InfoCommunications. *Applied Sciences*, 11(6):2646, 2021. DOI: 10.3390/app11062646.

[18] Kenyon, R. V., Phenany, M., Sandin, D., and Defanti, T. Accommodation and size-constancy of virtual objects. *Annals of Biomedical Engineering*, 36:342–348, 2008. DOI: 10.1007/s10439-007-9414-7.

[19] Korečko, Š., Hudák, M., Sobota, B., Marko, M., Cimrová, B., Farkaš, I., and Rosipal, R. Assessment and training of visuospatial cognitive functions in virtual reality: proposal and perspective. In *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 39–44. IEEE, 2018. DOI: 10.1109/CogInfoCom.2018.8639958.

[20] Kortum, P. *HCI Beyond the GUI: Design for Haptic, Speech, Olfactory, and Other Nontraditional Interfaces*. Elsevier, 2008.

[21] Kovari, A. CogInfoCom supported education: A review of CogInfoCom based conference papers. In *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 000233–000236. IEEE, 2018. DOI: 10.1109/COGINFOCOM.2018.8639879.

[22] Kövecses-Gősi, V. Cooperative learning in VR environment. *Acta Polytechnica Hungarica*, 15(3):205–224, 2018. DOI: 10.12700/aph.15.3.2018.3.12.

[23] Leyrer, M., Linkenauger, S. A., Bülthoff, H. H., Kloos, U., and Mohler, B. The influence of eye height and avatars on egocentric distance estimates in immersive virtual environments. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, pages 67–74, 2011. DOI: 10.1145/2077451.2077464.

[24] Luo, X., Kenyon, R. V., Kamper, D. G., DeFanti, T. A., et al. On the determinants of size-constancy in a virtual environment. *International Journal of Virtual Reality*, 8(1):43–51, 2009. DOI: 10.20870/ijvr.2009.8.1.2712.

[25] Mazyn, L. I., Lenoir, M., Montagne, G., Delaey, C., and Savelsbergh, G. J. Stereo vision enhances the learning of a catching skill. *Experimental Brain Research*, 179(4):723–726, 2007. DOI: 10.1007/s00221-007-0957-5.

[26] Miller, C. L. and Bertoline, G. R. Spatial visualization research and theories: Their importance in the development of an engineering and technical design graphics curriculum model. *Engineering Design Graphics Journal*, 55(3):5–14, 1991.

[27] Murgia, A. and Sharkey, P. M. Estimation of distances in virtual environments using size constancy. *International Journal of Virtual Reality*, 8(1):67–74, 2009. DOI: 10.20870/ijvr.2009.8.1.2714.

[28] Naceri, A. and Chellali, R. The effect of isolated disparity on depth perception in real and virtual environments. In *2012 IEEE Virtual Reality Workshops (VRW)*, pages 107–108. IEEE, 2012. DOI: 10.1109/VR.2012.6180905.

[29] Naceri, A., Chellali, R., and Hoinville, T. Depth perception within peripersonal space using head-mounted display. *Presence: Teleoperators and Virtual Environments*, 20(3):254–272, 2011. DOI: 10.1162/PRES_a_00048.

[30] Plumert, J. M., Kearney, J. K., Cremer, J. F., and Recker, K. Distance perception in real and virtual environments. *ACM Transactions on Applied Perception (TAP)*, 2(3):216–233, 2005. DOI: 10.1145/1077399.1077402.

[31] Renner, R. S., Velichkovsky, B. M., and Helmert, J. R. The perception of egocentric distances in virtual environments-a review. *ACM Computing Surveys (CSUR)*, 46(2):1–40, 2013. DOI: 10.1145/2543581.2543590.

[32] Schroeder, R., Heldal, I., and Tromp, J. The usability of collaborative virtual environments and methods for the analysis of interaction. *Presence: Teleoperators and Virtual Environments*, 15(6):655–667, 2006. DOI: 10.1162/pres.15.6.655.

[33] Sutcliffe, A. G., Poullis, C., Gregoriades, A., Katsouri, I., Tzanavari, A., and Herakleous, K. Reflecting on the design process for virtual reality applications. *International Journal of Human–Computer Interaction*, 35(2):168–179, 2019. DOI: 10.1080/10447318.2018.1443898.

[34] Viguier, A., Clement, G., and Trotter, Y. Distance perception within near visual space. *Perception*, 30(1):115–124, 2001. DOI: 10.1068/p3119.

[35] Willemsen, P. and Gooch, A. A. Perceived egocentric distances in real, image-based, and traditional virtual environments. In *Proceedings IEEE Virtual Reality 2002*, pages 275–276. IEEE, 2002. DOI: 10.1109/VR.2002.996536.

Acta Cybernetica **27** (2025) 43–51.

Effective Supervision of Students' Activity During Classroom Learning and Testing

Szabolcs Szilágyi[ab]

Abstract

Due to the lockdowns caused by the COVID-19 pandemic, the majority of educational institutions worldwide have been forced to switch to online education, which has created a significant challenge for teachers and students alike. In order to communicate effectively in the online space, educational institutions had a wide range of tools to choose from (e.g. Adobe Connect, Cisco Webex, Google Meet, Microsoft Teams, Skype, Zoom, etc.). The challenge for teachers was to learn how to use them, to teach practical subjects effectively and to provide a supervised examination environment. The return to face-to-face (in-class) teaching after the end of the COVID-19 pandemic has allowed the online collaborative environments listed above to fade into the background, but the supervision of interactive, computer-based practical lessons (e.g. teaching programming languages, network programming etc.) and proctored examinations can still be a challenge for teachers. This article reviews some screen monitoring systems developed for both corporate and educational environments. We present one of them in more detail, namely Veyon, which is available free of charge[1] and can be used on different operating systems, and whose applicability in both teaching and examination has been tested for almost a year at the Faculty of Informatics of the University of Debrecen.

Keywords: Classroom Management Software (CMS), digital learning, remote control, safe examination, screen monitoring, Veyon

1 Introduction

In the wake of COVID-19 pandemic, the world has seen a drastic transformation in how students learn and how teachers teach. As social distancing measures have forced students and educators alike to stay at home, digital learning has become the new norm. With this shift to online learning, classroom management software

[a]Faculty of Informatics, University of Debrecen, Hungary E-mail: szilagyi.szabolcs@inf.unideb.hu, ORCID: 0000-0003-3562-5062

[b]Faculty of Economics and Social Sciences, Partium Christian University, Romania, E-mail: szilagyi.szabolcs@partium.ro

[1]The basic version of Veyon is free, but you have to pay for the various desired add-on licenses.

DOI: 10.14232/actacyb.307809

(CMS) has become an essential tool for teachers to increase the efficiency of in-class learning and teaching. This paper discusses how, returning to the in-class teaching, CMS can improve the digital learning experience, keep students engaged, ensure interactive communication, track completion of practical tasks, and increase the safety and security of exams, sharing our experiences at the Faculty of Informatics of the University of Debrecen.

One of the key benefits of CMS is its ability to keep students interested in the learning experience. Traditional online classroom sessions can become monotonous and unengaging, leading students to become bored and disinterested. However, CMS allows teachers to incorporate interactive elements into their lessons, such as quizzes, games, and polls, to keep students engaged and motivated. Moreover, CMS can also ensure interactive communication between teachers and students. With chat boxes, they can communicate in real-time, providing opportunities for students to ask questions and teachers to provide feedback. This interactive communication not only fosters a supportive learning environment but also ensures that students are engaged in their learning and receive personalized attention from their teachers.

Another critical feature of CMS is its ability to track the completion of practical tasks, such as programming assignments. Tracking student progress in real-time can help teachers to identify areas where students may be struggling, allowing them to provide personalized support when necessary. Furthermore, it allows teachers to monitor students' progress, which can help them to adjust their instructional approach to cater to the needs of individual students more efficiently. CMS also enables restrictions on the use of permitted software and websites.

This feature is particularly useful in ensuring that students remain focused on the educational material during class time, reducing the likelihood of them becoming distracted by social media or other non-educational websites. CMS can also provide a reliable test and exam environment. With remote control and screen monitoring functionalities, teachers can ensure students' compliance with exam rules and minimize the risk of academic fraud. Additionally, these features provide a secure environment for students to take exams, ensuring fair and equal opportunities for all students.

2 Increasing the efficiency of in-class education using management software tools

Based on our previous experience in online education, we have tried to find online tools that can effectively address the problems and challenges listed in the Section 1. In the first instance, we looked online for solutions that could make in-class teaching more effective by providing the features enumerated in Table 1.

Seventeen relevant software products were found that could meet all or part of the needs we listed. Some of these are commonly used in corporate environments to monitor employees' online activities (Employee Monitoring Software – EMS), others are specifically developed for in-class educational supervision (CMS). Table 2 summarises the main characteristics of the 17 solutions we examined, such as which

Table 1: CMS key features and benefits [11]

Screen monitoring	When additional support or guided learning is needed, teachers can quickly view every student monitor in real time — or switch to view individual screens.
Remote control	Make the most of class time by remotely logging in to one or more student devices to install apps or updates while students continue to work.
Broadcast teacher screen	Increase engagement in the classroom by broadcasting one screen to student monitors. Full-screen view locks student devices, and windowed view lets them work along with the teacher.
Push website	Save precious class time by instantly launching the same website on every classroom device to ensure students are always on task, focused and ready to learn.
Launch app	Get even more from class time by troubleshooting navigation issues and launching the same application on selected student devices — or for the entire class.
Blank screen	With the touch of a button, educators can shift classroom attention to them by blanking student screens and locking their devices.
Snapshot	Quickly take screenshots to showcase exceptional student work in the classroom, which may even be used as evidence later.
Send/receive files	Send documents, reading assignments, templates and other materials to student devices. Digitally collect assignments when they're complete.
Messaging	Simplify classroom communication, increase engagement and redirect focus with messaging features. Receive students' questions and chat one-on-one or with the entire class.
Remote power on/off	Remotely power on or off student devices to redirect focus or to perform app updates and other maintenance.

operating systems the environment can be installed on, whether it is available on a web interface, and whether the service is free or paid. All in all, we chose Veyon because it had all the features we needed, could be installed on both Windows and Linux operating systems, and last but not least, was free of charge (see e.g. [23, 7, 22, 2, 13]). At the same time we searched the internet for the most recommended

CMS system for 2023. A possible ranking is available on the website below: https://www.g2.com/categories/classroom-management#grid.

Surprisingly, the Veyon we have chosen is not in the ranking list, as if it were a completely new and unknown solution. As it has performed well in our university during the last year of live testing, we decided to present its use to the teaching community in this paper.

2.1 Installing Veyon

Installing Veyon is very simple. You need to use the installer downloaded from the official website (see [23]) for both the instructor and student installations, with the following settings:

The three components (*Veyon Service*, *Veyon Master*, *Interception driver*) must be installed on the teachers' computer. After installation, the necessary system settings can be made in the *Veyon Configurator*. Under the *General* menu, the *Key file authentication* option must be selected as the authentication method, and then under *Authentication keys*, the key pair for authentication must be created. The private key will be used on the instructor machine, whereas a public key needs to be exported to the student computers. Then, under *Loction & computers*, we can create the rooms and specify the names and IP addresses of the machines per room. If we want to save the current settings or load a previous backup, we have the option to do so in the Veyon Configurator's *File* menu (*Load settings from file*, *Save settings to file*). Veyon stores the settings in JSON format files.

On the students' computers, the *Veyon Master* system component does not need to be installed (only the *Veyon Service* and *Interception diver* components should be installed). Under the *General* menu, the *Key file authentication* method must be selected as authentication method, as for the teacher's computer. Then, under the *Authentication keys* menu, we need to import the public authentication key exported from the teacher's computer in the previous step (this can be done either using a pendrive, a network shared directory or an FTP server). If we need the IP address of the local machine (e.g. to register it to the teacher's machine), we can do this by issuing the *ipconfig* command in Command Prompt in Windows operating system, or in a Linux environment using the *ifconfig* or *ip addr show dev [interface name]* commands.

2.2 Veyon in action

Veyon is the only open-source CMS available on the market at the moment. Previous versions were called iTalc but it has been completely rebranded a few years ago. They had given up development on iTalc and then for some reason now they have rebranded and started issuing new releases [12].

The current version of Veyon includes the following features, which are typically accessed through the main menu: Monitoring, Demo, Lock, Remote view, Remote control, Power on, Reboot, Power down, Log in, Log off, Text message, Start application, Open website, File transfer, Screenshot.

Table 2: EMS and CMS comparison

Software list	Type	Supported operating system						Availability		
										
AB Tutor [1]	CMS	✓	✗	✓	✗	✗	✗	✗	✓	Non-free
Dyknow [3]	CMS	✓	✗	✓	✓	✗	✗	✗	✓	Non-free
Employee Desktop Live Viewer [4]	EMS	✓	✗	✗	✗	✗	✗	✗	✓	Non-free
Epoptes [5]	CMS	✗	✓	✗	✗	✗	✗	✗	✗	**Free**
Faronics Insight [6]	CMS	✓	✗	✓	✓	✓	✓	✗	✓	Non-free
Impero Classroom Management Software [8]	CMS	✗	✗	✗	✗	✗	✗	✓	✓	Non-free
LanSchool Classic [10]	CMS	✓	✗	✗	✗	✗	✗	✗	✓	Non-free
LanSchool Air [9]	CMS	✗	✗	✗	✗	✗	✗	✓	✓	Non-free
Mythware Classroom Management Software [14]	CMS	✓	✓	✓	✓	✓	✓	✗	✓	Non-free
Net Monitor for Employees Professional [15]	EMS	✓	✓	✓	✗	✓	✓	✗	✓	Non-free
Netop Vision [16]	CMS	✓	✗	✓	✓	✗	✗	✗	✓	Non-free
NetSupport School [17]	CMS	✓	✓	✓	✗	✓	✓	✗	✓	Non-free
Quasar [18]	EMS	✓	✗	✗	✗	✗	✗	✗	✗	**Free**
SchoolVue Classroom Management [19]	CMS	✓	✓	✓	✓	✓	✓	✗	✓	Non-free
ScreenWatch [20]	CMS	✓	✓	✓	✓	✓	✗	✗	✓	Non-free
SurveilStar Employee Software [21]	EMS	✓	✗	✗	✗	✓	✗	✗	✓	Non-free
Veyon [23]	CMS	✓	✓	✗	✗	✗	✗	✗	✗	**Free**

The use of Veyon is explained in the administrator and user manuals available in several languages on the official website, as well as in several youtube videos to help the user community.

3 Experiences with Veyon

We started using Veyon in education at the Faculty of Informatics of the University of Debrecen in autumn 2022. As the academic year is slowly approaching, we would like to share some our positive experiences of using it. The installation of the system can be described as simple (see Section 2.1), although it is a two-man job for larger computer rooms. We have used it to monitor an average of 18-25 machines per room. The system is stable, and in almost a year of intensive use, we have not experienced any operational problems, and it has worked practically as expected. We can say from our experience that it has been a great help in teaching the practical lessons of Programming Languages, in monitoring the students' activities, in making the best use of the lessons, in writing the weekly tests, clearly contributing to the quality of teaching. Figure 1 shows a 16-machine supervision system. While writing a Cisco class test, one of the students wanted to watch a video. Figure 2 shows the use of a calculator during a test writing session. The timestamp on the Veyon screenshots comes in handy.

As regards the evolution of student performance, two groups were selected for the same class as an experiment. In the case of both groups, the number of students was 18. In one group we did not use Veyon, while in the other we introduced it. As shown in Figure 3, the latter group showed some deterioration in performance over the semester. While in the first case the students' scores on the online tests averaged above 90%, when using Veyon they fluctuated between 80 and 90%. Given that the minimum performance required to obtain most industry certificates is 70%, it can be said that students performed well using Veyon, and not least as an indication of their true level of knowledge.

4 Conclusion

In this paper we have presented the possibilities of using CMSs in education. Out of seventeen CMS and EMS, we finally chose Veyon because, despite being free of charge, it includes all the relevant features needed for tutor supervision. After nearly a year of live testing, we concluded that its stable operation makes it a safe choice for practical class supervision. The use of Veyon does not cause any significant deterioration in student performance, increases the quality of teaching of practical classes and facilitates the correctness of the assessment. Although Veyon itself cannot be used for video recording, the possibility of screen recording and the nature of the monitoring system make it worthwhile to inform students about the monitoring system used in the syllabus at the beginning of the semester.

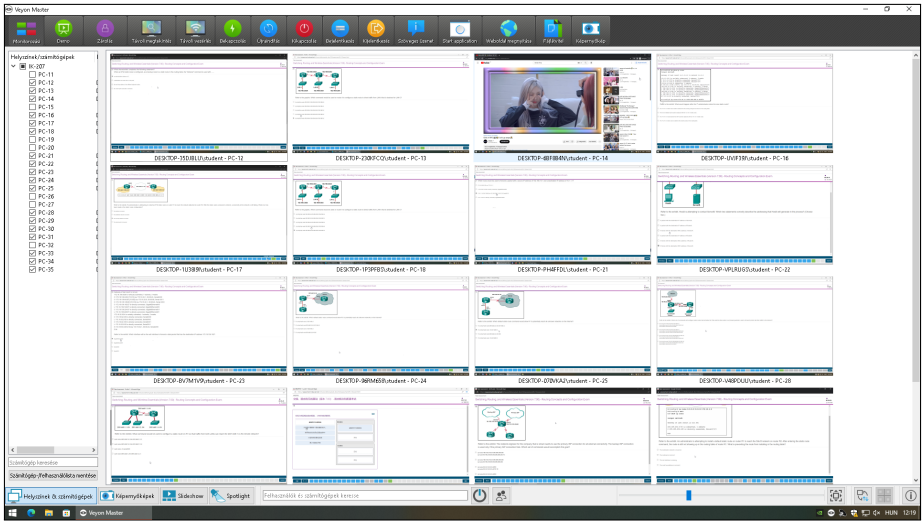


Figure 1: Veyon-supervised testing environment.

