

File Formats Security – Proprietary vs. Open-Source

Edmund Laugasson¹ and Kaido Kikkas^{1,2}

¹Tallinn University, Institute of Informatics, Narva Road 25, 10120 Tallinn, Estonia

²Estonian Information Technology College, Raja St 4c, 12616 Tallinn, Estonia

{Edmund.Laugasson,Kaido.Kikkas}@tlu.ee

Abstract. Privacy and confidentiality are important components of digital literacy. Yet nowadays documents can be found online, which apparently consist only of one or two pages yet have huge file size - even several megabytes. Such documents may contain sensitive data that has been deleted but actually is still there. Our study provides an analysis of such cases in public sector of Estonia. Based on experiments and public sector web page analysis we describe security threats and features of different file formats and offer suggestions for their use, e.g. we found that using open-source formats like OpenDocument may help prevention of accidental disclosure of data.

Keywords: privacy, confidentiality, file format, proprietary, open-source, vendor lock-in.

1 Background

Privacy is often considered an area of digital literacy that is in danger of being overlooked, compared with other, more "marketable" ICT skills [1]. Yet, as seen below, even official documents on public networks have privacy shortcoming - this paper looks at privacy issues surfacing due to widespread use of proprietary file formats.

Based on Estonian Interoperability Framework 2011 [2] only OpenDocument formats like ODT, ODS, ODP can be recommended for editing by both sides [3]. OpenDocument is also a standard in other countries [4]. EIF 2011 suggests to use PDF file format when editing by both sides is not needed – this is also common standard in the world [5].

Additionally, file sharing must not be based on import-export. For example, using DOCX means using MS Word, as trying to open it with LibreOffice Writer actually means importing it. Saving DOCX with LibreOffice Writer is possible but strongly not suggested as it is not native format for LibreOffice Writer and therefore problems may occur [6].

Also important is the version of MS Word – for example, editing DOCX file with MS Word 2013 and trying to open it with MS Word 2010 will run into incompatibility problems; some formatting may be lost and even some data (e.g. graphics).

We subscribe to the notion that using open file formats like OpenDocument is strongly suggested, as is creating them with LibreOffice as native editor. When using DOC or DOCX file formats then an appropriate MS Word version should be used.

When sharing files, the same office suite at both sides should be used. Mixing office suites will run into compatibility problems [7]. Keeping the same version with LibreOffice is not expensive as it is free and open-source software. But keeping MS Office on the same version across institutions might be too expensive and overwhelming for users.

OpenDocument adoption in the world has also begun already since OpenOffice.org times but now LibreOffice gives better user experience.

It has been suggested that when saving information as DOC, RTF or some other proprietary file format (e.g. OOXML - Office Open XML – docx, xlsx, pptx, etc), it will store deleted information [8]. It can be seen when something is deleted from such a file, the file then saved and closed - after reopening, the deleted data is still there and can be discovered using some plain text editor (e.g. Notepad2, Notepad++ in Windows or Kate, Geany etc in Linux). However, saving the same file in ODF format (odt, ods, odp etc) will reduce file size and remove hidden parts.

There is a recurring pattern in file handling, especially visible in public sector - users open an file, delete (some of) its content, add new elements and save the file under a new name. These results in document files having just one or two pages but the file size can be in megabytes. Also, large document template files can be found from public sector web pages - and these files can contain sensitive data.

Saving the same file into OpenDocument may reduce the file size for 10 or even more times. The file will still contain the same information but not any ballast data. Proprietary formats do also have problems with revisioning. Finally, there is the ever-visible vendor lock-in problem.

2 Methods

We analyzed documents originating from public sector web pages and carried out experiments. We used Google search by file type to download five random DOC files, five DOCX files and five RTF files from Estonian public sector web sites. DOC and RTF files were opened with MS Word 2003, all content were deleted and replaced with a single word "Hello", the file was then saved and closed. Afterwards, the files were studied using a text editor, searching for deleted text. We used the same process for DOCX files, with the exception of using MS Word 2010 as likely the most used version at the moment (note: as of the time of writing, MS Word 2013 is not widely used in Estonia yet, so the assumption was that most DOCX files available from public sector websites are still created using older - 2007 and 2010 - versions of MS Word).

