List Steganography Based on Syllable Patterns

David Martin, Ari Moesriami Barmawi
Informatics Engineering Graduate
School of Telkom University Bandung,
Indonesia
dm.aritonang@gmail.com, mbarmawi@melsa.net.id

Abstract—**List-based steganography or Listega is a steganography methodology with noiseless steganography paradigm, or Nostega. In this methodology, a message is embedded in the first characters of list items to create a list-cover. Listega uses filtered cover, which contains only the embedded list items. There are some rooms of improvements in Listega due to several problems, i.e. small embedding capacity and embedding failure in using several index of Latin square that it uses as mapping method. The proposed method uses syllable patterns of Indonesian language as a method of embedding message using Employee Birthday List. It also introduces an algorithm to make an unfiltered list-cover, which contains the embedded and non-embedded list items. This algorithm also makes possible to provide a grouping and a degree of sorting of list-covers. Furthermore, the results of the experiments show that the proposed method has higher capacity compares to Listega in filtered cover conditions. The use of unfiltered cover also reduces the suspicion from other parties.**

Keywords—Listega; Noiseless; steganography; syllable patterns

I. INTRODUCTION

In the era of communication, people need to share information to intended recipients. On the other hand, confidential information is vulnerable to eavesdropping by malicious parties. Therefore, efforts to secure information become necessary.

Protecting the information can be done using steganography or cryptography. Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video such that its presence cannot be detected, except by its intended recipients. Using steganography, the intended secret message does not attract attention to itself as an object of scrutiny. Likewise, Cryptography scrambles the message to conceal the information in contains [1]. Steganography is used when it is desirable to hide the message, instead of scramble it or when there are policy that ban the use of cryptography [1]. When it is necessary, both techniques can be combined to add multiple layers of security [2].

Based on the cover media used, steganography can be classified as image, video, audio, or text steganography [1], [2]. Among those cover media, text steganography is considered very challenging, because of the lack of redundancy in text [2]

Noiseless Steganography or Nostega paradigm is introduced by Desoky [3], [4], [5] which describes a paradigm for designing steganography system, which does not introduce noise to its cover, nor exploit noise as stego-carrier, as opposed to linguistic and non-linguistic steganography, which can be considered “noisy” as they either generate noise or conceal message in noise. There are several Nostega-based methodologies, such as Listega or List-based steganography, Graphstega or Graph steganography, Chestega or Chess steganography, Edustega or Education-centric steganography, etc.

List-Based Steganography or Listega [3], is one of several Nostega-based methodologies. This method takes advantage of textual list to camouflage data by exploiting textual lists of itemized data, e. g. book titles, CD titles, computer parts, etc. Listega encodes the secret messages by embedding it into the legitimate items to form a list-cover. Listega uses 4-bit slices to encode message into first characters of list-cover items. Since each character is in 8-bit ASCII representation, this method requires 2 rows of list- cover item to encode a character of message. This paper proposed a steganography method which exploits the use of syllable pattern to improve the embedding capacity of Listega.

II. RELATED WORK

As mentioned in previous section, Listega encodes the secret messages by embedding it into the legitimate items to form a list-cover. Listega architecture consists of four modules as shown in Fig.1 [3].

A. Listega Architecture

1) Domain Determination (Module 1): This module determines the domain suitable for concealing data. This module is only needed when building the listega for the first time.

2) Message Encoder (Module 2): In this module, the secret message is encoded into binary ASCII with a particular length of bits.

3) Message Camouflager (Module 3): This module consists of three sub-modules as follows:

a) Bank of textual items: A large database of textual items such as book or CD titles.

b) Selector: This module selects the database that will form the list-cover.

c) Cover generator: This module forms the list-cover, including the necessary formatting, such as header, footer, etc.

4) Communication Protocol (Module 4): This module functions to predetermine the particular Listega system, including its decoder, and covert channels among communicating parties. The covert channels itself can take form of exchanging emails, downloading files, etc.

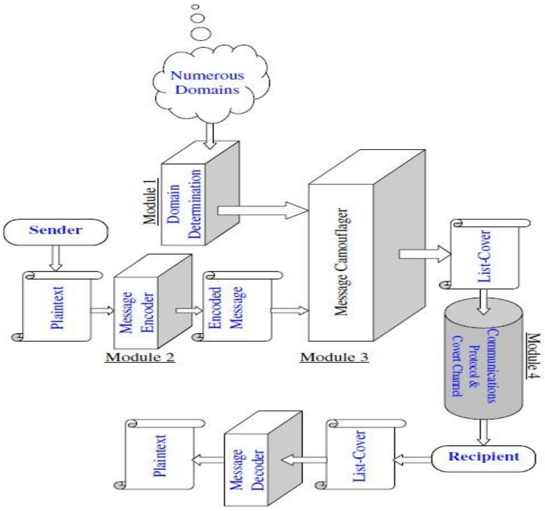


Fig. 1. Listega architecture [3]

B. *Example of Listega*

The following is an example of message encoding and list cover generating steps in Listega:

The plain text of secret message: “Stop”

Module 2 converts the message into binary ASCII and slices it into 4-bit slices:

Stop Æ01010011011101000110111101110000
0101 0011 0111 0100 0110 1111 0111 0000

The 4-bit slices are then mapped to the first characters of the list items. Listega opts to use Latin Square to add some randomness to this mapping, to prevent an adversary to correlate the pattern of the list-cover over time. The first time the encoder conceals a message, it uses the index 1 or the first row of latin square, which maps 0000 as A or Q, 0001 as B or R, 0010 as C or S, etc. For the second time, the encoder uses index 2, which maps 0000 as B or R, 0001 as C or S, and 0010 as D or T. The latin square used in Listega is shown in Fig. 2

For the example with the message “Stop”, the bits are mapped into a list item by the Message Camouflager using the row index 2 of latin square. The generated list-cover is shown in Table 2 [3].