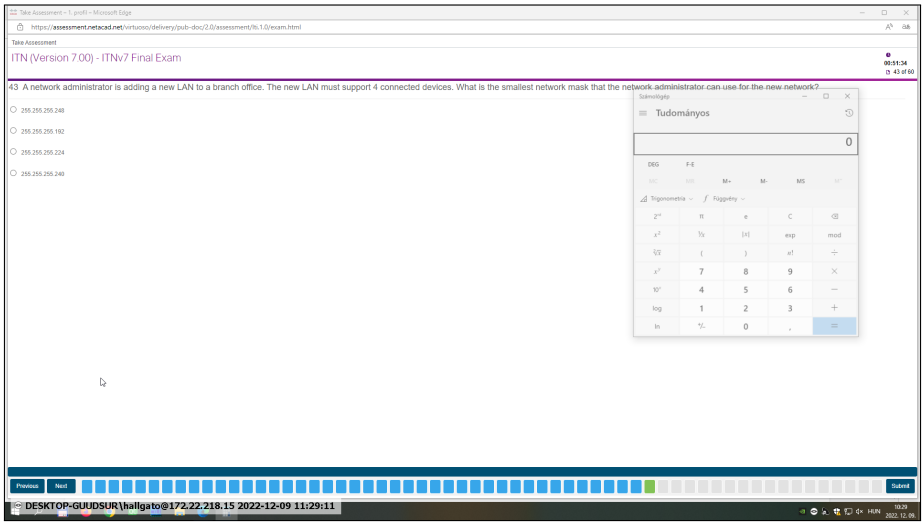


Figure 2: Monotoring a student screen with Veyon.

Acknowledgements

The author thanks to Ádám Szikra, the system administrator of the Faculty of Informatics of the University of Debrecen, who installed the Veyon software in many computer classrooms, enabling its testing and use in education.

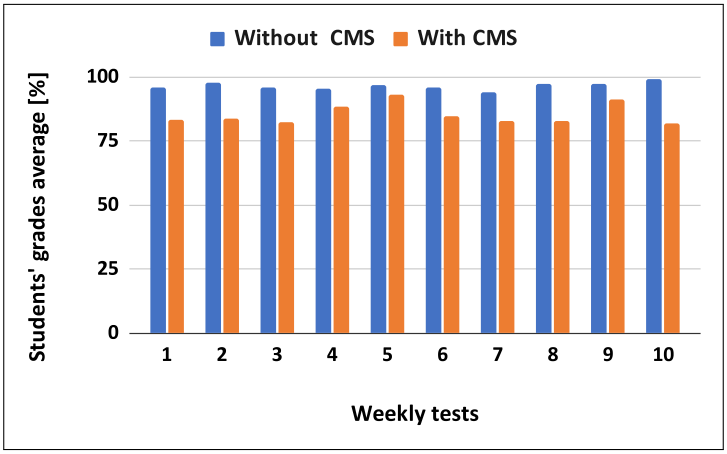


Figure 3: Comparing student's one semester performance with and without Veyon CMS in case of Cisco classes.

References

[1] Ab tutor official webpage. http://www.abtutor.com [Accessed on March, 2023].

[2] Bakonyi, V., Illés, Z., and Verma, C. Towards the real-time analysis of talks. In *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, pages 322–327, 2020. DOI: 10.1109/ICCAKM46823.2020.9051507.

[3] Dyknow official webpage. https://www.dyknow.com/ [Accessed on March, 2023].

[4] Employee desktop live viewer official webpage. https://www.nucleustechnologies.com/employee-desktop-live-viewer.html [Accessed on March, 2023].

[5] Epoptes official webpage. https://epoptes.org/ [Accessed on March, 2023].

[6] Faronics insight official webpage. https://www.faronics.com/en-uk/products/insight [Accessed on March, 2023].

[7] García, S., Gallardo, A., Larios, D. F., Personal, E., Mora-Merchán, J. M., and Parejo, A. Remote lab access: A powerful tool beyond the pandemic. In *25th Technologies Applied to Electronics Teaching Conference*, pages 1–5, 2022. DOI: 10.1109/TAEE54169.2022.9840672.

[8] Impero classroom management software official webpage. https://www.imperosoftware.com/impero-classroom-management-software/ [Accessed on March, 2023].

[9] Lanschool air official webpage. https://lanschoolair.com/ [Accessed on March, 2023].

[10] Lanschool classic official webpage. https://lanschool.com/solutions/classic/ [Accessed on March, 2023].

[11] Lanschool teaching and classroom management software for k-12 schools, 2023. https://lanschool.com [Accessed on 2023-04-06].

[12] Master teaching online. https://www.masterteachingmisc.com/veyon-vs-netsupport-classroom-is-paying-for-it-worth-it/ [Accessed on March, 2023].

[13] Molnár, G., Szűts, Z., and Balogh, Z. Modern digital web 2.0 devices and services supporting the teaching of technology and informatics. In *2019 IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 000089–000094, 2019. DOI: 10.1109/SISY47553.2019.9111555.

[14] Mythware classroom management software official webpage. https://www.planetteched.com/mythware-classroom-management-software/ [Accessed on March, 2023].

[15] Net monitor for employees professional official webpage. https://networklookout.com/ [Accessed on March, 2023].

[16] Netop vision official webpage. https://vision.netop.com/ [Accessed on March, 2023].

[17] Netsupport school official webpage. https://www.netsupportschool.com/ [Accessed on March, 2023].

[18] Quasar official webpage. https://github.com/quasar/Quasar [Accessed on March, 2023].

[19] Schoolvue classroom management official webpage. https://www.crosstecsoftware.com/schoolvue [Accessed on March, 2023].

[20] Screenwatch official webpage. https://www.acs-linksystems.com/products/screenwatch.cfm [Accessed on March, 2023].

[21] Surveilstar employee software official webpage. https://www.surveilstar.com/multi-screen-monitoring.html [Accessed on March, 2023].

[22] Uramová, J., Moravčík, M., Šterbák, M., and Remeň, J. Effective supervision of students' activity during misc distance learning and testing. In *INTED2022 Proceedings*, 16th International Technology, Education and Development Conference, pages 4815–4825. IATED, 2022. DOI: 10.21125/inted.2022.1260.

[23] Veyon official webpage. https://veyon.io/ [Accessed on March, 2023].

Acta Cybernetica **27** (2025) 53–65.

P4 Specific Refactoring Steps*

Máté Tejfel[ab], Dániel Lukács[ac], and Péter Hegyi[ad]

Abstract

P4 is a domain-specific programming language for programmable switches running inside next-generation computer networks. The language is designed to use the software defined networking (SDN) paradigm which separates the data plane and the control plane layers of the program. The paper introduces tool-supported refactoring steps for P4. The challenge in this task is that P4 has special domain-specific constructs that cannot be found in other languages and as such there is no existing methodology yet for refactoring these constructs. The proposed steps are implemented using P4Query, an analyzer framework dedicated to P4.

Keywords: P4 language, refactoring steps

1 Introduction

P4 [2] is a domain-specific programming language which enables a new approach to programming computer networks. It adopts the software defined networking (SDN) paradigm [10] which separates the data plane and the control plane layers of the program. P4 focuses on the data plane, while we need some other tool to create the control plane. It facilitates the implementation of the concept of fully programmable networks making possible the development of programmable data planes.

The paper introduces tool-supported refactoring steps for P4 with two-fold objectives. On one hand, we aim to assist developers to take full advantage of the programmability of P4, by providing standard refactoring services commonly found in IDEs of modern high-level languages. On the other hand, we want to enable P4 code optimizations that are aware of the unique make-up of this language.

*This research is in part supported by the project no. FK_21 138949, provided by the National Research, Development and Innovation Fund of Hungary. The research was partly supported by Ericsson Hungary.

[a]Faculty of Informatics, Eötvös Loránd University, ELTE, Budapest, Hungary
[b]E-mail: matej@inf.elte.hu, ORCID: 0000-0001-8982-1398
[c]E-mail: dlukacs@inf.elte.hu, ORCID: 0000-0001-9738-1134
[d]E-mail: immsrb@inf.elte.hu

DOI: 10.14232/actacyb.308085

The difficulty of the task is that P4 has many unique language constructs, for which there is no existing methodology for refactoring. One example is the declaration and application of lookup tables. As these components are performance-critical, their definitions are target-dependent: compilers have to consider the capabilities of their target architectures, and choose where to map tables to get the most efficient outcomes. Unfortunately, this is less straightforward than it looks, in particular because the intended content of the tables is unknown at compile-time, and so is the traffic that will be processed by these tables. As such, performing such optimizations is usually out of the scope of standard compilers, and it falls on specialized optimization tools, or maybe the well-informed developer, who can take into account these application-specific factors. We expect neither of these approaches to be prepared for handling machine-specific representations, and so it is preferable to perform the optimizing code transformation on the highest level, that is, directly on the P4 code.

1.1 Related work

There are currently many different tools that support the development of P4 programs. Some of them concentrate on error checking of P4 programs. For example, BF4 [3] is created as a P4C backend, which can not only detect error possibilities, but it is able to repair them by adding new keys to the lookup tables of the program and modify the table contents. Another tool p4-data-flow [1] uses data flow analysis to detect potential bugs in P4 switch codes.

Some other tools have been created for different purposes. For example, p4pktgen [9] uses symbolic execution for automatically generating test cases. SafeP4 [4] is a language which has precise semantics and a static type system that can be used to obtain guarantees about the validity of all headers which are used or modified by the program. The type checker of the language (P4Check) can also check P4 programs executing some static analysis on them.

P4 refactorings can be particularly useful for dataplane disaggregation, a problem recently addressed by Flightplan [13]. The objective here is to optimally segment P4 programs so that individual program segments can be assigned to resources, in turn transforming P4 into a programming language for the "one big switch" networking model. Flightplan can solve the allocation problem, and refactoring can help with the realization of program splitting, moreover, semantics-preserving program transformations may allow new, previously unseen forms of segmentations, enabling further optimization of allocations.

Another potential application for P4 refactorings can be found in [8]. The authors apply the normal form concept of relational database theory to lookup tables in P4 programs. Due to dependencies between their fields, large tables often contain significant amount of redundancy. By recognizing these dependencies, tables can be decomposed into smaller, irredundant tables. This irredundant representation is called a normal form, and normalization can be realized by vertically splitting tables. The authors find that on many targets normalization leads to better efficiency, because smaller, simpler tables can be updated by the controller with less work,

and because it is easier for compilers to find optimal representations for simpler tables.

The refactorings presented in this paper build on P4Query [7], whose program graph representation was mostly inspired by [5], a similar, but more established static analysis tool for Erlang, also developed at ELTE. A key idea in RefactorErl is that persisting pre-calculated semantic information in a database can both simplify and speed up refactorings. This also enables incremental refactorings where syntactical changes automatically trigger semantic analysis. Beyond being a tool for Erlang developers, RefactorErl is also a framework aiming to support quick and correct implementation of new refactorings.

2 P4 language

Every P4 program contains at least three main components: these are the parser, the match/action pipeline and the deparser [11]. The parser reads a packet from the network (as a bitstream) and builds up its header structure and its metadata information (while leaving the payload part of the packet unchanged). The match/action pipeline can read and modify the headers and the metadata (add, delete headers, modify header fields or metadata). While the deparser part creates the new packet using the original payload (for example, by changing the order of headers or omitting headers). This new packet will be sent forward on the network.

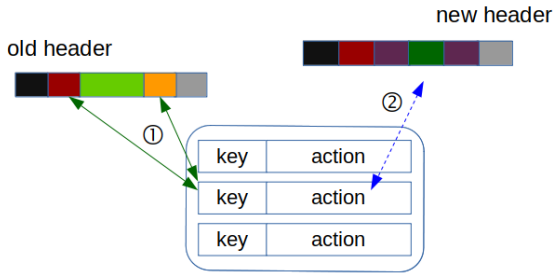


Figure 1: Match/Action table application

One of the most important part of the execution of a P4 program is the application of the match/action tables[1]. Figure 1 illustrates a match/action table application. The different colours in the headers represent different fields, which may vary in size. Every row in the table contains a key and an action part. The key refers to some fields of the header structure. The table application – when processing a specific packet – first searches the appropriate row in the table based on the concrete field values of the headers using a given lookup algorithm (exact, longest prefix match or ternary lookup). If the algorithm finds the appropriate

[1]More information about P4 can be found on the official website: https://p4.org/.

row, it will execute the action part of the row, which will modify the headers of the packet. If no appropriate line exists, the program executes the default action of the table. If the default action is not defined and no entry matches, then the table does not affect the packet.

It is worth mentioning that a P4 program defines only the data plane layer of a packet processing algorithm, namely it will define only the structure of match/action tables. Listing 1 introduces an example declaration of a match/action table in P4.

The declaration defines the key fields, the used lookup mechanisms, the possible actions, the maximum size (maximum number of rows), the default action and the const entries (the explicitly defined rows) of the table. The last three are optional. Specific data in the table (which actions will be executed for which field values) are specified by the control plane layer of the algorithm which is out of the scope of the P4 program.

```
table ipv4_lpm {
    key = {
        hdr.ipv4.srcAddr: exact;
        hdr.ipv4.dstAddr: lpm;
    }
    actions = {
        ipv4_forward;
        modify_dst;
        drop;
        NoAction;
    }
    size = 1024;
    default_action = drop();
    const entries = {
      ...
    }
}
```

Listing 1: Example of a match/action table declaration in P4

3 P4Query

The refactorings were realized with P4Query [7], a static analysis framework for P4[2]. The framework is centered around an extensible internal graph representation where the results of the different static analysis methods are stored also as part of the graph. The information in the knowledge graph is accessed using graph queries written in the Gremlin query language [12]. This way the framework guarantees unique standard representation both for the stored data, and for the data access mechanism. As the graph instance is detached from the code analysis, users and developers alike can access it by external tools for visualising, monitoring and

[2]The P4Query framework is available in GitHub: https://github.com/P4ELTE/P4Query.

validating purposes. Each static analysis has to declare which other static analyses it depends on, and a central component ensures that all analyses are executed in order, and without collisions. This also ensures that only necessary analyses are performed.

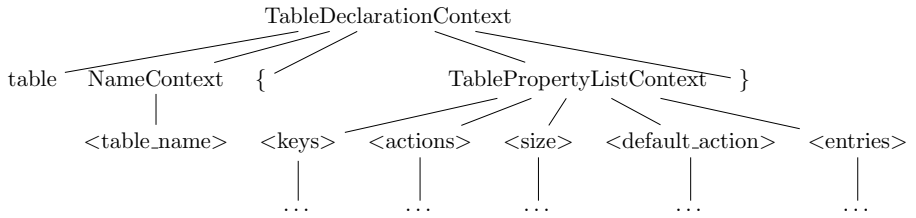


Figure 2: Representation of a match/action table in P4Query

Figure 2 illustrates the representation of a match/action table in P4Query. As the steps analyse and manipulate this representation, they can use the built-in analyzers of the tool to make checking the prerequisites much more easier.

4 Refactoring steps

In this section, we present the refactorings we defined for P4. Ultimately, refactorings are transformations of the syntax tree, satisfying the assumption that the resulting tree is semantically equivalent to the original one. Due to the complexity of the refactorings, circumspection must be exercised when executing the transformation, requiring to a considerable overhead. The general scheme of the refactoring is depicted by Algorithm 1, executing an R set of refactorings over a P4Query program graph G.

As we will see, more complex transformations often have various preconditions: for example, splitting a match/action table by its keys naturally requires a table with at least two keys. Thus, the first step of the algorithm is to check such transformation-specific preconditions. If the precondition is not satisfied, it does not make sense to start the refactoring. As P4 match/action tables are usually filled at runtime (by the SDN controller), it may be impossible to check all preconditions at analysis-time: in this case, the user should be warned that some preconditions could not be checked and it will be the responsibility of the user to ensure that the P4 program is being executed in the right environment. Additionally, the user should be subsequently given an option to cancel the refactoring, if they cannot guarantee this.

Checking the semantic equivalence between input and output syntax tree is one of the most important properties regarding implementation correctness. Unfortunately, exhaustive checking is not feasible due to the halting problem, which means we have to resort to non-exhaustive testing. While post-compile-time testing is the standard method for ensuring implementation correctness, it is difficult

Algorithm 1 General scheme of refactorings

Procedure Refactor(R, G):
Input: R is a list of refactorings
Input: G is graph, includes AST
Result: G conditionally transformed by refactorings in R

```
1:  for all r ∈ R do
2:    if ¬ r.CheckPreconditions(G) then
3:      Exit( "Preconditions failed:", r.FailedPreconditions(G))
4:    end if
5:    if r.HasExternPreconditions(G) then
6:      Warn( "Preconditions could not be checked:", r.FailedPreconditions(G))
7:    end if
8:    G.StartTransaction()
9:    r.Execute(G)
10:   if ¬CheckConsistency(G) then
11:     Warn( "Consistency test failed, reverting.")
12:     G.RollbackTransaction()
13:   else
14:     G.FinishTransaction()
15:   end if
16: end for
```

to sufficiently test complex systems with varied runtime parameters. For example, the graph backend under P4Query can be switched with relative ease, but this may have unexpected effects due to differences between graph backend implementations. Another issue could be parallelization (not yet a feature of P4Query, but could be a feature in graph backends) that – due to non-determinism – can also have unexpected effects. For this reason, the algorithm supports testing graph consistency after a refactoring was executed: if the test fails, the transformation is reverted so that the graph is left in a consistent state.

One of the promises of using a graph database in P4Query was that the database provides built-in support for operations such as atomic transactions and transaction rollback. Unfortunately, this is not always the case in practice: in the implementation, we had to implement rollbacks manually, since the in-memory database engine version (TinkerPop 3.4.4) used by default in P4Query does not support these features.

With having the general outline discussed, we can now focus on individual refactorings.