After this, the files were renamed for indexing. File sizes were registered for both original and changed files and the results compared (see the section "File size comparison" below).

Ubuntu 12.04 LTS 64-bit was used as the main testing platform (using separately added 64-bit kernel version 3.12.5) and file sizes were detected using bash command line and command `ls -l`. MS Word 2003 (11.5604.5606) and MS Word 2010 (14.0.7015.1000) were installed onto separate virtual machines using VirtualBox

4.3.4 r91027 with guest additions of the same version. The virtual machines used MS Windows XP Pro SP3 as operating system. All Microsoft software used was 32-bit.

For the file content investigation we used Emacs¹ 23.3.1, Midnight Commander² 4.8.1 - all available through regular repositories for Ubuntu 12.04 LTS (thus they were not of the latest versions but nevertheless rather up-to-date). For screen capturing Shutter³ 0.90.1 was used.

Also text editors like Kate, Geany, gEdit, Vim, Cream (modern Vim), wxHexEditor were tested, but Emacs was seen to give the best results in our case.

After using Word to delete the original content and replace it with "Hello", the file was saved into the original format and the latter opened with LibreOffice Writer (LibreOffice version 4.1.3.2 64-bit was used). We then saved the document into ODT-format and repeated the content analysis.

3 Results and Discussion

We started with two hypotheses: The first one was that deleted information is still available in the files and can cause leaks of sensitive information. The second hypothesis was saving DOC, DOCX, RTF file into ODT format will reduce file size. The results of the experiments are described below.

3.1 Used Files

Below is a brief overview of the files tested. The files were indexed for faster investigation and to test their integrity.

In experiment were three different filetypes used – DOC, DOCX, RTF – they were marked as described also in Table 1.

Files used in experiment is available here - <http://url.zeroconf.ee/hcii2014experiment>

3.2 File Size Comparison

Table 1 (below) shows the results of file investigation.

The new size and amount of characters were registered after deleting all content, replacing it by just one word: „Hello“ (5 characters) and then saving the file. Negative values in Table 1 mean file size increase when saving to ODT format - thus we see that a significant reduction of file size can be obtained with DOC and DOCX files by merely saving them in OpenDocument. Saving the RTF as ODT, however, resulted in file size increase in some cases (but the change was smaller than in other cases). This is described in more detail below.

¹ <http://www.gnu.org/software/emacs/>

² <http://www.midnight-commander.org/>

³ <http://shutter-project.org/>

Table 1. File size comparizon

File index	Original size, bytes	Original number of characters with spaces	New size, bytes (5 characters)	New size in ODT, bytes	Difference with new size saved into ODT, bytes	Difference with new size saved into ODT, %
DOC1	50688	1594	19968	8340	11628	41,77%
DOC2	32768	867	19968	8348	11620	41,81%
DOC3	28672	2028	19968	8365	11603	41,89%
DOC4	28672	8128	19968	8744	11224	43,79%
DOC5	30208	1767	20480	9006	11474	43,97%
RTF1	10899	4319	4696	8117	-3421	-57,85%
RTF2	8676	801	3651	7964	-4313	-45,84%
RTF3	82970	2122	4811	8259	-3448	-58,25%
RTF4	141610	3381	12471	9099	3372	72,96%
RTF5	53631	1232	5063	8347	-3284	-60,66%
DOCX1	19837	2139	15966	8706	7260	54,53%
DOCX2	21748	527	14836	8660	6176	58,37%
DOCX3	19876	810	15816	8388	7428	53,03%
DOCX4	25957	3454	19311	8827	10484	45,71%
DOCX5	40935	1512	17055	9153	7902	53,67%