With a large list, or a list which has list items begin with uniformly distributed first character, there might be no problem with the implementation of latin square in Listega. However, by using limited list size, there is a possibility of failure in message embedding.

Since the distribution of first character is not uniform and some 4-bit slices (1010, 1011, 1100, 1101, 1110 and 1111) are only represented by 1 character, list items which have the first character required by the mapping may be unavailable or very limited compares to other items. The failure can occur when an index requires book titles that begin with Q, X, Z, for example, which are less common than those that begin with A, I, E in Indonesian language.

By default, in original Listega, the list-cover contains only embedded list items. Optionally, the sender can mix the list items with non-embedded ones by following a particular sequence, such as read every fifth items, odd number, even number, etc.

	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9
Index →																										
1 ↓	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
3	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
4	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
5	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
6	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
7	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
8	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
9	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
10	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
11	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
12	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
13	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
14	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
15	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
16	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
17	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
18	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
19	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
20	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
21	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
22	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
23	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
24	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
25	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
26	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Fig. 2. Latin square for camouflaging 4-bit slices in Listega [3]

TABLE I. MAPPING OF 4-BIT SLICES TO THE FIRST LETTERS OF LIST ITEMS

4-bit Slice	**First Letters of List Items**	**List-cover**	
		List of books with year and author names	***List of books without year and author names***
0101	G or W	Warrior Heir, (2006). Axelrad, Catherine	Warrior Heir
0011	E or U	Ever (2008). Fitzgerald, M.	Ever
0111	I or Y	Year of Fog, (2008). Scott Sigler	Year of Fog
0100	F or V	Vengeful Virgin, (1958). Benjamin, Ross	Vengeful Virgin
0110	H or X	Hunting Wind, (2002). Smith, Melissa	Hunting Wind
1111	Q	Q is for Quarry (2002). Sue Grafton	Q is for Quarry
0111	I or Y	Inventing the Abbotts (1987). Joss, Morag	Inventing the Abbotts
0000	B or Y	Blood Is the Sky (2004). Steve Hamilton	Blood is the sky

But with this option, the length of the list-cover still depends on the length of the message. Using other list such as employee birthday list, this can raise suspicion, because the list has to contain all of the employees that has birthday in a particular period of time.

III. Modified List Steganography Based on Syllable Patterns

The proposed method exploits syllable pattern to encode message, while using list-cover to conceal data. This approach follows the Nostega paradigm in general, and Listega in particular, with the scope as follows. For implementing the proposed method, a company employee birthday case is used. The method uses a list of a company's employee birthday list, which has around 17,500 entries. The method uses two first syllables of the employee name, which is one of the columns of employee list. As the preliminary process, the names of the employees are syllabified according to the rules from Indonesian grammar, which will be described in details in the next section. The syllable patterns which resulted from the syllabification process are then used to embed the secret message.

The secret message is in Indonesian Language, while the characters used in the secret message are the upper-case alphabets without accent marks (A-Z) and SPACE.

To have a better understanding of the proposed method, consider the scenario below:

Bob and Alice work in the same company. Alice is placed in the overseas representative of the company. Bob sends unofficial company policies as secret messages to Alice every certain period of time. Bob and Alice have the base list of all the employees in the company. They agree on concealing messages in cover text in the form of employee birthday list by embedding them into the syllable patterns entries in the list to be published periodically. Alice is not the only recipient of Bob's list so that it will not raise suspicion.

This method can be implemented with filtered cover, i.e. the list-cover contains only embedded items, or with unfiltered cover, i.e. list-cover contains both embedded and non-embedded items. With the second mode, a protocol is needed to mix the embedded list items with the non-embedded ones, which will be explained in detail in the next section

A. Message Embedding Method

As discussed in previous section, the proposed method uses syllable pattern to conceal messages. The syllable patterns are generated from the employee names. The syllabification is made according to Indonesian grammar rule, as follows [6].

1) If in a word there are two consecutive vowels, then the separation is made between the two vowels.

2) If in a word there is a consonant between two vowels, then the separation is made before the consonant.

3) If in a word there are two or more consecutive consonants, then the separation is made after the first consonant.

Based on the rules, there are 11 variations of syllable patterns in Indonesian language, as shown in Table III. (C=consonants V= vowels).

Variations of syllable pattern in Indonesian language are shown in Table II.

After processing employee data list, there are only 7 syllable patterns found out of 11 variations in Indonesian language, namely V, CV, CVC, CCV, CCVC, VCC, and CC. These syllable patterns are obtained using the above syllabification rule. These syllable patterns are then to be used to encode the message.

TABLE II. Variations of syllable patterns In Indonesian Language [7]

No.	Syllable Pattern	Example
1	**V**	as **i** in ***i**-kan* (fish)
2	**VC**	as **ir** in *a-**ir*** (water)
3	**CV**	as **ku** in *a-**ku*** (I)
4	**CVC**	as **pan** in ***pan**-tai* (coast)
5	**VCC**	as **eks** in ***eks**-por* (export)
6	**CVCC**	as **teks** in ***teks*** (text)
7	**CCVCC**	as **pleks** in *kom -**pleks*** (complex)
8	**CCV**	as **pro** in ***pro**-sa* (prose)
9	**CCVC**	as **prak** in ***prak**-tek* (practice)
10	**CCCV**	as **stra** in ***stra**-tegi* (strategy)
11	**CCCVC**	as **struk** in ***struk**-tur* (structure)

The syllable pattern CC is not defined in Indonesian language, but is common in Indonesian names, such as "ny" in Den-ny, since the character "y" in Indonesian Language is defined as consonant.