4.1 Table structure modification

The P4 program can modify the incoming packets (namely the headers) by applying match/action tables. For a given program, determining the optimal table structure is a difficult task. It can often be the case that in a given hardware environment

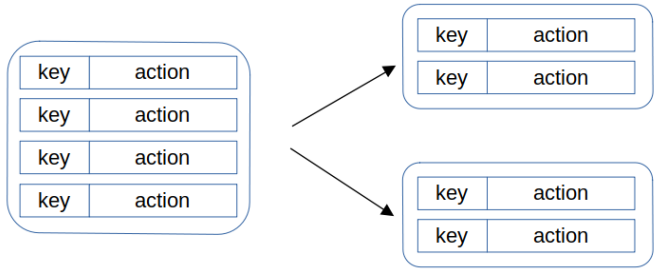


Figure 3: Horizontal splitting

using fewer but larger tables, while in another environment using more but smaller tables may yield better results. Therefore our implemented refactorings mainly focus on the modification of the match/action tables, namely horizontal and vertical splitting of tables, merging tables, changing the execution order of tables.

4.1.1 Horizontal table splitting

One of the simplest ways of table splitting is horizontal splitting introduced by Figure 3. In this case, the basic table structure remains the same, but we duplicate the table by halving the maximum size.
The prerequisites of the transformation are the following.

- The maximum size of the original table is explicitly defined.
- The table uses exact lookup mechanism.
- The application of the table appears in the match/action pipeline.

If the prerequisites hold, the following transformation steps must be taken.

1. Creating a new table with the same structure (using also the same default action).
2. Halving the maximum sizes.
3. Modifying the default action of the original table to `NoAction`. (A built-in action that changes nothing.)
4. If there exist more explicitly defined rows in the original table as the new maximum, copying the excess into the new table. (It is a possibility in P4 to defining explicit rows to a table.)
5. Searching every point in the match/action pipeline where the original table was applied. Modifying these applications to the execution of a sequence which first applies the original table and if no appropriate row was found during the lookup (the default action was executed) applies the newly created table.

4.1.2 Vertical table splitting

A much more complex case is when the table is split vertically. This step is executable if the table key contains two different fields. The parameter of the step is a value set which determines the likely values of the first key part which can appear in the table. It is worth noting that the control plane layer can change table contents dynamically, therefore this parameter has to be determined manually by an expert or based on some information coming from the control plane layer. Using this set, we can split the original table in such a way that first we just lookup the first key part in the determined set and then lookup the second key part in independent tables.

Considering more precisely the prerequisites of the transformation, we obtain the following conditions.

- The key of the table contains two different field values.
- The table uses the exact lookup mechanism for the first key part.
- The parameter set contains valid values for the first key part.
- The application of the table appears in the match/action pipeline.

If the prerequisites hold, the following transformation steps must be taken.

1. Creating a new dispenser table. Adding an explicit row for every values from the parameter set using a specific variant of the built-in `NoAction` as action.
2. Creating one executor table for every value from the parameter set. Copying the appropriate rows from the explicitly defined rows of the original table into the new tables.
3. Modifying the explicitly defined rows of the original table leaving only key values which do not appear in the parameter set.
4. Searching every point in the match/action pipeline where the original table was applied. Modifying these applications to the execution of a sequence which first applies the dispenser table and after that a `switch` branch based on the executed action, where every branch executes the application of the corresponding new executor table. Add the application of the original table as the default branch.

As an illustration, consider the table shown in Listing 1 and assume that the control plane layer will set the entries described in Table 1 into the table (for simplicity, we use only small numbers instead of real addresses in the example).

Executing a vertical splitting with the parameter set $\{1, 2, 3\}$, first we have to create three new specific version of the action `NoAction`. The new actions are introduced by Listing 2.

Table 1: Before vertical splitting

src_addr	*dst_addr*	action
1	1	$ipv4\_forward$
1	2	$ipv4\_forward$
1	3	$drop$
2	1	$drop$
2	2	$modify\_dst$
3	1	$modify\_dst$
4	4	$NoAction$

```
action case1() {  }
action case2() {  }
action case3() {  }
```

Listing 2: Three new empty actions

Then a new table (called dispenser) should be created which determines based on the first key value which new table will be used during the lookup. The table declaration of the dispenser table of our example is described by Listing 3.

```
table dispenser {
  key = {
    hdr.ipv4.srcAddr: exact;
  }
  actions = {
    case1;
    case2;
    case3;
  }
  size=1024;
}
```

Listing 3: Example dispenser table

After that we have to create a new table for every value in the parameter set (called executor tables) to execute the remaining part of the lookup (based on the second key value). Listing 4 defines the declaration of the first executor table in the example (the other two executor tables have very similar declarations).

```
table executor1 {
      key = {
          hdr.ipv4.dstAddr: lpm;
      }
      actions = {
          ipv4_forward;
          modify_dst;
```

```
            drop;
            NoAction;
        }
        size = 1024;
        default_action = drop();
    }
```

Listing 4: Example executor table

Finally the `ipv4_lpm.apply()` match/action table application should be replaced with the code snippet described by Listing 5 at each point in the program where it originally appears.

```
switch(dispenser.apply().action_run) {
    case1 : { executor1.apply(); }
    case2 : { executor2.apply(); }
    case3 : { executor3.apply(); }
    default : { ipv4_lpm.apply(); }
}
```

Listing 5: New code snippet

Using the new structure, we need to use the entries described by Table 2 and Table 3 in the new dispenser table, in the original table and in the new executor tables to provide the same functionality.

Table 2: Vertical dispenser and the original table after splitting

src_addr	action
1	$case_1$
2	$case_2$
3	$case_3$

src_addr	*dst_addr*	action
4	4	*NoAction*

Table 3: Executor tables

$executor_1$	
dst_addr	action
1	*ipv4_forward*
2	*ipv4_forward*
3	*drop*

$executor_2$	
dst_addr	action
1	*drop*
2	*modify_dst*

$executor_3$	
dst_addr	action
1	*modify_dst*

4.2 Changing the execution order of tables

In addition to the structure of the tables, P4 programs also define the order in which they are executed. However in many cases the applied tables use independent fields of the header structure, so changing the execution order of them will not modify the results of the program. Changing the order can lead to more optimal memory usage or can allow further modifications, such as table merging.

More formally we check the independence between tables based on the dependency relations defined by Lavanya et al. [6]:

- **Match dependency** where the actions of the first table can modify a field which is used as a key in the subsequent second table.
- **Action dependency** where the first table and a subsequent second table both can change the same field.
- **Successor dependency** where the match result of the first table determines whether a second table should be executed or not.
- **Reverse match dependency** where the first table matches on a field that can be modified by a subsequent second table, and the first table must finish matching before the second table changes the field.

Based on these definitions we have implemented a dependency analysis in P4Query which – using the existing control flow and data flow analyses – can determine the dependency between two tables. We have defined and implemented a simple version of execution order changing based on this analysis.

The prerequisites of the transformation are the following.

- The applications of the two table are successive applications in the match/action pipeline of the P4 program.
- There does not exist match, action, successor or reverse match dependency relation between the two tables.

If the prerequisites hold, the transformation swaps the application of the two table in the program.

4.3 Further refactoring step

In addition to those described above, we have defined additional P4 specific steps. One is the merging operation corresponding to the previously described splitting transformations. We also defined a variant of the horizontal splitting which split the table based not on the size, but on some much likely used action appearing in the table and using as parameter of the transformation. We have defined some generic (not P4 specific) transformation steps too (e.g. parameter renaming and magic number replacing).

5 Conclusion

We have presented the definition of refactoring steps for P4. The proposed transformations focus mainly on the manipulation of match/action tables which are basic language elements in P4. The steps were implemented using the P4 specific analyzer tool, P4Query. The transformations are executed on the level of the internal representation of the tool which helps in performing the analyses of the prerequisites. Our current solution applies some generic consistency check provided by P4Query on the resulted graph. In the future we plan to implement further, more complex refactoring steps using the created refactoring infrastructure and extend the actual consistency checks with more specific verification methods.

References

[1] Birnfeld, K., da Silva, D. C., Cordeiro, W., and de França, B. B. N. P4 switch code data flow analysis: Towards stronger verification of forwarding plane software. In *NOMS 2020 – 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–8, 2020. DOI: 10.1109/NOMS47738.2020.9110307.

[2] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and Walker, D. P4: Programming protocol-independent packet processors. *SIGCOMM Computer Communication Review*, 44(3):87–95, 2014. DOI: 10.1145/2656877.2656890.

[3] Dumitrescu, D., Stoenescu, R., Negreanu, L., and Raiciu, C. Bf4: Towards bug-free P4 programs. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '20, page 571–585, New York, NY, USA, 2020. Association for Computing Machinery. DOI: 10.1145/3387514.3405888.

[4] Eichholz, M., Campbell, E., Foster, N., Salvaneschi, G., and Mezini, M. How to avoid making a billion-dollar mistake: Type-safe data plane programming with SafeP4. In Donaldson, A. F., editor, *33rd European Conference on Object-Oriented Programming*, Volume 134 of *LIPIcs*, pages 12:1–12:28. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. DOI: 10.4230/LIPIcs.ECOOP.2019.12.

[5] Horváth, Z., Lövei, L., Kozsik, T., Kitlei, R., Tóth, M., Bozó, I., and Király, R. Modeling semantic knowledge in Erlang for refactoring. In *International Conference on Knowledge Engineering, Principles and Techniques*, pages 38–53, Cluj-Napoca, Romania, 2009.

[6] Jose, L., Yan, L., Varghese, G., and McKeown, N. Compiling packet programs to reconfigurable switches. In *12th USENIX Symposium on Networked Systems*

Design and Implementation, pages 103–115, Oakland, CA, 2015. USENIX Association. URL: https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/jose.

[7] Lukács, D., Tóth, G., and Tejfel, M. P4Query: Static analyser framework for P4. *Annales Mathematicae et Informaticae*, 2023. DOI: 10.33039/ami.2023.03.002.

[8] Németh, F., Chiesa, M., and Rétvári, G. Normal forms for match-action programs. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, CoNEXT '19, page 44–50, New York, NY, USA, 2019. Association for Computing Machinery. DOI: 10.1145/3359989.3365417.

[9] Nötzli, A., Khan, J., Fingerhut, A., Barrett, C., and Athanas, P. P4pktgen: Automated test case generation for P4 programs. In *Proceedings of the Symposium on SDN Research*, SOSR '18, New York, NY, USA, 2018. Association for Computing Machinery. DOI: 10.1145/3185467.3185497.

[10] Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K., and Turletti, T. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys and Tutorials*, 16(3):1617–1634, 2014. DOI: 10.1109/SURV.2014.012214.00180.

[11] P4 language specification, 2021. URL: https://p4.org/p4-spec/docs/P4-16-v1.2.2.html.

[12] Rodriguez, M. A. The Gremlin graph traversal machine and language. *Proceedings of the 15th Symposium on Database Programming Languages*, 2015. DOI: 10.1145/2815072.2815073.

[13] Sultana, N. and et al. Flightplan: Dataplane disaggregation and placement for P4 programs. In Mickens, J. and Teixeira, R., editors, *18th USENIX Symposium on Networked Systems Design and Implementation*, pages 571–592. USENIX Association, 2021. URL: https://www.usenix.org/conference/nsdi21/presentation/sultana.

Acta Cybernetica **27** (2025) 67–81.

Co-reference, Thematic, and Network Analysis of a Selected Hungarian Poem and Its English Translation (Füst Milán: A szőlőműves / The Vine-Dresser)*

István Károly Boda[a] and Erzsébet Tóth[b]

Abstract

In our paper we present a parallel co-reference analysis of the Hungarian poem "A szőlőműves"("The Vine-Dresser") by Milán Füst and its English translation. We explore the textual world of the poem and compare the Hungarian and English texts using formal linguistic tools based on semiotic textology. We also reveal the possible differences between the original Hungarian text and its English translation. The results of the analysis prove that co-reference analysis devised and elaborated by János S. Petőfi can be effectively applied in a polyglot environment.

We also introduce the hypertext implementation of the co-reference analysis of the selected poetic texts in a form of a web page. We would like to show that this has many practical advantages, for example, the analysis and its results can be transparent, accessible, and verifiable to everyone. In addition, the created web page provides additional aspects for the analysis. For example, considering the poetic text as a network, we can investigate whether the scale-free feature is also relevant in the textological environment – as it can be experienced in many other areas of reality.

Keywords: co-reference analysis, theme-rheme relationship, scale-free network, poetic text, textology

*This research has been supported by Virtual Reality Laboratory, Qos-HPC-IoT Laboratory and project TKP2021 NKTA-34 of the University of Debrecen, Hungary. Project no. TKP2021-NKTA-34 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the TKP2021-NKTA funding scheme.

[a]Department of Mathematics and Informatics, Debrecen Reformed Theological University, Debrecen, Hungary, E-mail: boda.istvan@drhe.hu, ORCID: 0000-0002-8827-6452

[b]Department of Data Science and Visualization, Faculty of Informatics, University of Debrecen, Debrecen, Hungary, E-mail: toth.erzsebet@inf.unideb.hu, ORCID: 0000-0003-1805-6283

DOI: 10.14232/actacyb.308016

1 Introduction and background

The background of our study is a 3D virtual library project which started in 2013, as part of cognitive infocommunications (CogInfoCom) direction of research [2, 3]. The content of the virtual library includes verbal and multimedia materials (mainly literary texts in English, and in some cases in Hungarian) which, among other things, can be excellently used for English language learning purposes. In the recent years, we gradually put the emphasis on bilingual language learning materials, adding Hungarian or English translations to selected texts of the virtual library. In the current implementation of the 3D virtual library we extensively use the 3D features of the MaxWhere Seminar System [14]. Nevertheless, from the beginning of the project we have been creating the hypertext representation of the library materials using the standard web technology (i.e. HTML/CSS/JavaScript etc.).

As regards the selection of the content of the library, we have always considered it our mission that the eternal cultural values which classical literary works can convey to the present-day culture would also be available for the users of the 3D virtual library, and especially for the young generations including the members of the "generation CE" (i.e. generation of cognitive entities) who "are already growing up in a kind of 'co-evolution' with ICT" [3]. Keeping in mind their wants and needs as well, we created a bilingual (English-Hungarian) language learning material developed for English language learners at an advanced level [10]. We attempted to organise the bilingual material to form a more or less scale-free network of interconnected nodes so as to provide an efficient and user-friendly learning environment [11].

As a further development of our 3D virtual library project we would like to provide access to *full texts*, both in Hungarian and in English, for the potential users of the bilingual language learning material. Because the mission of the virtual library project includes the accumulation and transmission of cultural values expressed in classical literary works, **we decided to select short poetic texts with high emotional content** which may involve attentive and intensive reading and, in turn, effective learning. Because of the high complexity of natural language texts, and especially poetic texts, it is crucial that the texts should be processed and prepared efficiently to satisfy the needs of the language learners. Therefore, in addition to the basic text processing and organising technologies which we have used so far, we also applied linguistic and textological tools for the comprehensive and intricate processing of poetic texts.

Co-reference analysis developed by János S. Petőfi [15, 16, 17] is a very effective textological and/or text-linguistic tool which enables us to explore some essential aspects of the data structure and organisation of natural language texts. In our research presented in this paper we applied co-reference analysis to a selected Hungarian poem and its English translation. The results of the analysis are presented on a web page the structure and organisation of which are fully compatible with the hypertext-based content of the 3D virtual library project. The web page whose details will be presented in the following sections of this paper was created as an integral part of the virtual library, therefore the content of the web page can be

displayed not only on the "traditional" 2D web but also in the MaxWhere virtual 3D environment.

2 General overview

In this paper, we will analyse and process a selected Hungarian poem and its English translation (Füst Milán: *A szőlőműves*; Milán Füst: *The Vine-Dresser*, translated by István Tóthfalusi; see [12] using the methods and notation of *co-reference analysis* the complex notation and terminology of which have been invented, developed and applied to the analysis of Hungarian literary texts by János S. Petőfi [15, 16, 17]. With reference to our bilingual language learning material, one of the main advantages of co-reference analysis is that it can serve as a kind of metalanguage which is independent of natural languages. That characteristic makes co-reference analysis particularly useful in computer-based text processing as well [4, 5, 6].

The central aim of the application of co-reference analysis for the polyglot investigation of selected poetic texts is to identify and organise the Hungarian and English keywords of the texts in order to create entries for the bilingual language learning material and explore the thematic structure of the analysed texts. As regards the methodology and use of co-reference analysis, we rely on the results of our previous studies mainly published in the series of Officina Textologica. In one study we investigated the theme-rheme relationships of the poem *"A szőlőműves"* by Milán Füst [7]; in other studies we carried out a parallel analysis of the Hungarian and English version of selected poems [8, 9]. Note that each study was based on the methodology of co-reference analysis with special emphasis on the formal description of poetic texts.

In our current study at first we present (part of) the parallel co-reference analysis of the poems *"A szőlőműves"* and *"The Vine-dresser"*, then we formally describe the theme-rheme (or topic-comment) relationships between the keywords of the texts, display their thematic structure, introduce a web page which establishes **a bilingual hypertext representation of both texts**, and examine, from a network theoretical perspective, the characteristics of the network of hypertext links that represent the network structure of the analysed texts.

Because of the complexity of natural language texts, it is essential to represent the linguistic knowledge as well as the background knowledge which are both necessary to understand and interpret the analysed texts. Taking a formal approach, we use several types of dictionaries as widely accepted sources of linguistic knowledge on the one hand; and we seek and select additional texts as relevant sources of background knowledge on the other hand. During the analysis of the original texts (i.e. the selected poems by Milán Füst) we add **commentaries to the communication units** that make up the text sentences of the analysed texts in order to formally represent the linguistic and background knowledge. Then we create **entries for all keywords** that occur in any of the text sentences in the role of theme (or topic). The resulting entries will contain the corresponding text sentences, their communication units, and the commentaries that complement them. The entries

are considered as **network nodes**, and the connections between them are mainly established by the keywords that occur in either the communication units or the commentaries (or both of them).

