DESIGN OF THE EXCEL VBA PASSWORD CHANGING MODULE

Jang, Seung-Ju¹ ¹Department of Computer Engineering, Dongeui University, Korea Sjjang@deu.ac.kr

ABSTRACT

In this paper, I design a module that can change password when Excel VBA password is set. Especially, I designed it to be applicable to VBA environment using macros in Excel. It often happens that you cannot access important file data due to password loss, after setting the VBA password. For these users, it is possible to access valuable data by more convenient VBA password using the proposed program. I actually implemented the proposed design in this paper. Experiments were performed on the actual contents of the excel file. Experimental results show that the contents of Excel VBA password can change.

Keywords: Excel VBA, Password Change, Password Recovery, VBA(Visual Basic for Application), Excel File Security.

1. INTRODUCTION

This paper relates to an Excel spreadsheet that we use daily in our daily life. VBA is often used in Excel spreadsheets for complex data processing. We design a module that enables changes to the password used in this VBA. Recent Excel spreadsheets have an OLE structure. Object Linking and Embedding (OLE) files (or COM Structured Storage) format has been used as document formats for Microsoft Office, Word, Excel, and PowerPoint. The OLE file format has the same file system structure as the internal hard disk. Therefore, there is the concept of folders and files inside the OLE file.

Since OLE files are one small file system, MS Office has backward compatibility and upward compatibility. For example, even if you create and save a document in MS Word 2007, you can read it in MS Word 2003. If you create a storage / stream that is used by the version of MS Word 2007 and a storage / stream that is used by the version of MS Word 2003 inside the OLE file, then the version of MS Word 2003 only refers to that version of storage and stream. Even if the MS Word 2007 related storage and stream exist, the version will not access the area, and it will have the effect as if the lower version software reads the document in the upper version without any problem [1, 2].

In this paper, I design a module that can change password when Excel VBA password is set. Especially in Excel, I designed it to be applicable to VBA environment using macros. After setting the VBA password, it often happens that you cannot access important file data due to password loss. By using programs for these users, it is possible to access valuable data through more convenient VBA password access. I actually implemented the proposed design in this paper. Experiments were performed on the actual contents of the excel file.

In this paper, Chapter 2 refers to related research, Chapter 3 refers to VBA password module design, Chapter 4 experiment, and Chapter 5 conclusion.

2. RELATED RESEARCH

The following Fig. 1 shows the internal structure of a doc file written in MS Excel simply by SSViewer (Structured Storage Viewer) program.

🔒 File El	ed Storage Viewer – Ltest ement PropertySet Decode	.doc. ers Windows Help				-121
or	DC	Folder				
Sur Sur Ta Dat Dat Ibc	ument Summary Information many Information bile mpObj s comentSummaryInformation mmaryInformation dDocument	Name Name Table CompObi Data DocumentSummayInformation DocumentSummayInformation DocumentSummayInformation WordDocument	Size [8] 11647 110 4096 4096 4096 140334	Created 1601-01-01 오건 90 1601-01-01 오건 90	Accessed 1601-01-01 오전 9.0 1601-01-01 오전 9.0 1601-01-01 오전 9.0 1601-01-01 오전 9.0 1601-01-01 오전 9.0 1601-01-01 오전 9.0 1601-01-01 오전 9.0	Modified 1601-01 1601-01 1601-01 1601-01 1601-01 1601-01
General Type Name	Storage test.doc					
Count Created	100,300 B 6 2012-07-10 오전 9:48:08					
Modified	2012-07-10 오전 9:48:15					
Accessed	2012-08-17 오전 10:47:56					
3 Checksur	ns					
CRC32	EB8878FB	and a second sec				
MD5	C47DD9DC22C4D74359D808	1				

Figure 1. Internal Structure of OLE Document using SSViewer

OLE file is divided into two big blocks. They are a header block and a data block. The header block has a size of 512 bytes, and the data block has 512 bytes or more. The header block contains the main information of the entire OLE file, and the data block has the following information.



- Properties
- Stream data
- Big Block Allocation Table (BBAT)
- Small Block Allocation Table (SBAT)

Most of the data blocks are occupied by the stream data.