For increasing the embedding capacity of original Listega, this paper proposed to use first two syllables instead of bit slicing for encoding process. Furthermore, the characters of secret messages is mapped into the syllable patterns in such way that the most occurred characters are to be mapped with the most occurred syllable patterns. For implementing the proposed method, the distribution of the syllable patterns and the distribution of the characters have to be provided. The distribution of syllable patterns is obtained from the rank of average occurrence of syllable patterns in 12 months employee birthday list. This is done because the data are intended to be sent periodically. The distribution of characters in Indonesian Language is obtained from the most occurred characters in the entries of "The Great Dictionary of Indonesian Language (Kamus Besar Bahasa Indonesia or KBBI)" [8]. The Distribution of Characters and Distribution of Syllable Patterns are shown in Table IV.

The distribution of SPACE is obtained based on average length of Indonesian words. The average length of Indonesian words is calculated using the average length of 31,410 word entries in KBBI which has the total length of 215,676 characters. Thus, the average word length is 6.87 characters. Then, the distribution of SPACE is counted as **1/(1+6.87)=12.71%**, which resides between A and I.

Using two first syllables to embed each character of the message, the embedding capacity is roughly twice as much as original Listega, which requires two rows of list to embed one character of message.

TABLE III. (A) DISTRIBUTION OF CHARACTERS IN INDONESIAN LANGUAGE (B) DISTRIBUTION OF SYLLABLE PATTERNS IN EMPLOYEE LIST

Letter	Percentage
A	14.06%
Space	12.71%
I	9.15%
E	8.29%
N	7.32%
R	6.44%
S	6.15%
T	5.70%
K	5.68%
O	5.20%
U	4.88%
L	4.86%
G	4.25%
M	4.13%
P	3.12%
D	2.44%
B	2.43%
H	1.59%
C	0.93%
F	0.92%
J	0.84%
W	0.58%
Y	0.54%
V	0.33%
Z	0.16%
X	0.01%
Q	0.01%

(a)

First-2-Syllable Pattern	Percentage
CV-CV	19.24%
CV-CVC	13.98%
CVC-CV	12.94%
CVC-CVC	6.49%
V-CV	6.16%
V-CVC	5.44%
VC-CVC	4.43%
VC-CV	3.30%
CV-V	2.50%
CCV-CV	2.30%
CCV-CVC	2.13%
CVC-CC	1.48%
CVC-CVCC	1.31%
CV-CC	1.23%
CV-CVCC	1.17%
CV-VC	0.95%
CVC-CCV	0.99%
VCC-CVC	0.90%
CCVC-CV	0.94%
CCV-V	0.81%
CCVC-CVC	0.78%
CV-CCV	0.72%
V-CVCC	0.77%
VC-CCV	0.71%
CVCC-CVC	0.65%
V-CC	0.58%
VC-CVCC	0.54%

(b)

From Table III (a) and (b), a character – syllable mapping can be made, such that the most occurred character is mapped to the most occurred syllable pattern, as shown in Table IV.

TABLE IV. CHARACTER – SYLLABLE PATTERN MAPPING

Char	Syllable Pattern	Char	Syllable Pattern	Char	Syllable Pattern
A	CV-CV	O	CCV-CV	C	CCVC-CV
SPACE	CV-CVC	U	CCV-CVC	F	CCV-V
I	CVC-CV	L	CVC-CC	J	CCVC-CVC
E	CVC-CVC	G	CVC-CVCC	W	CV-CCV
N	V-CV	M	CV-CC	Y	V-CVCC
R	V-CVC	P	CV-CVCC	V	VC-CCV
S	VC-CVC	D	CV-VC	Z	CVCC-CVC
T	VC-CV	B	CVC-CCV	X	V-CC
K	CV-V	H	VCC-CVC	Q	VC-CVCC

This Mapping Table is the agreed upon means to embed and extract message between Bob and Alice.

B. *Architecture of Modified List Steganography*

The architecture of modified List Steganography consists of the Message Embedding Module and Message Extracting module, as shown in Fig 3.

The message embedding process takes place in the cover generator, using the Employee Data from the pre-embedding process.

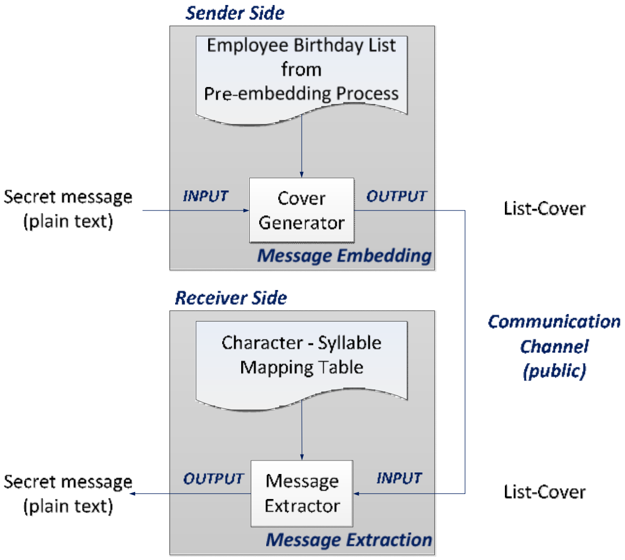
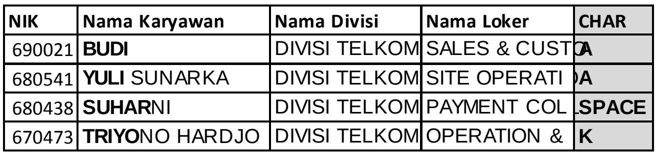