In this way, we can build **the hypertext (HTML) representation of the Hungarian and English texts** based on the system of entries and connections. Due to the interactivity of the hypertext implemented as a web page, this representation reflects the formal description and the hypertextual interpretation of the texts and makes it accessible to potential users. From a technical view, the accessibility of the hypertext links of the web page from a JavaScript program enables the examination of the global properties of the thematic structure of the texts including the scale-free characteristic of the network. During the creation of the hypertext representation as a web page, we pay great attention to using the same data structure as that of the 3D virtual library, so that the created web page can be fully integrated into the virtual library as an inherent part of it.

3 Parallel co-reference analysis of the poem *A szőlőműves* by Milán Füst and its English translation

In the following, we assign the co-reference indices [ixx] occurring in the communication units [kxx, cxx] of the text sentences [Kxx] and commentaries [Cxx] in each text sentence of the selected poetic text and its English translation. In addition, we also provide the formal description of the sentence structure of the communication units in table format [7].

First, we determine the text sentences in the selected poem by Milán Füst (for the sake of simplicity the text sentences will be presented only in the English translation of the poem[1]

Füst Milán: A szőlőműves [K00]

Lám, a Medve ragyog s fiát veri: csöndre tanitja.[K01]
S lejjebb lassan, valamint tavirózsa, leúszik a Hattyú.[K02]
Alant sötétül a kékség s a dús domboldalt beboritja,
Melyre fehér házat, kicsikét, százat egy óriás parittya
Fekete, tar venyigék közt össze-vissza szórt...[K03]
S tiszta éjjelen, mélyen a hold alatt repűl
És fénylő, gyors felhőket űz az őszi szél...[K04]
...Csak épp megnézi még hegyét, kicsit még jár körűl
S aztán bucsúzik ő is, ki a súlyos fürtöt óvta:[K05] *ím' hogy itt a tél,*

[1]Note that both the text of the original Hungarian poem and its English translation, as well as the web page containing the co-reference analysis of the poetic texts and all the tables and figures presented in this study are fully accessible on the internet at the following link: URL: https://bodaistvan.hu/callimachus/texts/Fust_Milan-Szolomuves.html (2024-02-04).

A szótlan szőlőműves is pihenni tér.[K06]
Jön, leballag a hegyről s hol borpincék nehéz szaga terjed,
Puttonyát s számos szerszámait hűs kamarába teszi vissza...[K07]
S míg felenged a tél s a hordók kotyogó bora erjed,
Vídáman heverész és derüs kedvvel borocskáit issza
S tiszta bölcsességnek örűl, amíg kívül hull a hó.[K08]

Milán Füst: The Vine-Dresser [K00]

See, the Bear shines and beats his son: teaches him to be still. [K01]
Lower and slowly, like a water-lily, downward floats the Swan. [K02]
Where 'mong black and bare vines small white houses, a hundred and one,
Lie as if scattered at random by a giant sling... [K03]
And at clear nights, flying deep under the moon
The autumn winds chase clouds that are bright and fast...[K04]
...He still looks round his hill, walks a bit and takes leave soon,
He too who used to guard the heavy bunches:[K05] *winter's here and it's best*
For the speechless vine-dresser, too, to take his rest.[K06]
He comes trudging downhill and where wine-vaults spread their heavy smell,
He puts down his butt and his numerous tools in the chilly shed...[K07]
And while the bubbling wine ferments in the casks and winter's frosts dwell,
He sips at his tasty wines in good humour, lolling on his bed
And takes pleasure in pure wisdom while it snows without.[K08]
(translated by István Tótfalusi)

Then we determine the co-reference indices in each text sentence and in each communication unit of the text sentences. (Because of length constraints it will be shown only for the first three text sentences.)

[K01]=[k01]&[k02]&[k03]

Table 1: The co-reference indices in the 1st text sentence of the poem

[k01] Lám, a Medve.[i01] ragyog [az őszi[i13] égbolton(i13)[i02]]
[k01] See, the Bear[i01] shines [in the autumn[i13] sky(i13)[i02]]
[k02] s [a Medve[i01] a] fiát(i01)[i03] veri:
[k02] and [the Bear[i01]] beats his[i01] son:(i01)[i03]
[k03] [a Medve[i01] a fiát(i01)[i03]] csöndre[i04a] tanítja.
[k03] [he[i01]] teaches him[i03] to be still.[i04b]

The *commentaries* in the communication units are put into square brackets. Although the translation of the poem in every case faithfully follows the origi-

nal poem, smaller differences occur. For example, the English translation of the "csöndre tanítja" Hungarian phrase in the first text sentence corresponds "to to be still" collocation that we would rather translate to "nyugalomra inti" in Hungarian. Therefore, besides [i04a] "csend, csönd" Hungarian keywords ("silence" in English) we introduce [i04b] "nyugalom" (békesség, mozdulatlanság etc.) Hungarian keywords which also correspond to "tranquility" (peacefulness, standstill, etc. in English).

According to our interpretation, in Milán Füst's poetic text "csönd"[i04a] Hungarian phrase refers to the late autumn night which, in turn, in a symbolic sense can refer to the silence of the approaching winter[2], but in the English translation of the poem "still"[i04b] primarily emphasizes the *tranquility* and the *standstill*. In a general sense, it can be connected to quiet evening/night, but it is not in consistency with the appearing "őszi szél" ("autumn winds" in English) in the 10th communication unit ("chase clouds that are ... fast" in English; "gyors felhőket űz" in Hungarian).

After this, let us introduce a commentary unit and using it we provide the background knowledge which is necessary for a better understanding of the poem.

[C01a]=[c04a]

Table 2: The 1st commentary unit [C01a] of the poem

[c04a] [Csendes,[i04a] tiszta este(i04a)[i05a] van.]
[c04a] [It is a quiet[i04b] and clear evening.(i04b)[i05a]]

The first commentary unit makes explicit what we get to know from the context of the poem: it is a quiet (compare to "csendre tanítja" in Hungarian, "teaches him to be still" in English) and clear evening (because the constellations can be seen in the sky). The fact is that the night has not come yet, and it turns out from the communication unit [k06] ("Alant sötétül a kékség" in Hungarian; "The blueness below turns dark" in English). If we also intend to complement the commentary with a place dimension, then we can attach "a szőlőhegyen[i08]" addition to the communication unit [c04a] which will become clear from the communication unit [k07] (namely the darkening night "enfolds the rich slopes of the hill", "a dús domboldalt beborítja" in Hungarian).

[K02]=[k05]

[2] cf. e.g. "Most tél van és csend és hó és halál. / A föld megőszült" (Vörösmarty Mihály: Előszó). Interestingly enough, the English translation of the poem we have found on the internet uses "stillness" (and not "silence") for "csend": "It's winter now and death and snow and stillness, / The earth turned white" (translated by Peter Zollman). URL: https://www.babelmatrix.org/works/hu/V%C3%B6r%C3%B6smarty_Mih%C3%A1ly-1800/El%C5%91sz%C3%B3/en/2123-Prologue (2024-02-14)

Table 3: The 2nd text sentence of the poem

[k05] S lejjebb [az égbolton[i02]] lassan, valamint tavirózsa, leúszik a Hattyú.[i06]
[k05] Lower [in the sky[i02]] and slowly, like a water-lily, downward floats the Swan.[i06]

The three constellations mentioned by Milán Füst can be observed in the *autumn sky*. First, we have to pay attention to the summer sky to understand the meaning of the Cygnus, the Swan constellation. The brightest stars of the summer constellations constitute the so-called Summer Triangle. On its top left point the brightest star of the Cygnus constellation, the Deneb can be found. In the autumn sky the Deneb can be observed to the west from the meridian, and in the winter sky it can be seen below, next to the horizon. The greatest part of the Cygnus constellation cannot be seen in the winter sky [13]. Namely with the approach of the winter – according to Milán Füst's own words – it is true that in the sky "downwards floats the Swan", "leúszik a Hattyú"in Hungarian.

[K03]=[k06]&[k07]&[k08]

Table 4: The 3rd text sentence of the poem

[k06] Alant[i02] sötétül a kékség[i07]
[k06] The blueness[i07] below[i02] turns dark,
[k07] s [a kékség[i07]] a dús domboldalt(i08)[i31] beboritja [a szőlőhegyen[i08]],
[k07] [The blueness[i07]] enfolds the rich slopes(i08)[i31] of the hill[i08],
[k08] Melyre(i08)[i31] fehér házat,(i31)[i09] kicsikét, százat egy óriás parittya
[k08] Fekete, tar venyigék[i10] közt össze-vissza szórt...
[k08] Where(i08)[i31] 'mong black and bare vines[i10] small white houses,(i31)[i09]
[k08] a hundred and one, / Lie as if scattered at random by a giant sling...

In the text sentence the "blueness"("kékség"in Hungarian) is the metonymy of the sky and its darkening indicates the twilight, the coming of the evening. In the context of the poem the "rich slope"("dús domboldal"in Hungarian) refers to the **hill** ("szőlőhegy"in Hungarian), the "black and bare wines"("fekete, tar venyigék"in Hungarian) refer to the **late autumn**, so the third text sentence also designates the place and time dimensions of the poetic text all at once.

Though in the presented co-reference analysis of the poetic text the stylistic aspects are not in the center of our investigations, but here it is worth making a detour. Namely the "giant sling"metaphor ("óriás parittya"in Hungarian) used by Milán Füst can be closely connected to the symbolic and allegorical interpretation of the poem. The slings and the arrows in Shakespeare's *Hamlet*'s mono-

logue are the tools of misfortune that cause problems and troubles for the people (cf. "The slings and arrows of outrageous fortune", Hamlet III.i.58). However in the allegorical context of *The Vine-Dresser* the ominous tone disappears, and the "hill" ("domboldal" in Hungarian) can symbolize the vine-dresser's world, the "sling" ("parittya" in Hungarian) can symbolize the unpredictability of the life and the human destiny – armed with the wind. ("The wind blows where it wishes, and you hear its sound, but you do not know where it comes from or where it goes. So it is with everyone who is born of the Spirit.", John 3:8 [3]

[C01b]=[c04b]

Table 5: The 2nd commentary unit [C01b] of the poem

[c04b] [Csendes,[i04a] tiszta éjszaka(i04a)[i05b] van.]
[c04b] [It is a quiet[i04b] and clear night.(i04b)[i05b]]

The second commentary unit (similarly to the first one) makes also explicit what we can know from the following communication unit [k09] ("And at clear nights..." in English; "S tiszta éjjelen..." in Hungarian): the night came.

After these steps let us arrange in a table the co-reference indices introduced in the analysis, the Hungarian and English keywords connected to them, and the frequency of the indices (Table 6; because of length constraints we can publish only a part of the table.)

4 Representation of the thematic (theme-rheme) structure of the poem

The thematic relationships in the poetic text were analysed primarily on the basis of the **co-reference indices which occurred** more than once in the text sentences and in the introduced commentary units (Table 7; part). In the text we regarded the first occurrence of the co-reference indices connecting the text sentences as *rheme*, and their further occurrences as *theme*.

On the basis of the above-mentioned table we can illustrate the theme-rheme relationships of the text sentences and the *thematic network* of the poem in a graph (Figure 1).

Note that the poem is divided into two sharply separated parts, namely two *thematic units*. In the first thematic unit the first four text sentences ([K01]–[K04]) of the poetic text occur, and in the second thematic unit the additional four text sentences ([K05]–[K08]) of the poem can be found. The relationship between them is established by the second commentary unit ([C02]).

[3] The Holy Bible, English Standard Version cop. 2001 by Crossway Bibles, a publishing ministry of Good News Publishers URL: https://www.bibleref.com/John/3/John-3-8.html (2024-02-07)

Table 6: Keywords and co-reference indices of the poem *The Vine-Dresser* by Milán Füst

Index (frequency)	**Hungarian Keywords**	**English keywords**
[i01] (6)	Nagy Medve; Göncölszekér	Great Bear; Ursa Major
[i02] (6)	ég; égbolt	sky
(i13)[i02]	*őszi égbolt*	*autumn sky*
[i03] (4)	Kis Medve; Ursa Minor	Little Bear; Ursa Minor
(i01)[i03]	*a Nagy Medve fia*	*son of the Great Bear*
[i04a] (3)	csend; csönd	silence
(i15)[i04a]	*téli csend*	*winter silence*
[i04b] (3)	nyugalom; békesség; mozdulatlanság	tranquility; peacefulness; standstill
[i05a] (2)	este	evening
(i04a)[i05a]	*csendes, tiszta este*	*silent and clear evening*
(i04b)[i05a]	*nyugodt, tiszta este*	*quiet and clear evening*
[i05b] (4)	éjszaka; éjjel	night
[i05b]*	*éjszakák; éjjelek*	*nights*
(i04a)[i05b]	*csendes, tiszta éjszaka*	*silent and clear night*
(i04b)[i05b]	*nyugodt, tiszta éjszaka*	*quiet and clear night*
[i06] (2)	Hattyú; Cygnus	Swan; Cygnus
[i07] (8)	ég; levegő	sky; skies; air
[i07]	*kék ég; kékség*	*blue sky; blueness*

Table 7: Theme-rheme relationships in the poem *The Vine-Dresser by Milán Füst*

Text sentences	**Themes**	**Rhemes**
[K00]		[i16]
[K01]		[i01], [i13], (i13)[i02], (i01)[i03], [i04a], [i04b]
[C01a]	[K01]→[i04a] [K01]→[i04b]	(i04a)[i05a],(i04b)[i05a]
[K02]	[K01]→[i02]	[i06]
[K03]	[K01]→[i02]	[i07], [i08], (i08)[i31], (i31)[i09], [i10]
[C01b]	[K01]→[i04a] [K01]→[i04b]	(i04a)[i05b], (i04b)[i05b]
[K04]	[K01]→[i13] [C01b]→[i05b] [K03]→[i07]	[i11], [i12], (i13)[i14]
[C02]	[K01]→[i13]	[i15]

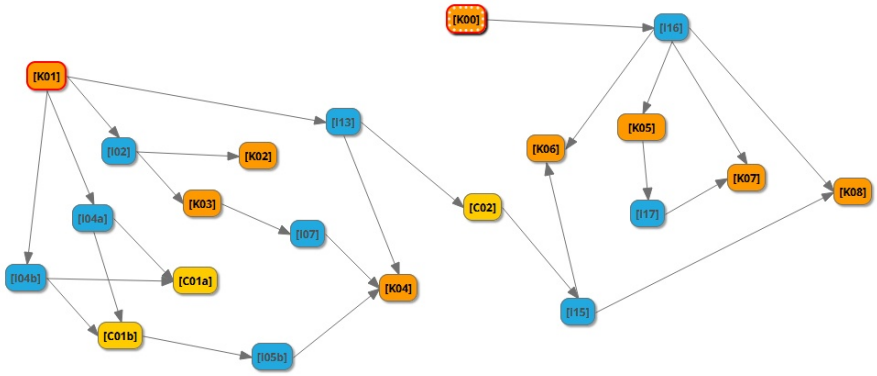


Figure 1: Theme-rheme relationships in the poem *The Vine-Dresser* by Milán Füst. Tool: https://www.mindmup.com (2024-02-06).

5 Examination of the hypertext representation of the poem from a network science perspective

As previously noted, the full co-reference analysis of the Hungarian and English versions of *The Vine-Dresser* poem is available on the internet[4]. We can represent the network relationships with hypertext links on the webpage (in a **source** → **target** form). On this webpage *those nodes that appear as the target ("anchor") of the various hypertext links will be characterized by the number of direct, explicit references to them called link strength.*

On the constructed webpage, the following relationships are represented by explicit hypertext links:

- The poetic text contains hypertext links to the text sentences.
- The communication units or commentaries include hypertext links to the co-reference indices.
- The co-reference indices provide hypertext links to their connected Hungarian and English keywords.
- The communication units or commentaries contain hypertext links to the keywords.
- The theme-rheme relationships in the text sentences are represented by hypertext links to their targets (in general they are themes) in the form of text sentences.

[4] https://bodaistvan.hu/callimachus/texts/Fust_Milan-Szolomuves.html (2024-02-12)

On the compiled webpage the theme-rheme relationships in the text sentences were shown in the **"Comments"** section after the text sentences. Let us observe it, for example, in the case of the first text sentence[K01] that we presented below the "Great Bear"[i01] English keyword on the webpage (Figure 2).

Figure 2: The first text sentence below the "Great Bear" keyword on the webpage

In the next row which follows the first communication unit[k01] of the first text sentence[K01] ("a Medve ragyog [az őszi égbolton(i13)[i02]]") there is an explanatory note about the Great Bear constellation (in Hungarian; omitted in Figure 2), and after it we can find the theme → rheme relationships: where the **source** of the link will be indicated by the "Nagy Medve" Hungarian keyword, and the **target** of the link will correspond to the "égbolt" Hungarian keyword. Then we provide two hypertext links (⇒ ⇒) which point directly to the second text sentence[K02] ("S lejjebb [az égbolton(i13)[i02]] lassan [...] leúszik a Hattyú."), and to the third text sentence[K03] ("Alant sötétül a kékség(i13)[i02]..."), respectively. These text sentences are in a theme-rheme relationship with the first text sentence through the "égbolt"(i13)[i02] keyword.

On the constructed webpage, the textual network structure represented by the hypertext links as relationships can be characterized by *the number of references pointing to the node, that is, by the link strength of the nodes.* From a network science perspective, it is particularly interesting how many nodes can be found with a given link strength (that is, the strength frequency). The investigation of this question in the case of *The Vine-Dresser* poem by Milán Füst has resulted in the following frequency distribution (Table 8).