2.1 Structure of OLE Block

An OLE header block is a block that contains the main information of an OLE file. The OLE file is divided into 512 bytes and the block number is assigned.



Figure 3. Structure of OLE Block

In this case, -1 block is a header block, and 0 ... n block is a data block. When opening an OLE file and reading it by 512 Bytes, it is like reading the header block. An OLE file is a file format used in various word documents. It consists of storage and streams similar to the FAT file system. Due to the limitations of the traditional file system, CFB(Compound File Binary) is structured according to the needs to put file and folder structure in one file [3, 4].

2.2 Storage and Stream

You can think of an OLE file as a normal file system. It consists of storage of folder role and stream of file role. It takes a very similar approach to the FAT file system to store storage and streams within OLE files. If you have a basic knowledge of the FAT file system, you can access it more easily. All information is stored in 512-byte sector units.



2.3 CFB(Compound File Binary) Components

To understand the compound file, you need to know the various components that make up the file. The Fig. 5 shows the structure of the compound file.



Figure 5. Structure of OLE Compound File

2.4 FAT(File Allocation Table) Array

A FAT Array is an array of 32-bit integers. The index of the array refers to the sector number, and the element corresponding to the index refers to the next index number. OLE file does not store the stream as a single line, just as a large file is not written to the disk from the beginning to the end. The storage is divided into sectors of 512 bytes size.

2.5 Directory Entry Array

A directory entry is where the meta information related to the storage stream is stored. A Directory Entry Array is an array of this Directory Entry. Directory Entry Arrays have meta information, so if you have Directory Entry information, you can easily access the information [7, 8, 9].

3. DESIGN OF VBA PASSWORD CHANGING MODULE

As we have seen in the related study, it is necessary to analyze the file structure of Excel in order to design Excel VBA password release module. The internal structure of Excel can be checked using SSViewer (Structure Storage Viewer).

⊡🍘 test2.xls	As HEX As T	ext A	s Picture	As F	RTF a	s HTML	1				
Document Summary Information	0x0000000	0908	1000	0006	0500	5438	CD07	D900	0200		^
Summary Information	0x00000010	0608	0000	E100	0200	B004	C100	0200	0000		
DocumentSummaryInformation	0x00000020	E200	0000	5000	7000	0200	0119	C215	C865		
	0x00000030	756E	2020	2020	2020	2020	2020	2020	2020	un	
📄 Workbook	0x0000040	2020	2020	2020	2020	2020	2020	2020	2020		
	0x00000050	2020	2020	2020	2020	2020	2020	2020	2020		
	0x0000060	2020	2020	2020	2020	2020	2020	2020	2020		
	0×00000070	2020	2020	2020	2020	2020	2020	2020	2020		
	0x0000080	2020	2020	2020	2020	2020	2020	2020	2020		
	0x00000090	2020	2020	2020	2020	4200	0200	B004	6101	B a .	
	0×000000A0	0200	0000	C001	0000	3D01	0200	0200	9000	=	
	0×00000080	0200	1100	1900	0200	0000	1200	0200	0000		
	0×00000000	1300	0200	0000	AF01	0200	0000	BC01	0200		
	0×00000000	0000	3000	1200	0000	0000	4038	C021	3800	= 0.8 1.8	
	0×000000E0	0000	0000	0100	F401	4000	0200	0000	8000	6	
	0×000000F0	0200	0000	2200	0200	0000	0 2 0 0	0200	0100		
	0x00000100	B701	0200	0000	DAGO	0200	0000	3100	1200		
	0x00000110	CBOO	0000	0800	9001	0000	0002	0000	0501		
	0x00000120	4100	5200	4900	4100	4000	3100	1200	C800	A R T A T. 1	
	0×00000130	0000	0800	9001	0000	0002	0000	0501	4100	д	
	0x00000140	5200	4900	4100	4000	3100	1200	CROOL	0000	D T A T. 1	
General	0x00000150	0800	9001	0000	0002	0000	0501	4100	5200	A D D	
Type Stream	0x00000160	4900	4100	4000	3100	1200	CBOOL	0000	0800	T A T. 1	
Name Workbook	0x00000170	90.01	0000	0002	0000	0501	4100	5200	4900	арт	
Path W	0100000180	4100	4000	2100	1300	C 8 0 0	0000	0200	9001	N T 1	
Size 129,654 B	0x00000190	0000	0003	0000	0501	4100	5200	4900	4100	A.D. 1	
Checksums	0x00000130	4000	2100	1400	2000	0000	5200	9001	0000	т. 1 П	
MD5 6C3731RE46EA990ARR4A3D7	0x00000180	0003	8100	0201	CBBS	COCE	3100	1400	CROO	1	
MD5 0C57218F40EA890A884A3D7	0x0000150	0000	0800	9001	0000	0003	8100	0201	CBB3		
	0x00000100	COCE	3100	1200	6801	0000	3500	9001	0000	1 b 5	