Sender Side
Employee Birthday List from Pre-embedding Process
Secret message (plain text)
INPUT
Cover Generator
OUTPUT
List-Cover
Message Embedding
Receiver Side
Communication Channel (public)
Character - Syllable Mapping Table
Secret message (plain text)
OUTPUT
Message Extractor
INPUT
List-Cover
Message Extraction

Fig. 3. The Architecture of list steganography based on syllable patterns

1) *Pre-embedding Process:*

Before the message can be embedded, there is a pre-embedding process, which intended to make the embedding process more straightforward. The pre-embedding process is merely a process to add a column in the employee data list, such that each row of data has a character it represents that is obtained from character-syllable mapping. The output of the pre-embedding process is as depicted in Fig. 4, with the added CHAR column is in grey. The original list has 7 columns, namely NIK, Nama Karyawan, Tanggal Lahir, Nama Divisi, Nama Posisi, Nama Loker, Nama Employee Sub Group (Employee Number, Employee name, Date of Birth, Division Name, Position Name, Location, Subgroup), but the columns can be selected based on need. In Fig. 4 there are only four columns used.



NIK	Nama Karyawan	Nama Divisi	Nama Loker	CHAR
690021	**BUDI**	DIVISI TELKOM	SALES & CUSTO	A
680541	**YULI** SUNARKA	DIVISI TELKOM	SITE OPERATI	A
680438	**SUHAR**NI	DIVISI TELKOM	PAYMENT COL	SPACE
670473	**TRIYO**NO HARDJO	DIVISI TELKOM	OPERATION &	K

Fig. 4. Insertion of CHAR column in the employee birthday list

2) *Cover Generator:*

As mentioned in previous section, the list-cover can take the form of filtered list or unfiltered list. Using a filtered list the cover generator simply picks the list items, i.e. employee names according to the message. The list-cover consists only embedded items can be sent directly to the receiver, which is the same as the original Listega.

Using an unfiltered list, both the embedded list items and non-embedded ones are sent to the receiver. Because the list used is the employee birthday list, the list can be divided by time intervals based on the employee birth date. An algorithm is introduced to mix the embedded list items and non-embedded ones based on the time intervals, e.g. daily, weekly, or monthly, grouped by division.

This algorithm also has the benefit that a message can be partitioned into message chunks that can fit in the covers

Message partition algorithm:

1. Select Employee Birthday List within a desired month (assumed that the range of selection period is from 1 day to 1 month).
2. Sort the selection by interval or division.
3. Insert markers and message characters, as follows:

a. Get the secret message.

b. Characters A and SPACE in the CHAR column are used to mark the beginning and the end of partitions of message or message chunks.

c. Count A and SPACE. If in certain interval/division count of A <1 and Space <2 then the embedding is done in the next interval/division.

d. Search character in CHAR column which matches each message character.

e. Move the row which has matched character after first

A in a particular interval/division until there is no matched character anymore.

f. If there is no matched character with the message character in the current interval/division, end with 2 SPACE and fill the next row with the rest of the non-

embedded list items until the next interval/division.

SPACE, then put SPACE after A in the next day. This is to keep the consistence 2 SPACE as the end of a message chunk.

h. When the all of the message characters has been embedded, end with 3 SPACE's and fill the next row

i. If the interval reach the end of the month, then the

The message partition process in the list grouping based

The cover generator then selects the employee names which have the corresponding syllable pattern, and then generates the cover according to the message partition

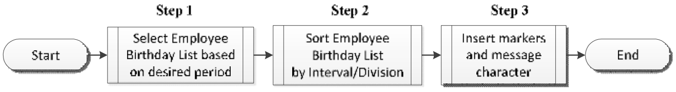

Step 1
Step 2
Step 3
Start
Select Employee Birthday List based on desired period
Sort Employee Birthday List by Interval/Division
Insert markers and message character
End

Fig. 5. Main algorithm of message partitioning

Count A and SPACE in CHAR_column procedure is described as the following pseudocode:

```
Set row to 1
Set A to 0
Set SPACE to 0
While row is less then number of rows
    Get the next A
    Add 1 to A
    Get the next SPACE
    Add 1 to SPACE
End while
```