In the left column of the table, the *Link strength* of each node is presented which means the number of references to nodes. In the second column of the table, the *Strength frequency* of the nodes with a certain Link strength can be seen, for example, there are 3 nodes in the established network which have 4 hypertext links

Table 8: Theme-rheme relationships in the poem *The Vine-Dresser* by Milán Füst

Link strength ('x'axis)	**Strength frequency ('y'axis)**	**Estimated value (c=42.338, γ=1.805, Δ/n≈0.268, $iter_n$=1050**
1	42	42.34
2	14	12.12
3	6	5.83
4	3	3.47
5	1	2.32
6	1	1.67
7	0	1.26
8	0	0.99
9	0	0.80
10	0	0.66
11	0	0.56
12	1	0.48

pointing to them[5].

In the third column of the table *(Estimated value)* we presented those calculated frequency values which we obtained supposing a *scale-free network* and a *power function* with the following formula:

$$c * x^{-\gamma} \tag{1}$$

It was fitted to the frequency values of the nodes [1]. From the table, it can be noticed that besides the given parameters of the power function (c $\approxeq$ 42.338, $\gamma \approxeq 1.805$), we received a relatively good fit to the values (the standard deviation of the values is Δ/n $\approxeq$ 0.268 where n=12). It is an additional contribution to the general statement that scale-free networks appear almost everywhere in the world surrounding us – in our case in the textual network of a poem by Milán Füst.

6 Summary, conclusions, and further development opportunities

In our paper we presented the *parallel co-reference analysis* of *The Vine-Dresser* poem by Milán Füst and its English translation. It enabled us to explore the textual world of the poem based on textology and to compare the two texts – Hungarian and English – using formal linguistic tools. Though the English translation of the poem

[5]Note that in the building of the network we took into consideration *together* the Hungarian and the English text of the poem which essentially duplicated the frequency of each node. We modified the frequency distribution of the network corresponding to this. In reality, there were 4 nodes with degree 7, and 2 nodes with degree 8, and taking their average, we obtained 4+6/2=3 value for the frequency of the nodes with degree 4. On the webpage, of course, we also presented the numbers of the real frequency distribution.

is slightly different from the original Hungarian text at some points, neither the number of, nor the textological aspects of these differences seem to be important. On the whole, we can conclude that the co-reference analysis devised and elaborated by János S. Petőfi can be very efficiently applied in a polyglot environment.

The hypertext implementation of the co-reference analysis of the poetic text has plenty of practical advantages, for example, this analysis and its results can be transparent, accessible to anyone and can be checked as well. In addition, the web page also provides additional aspects for the analysis. Considering the poetic text as a network, in our paper, we investigated whether the *scale-free feature* is relevant in the textological environment as well – as we can experience it in a number of other fields of reality. The results are promising, but note that in the building of the network we presented primarily the textological characteristics of the analysed text and did not take into account e.g. the linguistic competence possessed by the Hungarian and the English native readers of the poem. It offers almost an unlimited number of opportunities for the extension of the created hypertext structure, and makes possible for us to further analyse the extended textual network using network science tools.

References

[1] Barabási, A.-L. and Frangos, J. *Linked: The New Science of Networks*. Perseus Publishing, Cambridge, MA, 2002. ISBN: 978-0738206677.

[2] Baranyi, P. and Csapó, A. Definition and synergies of cognitive infocommunications. *Acta Polytechnica Hungarica*, 9(1):67–83, 2012. URL: http://www.uni-obuda.hu/journal/Baranyi_Csapo_33.pdf.

[3] Baranyi, P., Csapó, A., and Sallai, G. *Cognitive Infocommunications (CogInfoCom)*. Springer, Berlin, Heidelberg, 2015. DOI: 10.1007/978-3-319-19608-4.

[4] Boda, I. K. and Porkoláb, J. *Koreferenciális kifejezések és koreferencia-relációk. Példaszöveg: Szent János Apostol Jelenéseinek könyve, 21:9-23. Az új Jeruzsálem. [Co-referential Phrases and Co-referential Relations. Example Text: Revelation 21:9-23. The New Jerusalem.]*. In *Officina Textologica 2*, pages 32–56. University of Debrecen, 1998.

[5] Boda, I. K. and Porkoláb, J. *A korreferencia kérdései a számítógépes szövegfeldolgozás szempontjából [Issues of Co-reference From the Aspect of Computer-Aided Text Processing]*. In *Officina Textologica 4*, pages 150–180. University of Debrecen, 2000.

[6] Boda, I. K. and Porkoláb, J. Co-reference analysis and the structure of natural language texts. In Andor, J., Benkes, Z., and A., B., editors, *Szöveg az egész világ — Petőfi Sándor János 70. születésnapjára*, pages 81–100. Tinta

Publishing, 2002. URL: https://m2.mtmt.hu/gui2/?mode=browse&params=publication;2380442.

[7] Boda, I. K. and Porkoláb, J. *Téma-réma kapcsolatok vizsgálata egy kiválasztott versszövegben korreferencia-elemzés segítségével [Investigation of the Theme-Rheme Relationships in a Selected Poetic Text using Co-reference Analysis]*. In *Officina Textologica 7*, pages 93–112. University of Debrecen, 2002. URL: https://m2.mtmt.hu/api/publication/2380436.

[8] Boda, I. K. and Porkoláb, J. *Egy angol vers és magyar fordításainak összevetése a korreferencialitás szempontjából [Comparison of an English Poem and Its Hungarian Translations from the Aspect of Co-reference]*. In *Officina Textologica 12*, pages 83–99. University of Debrecen, 2005. URL: https://mnytud.arts.unideb.hu/ot/12/ot12_5boda.pdf.

[9] Boda, I. K. and Porkoláb, J. *Füst Milán Köd előttem, köd utánam... című versének és angol fordításának korreferenciális elemzése [Co-reference Analysis of the poem "Köd előttem, köd utánam..."by Milán Füst and Its English Translation]*. In *Officina Textologica 21*, pages 31–42. University of Debrecen, 2020. URL: https://mnytud.arts.unideb.hu/ot/21/boda-porkolab.pdf.

[10] Boda, I. K. and Tóth, E. English language learning by visualizing the literary content of a knowledge base in the three-dimensional space. *Annales Mathematicae et Informaticae*, 53:45–59, 2021. DOI: 10.33039/ami.2021.04.003.

[11] Boda, I. K., Tóth, E., and Nagy, L. T. Improving a bilingual learning material in the three-dimensional space using Google Translate. In *Proceedings of the 13th IEEE International Conference on Cognitive Infocommunications*, pages 93–98. IEEE, 2022. DOI: 10.1109/CogInfoCom55841.2022.10081827.

[12] Füst, M. *25 Poems – 25 vers (Translated by István Tótfalusi)*. Maecenas, Budapest, 1990. ISBN: 9637425233.

[13] Herrmann, J. *SH atlasz — Csillagászat [SH Map — Astrology]*. Springer Hungarica, Budapest, 1992.

[14] MaxWhere VR Even more. URL: http://www.maxwhere.com/, Accessed: (2023-06-30).

[15] Petőfi, J. S. *Egy poliglott szövegnyelvészeti-szövegtani kutatóprogram [A Polyglot Linguistic and Textological Research Project]*. In *Officina Textologica 1*. University of Debrecen, 1997. URL: https://mek.oszk.hu/01700/01777/.

[16] Petőfi, J. S. *Korreferenciális elemek és korreferenciarelációk. Példaszöveg: Mt. 9,9-13. Máté meghívása. [Co-referential Elements and Co-referential Relations. Example Text: Matthew 9:9-13 Matthew's Invitation.]*. In *Officina Textologica 2*, pages 15–31. University of Debrecen, 1998. URL: https://mek.oszk.hu/01700/01757/01757.pdf.

[17] Petőfi, J. S. *A szöveg mint komplex jel. Bevezetés a szemiotikai-textológiai szövegszemléletbe. [Text as a Complex Sign. An Introduction to the Semiotic Textological Approach.].* Akadémiai Kiadó, Budapest, 2004. ISBN: 9630581264.

Acta Cybernetica **27** (2025) 83–91.

Hungarian Sentence Analysis Learning Application with Transformer Models

Noémi Evelin Tóth[a], Beatrix Oszkó[bcd], and Zijian Győző Yang[be]

Abstract

The purpose of our research is to present a project in which we started to develop an educational support tool that helps primary and high school students to use the correct techniques of sentence analysis based on the rules of Hungarian grammar taught in school. The aim was to create an application called LMEZZ that would help students of the Hungarian education system to practise tasks related to native language lessons. In this way, we expect them to have a more accurate understanding of the grammar rules. The application allows them to learn in the comfort of their own homes by providing immediate and accurate feedback on the solutions to various tasks. Natural language processing has made spectacular progress with the application of neural network technology, especially the contextual transformer model. In our research, Hungarian transformer-based BERT models were trained for our sentence analyser task. The results showed that the transformer models were much more condescending than the previously trained convolutional neural network based SpaCy models. This allowed us to increase the reliability of our software.

Keywords: learning application, Hungarian grammar, sentence analysis, SpaCy, transformer models, BERT

1 Background

These days, there is a growing demand for self-studying. With the current state of technology, learning is increasingly accessible through mobile phones, tablets and laptops. Young people are familiar with this type of technology and use it daily. There are a lot of applications targeted at learning, such as Duolingo[1], Kahoot[2],

[a]Eszterházy Károly Catholic University, Eger, Hungary, E-mail: noemitth.10@gmail.com, ORCID: 0009-0006-4919-4338
[b]HUN-REN Hungarian Research Centre for Linguistics, Budapest, Hungary
[c]University of Novi Sad, Serbia
[d]E-mail: oszko.beatrix@nytud.hun-ren.hu, ORCID: 0000-0002-0169-4505
[e]E-mail: yang.zijian.gyozo@nytud.hun-ren.hu, ORCID: 0000-0001-9955-860X
[1]https://www.duolingo.com/
[2]https://kahoot.com/

DOI: 10.14232/actacyb.307878

Mateking[3] and many others. With this research, we wanted to find a good use of the results of computational linguistics and help Hungarian students learn Hungarian grammar, especially sentence analysis. Our main target group was primary and secondary school students. The rules of Hungarian sentence analysis are often not self-explanatory, and there is little time to practice them in class. Our idea was to provide a tool that would allow them to analyse any sentence in real time.

Thanks to the development of computational linguistics, the solution for many language and communication problems can now be automated. Therefore, to avoid having to constantly verify sentences and their associated handwritten analysis by hand, the central theme of the research - in addition to application development - is the teaching and testing of linguistic analytical models to analyse raw sentences.

In the early stage of development, we only used SpaCy to train two models and we compared the results [8]. Dependency analysis was used as the basis for the preparation of the source material, as it was most similar to the school analysis. We distinguished the two models based on the label set we defined. In the case of the smaller model, we were only interested in the most important and basic labels, while in the case of the extended model, we also covered the analysis of different types of adjectives and indicators. In the current research, we further developed our application with the new generation deep contextual transformer language models.

2 Rules of sentence analysis

In order to develop the application and teach several neural network models, we first must consider how sentence analysis is taught in school. The relationship between linguistics and the teaching of Hungarian grammar is not always clear and consistent [3]. There is a significant difference and gap between the methods of scientific linguistics and the material taught. Scientific linguistics is always slightly ahead of school grammar, as the latter always tries to teach theories and models that have already been proven. In today's modern syntax, generative grammar does not categorise linguistic structures and their functions, but promotes applicable knowledge and critical and analytical thinking. In its model, of course, regularities and ways of describing and defining sentences are present, but it focuses on linguistic skills and competence. In contrast, school grammar is built on the traditional levels of language, so that knowledge of sentences can be acquired through the knowledge of phonemes, morphemes, lexemes and synagems [10].

School grammar has a dependency approach, the aim is to establish the relationship between the syntagmas or words that make up the sentence. The order of the words is rather loose, given the characteristics of the Hungarian language, so it is not dealt with in school; the dependencies form the hierarchy of the words and thus ensure the meaning of the sentence. Sentences can be classified in many ways, for example according to their structure or logical quality. In this paper, we will look in detail at the structure and analysis of simple sentences. They consist of a single clause, i.e. a single statement. Sentences can be further broken down into

[3] https://www.mateking.hu/

word structures, which are formed by the grammatical combination of two words of a basic word type that are closely related. The subject and predicate form such a syntagm. They are the main parts of the sentence and form the grammatical, semantic and logical core of the sentence. In addition, the analysis usually takes into account other elements that are extensions of the sentence. Extensions are the subject, adverbs and adverbials.

To understand this, take a look at a real example. The sentence is the following in Hungarian: *A hatalmas jegesmedve az Északi-sarkon él.* Which means: *the giant polar bear lives in the Arctic.* The predicate of the sentence is '*lives*', and the subject is the '*polar bear*' itself. The word '*giant*' is an indicator of quality, and the word '*in the Arctic*' is a locative part of the sentence. It locates where the polar bear lives.

In computer linguistics, dependency analysis [1] has been used to represent sentence structure. The approach of traditional Hungarian grammar is vastly similar to this. The sentence structure is represented as a tree and the starting point is always the predicate. An important difference, however, is that while dependency analysis works with tokens, the grammar targeted in this research uses syntactic words. A syntactic word can sometimes consist of several tokens, but traditional analysis does not establish any further relationship between the individual elements, they simply appear as a node of several words in the tree. Another important difference is that the traditional analysis ignores certain words. These are typically function words: article words, conjunctions, participles, etc. School grammar does not take punctuation into account either, but in computer analysis these are also present as separate tokens. Later, these tokens will have their own label, different from the ones we discussed in this section.

3 Methods

Natural language processing is a branch of artificial intelligence based on linguistic research.The goal is to reduce the gap between the computer and the human as much as possible, so that the computer can read, interpret and process human language [4]. The first problem to be solved by using natural language processing tools was to translate a text into another language. To do this, the computer must be able to understand the rules, morphology and syntax. The latter requires knowledge of the semantics and vocabulary of the language. Today, machine translation is only a small part of computational linguistics and can be found in many different areas of life, with intelligent assistants and chatbots, it powers search engines and spell checkers, and there are now many people involved in computer processing of various textual data and its use in other disciplines.

We can say that there are different types of neural networks that are optimised for different data. As we have already mentioned, when we started developing LMEZZ, we only used HuSpaCy [7] to train two models and compared them. At that time SpaCy used convolutional neural networks to achieve its goals [6]. It was optimised for industrial use. However, as we will see later in the article, we reached certain limits with the convolutional neural networks. In this paper, our goal was

to try to train transformer models to get more reliable results with our previously created label set.

Before transformer models, recurrent neural networks were used to processing texts [9]. The problem with these types of models is that they work sequentially and can only analyse one word at a time, in order. But in human languages, word order is a big part of the meaning of the text. For this reason, recurrent neural networks are difficult to train. To solve this problem, the researchers developed the first transformer model in 2017 [11], which was originally designed for translation. It was a model that could scale up to a huge dataset. Transformer models are based on three main ideas: positional encoding, attention, and self-attention. With positional encoding, instead of looking at words sequentially, we store information about word order so that the model learns the meaning of word order directly from the data. Attention allows the model to look at each word in the original sentence when making a decision about how to translate a word in the output sentence. Self-attention allows a model to understand a word in the context of the words around it.

4 Models

At the start of this research, we collected and analysed sentences by hand for the training and testing. Most of the sentences were collected from a Hungarian grammar textbook, called *Magyar nyelv a középiskolások számára 9.* written by Adrienne Fráter. Other sentences were chosen from textbooks for secondary school students, written by Ágnes Szabó Antalné and Judit Raátz. We also used examples from the Grammarly Practice Book.[4] The corpus consisted of 268 sentences at the beginning. We used 82% of the dataset for training and the remaining part for testing. We defined a set of labels. These labels and their explanation are shown in Table 1. Then we trained two HuSpaCy models. The models had some problems identifying labels, which were not as common in the corpus as predicates or subjects. In addition to collecting more sentences for our corpus, we wanted to improve our application with this research in order to find a better model with more reliable results for each label.

In recent years, natural language processing tasks can be solved with high performance by fine-tuning a pre-trained transformer language model. One of the most popular transformer based language model is BERT. BERT (Bidirectional Encoder Representations from Transformer) is defined as a multi-level, bidirectional transformer encoder architecture [2]. The BERT model is pre-trained on two language modeling tasks: word masking and next sentence prediction. In the recent years, two state of art BERT models have been trained for Hungarian: huBERT [5] (BERT base model – 110 million parameter) and PULI BERT-Large [12] (BERT large model – 345 million parameter).

[4] https://gepeskonyv.btk.elte.hu/adatok/Magyar/31Lakatos/Digi_TK_v2/Gyakorlokonyv.html

Table 1: Label set we used to train the models

Label	Meaning of the label
ROOT	Predicate
A	Subject
T	Object
H	Adverbial
J	Indicator
P	Element of a multi-word group of the sentence
X	Not analysed part of the sentence

We solved this sentence analysis problem as a token classification task. To fine-tune the Hungarian BERT models, we used the code provided by Hugging Face.[5]

5 Results

In Table 2, you can see the results of the fine-tuned transformer models compared with the results of one of the HuSpaCy models.

Table 2: F-Score results

	HuSpaCy	huBERT	PULI BERT-Large
Predicate (R)	94.12%	100%	100%
Subject (A)	73.91%	93.02%	90.48%
Object (T)	86.75%	100%	100%
Adverbial (H)	78.87%	96.15%	96.15 %
Indicator (J)	76.92%	96.97%	78.57%
P	58.33%	86.49%	94.44%
X	95.96%	100%	100%

In the case of the models we have previously produced, we found that the label set was too large for the size for the corpus we used, resulting in too many rare labels. Therefore, in this comparison, we only use the results of the small model, which did not take into account the different types of adverbs and indicators. Finally, this model was implemented for the first time in the application. The code of the application itself can be found here[6].