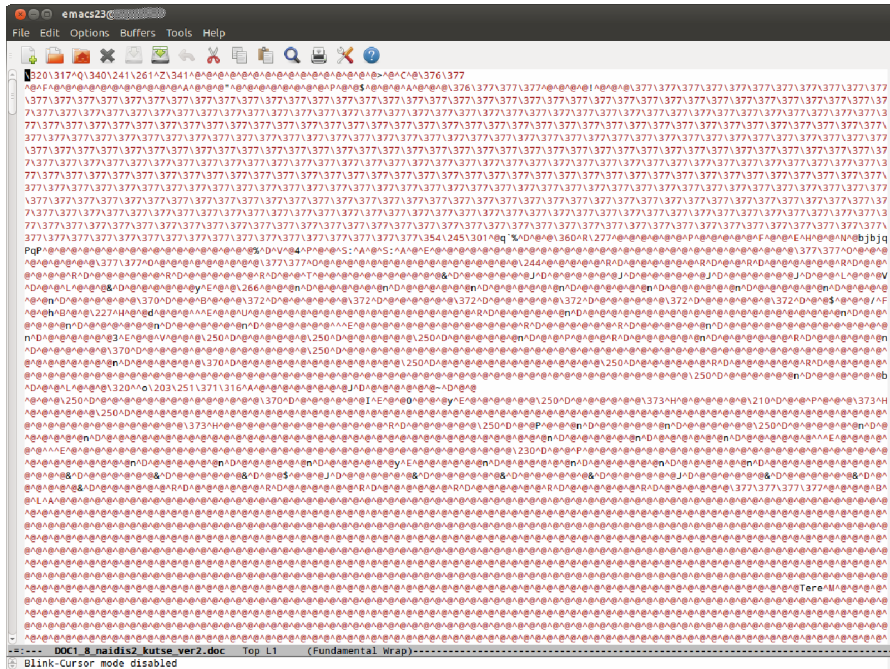


Fig. 1. Emacs display of “DOC1” file

```

DOC1_8_naidis2_kutse.doc
bjbjqPqP
Hello
  I D I S  2
Alland Parman
Normal.dot
Microsoft Office Word
  I D I S  2
Tiitel
Microsoft Office Wordi dokument
MSWordDoc
Word.Document.8

```

Fig. 2. Midnight Commander view of “DOC2”

Experiments with „DOC1“, „DOCX1“, „RTF1“ Files.

After opening „DOC1“ file with MS Word 2003, replacing everything with „Hello“ and saving it, the file size was 19 968 bytes. Using the text editor Emacs on the file still shows a lot of content (Figure 1). We got similar results with other DOC files.

Due DOC being binary format, the text is unreadable. But Midnight Commander shows the content as more legible (Figure 2).

As we can see, there are some more information visible in the Midnight Commander view than might be expected. This is one point where sensitive data may leak.

When opening DOCX file with Emacs, it shows archived XML-files at the beginning - these can be opened directly like hyperlinks (Figure 3).

M	Filemode	Length	Date	Time	File
-	-rw-rw-rw-	1819	1-Jan-1980	00:00:00	[Content_Types].xml
-	-rw-rw-rw-	590	1-Jan-1980	00:00:00	_rels/.rels
-	-rw-rw-rw-	1350	1-Jan-1980	00:00:00	word/_rels/document.xml.rels
-	-rw-rw-rw-	1628	1-Jan-1980	00:00:00	word/document.xml
-	-rw-rw-rw-	1466	1-Jan-1980	00:00:00	word/endnotes.xml
-	-rw-rw-rw-	1472	1-Jan-1980	00:00:00	word/footnotes.xml
-	-rw-rw-rw-	6992	1-Jan-1980	00:00:00	word/theme/theme1.xml
-	-rw-rw-rw-	3313	1-Jan-1980	00:00:00	word/settings.xml
-	-rw-rw-rw-	1451	1-Jan-1980	00:00:00	word/fontTable.xml
-	-rw-rw-rw-	19000	1-Jan-1980	00:00:00	word/styles.xml
-	-rw-rw-rw-	15713	1-Jan-1980	00:00:00	word/stylesWithEffects.xml
-	-rw-rw-rw-	986	1-Jan-1980	00:00:00	docProps/app.xml
-	-rw-rw-rw-	630	1-Jan-1980	00:00:00	docProps/core.xml
-	-rw-rw-rw-	5686	1-Jan-1980	00:00:00	word/webSettings.xml
-	-rw-rw-rw-	1639	1-Jan-1980	00:00:00	word/numbering.xml
-	-	63735	-	-	15 files