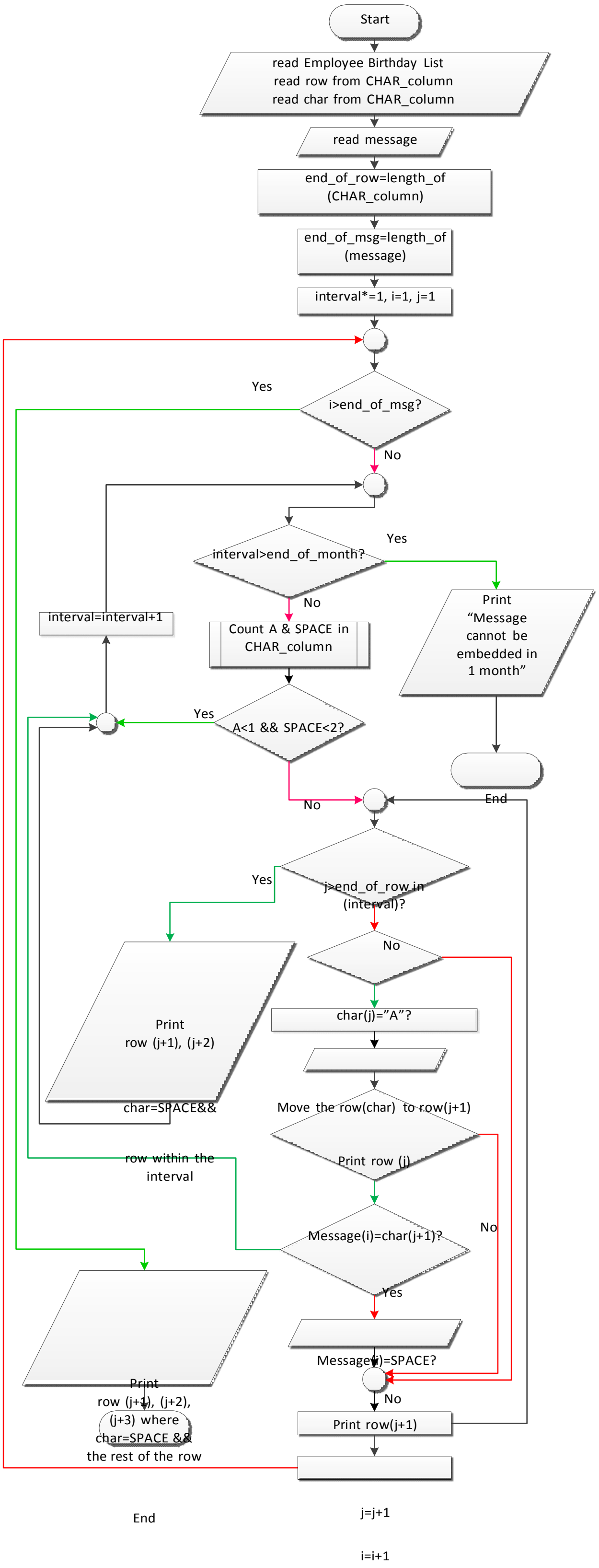
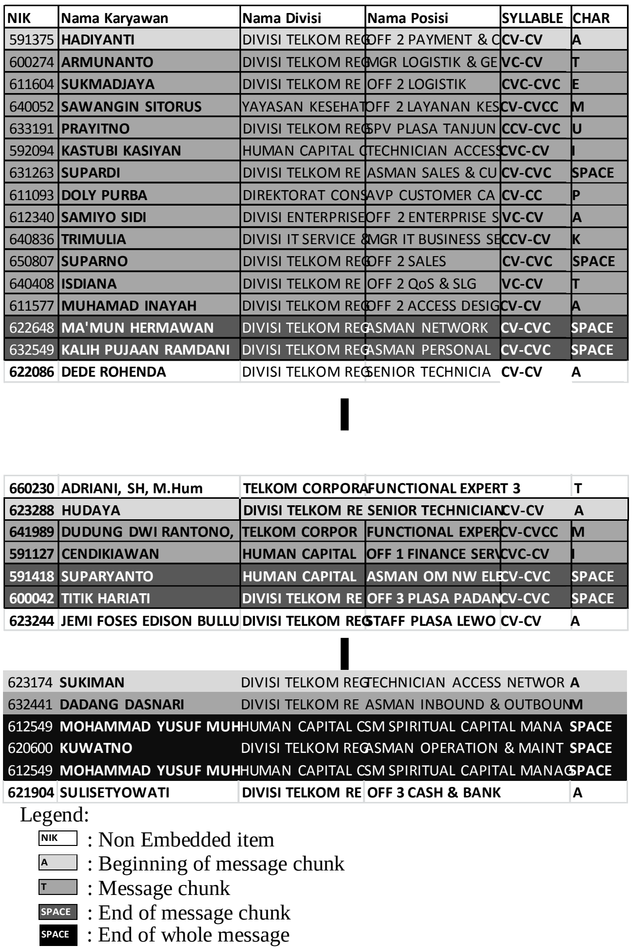

Start
read Employee Birthday List
read row from CHAR_column
read char from CHAR_column
read message
end_of_row=length_of (CHAR_column)
end_of_msg=length_of (message)
interval*=1, i=1, j=1
Yes
i>end_of_msg?
No
Yes
interval>end_of_month?
No
interval=interval+1
Count A & SPACE in CHAR_column
Print "Message cannot be embedded in 1 month"
Yes
A<1 && SPACE<2?
No
End
Yes
j>end_of_row in (interval)?
No
Print row (j+1), (j+2)
char=SPACE&&
char(j)="A"?
Move the row(char) to row(j+1)
Print row (j)
row within the interval
Message(i)=char(j+1)?
No
Yes
Message(i)=SPACE?
Print row (j+1), (j+2), (j+3) where char=SPACE && the rest of the row
No
Print row(j+1)
End
j=j+1
i=i+1

Fig. 6. Detailed algorithm of step 3 of message partitioning

Below is the example of message embedding process. Secret message: TEMUI PAK TAMIM (See Mr. Tamim). The mapping of the secret message as syllable patterns is as follows:

TÆVC-CV, EÆCVC-CVC, MÆCV-CVCC, UÆCCV-CVC, IÆCVC-CV, SPACEÆCV-CVC, PÆCV-CC, AÆCV-CV, KÆCCV-CV, SPACEÆCV-CVC.