The results of the contextual transformer models turned out far more condescending than the previously trained models. The new models predicted with 100%

[5] https://github.com/huggingface/transformers
[6] https://github.com/noemitth10/Learning-App

accuracy the labels of the predicate, the object and the X. We also achieved results above 90% for the subject and adverbs. However, we still need to achieve some improvement with the indicators. The PULI BERT-Large model only achieved a two percent improvement over the previous model. This result is due to the fact that only a very few test sentences contained any indicator type. The same applies to words marked with a P label.

6 The Application

LMEZZ is a web application built with React and Javascript. Based on our target audience it has a user-friendly interface and a colourful design. For this phase of the development, our main goal was to make it responsible, besides teaching the new transformer models. This way, students can easily use the application from their phones and tablets. Most of the services on the site can be used by a registered user. If someone does not have an account, they can create one using the Register option by entering their email address, password, name and other details. After logging in, the *Elemezz!* option becomes available. By clicking on it, the site redirects the user to the model that can analyse any given sentence. After entering the sentence, the user has to click on the *Kész* button. The application will return the analysed sentence in a minute. Figure 1 shows an example of how this works. The example is the following: *Peti könyvet olvas a verandán.* Which means *Peti reads a book on the porch.* Peti is the subject of the sentence, the könyv is the object. He reads the book, so olvas is the predicate. The porch, is the adverbial of the sentence, which gives us information about the location, where Peti reads the book. Other parts of the sentences are labeled with X, the app will simply not analyze those parts. The newly implemented transformer model is running on a different server, the application just calls the API when a certain sentence is given.

7 Conclusion

In summary, we can say that transformer models are much better suited to the problem of analysing sentences. They have reached a more reliable state when it comes to analysing new sentences. This is a really important factor in teaching. We need to avoid the possibility of giving students incorrect information. However, they still have shortcomings due to the low number of test data. In the future, we need to increase the size of the corpus. In parallel with this research, we have already collected 1000 new raw sentences that need to be annotated. After that we can retrain the transformer models to further improve the results. In the next step we can test again the extended label set with the newly collected data to see where the model needs further improvements.

Figure 1: Screenshot of the application in mobile view

References

[1] De Marneffe, M.-C. and Nivre, J. Dependency grammar. *Annual Review of Linguistics*, 5:197–218, 2019. DOI: 10.1146/annurev-linguistics-011718-011842.

[2] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. DOI: 10.18653/v1%2FN19-1423.

[3] Gábor, T. N. A magyar nyelv leírása és iskolai oktatása (the description and education of the hungarian language), 2002. URL: https://mta.hu/data/dokumentumok/i_osztaly/1_Eloadasok_tara/Magyar_nyelv_es_kutatasa_20020502/TolcsvaiN_leiras_oktatas_20020502.pdf.

[4] Nadkarni, P. M., Ohno-Machado, L., and Chapman, W. W. Natural language processing: An introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011. DOI: 10.1136/amiajnl-2011-000464.

[5] Nemeskey, D. M. Introducing huBERT. In *XVII. Hungarian Computational Linguistics Conferences*, pages 3–14, Szeged, Hungary, 2021. Institute of Informatics, University of Szeged.

[6] Orosz, G., Szabó, G., Berkecz, P., Szántó, Z., and Farkas, R. Advancing Hungarian text processing with HuSpaCy: Efficient and accurate NLP pipelines. In Ekštein, K., Pártl, F., and Konopík, M., editors, *Text, Speech, and Dialogue*, pages 58–69, Cham, 2023. Springer Nature Switzerland. DOI: 10.1007/978-3-031-40498-6_6.

[7] Orosz, G., Szántó, Z., Berkecz, P., Szabó, G., and Farkas, R. HuSpaCy: an industrial-strength Hungarian natural language processing toolkit. In *XVIII. Hungarian Computational Linguistics Conferences*, pages 59–73, 2022. URL: https://arxiv.org/abs/2201.01956.

[8] Oszkó, B., Tóth, N. E., and Yang, Z. G. Az általános és középiskolai magyar nyelvtan tananyag elsajátítását segítő alkalmazás (Supporting application for learning Hungarian grammar in elementary and secondary schools). In *A digitális oktatás nyelvi dimenziói: Válogatás a PeLiKon2020 oktatásnyelvészeti konferencia kerekasztal-beszélgetéseiből és előadásaiból (Linguistic dimensions of digital education. Selected papers of lectures and roundtable discussions of the PeLiKOn 2020 conference)*, pages 145–157, Eger, Hungary, 2022. Eszterházy Károly Catholic University. URL: http://publikacio.uni-eszterhazy.hu/id/eprint/7566.

[9] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics. DOI: 10.18653/v1/N18-1202.

[10] Szabó, V. A magyar mondat modelljei a nyelvtanoktatásban (The models of the Hungarian sentence in grammar education), 2010. Manuscript, University of Pécs.

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[12] Yang, Z. G., Dodé, R., Ferenczi, G., Héja, E., Jelencsik-Mátyus, K., Kőrös, d., Laki, L. J., Ligeti-Nagy, N., Vadász, N., and Váradi, T. Jönnek a nagyok! BERT-Large, GPT-2 és GPT-3 nyelvmodellek magyar nyelvre (The Heavy

Guys are Coming! BERT-Large, GPT-2 and GPT-3 Language Models for Hungarian). In *XIX. Hungarian Computational Linguistics Conferences (MSZNY 2023)*, pages 247–262, Szeged, Hungary, 2023. Institute of Informatics, University of Szeged. URL: https://acta.bibl.u-szeged.hu/78417/.

Acta Cybernetica **27** (2025) 93–112.

Scalix Mix Network*

Ádám Vécsi[ab] and Attila Pethő[ac]

Abstract

Mix networks have now advanced to a level where they can compete in the domain of low-latency anonymous communication, owing to their "strong" anonymity design advantage. However, a few bottlenecks still exist, primarily because users are required to select the complete message path. This characteristic of mix networks hinders the implementation of load balancing and necessitates the use of an extensive shared database. To address this issue, we introduce Scalix, which presents a new topology featuring load balancers, modified path selection, and a mix package based on identity-based encryption and attribute-based encryption. Additionally, Scalix can serve as an anonymous return channel with minimal modifications.

Keywords: identity-based cryptography, attribute-based cryptography, mix network, anonymity

1 Introduction

When we talk about secure communication, we often mean encrypting messages with a cryptographic method to protect sensitive information. But communication also involves metadata, which can be valuable to observers. Metadata is information about the communication itself, such as who sent it, when it was sent, and where it was sent from. In this work, we focus on protecting the identities of the parties involved in communication by providing anonymity through a mix network.

The most well-known tool for anonymous communication is Tor [10], which uses onion routing to provide low-latency anonymous communication. However, this system only provides anonymity when the adversary can only observe a part of the system and not the entire network. If the messages go through servers that are not observed, the communication will be anonymous. VPN services are another popular solution, but they provide only superficial anonymity that hides the user's IP address, geolocation, and identity from requested sites and the ISP.

*The research was supported by the 2018-1.2.1-NKP-2018-00004 Security Enhancing Technologies for the Internet of Things project.

[a]Department of Computer Science, Faculty of Informatics, University of Debrecen, Hungary
[b]E-mail: vecsi.adam@inf.unideb.hu, ORCID: 0009-0003-5813-6111
[c]E-mail: petho.attila@inf.unideb.hu, ORCID: 0000-0002-9764-1570

DOI: 10.14232/actacyb.307671

For anonymity against a strong adversary that can observe the entire network, mix-based architectures are the best choice. Such architectures are useful in applications like voting, exam and assessment systems [14, 18, 13]. A mix network is a system that includes multiple stages of mixes, where every stage receives multiple messages, performs some cryptographic transformation for each message, and permutes them. After every mix, tracking the path of the messages gets more complex, achieving untraceability. However, mix networks used to have a trade off of high latency in communication. This issue seems to be improving rapidly thanks to Loopix [17], which is built on the Sphinx [7] mix format and can achieve low latency in communication, narrowing the delay to milliseconds.

Despite the promising benchmarks for Loopix, we believe there are two bottlenecks of the protocol. The first aspect involves load balancing among the nodes in each stage, which relies on theoretical random generation. This approach can lead to the possibility of an overloaded node if it is selected more frequently, or an underloaded node if it is chosen less frequently. The system lacks control over this outcome. The second is that the sender must pick the full path of the message, which requires a shared database with information about every mix node and could become costly to maintain with many mix nodes.

Our protocol addresses these issues by applying identity-based cryptography (IBC) and attribute-based cryptography (ABC) methods. In IBC, the public key is a known identifier string of an entity, such as an email address or phone number. ABC holds similar novelties but with even more flexible keys, allowing fine-grained cryptographic access control. Despite using IBC, our protocol allows the use-case as an anonymous return channel, enabling senders to communicate with a receiver in a way that the receiver will not know the sender's identity but can still reply to the messages. For more details, we refer to Golle and Jakobsson [11].

Our architecture aims to provide low-latency anonymous communication for users against adversaries who can monitor the entire network. We also aim to improve efficient scalability with reduced data sharing and avoid possible bottlenecks by implementing load balancers in the system. Our goal is to keep all the anonymity properties of Loopix. The sender-receiver third-party unlinkability ensures that it is impossible for an adversary to connect the honest sender of a message to its honest receiver. Sender online unobservability means that an adversary can't decide if an online honest sender is communicating with any receiver or not. Meanwhile, receiver unobservability means the inability of an adversary to decide if any sender is communicating with a specific online or offline honest receiver.

The rest of the paper is organized as follows. Section 2 introduces the core building blocks of our system, which are identity-based cryptography and attribute-based cryptography, and presents the main idea of mix networks. The related work is found in Section 3, which mainly focuses on Loopix, a mix network that provided a good foundation for our system. Section 4 presents the details of our construction, and finally, Section 5 concludes the paper.

2 Preliminaries

2.1 Mix network

Mix networks are designed to provide strong anonymity for communicating parties by performing multistage cryptographic transformations and permutations on messages called mixing. This mix operation changes the appearance of messages and their order of transmission, making it difficult for even a strong adversary who can observe all communications in the network to trace messages.

The first mixnet was created by Chaum [6], using a single cascade of n mix nodes where each message was sealed multiple times, addressing each mix node and the receiver of the communication. While this system was able to provide anonymous emailing for those who prioritize anonymity, it suffered from high latency due to the required permutation on each node and the need to wait for a batch of messages before forwarding them. Additionally, the design was disadvantageous since a single malfunctioning node could cause the entire service to stop working. These shortcomings led to the emergence of other anonymity protocols like Tor [10], which provided low-latency with slightly weaker anonymity.

However, recent research has made mixnets more promising. There have been advancements in latency and fault tolerance, with different types of topologies and mix strategies. Stratified topology is the most widespread topology used in mixnet research due to its scalability and fault tolerance. As Figure 1 shows, it is organized into layers where nodes communicate with the nodes in the next layer, allowing horizontal scaling of the system if message traffic requires it. Diaz, Murdoch, and Troncoso's work [9] provides a clear analysis of mixnet topologies and suggests that stratified and restricted stratified topologies are the best choices for balancing anonymity and latency tradeoffs.

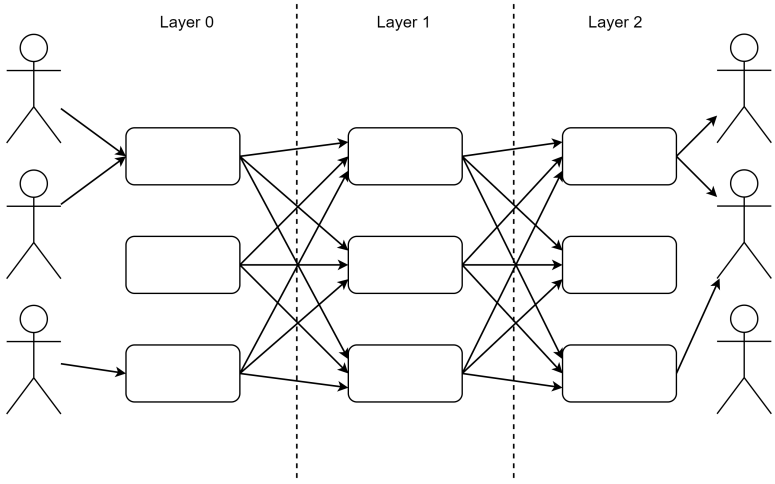


Figure 1: Stratified topology

The other area of mixnet research is mix strategies for nodes. There are various approaches to this topic, including the use of a message threshold, which requires nodes to wait until they receive a certain number of messages before forwarding them. However, this approach may not be suitable for low-latency networks as it could result in bottlenecks if nodes do not receive enough messages in time. Another approach, which had practical use too [21], is the message pool, which requires nodes to wait until they have a certain number of messages in their pool before performing permutation and then with a certain probability forwarding them to the next node. This could also lead to high latency if a message is unlucky and doesn't get forwarded. There are other ways to perform mixing [16], but the Stop and Go strategy [15] is the most promising for low-latency mix networks. In this strategy, the sender determines the amount of delay for each mix node before releasing the message, ensuring low-latency while still providing anonymity. However, if a node receives only one message, it won't strengthen anonymity. This issue can be addressed by using cover messages in the correct manner.

Overall, mix networks continue to evolve and improve, with new advancements being made in both topology and mix strategies.

2.2 Identity-based Cryptography

The original concept behind IBC was coined by Shamir in 1984 [20], who managed to build an identity-based signature scheme based on factorization. However, identity-based encryption (IBE) remained an unsolved problem until Boneh, and Franklin created their pairing-based scheme in 2001 [4], providing feasible performance for practical use.

The uniqueness of IBC lies in the fact that its public key is a string that identifies an entity in a particular domain. One may think about an email address, a username, or a phone number. This novelty directly connects with the core idea of the IBC, which was to simplify certificate management and eliminate the need for certification authorities. In the public key infrastructure scenario, public keys and user identities are bound together with certificates. With IBC, however, there is no need for such certificates since the public key corresponds directly to the user identity.

Furthermore, the public key may contain more information than just the identity of the user. This extension of the public key with domain-specific data enables a wide spectrum of advanced use cases where fine-grained access control is necessary.

Since this protocol family eliminates the need for certification authorities, it requires a trusted third party responsible for the user key generation, called the private key generator (PKG). The PKG responds to every extraction request based on the user identity, system parameters, and the master secret. Typically, in the IBC model, only the PKG knows the master secret. Otherwise, all user's private keys would be in danger. A reference implementation in C and WebAssembly can be found in [22].

2.3 Attribute-based cryptography

Attribute-based cryptography (ABC) is an extension of the idea of fuzzy identity-based cryptography [19] that provides a solution for fine-grained access control through the use of access structures. These structures allow the use of logical operators between attributes to define the authorization policy, such as (("Public Corruption Office" AND ("Knoxville" OR "San Francisco")) OR (management-level $>$ 5) OR "Name: Charlie Eppes"). ABC allows the targeting of a specific group of people with cryptographic access policies.

There are two types of ABC schemes: key-policy [12], in which the policy is associated with the user's secret keys and the ciphertexts with sets of descriptive attributes, and ciphertext-policy [3], in which the secret key is associated with sets of descriptive attributes and the ciphers with the policies. Ciphertext-policy provides the encryptor with control over who can access the data, while key-policy requires the encryptor to trust that the key-issuer has issued appropriate keys.

ABC also solves the problem of the encryptor not knowing the exact identities of all the people who should access the data. With ciphertext-policy attribute-based encryption, identities that the encryptor knows can be included, and the rest of the permitted people can be given by the attribute policy.

The key generation for ABC is similar to that of identity-based cryptography. As it can be seen on Figure 2, a trusted central authority creates decryption keys for users using a master secret key and additional descriptive information. With these keys, users can decrypt the ciphertexts if their attributes satisfy the policy associated with the ciphertext.

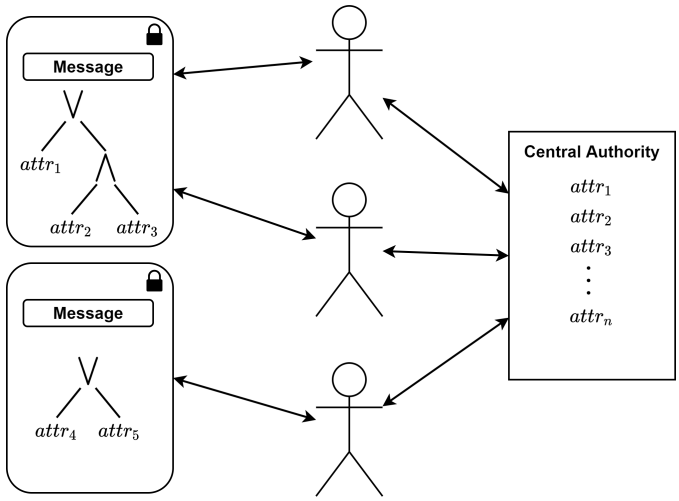


Figure 2: Ciphertext-policy attribute-based encryption

Multi-authority attribute-based encryption

Traditional attribute-based cryptography suffers from several flaws when it comes to the design of a single attribute authority. This can cause a bottleneck in terms of performance, as a single authority has to provide service to all the requests. Additionally, having a single authority creates a security risk, as one trusted party is responsible for all decryption key generation, which leads to the key escrow problem and requires a lot of trust.

To address these issues, Chase introduced a solution, multi-authority attribute-based encryption (MA-ABE) [5]. This concept allows for distributed management of attributes with independent authorities, and any number of authorities can be utilized, even if some of them are corrupt.

By increasing the number of authorities, the performance issue can be resolved since they can operate simultaneously, as depicted in Figure 3. Moreover, if the authorities are covering different parts of the attribute universe, the security will be increased, as the attributes are managed in a distributed way between authorities.