Fig. 3. Emacs view of “DOCX1” file structure

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<w:document xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:wp14="http://schemas.microsoft.com/office/word/2010/wordprocessingDrawing"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml"
xmlns:wpg="http://schemas.microsoft.com/office/word/2010/wordprocessingGroup"
xmlns:wpi="http://schemas.microsoft.com/office/word/2010/wordprocessingInk"
xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"
xmlns:wps="http://schemas.microsoft.com/office/word/2010/wordprocessingShape" mc:Ignorable="w14 wp14">
w:body>w:p
w:rsidR="00B6A88" w:rsidRPr="00374D62" w:rsidRDefault="00374D62"
w:rsidP="00374D62">w:r>w:t>Hello</w:t></w:r>w:bookmarkStart w:id="0" w:name="GoBack">w:bookmarkEnd
w:id="0"</w:p><w:sectPr w:rsidR="00B6A88" w:rsidRPr="00374D62" w:rsidSect="00FB3C63">w:pgSz w:w="11906"
w:h="16838">w:pgMar w:top="1417" w:right="1417" w:bottom="1417" w:left="1417" w:header="708" w:footer="708"
w:gutter="0"</w:cols w:space="708">w:docGrid w:linePitch="360"</w:sectPr></w:body></w:document>
```

Fig. 4. Emacs view of “DOCX1” file component “document.xml”

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<w:document xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:wp14="http://schemas.microsoft.com/office/word/2010/wordprocessingDrawing"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml"
xmlns:wpg="http://schemas.microsoft.com/office/word/2010/wordprocessingGroup"
xmlns:wpi="http://schemas.microsoft.com/office/word/2010/wordprocessingInk"
xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"
xmlns:wps="http://schemas.microsoft.com/office/word/2010/wordprocessingShape" mc:Ignorable="w14 wp14">
w:body>w:p
w:rsidR="00B6A88" w:rsidRPr="00374D62" w:rsidRDefault="00374D62"
w:rsidP="00374D62">w:r>w:t>Hello</w:t></w:r>w:bookmarkStart w:id="0" w:name="GoBack">w:bookmarkEnd
w:id="0"</w:p><w:sectPr w:rsidR="00B6A88" w:rsidRPr="00374D62" w:rsidSect="00FB3C63">w:pgSz w:w="11906"
w:h="16838">w:pgMar w:top="1417" w:right="1417" w:bottom="1417" w:left="1417" w:header="708" w:footer="708"
w:gutter="0"</w:cols w:space="708">w:docGrid w:linePitch="360"</w:sectPr></w:body></w:document>
```

Fig. 5. Emacs view of “RTF1” file

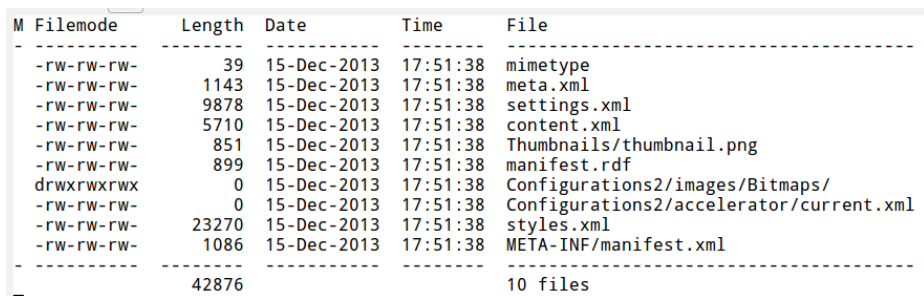
Each of these XML files does contain data about formatting and also content. The most important component is document.xml, which contains the content of document (Figure 4).

Note: we can also rename DOCX file to ZIP and uncompress it but Emacs does it on the fly, which is much more comfortable. Emacs is available for all popular operating systems and is free software.

When looking inside the RTF-file we can finally read the content (Figure 5). It contains a lot of information but only one word entered by the user – the word „Hello“ seen at the end of RTF block.