NIK	Nama Karyawan	Nama Divisi	Nama Posisi	SYLLABLE	CHAR
591375	HADIYANTI	DIVISI TELKOM REG	OFF 2 PAYMENT & C	CV-CV	A
600274	ARMUNANTO	DIVISI TELKOM REG	MGR LOGISTIK & GE	VC-CV	T
611604	SUKMADJAYA	DIVISI TELKOM RE	OFF 2 LOGISTIK	CVC-CVC	E
640052	SAWANGIN SITORUS	YAYASAN KESEHAT	OFF 2 LAYANAN KES	CV-CVCC	M
633191	PRAYITNO	DIVISI TELKOM REG	SPV PLASA TANJUN	CCV-CVC	U
592094	KASTUBI KASIYAN	HUMAN CAPITAL C	TECHNICIAN ACCESS	CVC-CV	I
631263	SUPARDI	DIVISI TELKOM RE	ASMAN SALES & CU	CV-CVC	SPACE
611093	DOLY PURBA	DIREKTORAT CONS	AVP CUSTOMER CA	CV-CC	P
612340	SAMIYO SIDI	DIVISI ENTERPRISE	OFF 2 ENTERPRISE S	VC-CV	A
640836	TRIMULIA	DIVISI IT SERVICE &	MGR IT BUSINESS SE	CCV-CV	K
650807	SUPARNO	DIVISI TELKOM REG	OFF 2 SALES	CV-CVC	SPACE
640408	ISDIANA	DIVISI TELKOM RE	OFF 2 QoS & SLG	VC-CV	T
611577	MUHAMAD INAYAH	DIVISI TELKOM REG	OFF 2 ACCESS DESIG	CV-CV	A
622648	MA'MUN HERMAWAN	DIVISI TELKOM REG	ASMAN NETWORK	CV-CVC	SPACE
632549	KALIH PUJAAN RAMDANI	DIVISI TELKOM REG	ASMAN PERSONAL	CV-CVC	SPACE
622086	DEDE ROHENDA	DIVISI TELKOM REG	SENIOR TECHNICIA	CV-CV	A

660230	ADRIANI, SH, M.Hum	TELKOM CORPORA	FUNCTIONAL EXPERT 3		T
623288	HUDAYA	DIVISI TELKOM RE	SENIOR TECHNICIAN	CV-CV	A
641989	DUDUNG DWI RANTONO,	TELKOM CORPOR	FUNCTIONAL EXPER	CV-CVCC	M
591127	CENDIKIAWAN	HUMAN CAPITAL	OFF 1 FINANCE SERV	CVC-CV	I
591418	SUPARYANTO	HUMAN CAPITAL	ASMAN OM NW ELE	CV-CVC	SPACE
600042	TITIK HARIATI	DIVISI TELKOM RE	OFF 3 PLASA PADAN	CV-CVC	SPACE
623244	JEMI FOSES EDISON BULLU	DIVISI TELKOM REG	STAFF PLASA LEWO	CV-CV	A

623174	SUKIMAN	DIVISI TELKOM REG	TECHNICIAN ACCESS NETWOR		A
632441	DADANG DASNARI	DIVISI TELKOM RE	ASMAN INBOUND & OUTBOUN		M
612549	MOHAMMAD YUSUF MUH	HUMAN CAPITAL C	SM SPIRITUAL CAPITAL MANA		SPACE
620600	KUWATNO	DIVISI TELKOM REG	ASMAN OPERATION & MAINT		SPACE
612549	MOHAMMAD YUSUF MUH	HUMAN CAPITAL C	SM SPIRITUAL CAPITAL MANAG		SPACE
621904	SULISETYOWATI	DIVISI TELKOM RE	OFF 3 CASH & BANK		A

Legend:

NIK	: Non Embedded item
A	: Beginning of message chunk
T	: Message chunk
SPACE	: End of message chunk
SPACE	: End of whole message

Fig. 7. The Embedding of Message "TEMUI PAK TAMIM"

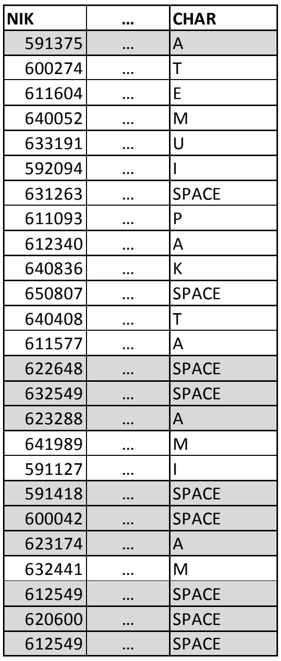
3) *Message Extracting Module*

As the Character - Syllable Mapping Table is the agreed upon means to embed and extract the message, the receiver can also be assumed to have the syllabification module like in the sender side to convert the employee names into syllable patterns. The message extraction algorithm is then implemented to extract the whole message.

Message extraction algorithm:

1. Open the mapping table and the list-cover.
2. Find for the first "A" in the list-cover. If there is no "A" then go to next interval/division, else go to step 3.
3. Check whether 2 characters after "A" are not "SPACE".
   If true go to step 4, else go to next interval/division.
4. The character after "A" of list-cover is the message character.
5. Go to next character of message.
6. Check whether 3 characters after "A" are "SPACE".
   If true go print the character of the message, else go to step 7
7. If there are two consecutive "SPACE", then go the next interval or division.

With the algorithm, the message "TEMUI PAK TAMIM" can be extracted, by omitting the markers, as shown in Fig. 8.



NIK	…	CHAR
591375	…	A
600274	…	T
611604	…	E
640052	…	M
633191	…	U
592094	…	I
631263	…	SPACE
611093	…	P
612340	…	A
640836	…	K
650807	…	SPACE
640408	…	T
611577	…	A
622648	…	SPACE
632549	…	SPACE
623288	…	A
641989	…	M
591127	…	I
591418	…	SPACE
600042	…	SPACE
623174	…	A
632441	…	M
612549	…	SPACE
620600	…	SPACE
612549	…	SPACE

Fig. 8. The Extraction of Message "TEMUI PAK TAMIM"

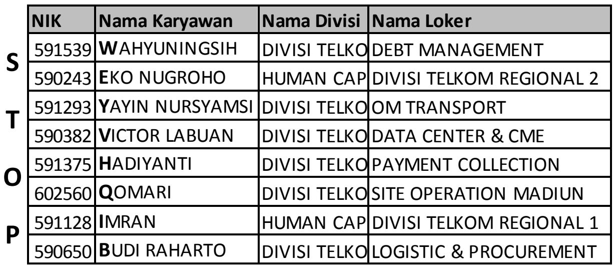
IV. Experiment Results

Several experiments are conducted to compare the performance of original Listega and modified List Steganography with filtered cover. Then the performances of modified List Steganography with unfiltered cover using various groupings were observed to find the optimum grouping to embed message.