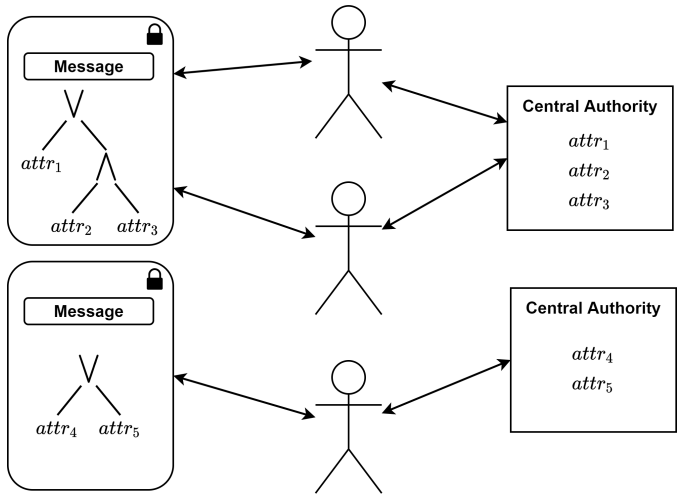


Figure 3: Multi-authority ciphertext-policy attribute-based encryption

3 Related work

3.1 Loopix

Loopix [17] is an academic mixnet that focuses on improving low-latency mix networks. It introduces several innovative features that, when combined, result in a low-latency solution for mix networks.

Loopix is built on a stratified topology, which includes a new entity called Provider. The idea of Providers comes from real-world messaging, where service

providers act as intermediaries between end-users. Similarly, all messages in Loopix go through Providers, which also act as storage for messages when end-users are offline. Providers are in a long-term relationship with users, and one provider can serve multiple users. The Provider layer is in connection with all the mix nodes from the first and the last layer to send and receive messages. The topology of Loopix can be seen in Figure 4.

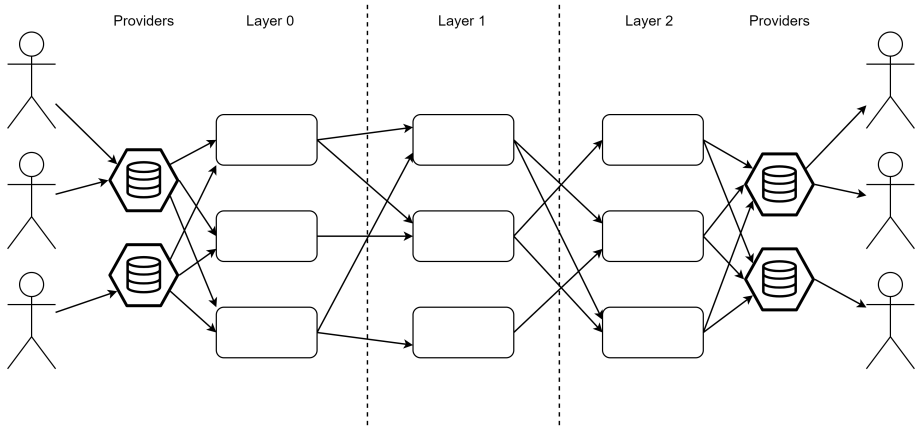


Figure 4: Loopix's topology

Mix networks' security and the latency of their messages are strongly linked, as all mix strategies depend on the number of messages flowing through the system. Loopix generates cover traffic to ensure message delivery and prevent bottlenecks of mix strategies. It includes three categories of cover traffic.

First, we need to understand how the message sending works in the system. When a user wants to send an anonymous message, they must place it into a buffer. The system periodically checks this buffer and sends any message it contains. If the buffer is empty, the system generates the first category of cover traffic and sends it to the user's Provider. This cover traffic looks like a real message, so it goes through the mixnet to a random Provider, which then discards it. With this periodic message and cover traffic, the system hides the information when the users are communicating, as real messages and this type of cover traffic are indistinguishable.

Users also send a second type of cover message that loops back to them, specifying themselves as the recipient. This category of messages provides cover for users when they receive real messages.

Finally, each mix node injects its own loop cover traffic that goes through the system to a random Provider and back to the mix node itself.

With these guaranteed messages, Loopix can use a simplified form of the Stop and Go mix strategy, which greatly contributes to low-latency communication. It is called Poisson mix, named after the fact that users and mixes send messages and cover traffic according to a Poisson process. Additionally, messages are indepen-

dently delayed using an exponential distribution. These factors make the system modelable in steady states with a Poisson distribution, allowing us to calculate the number of messages mixed together at any time.

With these innovations and the goal of achieving low-latency mix networks, Loopix has influenced multiple current mixnet solutions, such as Katzenpost [2], Nym [8], and Scalix (our system).

4 Scalix

Many mixnet protocols rely on maintaining a database of information about mix nodes, which allows message senders to select the route of their message. However, as the number of mix nodes scales, the size of the database required to store this information also increases, making it more costly to maintain. Additionally, scaling the number of users results in more queries on this database, making maintenance even more challenging and expensive. To address this issue, our model utilizes identity-based (IB) and attribute-based (AB) methods to provide a solution.

4.1 Topology

Loopix gave us many great ideas and a working low-latency mixnet. Our aim was to build upon this existing solution and extend it with our ideas. As seen in Figure 5 scalix's topology is similar to that of Loopix. However, we included load balancers (marked as $AA + LB$) between each layer of mix nodes, to guarantee the proper distribution of messages between mix nodes herewith maximizing the permeability.

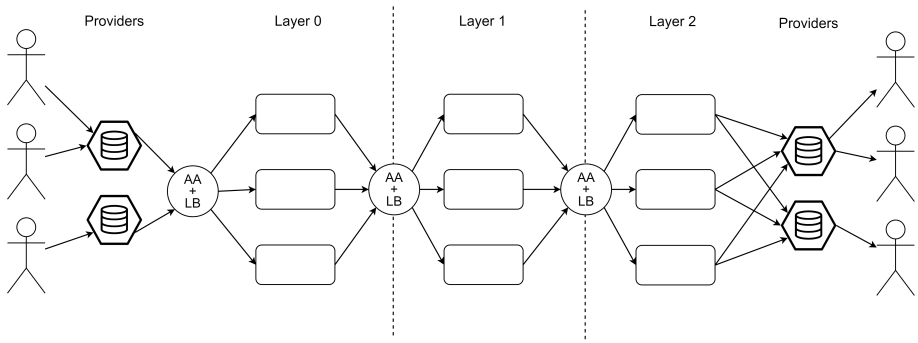


Figure 5: Scalix's topology

Attribute authority and load balancer nodes

Figure 5 depicts nodes labeled as $AA + LB$ between each mix layer, which serve two purposes in the system - attribute authority (AA) and load balancer (LB).

In our system, users target the $AA + LB$ nodes instead of individual mix nodes to determine the path through which their packets will be mixed. This ensures that packets always pass through an $AA + LB$ node before entering the mix layer, which is responsible for selecting the mix node to receive the packet based on load balancing methodologies.

The attribute authority functionality of these nodes serves a security purpose by allowing the use of decryption keys as single-use keys with minimal computation and without sharing unnecessary information with the encryptor user. To accomplish this, the attribute authority generates the single-use part of the decryptor key and sends it to the selected mix node. Consequently, even if the encryptor targets a whole layer of mix nodes with their encryption, only the mix node selected by the $AA + LB$ node can perform the mix operation. Further details on how this mechanism works are available in section 4.2.

Providers

The providers, which are the final layer of the mix network before the users, have similar functions as those defined in Loopix [17], and we have not introduced any new services to these entities.

4.2 ABE requirements

In our protocol, we utilize ABE that supports multiple authorities, constant-size ciphertext, and user revocation.

The use of multiple authorities aims to reduce the trust placed on any single authority while improving the system's overall performance.

The constant-size ciphertext is crucial for ensuring that packets are of the same size at all points in their path and indistinguishable from other packets in the mix network.

The requirement for user revocation is closely linked to the load balancer's role in selecting the mix node for a particular layer. If every mix node in a layer can perform the mix operation, the packet may be vulnerable. However, by incorporating the concept of single-use keys into user revocation, we can target an unknown node.

Encrypt to an unknown target in a group

Figure 6 illustrates how this concept works in our system. Each node has a long-term attribute, $attr_{layer}$, which specifies the layer to which it is assigned upon registration, as well as a single-use attribute, $attr_{single}$, that is active for only one decryption of a specific packet and is revoked instantly.

When a message sender creates a packet, he encrypts it with a policy that requires both attributes. This ensures that even if the packet is leaked, it cannot be decrypted because no one satisfies the $attr_{single}$ part of the policy.

Once the packet is created, it is sent to the $AA + LB$, which selects a mix node and assigns the $attr_{single}$ to it. After the mix node completes its work, the attribute is revoked.

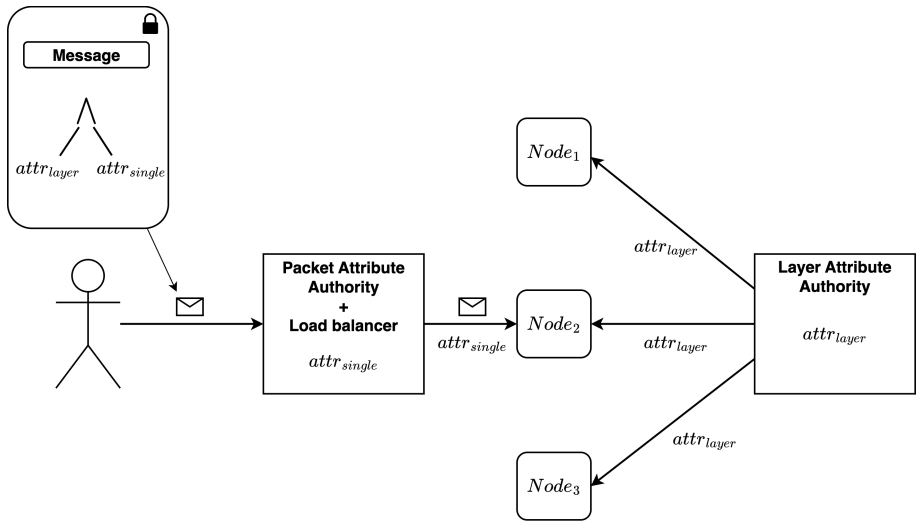


Figure 6: Encrypt to an unknown target in a group

To implement this concept in practice, we use the scheme proposed by Zhang et al. [23], which satisfies the requirements of multiple authorities, constant-size ciphertext, and user revocation.

In the scheme U denotes the set of all attributes. During the setup phase of the scheme, an injective τ encoding is chosen, which sends each of the m attributes $at \in U$ onto an element $\tau(at) = x \in \mathbb{Z}_p^*$. In the proposed use of the scheme, the attribute authorities would generate decryption keys using τ as the attribute's encoding function, since the documents are encrypted using that. Then, when a user gets revoked, they change the encoding function to τ' and create an update key, with which the ciphertexts and the decryption keys of the non revoked users can be updated.

In our case, the default τ is used only for the encryption and for the $attr_{layer}$ decryption key part generation. This way, the encryptors can encrypt targeting a specific layer, but no one can decrypt, since $attr_{single}$ is not generated using τ.

As mentioned earlier, an $AA + LB$ node decides which mix node gets the encrypted packet in our system. Since this node handles the layer's single-use attribute, which should be included in the encryption phase, it can compute to a τ' encoding function the update key, update the packet, and send out the corresponding decryption key part to the picked mix node. After this, the $AA + LB$ node should choose a different encoding function to protect the next packet from the recently selected mix node, operating with single-use keys, which means instant user revocation.

4.3 Packet

Requirements

We have set the following requirements for our packet design to fulfil.

- *Secrecy*: During the mixing process, packets should only reveal the information necessary for each layer, in order to prevent malicious mix nodes from gaining access to extra information. This helps to ensure anonymity and protection against malicious nodes.
- *Indistinguishability*: To prevent adversaries from tracking the route of messages, packets in the network should appear random and maintain a consistent length throughout their journey.
- *Infinite loop resistance*: If the number of hops is not limited, adversaries could flood the network with messages, causing significant delays or even preventing the system from operating. Therefore, limiting the path length is necessary to prevent such attacks.
- *Support for flags*: The packet must be designed to provide information to nodes that can aid their operation or enable additional features, such as drop flags that enable the use of cover traffic.
- *Expandability*: "The packet should also be expandable to accommodate additional information that may be required for future features, without requiring changes to the packet building method.

Notations

Since our method of encrypting to an unknown target is based on ABE, to understand the construction of the packet, first we will introduce the encryption function and its notations from [23].

To encrypt a message M the sender sets an access policy, which in our case as introduced in Section 4.2 is the necessity of the layer attribute and the single-use attribute. The layer attribute should be identical for every node, with distinct value for each layer. On the other hand, the single-use attribute must be a different attribute for each layer. This way one layer's $AA + LB$ node can only generate the decryption key part for the intended layer (for the single-use attribute of that layer).

With encryption, we always target one layer. In our case, two AAs handle the decryption key. Let I_{AA} be the set of these two AAs, and $aid \in I_{AA}$ represent any of the AAs. Additionally, let $\mathbb{G}$ and $\mathbb{G}_T$ denote two multiplicative groups with the same prime order p. The value $v_{aid} \in \mathbb{G}_T$ is introduced in the global public key of the protocol. $A_{aid} \in \mathbb{Z}_p^*$ are chosen randomly by the central authority, and the set of these values forms the global master secret key. AA aid randomly selects $B_{aid}, Y_{aid} \in \mathbb{Z}_p^*$, which serve as the local master secret key for AA aid. (In the referred paper, $A_{aid}, B_{aid}, Y_{aid}$ are denoted as $\alpha_{aid}, \beta_{aid}, \gamma_{aid}$, respectively.

However, we have changed the notations here to avoid any potential confusion or collisions.) n_{aid} denotes the upper bound of the size of allowed decryption policy for AA aid, and in this setup, D_{aid} should consist of only one element from $\mathbb{Z}_p^*$, therefore $D_{aid} = \{d_{aid,1}\}$ and $n_{aid} = 2$ in our case. Similarly, S_{aid} should also be a set with only one attribute, resulting in $s_{aid} = 1$. In addition, the notation $D_{aid,i}$ represents the set $\{d_{aid,1}, d_{aid,2}, ..., d_{aid,i}\}$. Finally, h is a generator element of $\mathbb{G}$, and $u_{aid} \in \mathbb{G}$ represents a public key component of aid.

Important to mention that the encryption supports threshold access structure too, but in our case that property is not needed. Therefore t_{aid} should be the same value as s_{aid} (which is 1 in our case).

The sender picks a random $\kappa \in \mathbb{Z}_p^*$ and can perform the encryption which's result should be the following for $aid \in I_{AA}$:

$$CT = \begin{cases} C = M \prod\limits_{aid \in I_{AA}} v_{aid}^{\kappa} \\ C_{1,aid} = h^{\kappa \cdot \frac{A_{aid}}{B_{aid}} \cdot X} \\ C_{2,aid} = u_{aid}^{-\kappa} \end{cases}$$

where

$X = (Y_{aid} + \tau(at))(Y_{aid} + d_{aid,1})$ and $at \in S_{aid}$.

The notations of the packet are the following.

Let l be the number of mix nodes that a Scalix mix message will traverse before delivered to the receiver.

Let CH_i be the ciphertext parts denoted as $C_{1,aid}$ and $C_{2,aid}$ after attribute-based encryption for the ith layer.

H_i is the mixing header for the ith layer.

Let n be the size of CH_i and m be the size of H_i for $i \in 1, 2, ..., l$.

Let $r = (l-1) \cdot n$ and $s = (l-1) \cdot m$.

Our system also uses two hash functions to produce a packet:

- $h_1 : \mathbb{G}_T \to \{0,1\}^r \{0\}^n$
- $h_2 : \mathbb{G}_T \to \{0,1\}^{l \cdot m}$

We are using identity-based encryption (IBE), multi-authority ciphertext-policy attribute-based encryption (MA-ABE) and Advanced Encryption Standard (AES) as building blocks of our system.

M denotes a plaintext for each occurrence. Let $E_{IB}(ID, M)$ denote the encryption method of the IBE scheme, where ID is a public key. $E_{AB}(ATTR, M)$ is the encryption method of the MA-ABE with constant-size ciphertexts [23], where $ATTR$ is the access policy (The constant-size ciphertext is an important property, so the packets will be indistinguishable by their size). Finally, $E_{AES}(K, M)$ denotes an AES encryption with the key K.

The most common access policy notations are in the form of L_i which is a policy that contains a specific layer and a single-use attribute. R is the receiver's identity and P_R is the receiver's provider's identity.

Building a packet

Our packets are built in an onion structure, where each layer only reveals the information necessary for the node to perform the mix operation.

The necessary data to perform a mix are the amount of delay, the address where to forward the packet and the drop flag. We made it possible with the use of ABE and XOR cipher. The packet is built from three parts, the header, the cipherheader and the body. The header holds all the mix information listed above, the cipherheader has the CH_i part of the ABE cipher and the body is the rest of the ABE cipher.

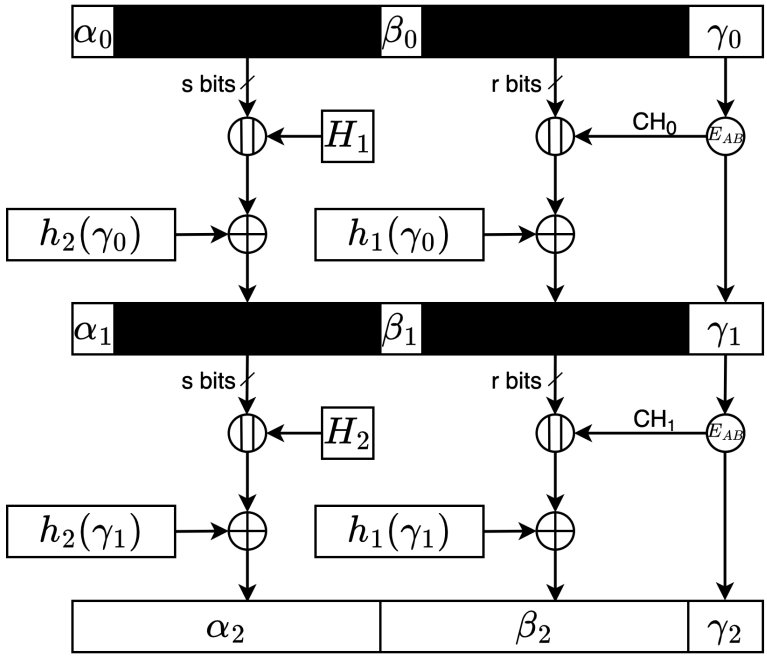


Figure 7: The construction of the Scalix packet for two mix layers

Figure 7 presents how our packet is built for the layers that it will traverse through. The core ($0th$) layer is for the receiver's provider, so it does not hold any mix information.

