available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/diin

Digital Investigation

Lessons learned from the construction of a Korean software reference data set for digital forensics

Kibom Kim[a], *Sangseo Park*[a,*], *Taejoo Chang*[a], *Cheolwon Lee*[a], *Sungjai Baek*[b]

[a]*The Attached Institute of ETRI, Deajeon, Republic of Korea*
[b]*Supreme Prosecutor's Office, Seoul, Republic of Korea*

Keywords:
Digital forensic
Reference data set (RDS)
Hash data set
National software reference library (NSRL)
Korean RDS (KRDS)

ABSTRACT

This paper analyzes the National Software Reference Library Reference Data Set (NSRL RDS) and constructs a Korean RDS (KRDS) based on it. The fact that NSRL RDS offers the largest amount of hash data sets has led to its widespread adoption. However, the effectiveness analysis of NSRL RDS indicates that there are both duplicate/obsolete data that can be eliminated and unused metadata that can be deleted. Moreover, language-specific software and domestic software that has been widely used for years have to be added. Bearing these issues in mind, we develop a strategy and model for both importing effective NSRL RDS and adding Korea-specific data sets. We then construct initial KRDS using proprietary software designed to manage the entire process of analysis and construction. Lessons learned during this work are believed to be useful for those who need to construct their own RDS (based on NSRL RDS or not) and later upgrade of the NSRL RDS.

1. Introduction

Effective and efficient digital forensic investigation requires narrowing the scope using known files, as digital storage capacity increases (NISTa; Mead 2006). Realization of this goal requires either focusing on specific file(s) of interest or eliminating unnecessary one(s). An example of the former is the investigation of child pornographies or violations of intellectual property, such as unauthorized installation of software. An example of the latter is computer virus examination, which requires elimination of normal, uninfected, operating system, and application software files from the investigation list.

Hash values have been used for identification of specific known files (White and Ogata, 2005). A few institutions have constructed their RDS using hash values of the file's content. The most prominent RDS is the NSRL RDS which is distributed by US National Institute of Standards and Technology (NIST). NIST extracts key information, builds metadata, and computes hash values for widely used commercial software. It uses SHA-1, MD5, and CRC32 algorithms. The most recent version is RDS 2.24 released on March 2009 on the Internet for public use, and it contains 15,722,076 unique SHA-1 values for 50,121,818 files.

(Farrell et al., 2008) examined NSRL RDS from the viewpoint of high speed search for deciding whether specified file is known by the RDS or not by using Bloom Filters. In this study, how many files are covered in the NSRL RDS also analyzed by comparing hash values of NSRL RDS and newly installed Windows respectively. However, the limitation of the research is it assumed the consistency of NSRL RDS.

Due to the huge size of NSRL RDS, it would be more efficient if an investigator can extract some appropriate subset of RDS based on specific product, language, or operating system

[*] *Corresponding author*. The Attached Institute of ETRI, 909 Jeonmin-Dong, Deajeon, 305-390 Republic of Korea.
E-mail addresses: kibom@ensec.re.kr (K. Kim), spark@ensec.re.kr (S. Park), tchang@ensec.re.kr (T. Chang), cheolee@ensec.re.kr (C. Lee), bryan.baik@gmail.com (S. Baek).

1742-2876/$ – see front matter © 2009 Digital Forensic Research Workshop. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2009.06.005

version in accordance with each case. In order for this, it is required to check the effectiveness of NSRL RDS in terms of file record and metadata.

Interest in digital forensics in Korea has dramatically increased in the past few years due to several nation-wide issues. It is thought that import of NSRL RDS, which offers the largest amount of data of any RDS, is a simple and time-saving alternative rather than constructing a new RDS from scratch. However, upon careful examination, it was found that some of NSRL RDS are inappropriate for direct import and subsequent use due to reasons of integrity. In addition, these analyses also identified that more than half of the data sets associated with the Windows XP, the most prevailing operating system for PC, and software supporting Hangul, the Korean, are also inadequate for the same reason. Moreover, NSRL RDS rarely contains data sets pertaining to domestic software. As a result, construction of KRDS has been attempted based on two phases: tailoring and importing NSRL RDS to form a basis, and then adding Korea-specific data sets onto it.

The aim of this research is to inspect which NSRL RDS data are adequate for import and to discover how efficiently Korea-specific data sets can be added. We analyze NSRL RDS version 2.17 and find about 60% of the metadata and 20% of hash data sets to be insufficient. We also suggest a new methodology of efficient construction of data sets. Based on the analysis and method, we develop a supporting tool and finally construct about 34.7 million hash data sets.