A. *The Performance of Modified List Steganography with Filtered Cover*

The objective of this experiment is to compare the performance of the modified List Steganography with the original Listega, in term of embedding capacity and success ratio. Embedding capacity is the amount of data that can be hidden in a cover, compared to the size of the cover [3]. The success ratio is the ratio between successful indices (indices that can produce successful embedings) and total of 26 indices. The scenario of this experiment is to embed the message using the original Listega with the employee birthday list, and then compare the results with modified List Steganography using the same list. The first experiment used the message from the examples of the original Listega, which were "Stop" and "get him". Since the modified List Steganography uses only capital letters, the message used are adapted as "STOP" and "GET HIM". "STOP" was encoded using the original Listega with index 2 of latin square in Table I was used for the mapping matrix with Employee birthday list as the cover. The result is shown in Fig. 9.

With the original Listega, "STOP" needs 8 rows for embedding. Using the list cover with four columns (including the column titles): NIK (employee number), Nama Karyawan (employee name), Nama Divisi (Division Name), Nama loker (Department Name), which consists of 505 characters, the embedding capacity is **4/505** =**0.79%**, while using the list cover of with two columns, NIK and Nama Karyawan, which consists of 147 characters, the embedding capacity is **4/147** = **2.72%.** In the case of "GET HIM", the experiment yielded the embedding capacity of **0.78%** and **2.51%**.



	NIK	Nama Karyawan	Nama Divisi	Nama Loker
S	591539	**W**AHYUNINGSIH	DIVISI TELKO	DEBT MANAGEMENT
	590243	**E**KO NUGROHO	HUMAN CAP	DIVISI TELKOM REGIONAL 2
T	591293	**Y**AYIN NURSYAMSI	DIVISI TELKO	OM TRANSPORT
	590382	**V**ICTOR LABUAN	DIVISI TELKO	DATA CENTER & CME
O	591375	**H**ADIYANTI	DIVISI TELKO	PAYMENT COLLECTION
	602560	**Q**OMARI	DIVISI TELKO	SITE OPERATION MADIUN
P	591128	**I**MRAN	HUMAN CAP	DIVISI TELKOM REGIONAL 1
	590650	**B**UDI RAHARTO	DIVISI TELKO	LOGISTIC & PROCUREMENT

Fig. 9. List-cover of original Listega generated using employee birthday list to embed the message “STOP”

The implementation of latin square with limited list may result failures in message embedding (as mentioned in section II.B). In the case of employee birthday list, the message embedding always failed in indices of latin square that requires X as the first letter of employee name, because no employee name begins with letter X, and occasionally failed when requires Q and V. This happens because employee names that begin with letters are very limited. Table V shows the message embedding success ratio in employee birthday list segmented in 12 months.

Using the modified List Steganography, the first message only needed half the rows of the previous method, which were 4 rows. Embedding the message STOP using four and two columns yielded the embedding capacity of **1.37%** and **4.76%**, respectively. While the message GET HIM takes 7 rows, which yielded embedding capacity of **1.58%** and **4.96%**. Both results are roughly twice as much as the original Listega. This is due to direct mapping between character and syllable pattern in modified List Steganography, as opposed to mapping half character (4-bit slice) to first letter of list item in original Listega.

TABLE V. SUCCESS RATIO OF EMBEDDING WITH ORIGINAL LISTEGA WITH VARIOUS INDICES OF LATIN SQUARE

	Msg= "STOP"		**Msg= "GET HIM"**	
	Successful Indices	***Success ratio***	***Successful Indices***	***Success ratio***
JAN	23	88.46%	23	88.46%
FEB	25	96.15%	25	96.15%
MAR	24	92.31%	24	92.31%
APR	24	92.31%	24	92.31%
MAY	24	92.31%	24	92.31%
JUN	25	96.15%	25	96.15%
JUL	24	92.31%	24	92.31%
AUG	25	96.15%	25	96.15%
SEP	25	96.15%	25	96.15%
OCT	24	92.31%	24	92.31%
NOV	25	96.15%	25	96.15%
DEC	24	92.31%	24	92.31%

B. The Performance of Modified List Steganography with Unfiltered Cover

The objective of the experiments is to compare the performance of modified List Steganography using unfiltered cover and to find out the best interval. The experiment consists of two scenarios. The first scenario is to embed messages in various intervals, i.e. daily, every 2 days, every 3 days, weekly, every 10 and compare the results. The second scenario is to embed messages using the grouping the list items by division in alphabetical and by division with adjustment.

To conduct the experiment in both scenarios, four messages are used, as follows.

- Message 1: AMANKAN MAKANAN (15 characters, 5 variables or distinct characters)
- Message 2: NAMA DAN ALAMAT (15 char, 7 var)
- Message 3: TEMUI PAK TAMIM (15 char, 9 var)
- Message 4: KIRIM VIA FAX USULAN BOQ SWITCH ZTE YANG AKAN DIPASANG DI JAKARTA (65 char, 27 var)

The results of the experiment with first scenario are seen in Fig. 10. Using interval of every 10 day, each message can be delivered in one chunk, which means no partition needed to embed message. While other intervals have significant raise in the number of chunks in embedding Message 4 (65 char, 27 var).