After saving the previously modified RTF-files into ODT-format, the situation changed. Some RTF-files were even bigger in ODT than in RTF-format itself but this is not always so as the „RTF4“ got smaller when saved into ODT format (the result of file content comparison is shown in Appendix 1). Saving „RTF4“ into ODT format saves a lot of file size. This also depends on the MS Word version that was used previously to modify the RTF-file. Each MS Word will leave its additions into file and the file gets bigger. All these additions will be removed when saving into ODT format.

Even if usually RTF comes smaller than ODT, it is not a good idea for use as a document format – usually it cannot support more complex formatting like ODT, DOC or DOCX does.



M	Filemode	Length	Date	Time	File
	-rw-rw-rw-	39	15-Dec-2013	17:51:38	mimetype
	-rw-rw-rw-	1143	15-Dec-2013	17:51:38	meta.xml
	-rw-rw-rw-	9878	15-Dec-2013	17:51:38	settings.xml
	-rw-rw-rw-	5710	15-Dec-2013	17:51:38	content.xml
	-rw-rw-rw-	851	15-Dec-2013	17:51:38	Thumbnails/thumbnail.png
	-rw-rw-rw-	899	15-Dec-2013	17:51:38	manifest.rdf
	drwxrwxrwx	0	15-Dec-2013	17:51:38	Configurations2/images/Bitmaps/
	-rw-rw-rw-	0	15-Dec-2013	17:51:38	Configurations2/accelerator/current.xml
	-rw-rw-rw-	23270	15-Dec-2013	17:51:38	styles.xml
	-rw-rw-rw-	1086	15-Dec-2013	17:51:38	META-INF/manifest.xml
		42876			10 files

Fig. 6. Emacs view of “RTF4” in ODT format

In ODT file the „content.xml“ contains the file content itself (Figure 6).

4 Conclusion

The experiments revealed some problems with the files. The DOCX files were sometimes opened in compatibility mode, which suggests an earlier version of DOCX than MS Word 2010 (usually MS Word 2007). This means that DOC and DOCX files stored at Estonian public sector websites were created using earlier versions of MS Word than the current 2010 used for our experiment. The same situation is seen with MS Word 2010 and 2013 – the 2010 does not support all features available in 2013, so the compatibility mode will be used and some data may be lost due to this.

The first hypothesis – deleted information will be stored inside the file – was not completely confirmed. We may say that at least MS Word 2010 will permanently delete almost all content. There might be remain some small parts but these are not noticeable. But as Chinese researchers suggest: „Experiments show that 0.44 bit is embedded into each word and 1/151 bit is embedded into each bit of the document on average, which is higher than contemporary linguistic steganography approaches“ then still we may say that MS Word is not cleanly deleting all the data [8]. Even Microsoft gives suggestions how to discover [9] and remove unnecessary parts of file

[10]. Considering all this, Estonian public sector shows similar vulnerabilities like those described by van Hamel [1] and should likewise strive towards an established set of privacy competencies to maintain confidential information online.

The second hypothesis – RTF, DOC, DOCX file sizes will be reduced when saving into ODT – were completely confirmed in part of DOC, DOCX files. Even if RTF-files were smaller than ODT-files, there is not recommended to use it due to lack of support more complex functions available nowadays modern office suites.

Using correct file formats with appropriate programs when working collaboratively – this all makes an important part of nowadays digital literacy. Using importing-exporting documents in foreign programs may run into incompatibility issues and even security leaks. Therefore interoperability frameworks are created and strongly suggested to follow to ensure hassle free collaboration between people.

Acknowledgements. This research was supported by the Tiger University Program of the Information Technology Foundation for Education.