The rest of this paper consists of three parts: a model for constructing KRDS, analysis of NSRL RDS and suggestion for improvement of RDS construction, and result of the KRDS construction.

2. A model for KRDS construction

Our strategy for constructing KRDS is *tailor-and-add*, in short. Specifically, it means importing NSRL RDS first and then adding Korea-specific information with regard to that data while verifying effectiveness of the data sets at each phase. This strategy is based on the *majority*, *specialty*, *integrity*, and *efficiency* principles.

- NSRL RDS offers the largest amount of data among other data sets which include Microsoft's products that occupy most of the personal computer market in Korea. Therefore, NSRL RDS has to be selected for the base of KRDS.
- Some software has been widely used only in Korea. This implies that they may not be reflected adequately in NSRL RDS. Thus, data sets unique to Korea need to be constructed. One such category of software is those supporting Hangul, such as Windows XP Korean and Microsoft's Office Korean. Another is domestic software developed and widely used in Korea over the past few years. Examples include the Hangul word processor (HWP) (HaanSoft), the compression/decompression software Alzip (ESTSoft) and the anti-virus software V3 (AhnLab). A third type is related to freeware and/or shareware, such as Google Earth and Total Commander.
- Inconsistent data should be excluded from the KRDS. Therefore, not only NSRL RDS but also Korea-specific data sets need to be examined for their integrity and then filtered out if they are inconsistent.
- Files proved to be improper for inclusion in KRDS should be filtered out from the beginning.

Based on the above strategies and principles, we built a model for construction of KRDS (Fig. 1).

The model is composed of two phases. Phase I includes importing NSRL RDS into KRDS. Here, only effectiveness-verified data sets are imported. In Phase II, generating and appending new hash data for Korea-specific software is undertaken. The core of this phase is specialty principle for inclusion of Korea-specific software. Integrity and efficiency principles are applied so as to exclude unnecessary files and hash data as well.

3. Effectiveness analysis and filtering methods

3.1. NSRL RDS data formats

The NSRL RDS data set is composed of five logical records NISTb: Version Record, Product Record (PR), Operating System Record (OSR), Manufacturer Record (MR), and File Record (FR). The Version Record contains the version of the current NSRL RDS and overall SHA-1 value. The PR contains the product code (pcode), product name, product version, manufacturer code (mfgcode), operating system code (oscode), and language, etc. The OSR contains the oscode, name, version of operating system, etc. The MR contains the mfgcode and name. Finally, the FR contains the SHA-1, MD5, CRC32, file name (fname), file size (fsize), pcode, oscode, and special code.

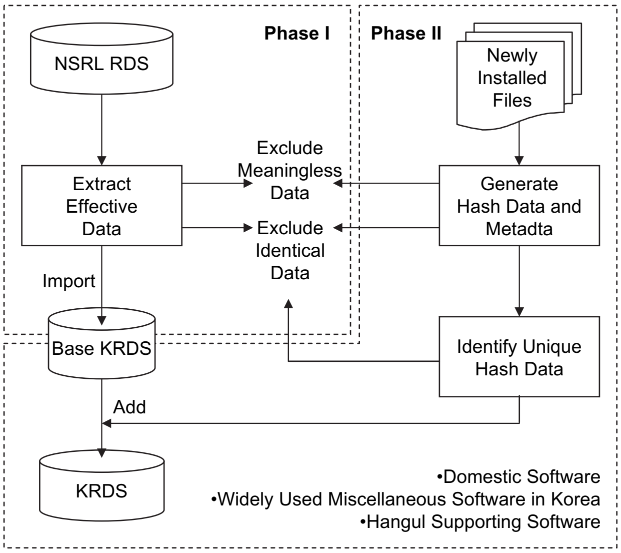

Fig. 1 – KRDS construction model.