Figure 6. Structure of Test2.xlsm File

The Excel file has an OLE structure as shown Fig. 6 above. The OLE structure consists of a header and a block. For Excel files that use macros, the extension is in .xlsm format. If you open Excel's xlsm file format with Notepad, it is shown like Fig. 7 below. As shown in Fig. 7, the Excel file format has an unknown structure. This is because the Excel file is compressed. In the case of an Excel file that uses a macro, the format of the file is a compressed file.



In this way, macro-format files in Excel are basically compressed. Therefore, the macro file of Excel can be opened by changing the extension to compressed file format. The following Fig. 8 is the result of opening a macro format Excel file using a compression program.

By using these results, it is possible to recover the password error environment in VBA more conveniently by allowing the users to change the extension name etc. by using the program without manually performing this process.

When opening the above compressed file, there is a .bin file. The VBA program is stored in this file. In this paper, we change the .xlsm extension to .zip and open the modified compressed file using a compression program. This process is designed to be done through the program. If the compression program opens successfully, it looks for the .bin file. When you find the .bin file, it is extracted from the compressed file.

It then programs the .bin file extracted from the compressed file to use binary edits. It runs binary edits so that you can edit the .bin file extracted from the compressed file. It is written in a program so that these processes can be operated.

When you run binary editing of the .bin file, it modifies the DPB string in the .bin file. It changes DPB string to DPx string. It will then save the .bin file. This process is shown in Fig. 8.

Change extension .xlsm \rightarrow .zip
①
Run compressed file
Ū.
Open .zip file
Extract .bin file
$\overline{\mathbf{t}}$
Run binary edit
Open .bin file
$\overline{\Gamma}$
Search DPB string
Ϋ́
Change DPB →DP
$\overline{\Gamma}$
Save .bin file

Figure 8. Decryption Process

4. EXPERIMENT

I design the contents in this paper using C program. The operation of the implemented program is shown in Fig. 9.

	ajeet in been gritting set the					
윕파일 = C:WUsersWuser1WDesktopWtmp#	/test.zip					^
🔬 📾 📂 😳 🗉 🗉	test	.zip - 알집		- 1	- ×	
파일 홈 편집 도구	보기 도움말				⇔ 🕜	
	📂 📄 💼 파일삭제	III 아이콘 👘				
압축물기 바로실행 새로압축 입	·축열기 파일추가 🗔 이용번경	는 간단히 마운트 문자세히				
해제 실행 압축	파일	표시 마운트				
물더 필터 검색 🗡	파일명	압축크기	원본크기 압축	물 중류 문자 민소국		
v fill test.zip	docProps			로칠 디스크		
docProps	Content_Types1.xml	435	1.602 73	로컬 디스크 % XML 문서		
> • AI						
						~
					I	
					I	
					I	
					I	
	<				2	

Figure 9. Screen of Program Operation

Fig. 9 above shows the program screen that executes the proposed design in this paper. The program performs the function of changing the .xlsm file to a .zip file.