$\gamma_0 = E_{AES}(K, M)||E_{IB}(R, K)||E_{IB}(P_R, R)$
$\beta_0 = \{0,1\}^{l \cdot n}$ random bits
$\alpha_0 = \{0,1\}^{(l \cdot m)-1}$ random bits, and the last bit is the drop flag.
For each $i \in \{1, 2, ..., l\}$:
$\gamma_i = E_{AB}(L_i, \gamma_{i-1}) \setminus CH_{i-1}$
$\beta_i = \beta_{i-1[n..l \cdot n]}||CH_{i-1} \oplus h_1(\gamma_{i-1})$

$\alpha_i = \alpha_{i-1[m..l \cdot m]} || H_i \oplus h_2(\gamma_{i-1})$

How this construction keeps the information secure and reveals only for the right layer will be discussed more in Section 4.4.

The size of our packet is constant. This is possible by using constant-size ciphertext ABE, which makes the body part (γ_i) the same size every time and also the cipherheader fragments (CH_i). Since the mix information is the same for every layer, the size of it is easy to be standardized and make it constant in the system. In addition, since the number of mixes are set by the system to l, and now we know all the fragments are constant, n and m bits for CH_i and H_i respectively. β_i and α_i can be set to a constant size of $l \cdot n$ and $l \cdot m$ respectively.

The path length cannot be more than l. Since α_i is $l \cdot m$ bits long and every time a new layer is added to the packet, we cut the first m bits of the header and then we concatenate the new part, after l steps we will start to cut useful information, keeping the path lenght l, just making it unable to reach its intended destination.

Also, since the information the header holds is defined by the system and the only requirement is that it be standardized with constant size m, it also gives the option of easy expansion making it flexible for future features if needed.

4.4 Mix operation

During the mixing operation, a mix node can only extract the intended information and no more. As a result, dishonest mix nodes are unable to compromise the anonymity of the system as long as there is at least one honest mix node present.

Our system uses the Poisson mix strategy defined in the paper of Loopix [17]. The layer's extraction operation works as shown on Figure 8.

If a received packet is not of uniform length, it indicates that the previous mix node did not follow the mixing method. In such cases, the packet should be discarded.

For each $i \in \{l, l-1, ..., 1\}$, assuming the node is authorized for decryption (the attributes are correct):

The mix node finds at the end of β_i the CH_{i-1} part of the ABE cipher. Using CH_{i-1} and γ_i it can perform a decryption, receiving γ_{i-1}.

Once it obtained γ_{i-1} it should XOR β_i with $h_1(\gamma_{i-1})$. Then the result will be right shifted with n bits to throw away the used CH_{i-1} and concatenate it to n random bits, to keep it the same size. At the end of this sequence it will receive β_{i-1}.

To acquire H_i, the mix node will XOR α_i with $h_2(\gamma_{i-1})$. The last m bits of the result is H_i. After that it should right shift the result of the XOR with m bits and concatenate it to m random bits. Once its done, the result is α_{i-1}.

At the end it should just wait for the amount of the delay and after that forward the result to $AA + LB_{i-1}$.

In the case when the node is not authorized for decryption, it cannot perform the attribute-based decryption, therefore can't obtain any information. This also protects the inner layers from the current mix node.

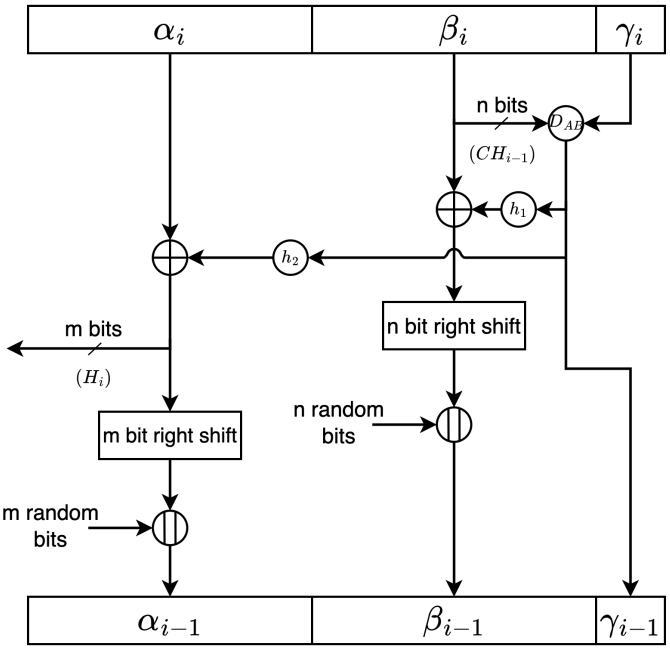


Figure 8: The mix operation of the Scalix packet

4.5 Mixnet as reusable anonymous return channel

We have mentioned that one of our goals is to support the use of our system as a reusable anonymous return channel, which allows the recipients of anonymous messages to send anonymous replies without knowing who they are replying to. This provides complete anonymity to the sender. A mix network solution for this purpose was proposed by Golle and Jakobsson in [11]. In order to offer this service in our system, we need to modify the content of the core of the packet body.

The original was: $\gamma_0 = E_{AES}(K, M)||E_{IB}(R, K)||E_{IB}(P_R, R)$

Let S be the notation of the sender's ID and P_S is the sender's provider's ID.

If the core of the body part is

$\gamma_0 = E_{AES}(K, M)||E_{IB}(R, K)||E_{IB}(P_R, R)||E_{IB}(S, K)||E_{IB}(P_S, S)||E_{AES}(K, P_S)$,

and the rest of the packet building and the mixing operation stays the same, Scalix will function as a reusable anonymous return channel.

If the receiver wants to reply, he will encrypt his message with the received K encryption key and attach $E_{IB}(S, K)||E_{IB}(P_S, S)$ to it which were included in γ_0. This way building the original part of a mix packet. γ_0 also includes the AES encryption of P_S, which has to be included in the packet header to be routed to the correct provider, since a provider gives service to multiple users, this won't break the anonymity of the sender. The rest of the construction is the same as explained

earlier.

Assuming the IBE used is secure, it is not possible to decrypt S without the provider's decryption key. When the receiver of γ_0 obtains P_S, it narrows down the set of possible senders. However, since every provider serves multiple senders and the system is designed such that each user sends messages periodically and receives cover messages, the sender can remain anonymous.

4.6 Proof of security

Given that our architecture is based on Loopix, our aim is to achieve similar security properties against the same threat model. Specifically, our goal is to prevent adversaries from compromising the anonymity of the communicating parties by attempting to link message senders with the correct receivers.

To achieve their objective, adversaries are capable of observing all traffic within the network. Some may also compromise $AA+LB$ nodes or mix nodes or providers, allowing them to take full control. Furthermore, some adversaries are able to participate as a compromised sender or receiver.

Since our mix strategy is the same as that used in Loopix, for security analysis we refer to [17]. However, our topology includes a group of nodes called $AA + LB$, which requires additional security evaluation.

In terms of packet security, we are partially following the principles outlined in Sphinx [7], which include correctness, integrity, and security. We have excluded wrap-resistance, as we believe that it is not a useful property for mix networks, since the adversary is already capable of monitoring the entire network and will therefore not perform a wrapping attack.

$AA + LB$ nodes

If the $AA + LB$ node is compromised, it can be subject to three possible attacks: forwarding messages improperly, ceasing to forward any messages, or attempting to read or modify the packet.

In the first scenario, if the $AA + LB$ node is compromised, it will not function properly as a load balancer and will select certain packets to not be forwarded in a balanced way. However, due to our usage of the Poisson mix strategy, the system can be modeled in its steady states with a Poisson distribution. This allows us to calculate the number of messages that are mixed together at any given time. As a result, mix nodes can establish a range for the number of messages that they expect to receive. If the number of received messages is not within this range, they can report the $AA + LB$ node, and the service provider can take steps to fix the malicious node.

In our current topology, if an $AA+LB$ node stops forwarding messages, it could create a bottleneck in the system since these nodes represent a single point of failure. Consequently, any messages routed through the malfunctioning node would not be delivered. However, we use multi-authority attribute-based encryption, which can

be combined with distributed load balancing techniques. This way the $AA + LB$ would act as a distributed system, eliminating the risk of a single point of failure.

In the third attack, the malicious $AA + LB$ node would act as an honest node while attempting to extract as much information from the packet as possible. However, in the normal case, this node is weaker than a malicious mix node because it cannot decrypt the attribute-based cipher and, therefore, has no means of gathering or altering information. Let's assume that the malicious $AA+LB$ node manages to obtain a decryption key, which allows it to access the information intended for that layer's mix node. With this, the node could potentially modify the delay information or the forward address of the packet. However, changing the forward address would be ineffective since it would be received by a layer that cannot decrypt the attribute-based encryption part and therefore cannot forward the packet. On the other hand, modifying or reading the delay information could allow the adversary to track the packet for one hop. Nonetheless, as proven in [17], a single honest Poisson mix provides a measure of sender-receiver unlinkability.

The $AA + LB$ node is subject to overload attacks as well. Nevertheless, our system does not restrict the $AA + LB$ node to a single server. A potential solution is to distribute the ABE protocol's parameters and keys across multiple servers and utilize a dynamic distributed cooperative load balancing algorithm. This approach can help mitigate the impact of the attack. In a cooperative load balancing algorithm, the servers work together collaboratively to balance the load. They share information and cooperate in the decision-making process to ensure an optimal distribution of workload. This can involve exchanging data about server capacities, current loads, and performance metrics [1].

Correctness

It is evident that our system works correctly if there is no adversary and all the entities are honest. That is, all the $AA+LB$ nodes produces the correct decryption key parts and forwards the packets to the mix nodes, which processes the packet correctly and finally the correct message is sent to the correct receiver.

Integrity

Our system has a predetermined maximum number of layers that a mix packet can travel through during its path, which is set during the setup. As detailed in Section 4.3, if an adversary attempts to create a longer path by adding more layers, information loss will occur in the header and cipherheader parts of the packet. This causes the length to remain at the maximum amount, but results in the message being lost and the final destination being changed with each additional layer added.

Sender-receiver third-party unlinkability, sender and receiver unobservability and mix security

The proofs of sender-receiver third-party unlinkability, the sender and receiver unobservability and mix security are based on the fact that our system is utilizing

Poisson mix and employing the same cover message strategies as described in [17]. As the proofs of these properties are already given in the paper of Loopix, we refer to the description provided there.

Message Indistinguishability

If we assume that the MA-ABE scheme used in our system is secure (meaning that an adversary without the private key cannot perform decryption), any attempt to extract useful information from a packet would be futile. This is because the information is hidden and even the number of hops traveled by the packet is concealed, since the packet size remains consistent and the bits appear random to any inspector. Moreover, since all packets are designed to have the same size, it is impossible to distinguish a real message from a cover message.

5 Conclusion

In this paper, we have introduced a mix network that incorporates practical load balancing and significantly reduces the size of the shared database of node information. We have utilized identity-based and attribute-based encryption as effective tools to achieve these objectives. These encryption methods have been particularly useful in scenarios where we need to encrypt to entities whose identity we know as well as entities whose participation in the communication process is uncertain. Our solution is also unique in this field of cryptography as it provides an identity-based mix network that can be utilized as a reusable anonymous return channel, thus bridging a gap in this cryptographic family.

References

[1] Alakeel, A. A guide to dynamic load balancing in distributed computer systems. *International Journal of Computer Science and Network Security*, 10(6):153–160, 2009. URL: http://paper.ijcsns.org/07_book/201006/20100619.pdf.

[2] Angel, Y., Danezis, G., Diaz, C., Piotrowska, A., and Stainton, D. Katzenpost Mix Network specification, 2019. URL: https://github.com/Katzenpost/docs/blob/master/specs/mixnet.rst.

[3] Bethencourt, J., Sahai, A., and Waters, B. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*. IEEE, 2007. DOI: 10.1109/sp.2007.11.

[4] Boneh, D. and Franklin, M. Identity-based encryption from the weil pairing. In *Advances in Cryptology — CRYPTO 2001*, Lecture Notes in Computer Science, page 213–229. Springer Berlin Heidelberg, 2001. DOI: 10.1007/3-540-44647-8_13.

[5] Chase, M. Multi-authority attribute based encryption. In *Theory of Cryptography*, pages 515–534. Springer Berlin Heidelberg, 2007. DOI: 10.1007/978-3-540-70936-7_28.

[6] Chaum, D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981. DOI: 10.1145/358549.358563.

[7] Danezis, G. and Goldberg, I. Sphinx: A compact and provably secure mix format. In *2009 30th IEEE Symposium on Security and Privacy*. IEEE, 2009. DOI: 10.1109/sp.2009.15.

[8] Diaz, C., Halpin, H., and Kiayias, A. The Nym network — The next generation of privacy infrastructure, 2021. Whitepaper, URL: https://nymtech.net/nym-whitepaper.pdf.

[9] Diaz, C., Murdoch, S. J., and Troncoso, C. Impact of network topology on anonymity and overhead in low-latency anonymity networks. In *Privacy Enhancing Technologies*, pages 184–201. Springer Berlin Heidelberg, 2010. DOI: 10.1007/978-3-642-14527-8_11.

[10] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The second-generation onion router, 2004. DOI: 10.21236/ada465464.

[11] Golle, P. and Jakobsson, M. Reusable anonymous return channels. In *Proceeding of the ACM Workshop on Privacy in the Electronic Society*. ACM Press, 2003. DOI: 10.1145/1005140.1005155.

[12] Goyal, V., Pandey, O., Sahai, A., and Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and communications Security*, pages 89–98. ACM Press, 2006. DOI: 10.1145/1180405.1180418.

[13] Huszti, A. A homomorphic encryption-based secure electronic voting scheme, 2011. DOI: 10.5486/pmd.2011.5142.

[14] Huszti, A. and Pethő, A. A secure electronic exam system. *Publicationes Mathematicae Debrecen*, 77:299–312, 2010. DOI: 10.5486/pmd.2010.4682.

[15] Kesdogan, D., Egner, J., and Büschkes, R. Stop-and-Go-MIXes providing probabilistic anonymity in an open system. In *Information Hiding*, Lecture Notes in Computer Science, pages 83–98. Springer Berlin Heidelberg, 1998. DOI: 10.1007/3-540-49380-8_7.

[16] Oujani, A. Tools and protocols for anonymity on the internet, 2011. URL: https://www.cse.wustl.edu/~jain/cse571-11/ftp/anonym/.

[17] Piotrowska, A. M., Hayes, J., Elahi, T., Meiser, S., and Danezis, G. The Loopix Anonymity System. In *26th USENIX Security Symposium*, pages 1199–1216, Vancouver, BC, 2017. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/piotrowska.

[18] Rjaskova, Z. Electronic voting schemes. Master's thesis, Comenius University, Bratislava, 2002.

[19] Sahai, A. and Waters, B. Fuzzy identity-based encryption. In *Advances in Cryptology — EUROCRYPT 2005*, Lecture Notes in Computer Science, pages 457–473. Springer Berlin Heidelberg, 2005. DOI: 10.1007/11426639_27.

[20] Shamir, A. Identity-based cryptosystems and signature schemes. In *Advances in Cryptography (CRYPTO 1984)*, Lecture Notes in Computer Science, page 47–53. Springer Berlin Heidelberg, 1984. DOI: 10.1007/3-540-39568-7_5.

[21] Tuckley, C., Arneson, E., Kirk, A., Crook, S., and Palfrader, P. Mixmaster. URL: http://mixmaster.sourceforge.net/.

[22] Vécsi, A., Bagossy, A., and Pethő, A. Cross-platform identity-based cryptography using WebAssembly. *Infocommunications Journal*, 11(4):31–38, 2019. DOI: 10.36244/icj.2019.4.5.

[23] Zhang, X., Wu, F., Yao, W., Wang, Z., and Wang, W. Multi-authority attribute-based encryption scheme with constant-size ciphertexts and user revocation. *Concurrency and Computation: Practice and Experience*, 31(21), 2018. DOI: 10.1002/cpe.4678.

Contents

International Conference on Applied Informatics 2023 **1**
Imre Varga and Gergely Kovásznai: Preface 3
Tibor Ásványi: Invariants and String Properties in the Analysis of the Knuth-Morris-Pratt Algorithm 5
Tibor Guzsvinecz, Judit Szűcs, and Erika Perge: How Egocentric Distance Estimation Changes in Virtual Environments by Using a Desktop Display or the Gear VR 15
Szabolcs Szilágyi: Effective Supervision of Students' Activity During Classroom Learning and Testing 43
Máté Tejfel, Dániel Lukács, and Péter Hegyi: P4 Specific Refactoring Steps . 53
István Károly Boda and Erzsébet Tóth: Co-reference, Thematic, and Network Analysis of a Selected Hungarian Poem and Its English Translation (Füst Milán: A szőlőműves / The Vine-Dresser) 67
Noémi Evelin Tóth, Beatrix Oszkó, and Zijian Győző Yang: Hungarian Sentence Analysis Learning Application with Transformer Models 83
Ádám Vécsi and Attila Pethő: Scalix Mix Network 93

ISSN 0324—721 X (Print)
ISSN 2676—993 X (Online)

Editor-in-Chief: Tibor Csendes