3.2. *RDS effectiveness analysis*

Let RDS be a set composed of PR, OSR, MR, and FR as follows:

$$\begin{aligned}&\mathrm{RDS}{:=}\{\mathrm{FR}\}\bigcup\{\mathrm{PR}\}\bigcup\{\mathrm{OSR}\}\bigcup\{\mathrm{MR}\},\\&\text{where } \mathrm{FR}{:=}\{\mathrm{fr}_i|1\le i\le n_{\mathrm{FR}}\},\ \mathrm{PR}{:=}\{\mathrm{pr}_j|1\le j\le n_{\mathrm{PR}}\},\\&\mathrm{OSR}{:=}\{\mathrm{osr}_k|1\le k\le n_{\mathrm{OSR}}\},\ \mathrm{MR}{:=}\{\mathrm{mr}_h|1\le h\le n_{\mathrm{MR}}\}.\end{aligned}$$

The RDS effectiveness check is composed of three steps. The purpose of first step is to filter out hash data that do not have any matching metadata. It is thus necessary to examine the hash data from the viewpoint of metadata to check if the oscode and pcode in each fr_i exist in the associated metadata. This can be described as follows:

$$\begin{aligned}&\mathrm{FR}^{(1)}{:=}\{\mathrm{fr}_i{:=}{<}\mathrm{fr}_{i,\mathrm{pcode}},\ \mathrm{fr}_{i,\mathrm{oscode}},\\&\ldots{>}\in \mathrm{FR}^{(0)}|\mathrm{fr}_{i,\mathrm{pcode}}=\mathrm{pr}_{j,\mathrm{pcode}}\in \mathrm{PR}^{(0)}\text{ for some }1\le j\le n_{\mathrm{PR}},\\&\mathrm{fr}_{i,\mathrm{oscode}}=\mathrm{osr}_{k,\mathrm{oscode}}\in \mathrm{OSR}^{(0)}\text{ for some }1\le k\le n_{\mathrm{OSR}}\},\end{aligned}$$

where $\mathrm{FR}^{(0)}$ is initial set of FR, $\mathrm{fr}_{i,\mathrm{pcode}}$ and $\mathrm{pr}_{j,\mathrm{pcode}}$ are pcode in fr_i and pr_j respectively. $\mathrm{fr}_{i,\mathrm{oscode}}$ and $\mathrm{osr}_{k,\mathrm{oscode}}$ are oscode in fr_i and osr_k respectively.

The purpose of second step is to filter out metadata that do not have any associated hash data. This step is to examine metadata from the hash data point of view to check if each data in metadata (PR, OSR and MR) is associated with matching information fr_i in $\mathrm{FR}^{(1)}$. It can be described as follows:

$$\begin{aligned}&\mathrm{PR}^{(1)}{:=}\{\mathrm{pr}_j{:=}{<}\mathrm{pr}_{j,\mathrm{pcode}},\ \ldots{>}\in \mathrm{PR}^{(0)}|\mathrm{pr}_{j,\mathrm{pcode}}=\mathrm{fr}_{i,\mathrm{pcode}}\in \mathrm{FR}^{(1)}\\&\text{for some }1\le i\le n_{\mathrm{FR}}\},\\&\mathrm{OSR}^{(1)}{:=}\{\mathrm{osr}_k{:=}{<}\mathrm{osr}_{k,\mathrm{oscode}},\\&\ldots{>}\in \mathrm{OSR}^{(0)}|\mathrm{osr}_{k,\mathrm{oscode}}=\mathrm{fr}_{i,\mathrm{oscode}}\in \mathrm{FR}^{(1)}\text{ for some}\\&1\le i\le n_{\mathrm{FR}}\},\\&\mathrm{MR}^{(1)}{:=}\{\mathrm{mr}_h{:=}\mathrm{mr}_{h,\mathrm{mfgcode}},\\&\ldots{>}\in \mathrm{MR}^{(0)}|\mathrm{mr}_{h,\mathrm{mfgcode}}=\mathrm{pr}_{j.\mathrm{mfgcode}}\in \mathrm{PR}^{(1)}\text{ for some}\\&1\le j\le n_{\mathrm{PR}}\}.\end{aligned}$$

The purpose of third step is to discard duplicate records so as to satisfy the following condition:

$$\begin{aligned}&{<}\ \mathrm{fr}_{i,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{i,\mathrm{fname}},\ \mathrm{fr}_{i,\mathrm{pcode}},\ \mathrm{fr}_{i,\mathrm{oscode}}{>}\neq{<}\mathrm{fr}_{j,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{j,\mathrm{fname}},\\&\quad \mathrm{fr}_{j,\mathrm{pcode}},\ \mathrm{fr}_{j,\mathrm{oscode}}{>},\text{ if } i\neq j.\end{aligned}$$

3.3. *Filtering meaningless files and duplicated records when generating hash data*

Software has to be first installed on base machines so that addition of Korea-specific information onto the base KRDS is possible. The purpose of the hash data generation in Phase II is two-fold: (1) filter out files that do not need to have hash values from the beginning, and (2) for files having hash values, discard identical ones having the same hash value. Therefore, in order to fulfill the first purpose, before generating hash value, each file should be excluded based on the following:

- if its file size is "0"
- if it is a temporary file used for installation purpose or software's running, or
- if it is being altered when the software runs.