Figure 10. Screen of Compression Program Execution

····································
물더 필터 검색 × 파일명 압축크기 원본크기 압축률 통류 · jrels docProps · Jrels drawings · Jrels drawings · Jrels drawings · Jrels drawings · Jrels · Jre

Fig. 10 shows the screen of compression program execution. It extracts the vbaProject.bin file from the compressed program. The program is that binary edits can be performed on the extracted vbaProject.bin file.

~~~ II	- 10	$\sim$	0	1.2		m	0	- 1	ė. –	-	8		•																								
B		$\sim$	Ъ	9	5 (		4	_ <b>Q</b>	.≁ [:	<b>T</b>	U		ſ									_															
3CD23	6E	43	6F	бD	70	61	74	69	62	6C	65	33	32	зD	22	33	39	33	32	32	32	n	C o	m	P	a t	÷.	ъ	L e	: 3	2	- "	3	9 3	3 2	2	2
3CD38	30	30	30	22	OD	0A	43	4D	47	зD	22	32	30	32	32	38	43	43	41	39	30	0	0 0			C	M	G٠	- "	2	0	2 2	8	co	C A	9	¢
3CD4D	43	41	39	30	43	46	39	35	43	46	39	35	22	OD	OA	44	50	78	зD	22	44	C	A 9	0	С	FS	5	C	ē 9	5			D	P	- 2	••	I
3CD62	41	44	38	37	36	33	45	32	44	35	42	32	44	35	42	44	32	41	35	32	45	A	D 8	7	6	3 E	2	D	5 B	: 2	D	5 E	Ð	2 2	¥ 5	2	E
3CD77	35	42	36	39	44	37	42	37	39	32	31	31	38	43	37	35	32	30	36	41	46	5	ве	9	D	7 E	7	9	2 1	. 1	8	C 7	5	2 0	9 6	A	3
SCDSC	41	32	43	45	33	39	30	42	42	45	35	38	39	31	35	41	42	36	44	35	35	A	2 C	E	з	9 0	в	в	2 5	8	9	1 8	A	в	5 D	5	٤
3CDA1	39	36	36	46	42	35	43	31	44	31	22	OD	OA	47	43	зD	22	39	34	39	36	9	6 6	F	в	s c	: 1	D:	L "			GC		" 9	9 4	9	6
3CDB6	33	38	46	38	34	38	42	33	34	39	42	33	34	39	42	33	22	OD	0A	OD	OA.	3	8 F	8	4	8 E	3	4 :	эв	3	4	9 E	3		T		
SCDCB	5B	48	6F	73	74	20	45	78	74	65	6E	64	65	72	20	49	6E	66	6F	5D	oD	E	нo		t	Ε	: *	t e	e n	d	•	r	Ι	n 1	r 0	1	
SCDEO	OA	26	48	30	30	30	30	30	30	30	31	зD	7B	33	38	33	32	44	36	34	30		& H	0	0	0 0	0	0	0 1	. =	- (	3 8	3	2 1	5 6	4	¢
3CDF5	2D	43	46	39	30	2D	31	31	43	46	2D	38	45	34	33	2D	30	30	41	30	43	-	CF	. 9	0	- 1	. 1	C	e -	8	E	4 3	- 1	0 0	A C	0	c
SCEOA	39	31	31	30	30	35	41	7D	зв	56	42	45	зв	26	48	30	30	30	30	30	30	9	1 1	0	0	5 A	}	2.1	JВ	E	7	6 H	0 1	0 0	ہ د	0	c
3CE1F	30	30	OD	OA	OD	0A	5B	57	6F	72	6В	73	70	61	63	65	SD	OD	0A	54	68	0	0	Г		1	w	•	c k	: 3	P	ac	: e	3	T	т	ŀ
SCE34	69	73	57	6F	72	6В	62	6F	65	6В	зD	31	35	30	20	20	31	36	30	20	20	1	s W	0	r	k E	0	0	k =	: 1	5	ο,		1 4	5 0		
3CE49	39	30	38	2C	20	35	32	38	2C	20	43	OD	OA	53	68	65	65	74	31	зD	31	9	0 8			5 2	8		C	:		s r		• 1	- 1	=	1
3CE5E	32	30	2C	20	31	32	38	2C	20	38	37	38	2C	20	34	39	38	2C	20	OD	OA	2	ο,		1	2 8		1	3 7	8		4	9	8,			
3CE73	4D	6F	64	75	60	65	31	зD	31	32	39	20	20	34	32	20	20	31	31	38	30	м	o d	l u	1	e 1	-	1:	2 9			4 2		2	1 1	8	C
SCE88	20	20	35	35	32	20	20	OD	OA	52	65	67	72	65	73	73	69	6F	6E	зD	31		5	5	2			3	R e		r	e =		i c	o n	-	1
SCE9D	30	34	20	20	31	30	34	20	20	38	30	32	20	20	35	38	39	20	20	20	20	0	4,		1	0 4		1	во	2		5	8	э,			
3CEB2	37	38	20	20	37	38	20	20	35	30	34	20	20	35	36	33	20	20	OD	OA	4D	7	в,		7	в,		5 (	9 4			5 6	3				Þ
3CEC7	6F	64	75	6C	65	32	зD	31	30	34	2C	20	31	30	34	20	20	31	33	30	32	0	d u	1	•	2 =	1	0	۹,	$\square$	1	0 4		2	1 3	0	2
SCEDC	2C	20	34	33	33	2C	20	OD	OA	4E	40	52	65	67	72	65	73	73	69	6F	6E		4	3	3			1	I I	R	-	gz	-		s i	•	r
SCEF1	3D	31	35	36	20	20	31	35	36	20	20	31	33	32	38	20	20	35	33	35	2C	-	1 5	6		1	. 5	6		1	3	2 8		5	5 3	5	