References

1. van Hamel, A.: The Privacy Piece: Report on Privacy Competencies in Digital Literacy Programs in Canada, Britain, Australia, America, and Brazil. University of Ottawa (November 2011), https://www.priv.gc.ca/information/research-recherche/2011/hamel_201111_e.asp (retrieved)
2. RISO. Estonian Interoperability Framework (2011), <http://www.riso.ee/en/estonian-interoperability-framework> (retrieved)
3. RISO. Estonian state it architecture (2007), <http://www.riso.ee/et/koosvoime/arhitektuur> (retrieved)
4. Weir, R.: OpenDocument Format: The Standard for Office Documents. IEEE Internet Computing 13(2), 83 (2009)
5. PC Magazine. PDF File Format to Become Open Standard (2007); PC Magazine Online (2007), <http://www.pcmag.com/article2/0,2817,2088283,00.asp> (retrieved)
6. Park, E.G., Oh, S.: Examining Attributes of Open Standard File Formats for Long-term Preservation and Open Access. Information Technology & Libraries 31(4), 44–65 (2012)
7. Microsoft, 3. Differences between the opendocument text (.odt) format and the word (.docx) format (2013), <http://office.microsoft.com/en-ca/word-help/differences-between-the-opendocument-text-odt-format-and-.the-word-docx-format-HA010355788.aspx> (retrieved)
8. Fu, Z., Liu, Y., Li, B., Sun, X.: Text split-based steganography in OOXML format documents for covert communication. Security and Communication Networks 5(9), 957–968 (2012), doi:10.1002/sec.378
9. Microsoft, 1. Inspect documents for hidden data and personal information (2013), <http://office.microsoft.com/en-us/word-help/inspect-documents-for-hidden-data-and-personal-information-HA010074435.aspx> (retrieved)
10. Microsoft, 2. Remove hidden data and personal information by inspecting documents (2013), <http://office.microsoft.com/en-us/word-help/remove-hidden-data-and-personal-information-by-inspecting-documents-HA010354329.aspx> (retrieved)