The number of overheads is directly proportional to the number of the chunks. Since each chunk contains 1 A and 2 SPACE’s and another 1 SPACE to mark the end of whole message, then if the number of chunks equals N, then the number of overheads can be counted as follows:

$$\sum Overheads\ (char) = 3N + 1 \qquad (1)$$

The interval of 10 days has the smallest number of overheads, i.e. 4 characters. While the daily interval has the highest number of overheads, with 85 characters. This is due to the higher availability of employee names to embed the message in the interval of 10 days compared to daily interval. And the availability of employee names makes less overheads needed

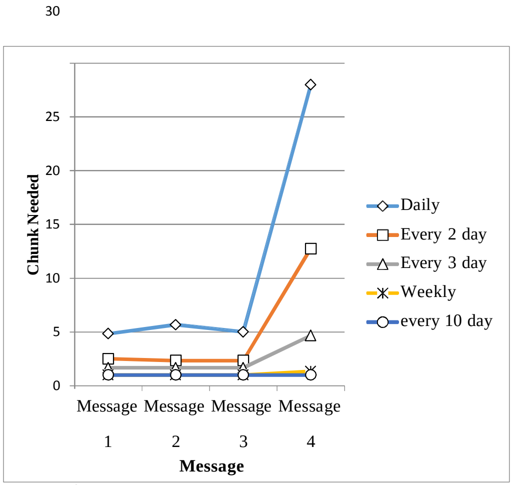


Fig. 10. Graphic of Message Chunks Needed to Embed Message 1 – 4 in various intervals.

The results of second scenario are seen in Fig. 11. For embedding Message 1, 2, and 3 by grouping list items by division in alphabetical order, the average 22.56 chunks is needed. The failure of message embedding is because it is just based on 1 month limit. It will be able to embed with extended interval, i.e. more than 1 month.

The adjustment is made by placing 7 divisions in the beginning of the list, namely Divisi Regional 1 until 7. The divisions is chosen because the divisions have the largest employees and also there are still regularity in the division names.

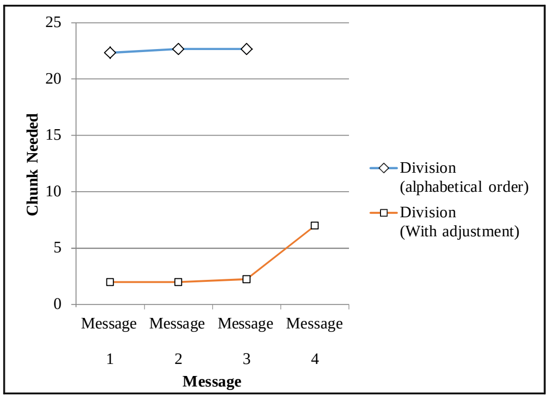


Fig. 11. Comparison between embedding message based on alphabetically division grouping and adjusted division grouping.

The average number of chunks needed to deliver Message 1, 2 and 3 is 2.08 chunks. While the embedding of Message 4 needed 7 chunks. The improvement is gained due to the greater availability of employee names needed to embed the message in the 7 divisions in the beginning compared to those in alphabetical order.

Based on the two scenarios, the grouping of list items based on interval of 10 days are considered the best embedding, because it needs the least overheads and always succeed to embed message up to 65 characters with 27 distincts characters. In term of message integrity, this interval also give the best performance, as the whole message can be delivered without having to be divided into smaller chunks.

V. Conclusions and Future Works

A. Conclusions

In original Listega, there is no correlation between cover and the mapping of the character. Therefore, in original Listega failures in embedding message can occur due to the requirement of rarely-used characters in the mapping using latin square. For overcoming this problem, the modified List Steganography built the mapping used for embedding the messages based on the list that will be embedded such that the needed characters in the modified List Steganography are always available. The failure in embedding message occurs only when there is a shortage of character needed by the message.

Mixing the embedded data with non-embedded data as implemented in the modified List Steganography with unfiltered cover can reduce suspicion form other parties, as the list-cover contains the whole natural cover of employee birthday list. Meanwhile, original Listega contains only part of the list such that it can lead to suspicion when there are some employee names missing from the list. Thus, the modified Listega is better than the Original Listega in term of reducing the suspicion from other parties.

B. Future Works

In this research, only one column of a list, i.e. employee names, is used to embed messages. It should be investigated whether using more than a column can enhance the embedding capacity.

There is also possibility to develop the method without using a list.

References

[1] Agarwal, M., “Text steganographic approaches: a comparison”, International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1, January 2013.

[2] Thampi, S. M., “Information hiding techniques: a tutorial review”, ISTE-STTP on Network Security & Cryptography, LBSCE, 2004.

[3] Desoky, A., “Listega: list-based steganography methodology”, Int. J. Inf. Secur. 8, 4 (August 2009), 247-261, 2009.

[4] Desoky, A., “Matlist: Mature linguistic steganography methodology”. Security Comm. Networks, 4: 697–718, 2011.

[5] Desoky, A., “NORMALS: normal linguistic steganography methodology”, Journal of Information Hiding and Multimedia Signal Processing, Volume 1, Number 3, July 2010.

[6] Sinduwiryo, S., “The standard forms of the base Indonesian words”, The First Asia International Lexicography Conference, Manila, Philippines, 1992.

[7] Basuki, T. A., “Pengenalan suku kata bahasa Indonesia menggunakan finite-state automata”, Integral, vol. 5 no. 2, 2000.

[8] Indonesian Department of Education, “Kamus besar bahasa Indonesia” (The Great Dictionary of Indonesian Language), Fourth Edition, Gramedia Pustaka Utama Publisher, 2008.