3CF06	20	43	20	20	31	30	34	20	20	31	30	34	20	20	31	32	37	36	20	20	34		с,	1	1	0 4			1 0	4		1	2	7 4	δ,		4
3CF1B	38	33	20	20	OD	OA	01	00	00	4A	01	00	00	4B	01	00	00	40	01	00	00	8	з,			r	-		J -			ĸ		1	- L		
3CF30	FE	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	01	00	02	00	00	Þ	2 2	2	ÿ	9 S	2	ÿ :	2 2	ÿ	ÿ	2 5	ÿ	-	-		
3CF45	00	22	45	02	00	00	00	00	00	CO	00	00	00	00	00	00	46	00	00	00	00	É	" E	15		1	1	1	A	1	E T		1	F	Ť		
SCESA	00	00	00	00	12	45	02	00	00	00	00	00	CO	00	00	00	00	00	00	46	00			1 ·	T		-		-	-	à	_				-	

Figure 11. Screen of Starting Binary Edit on vbaProject.bin

After performing the binary edit of the vbaProject.bin file, it looks for the DPB string. After finding, it modifies the DPB string with DPx. The process of modifying with DPx. After

modifying the DPB string, it saves this file. When the vbaProject.bin file has been modified, the vbaProject.bin file is added to the compressed file.

파일 홈 편집 도구	보기 도움말					۵ (	8
값 <th></th> <th>이콘 500 단히 마운트 네히 마운트</th> <th></th> <th></th> <th></th> <th></th> <th></th>		이콘 500 단히 마운트 네히 마운트					
불더 필터 검색 × ♥ @ test.⊐ip ↓ docProps > ↓ 제	파일명 - rels -	양축크기 999 421 1.055 92,549 작업선택 월하는 작업 이 발접신 양제파 입양호/	원본크기 6,564 1,748 4,284 256,512 를 선택하십시오 로 알기(P) 축하기(N) 일에 추가하기(A) 사용(W)	압축률 85% 76% 75% 64%	종류 로컬 디스크 로컬 디스크 로컬 디스크 로컬 디스크 로컬 디스크 로컬 디스크 로컬 디스크 로컬 디스크 로컬 디스크 로 전 고 고 오 코 오 고 오 크 고 오 크 고 오 크 고 오 크 고 오 크 고 오 크 고 오 크 고 오 크 고 오 크 고 오 크 고 오 크 고 오 크 오 그 오 크 고 오 크 오 그 오 크 오 그 오 크 오 그 오 크 오 그 오 크 오 그 오 크 오 그 오 크 오 크	×	
	<			확인	ZIP 2.0 호환 취소		>

Figure 12. Add the Modified vbaProject.bin File to the Compressed File

After adding the modified vbaProject.bin file to the compressed file as shown in Fig. 12, it returns the .zip extension name back to .xlsm. This process is shown in Fig. 13 below.

파일 홈 공유 보기 압축 풀기	7]				~ 🔞
★         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲         ▲	명기 위치 · 위치 · · · · · · · · · · · · · · · ·	[] [] [] [] [] [] [] [] [] [] [] [] [] [	<ul> <li>▲ 월 일기 ▼</li> <li>▲ 월 편집</li> <li>▼ 월 히스토리</li> </ul>	모두 선택 선택 안 함 선택 영역 반전	
클립보드	구성	새로 만들기	열기	선택	
$\leftrightarrow \rightarrow \neg \uparrow $ $\blacktriangleright$ tmp			ٽ ~	tmp 검색	Q
A	이름		수정한 \	at w	유형 ^
☆ 마도 가기 ■ HL타 히며	📴 test.xlsm		2017-04	1-24 오후 2:32	ALZip ZIP File
	vbaProject.bin		2017-04	-24 오후 2:31	BIN 파일

Figure 13. Return Extension Name to .xlsm

After returning the extension name to .xlsm as shown in Fig. 13, you open the Excel file. If you open this Excel file, the file will open. Then you can set a new project property password in the security function area. I set "123" as the new password. Then, we store the Excel file. When we exit Excel and open it again, it works. When you run the VBA module, you must enter the password. At this time, if you input the newly changed password "123", you can check that the VBA module opened normally.

### 5. CONCLUSION

In this paper, I designed a system that could change the password when using Excel VBA. Especially, I designed it to be applicable to VBA environment using macros in Excel. It often happens that you cannot access important file data after setting the VBA password, due to password loss. The proposed program can make it possible for these users to access valuable data using more convenient VBA password change module.