ODT Content of „RTF4“ File

```
<?xml version="1.0" encoding="UTF-8"?>
<office:document-content xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"
  xmlns:style="urn:oasis:names:tc:opendocument:xmlns:style:1.0" xmlns:text="urn:oasis:names:tc:opendocument:xmlns:text:1.0"
  xmlns:table="urn:oasis:names:tc:opendocument:xmlns:table:1.0" xmlns:draw="urn:oasis:names:tc:opendocument:xmlns:drawing:1.0"
  xmlns:form="urn:oasis:names:tc:opendocument:xmlns:sl:fo-compatible:1.0" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:meta="urn:oasis:names:tc:opendocument:xmlns:meta:1.0"
  xmlns:number="urn:oasis:names:tc:opendocument:xmlns:datatypes:1.0" xmlns:svg="urn:oasis:names:tc:opendocument:xmlns:svg-compatible:1.0"
  xmlns:chart="urn:oasis:names:tc:opendocument:xmlns:chart:1.0" xmlns:dr3d="urn:oasis:names:tc:opendocument:xmlns:dr3d:1.0"
  xmlns:math="http://www.w3.org/1998/Math/MathML" xmlns:form="urn:oasis:names:tc:opendocument:xmlns:form:1.0"
  xmlns:script="urn:oasis:names:tc:opendocument:xmlns:script:1.0" xmlns:ooo="http://openoffice.org/2004/office"
  xmlns:ooon="http://openoffice.org/2004/writer" xmlns:oooc="http://openoffice.org/2004/calc" xmlns:odom="http://www.w3.org/2001/xml-events"
  xmlns:forms="http://www.w3.org/2002/xforms" xmlns:xsde="http://www.w3.org/2001/XMLSchema"
  xmlns:si="http://www.w3.org/2001/XMLSchema-instance" xmlns:rpt="http://openoffice.org/2005/report"
  xmlns:of="urn:oasis:names:tc:opendocument:xmlns:of:1.2" xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns:grddl="http://www.w3.org/2003/ga/data-view#" xmlns:officeooo="http://openoffice.org/2009/office"
  xmlns:tableooo="http://openoffice.org/2009/table" xmlns:drawooo="http://openoffice.org/2010/draw"
  xmlns:calcext="urn:org:documentfoundation:names:experimental:calc:xmlns:calcext:1.0"
  xmlns:field="urn:openoffice:names:experimental:ooo-ms-interop:xmlns:field:1.0"
  xmlns:formx="urn:openoffice:names:experimental:ooxml-odf-interop:xmlns:form:1.0" xmlns:css3t="http://www.w3.org/TR/css3-text/"
  office:versions="1.2">
  <office:scripts/>
  <office:font-decls>
  <style:font-face style:name="Lohit Hindi" svg:font-family="Lohit Hindi"&apos;"/>
  <style:font-face style:name="Arial" svg:font-family="Arial" style:font-family-generic="swiss"/>
  <style:font-face style:name="Arial Unicode MS" svg:font-family="Arial Unicode MS&apos;"/>
  <style:font-face style:name="Lohit Hindi" style:font-family-generic="swiss"/>
  <style:font-face style:name="Tahoma" svg:font-family="Tahoma" style:font-family-generic="swiss"/>
  <style:font-face style:name="Times New Roman" svg:font-family="Times New Roman&apos;"/>
  <style:font-face style:name="Tahoma Greek" svg:font-family="Tahoma Greek&apos;"/>
  <style:font-face style:name="Arial" style:font-family-generic="system"/>
  <style:font-face style:name="Arial1" style:font-family="Arial1" style:font-family-generic="swiss"/>
  <style:font-face style:name="Droid Sans" style:font-family="Droid Sans" style:font-family-generic="swiss"/>
  <style:font-face style:name="Lohit Hindi" style:font-family="Lohit Hindi" style:font-family-generic="system"/>
  <style:font-face style:name="Lohit Hindi" style:font-family="Lohit Hindi" style:font-family-generic="system"/>
  </office:font-decls>
  <office:automatic-styles>
  <style:style style:name="P1" style:family="paragraph" style:parent-style-name="Standard" style:master-page-name="Standard">
  <style:paragraph-properties fo:margin-left="0cm" fo:margin-right="0cm" fo:text-align="start" style:justify-single-word="false" fo:orphans="2" fo:widows="2" fo:text-indent="0cm" style:auto-text-indent="false" style:page-number="auto" style:vertical-align="auto"/>
  </style>
  <style:style style:name="P2" style:family="paragraph" style:parent-style-name="Footer">
  <style:paragraph-properties fo:margin-left="0cm" fo:margin-right="0cm" fo:text-align="end" style:justify-single-word="false" fo:orphans="2" fo:widows="2" fo:text-indent="0cm" style:auto-text-indent="false" style:vertical-align="auto">
  <style:tab-stops>
  <style:tab-stop style:position="8.001cm" style:type="center"/>
  </style:tab-stop>
  <style:style style:name="P3" style:family="paragraph" style:parent-style-name="Footer">
  <style:paragraph-properties fo:margin-left="0cm" fo:margin-right="0cm" fo:text-align="center" style:justify-single-word="false" fo:orphans="2" fo:widows="2" fo:text-indent="0cm" style:auto-text-indent="false" style:vertical-align="auto">
  <style:tab-stops>
  <style:tab-stop style:position="8.001cm" style:type="center"/>
  </style:tab-stop>
  <style:style style:name="P4" style:family="text" style:parent-style-name="Text">
  <style:text-properties fo:font-size="12pt" fo:language="et" fo:country="EE" style:font-size-Asian="12pt" style:language-Asian="et" style:country-Asian="EE"/>
  </style>
  <style:style style:name="T1" style:family="text">
  <style:text-properties fo:font-size="12pt" fo:language="et" fo:country="EE" style:font-size-Asian="12pt" style:language-Asian="et" style:country-Asian="EE"/>
  </style>
  <style:style style:name="T2" style:family="text">
  <style:text-properties fo:language="et" fo:country="EE"/>
  </style>
  <style:style style:name="T3" style:family="text">
  <style:sequence-decls>
  <text:sequence-decl text:display-outline-level="0" text:name="Illustration"/>
  <text:sequence-decl text:display-outline-level="0" text:name="Table"/>
  <text:sequence-decl text:display-outline-level="0" text:name="Text"/>
  <text:sequence-decl text:display-outline-level="0" text:name="Drawing"/>
  </text:sequence-decls>
  <text:p text:style-name="P1">
  <text:span text:style-name="T1">Hello</text:span></text:p>
  </office:body>
  </office:document-content>
  </office:document-content>
```

Fig. 8. ODT content of “RTF4” file