I designed a program that makes it easier to release the password for the file that is set VBA password. The structure of this program is roughly divided into two parts. One is the process of extracting the .bin file from the Excel xlsm file. The other is the configuration process to modify the VBA password through changing the DPB keyword in the extracted .bin file.

I designed the program so that the above two processes can be used easily and conveniently. It changes the extension of the xlsm file to zip. After changing to zip, it starts a program that can decompress. When the decompression program is started, it extracts the vbaProject.bin file from the compressed file.

Next, it performs to edit binary of the vbaProject.bin file extracted from the compressed file. Then it runs the binary edit program to edit the vbaProject.bin file. When the binary edit program starts, it loads the vbaProject.bin file. If the vbaProject.bin file is opened normally, DPB string modification work will be done in this file. When the string modification is completed normally, the changed vbaProject.bin file is saved.

Once the modification of vbaProject.bin file has been completed, it will be saved in the original compressed file. When the vbaProject.bin file is newly saved, the extension zip of the compressed file is modified to the original extension xlsm.

### REFERENCES

- [1] Robin Abraham, Margaret Burnett, Martin Erwig, (2009) Spreadsheet Programming, Wiley Encyclopedia of Computer Science and Engineering.
- [2] Thomas A. Grossman, (2008) "Source Code Protection for Applications Written in Microsoft Excel and Google Spreadsheet", arXiv.org > cs > arXiv:0801.4774 Software Engineering.
- [3] Jocelyn Paine, (2009) "Documenting Spreadsheets with Pseudo-Code: an Exercise with Cash-Flow and Loans", arXiv.org > cs > arXiv:0908.1187 Software Engineering.
- [4] Larry J. LeBlanc, (2007) "Implementing Large-Scale Optimization Models in Excel Using VBA", *Interface*, Volume 37, Issue 4, pp. 370 382.
- [5] J Pichitlamken Email author, S Kajkamhaeng, P Uthayopas, R Kaewpuang, (2011) "High performance spreadsheet simulation on a desktop grid", *Journal of Simulation*, Volume 5, Issue 4, pp 266–278.
- [6] Michael R. Bartolacci, Larry J. LeBlanc, Yasanur Kayikci, Thomas A. Grossman, (2012) Optimization Modeling for Logistics: Options and Implementations, *Journal of Business Logistics*, Volume 33, Issue 2, pp. 118–127.
- [7] Mangesh S Gharote and Dr. Sachin Lodha, (2010) "ExBOS Excel Based Optimization Software Framework", SOM 2010 XIV Annual International Conference of the Society of Operations Management, pp.1-8.
- [8] http://www.reversenote.info/ole-file-format/

[9] Pichitlamken1, S Kajkamhaeng, P Uthayopas and R Kaewpuang, (2011) "High performance spreadsheet simulation on a desktop grid", *Journal of Simulation*, no. 5, pp. 266–278.