The remaining files are adequate ones to calculate hash values. To achieve second purpose, record(s) should be excluded so as to satisfy the following condition:

$${<}\ \mathrm{fr}_{i,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{i,\mathrm{fname},}\ \ldots{>}\neq{<}\mathrm{fr}_{j,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{j,\mathrm{fname},\ldots}{>},\text{ if } i\neq j.$$

An algorithm to identify meaningful files and exclude identical records for a software can be described as follows:

```
Candidate file $\mathrm{CF}^{(0)}{:=}\{\mathrm{cf}_i|\mathrm{cf}_i$ is a file existed in base machine for some $1\le i\le n_{\mathrm{CF}}^{(0)}\}$
Create the metadata for a software($\mathrm{sw}_j$) if it doesn't exist
Install $\mathrm{sw}_j$ on base machine
$\mathrm{CF}^{(1)}=\{\mathrm{cf}_i|\mathrm{cf}_i$ is a file existed in the machine after $\mathrm{sw}_j$ installation for some $1\le i\le n_{\mathrm{CF}}^{(1)}\}$
$\mathrm{CF}^{(2)}:=\mathrm{CF}^{(1)}-\mathrm{CF}^{(0)}$
$\mathrm{CF}^{(3)}:=\mathrm{CF}^{(2)}-\{\mathrm{cf}_i\in \mathrm{CF}^{(2)}|\mathrm{SIZE}(\mathrm{cf}_i)=0$ or $\mathrm{FEATURE}(\mathrm{cf}_i)=\mathrm{TempForInstallOrRun}|\mathrm{AlterWhenRun}$ for all $1\le i\le n_{\mathrm{CF}}^{(2)}\}$
KRDS file record $\mathrm{KR}^{(1)}{:=}\{\mathrm{fr}_i{:=}{<}\mathrm{fr}_{i,\mathrm{fname}},\ \mathrm{fr}_{i,\mathrm{fsize}},\ \mathrm{fr}_{i,\mathrm{crc32}},\ \mathrm{fr}_{i,\mathrm{md5}},\ \mathrm{fr}_{i,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{i,\mathrm{sha\text{-}256}},\ \mathrm{fr}_{i,\mathrm{pcode}},\ \mathrm{fr}_{i,\mathrm{oscode}}{>}$ for all $1\le i\le n_{\mathrm{CF}}^{(3)}\}$
$\mathrm{KR}^{(2)}{:=}\mathrm{KR}^{(1)}-\{\mathrm{fr}_i\in \mathrm{KR}^{(1)}|{<}\mathrm{fr}_{i,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{i,\mathrm{fname},}\ \ldots{>}={<}\mathrm{fr}_{j,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{j,\mathrm{fname},\ldots}{>}$ for some $1\le i<j\le n_{\mathrm{KR}}^{(1)}\}$
```

3.4. *Filtering operating system independent records for identification of unique ones*

This step discards operating system independent records associated with commercial software, freeware, or shareware. The target to be discarded is a file that runs on more than two operating systems. A record is selected when another record has same sha-1, file name, and product code but different operating system code with it. This is required for each record in KRDS to confirm its exclusiveness before being inserted into KRDS. Thus the record fr_i should be discarded if it satisfies the following condition:

$$\begin{aligned}&{<}\ \mathrm{fr}_{i,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{i,\mathrm{fname}},\ \mathrm{fr}_{i,\mathrm{pcode}}{>}={<}\mathrm{fr}_{j,\mathrm{sha\text{-}1}},\ \mathrm{fr}_{j,\mathrm{fname}},\ \mathrm{fr}_{j,\mathrm{pcode}}{>}\\&\quad\text{but }\mathrm{fr}_{i,\mathrm{oscode}}\neq \mathrm{fr}_{j,\mathrm{oscode}},\text{ if } i\neq j,\end{aligned}$$

where $\mathrm{KR}^{(\mathrm{KRDS})}$ is Korea-specific hash data records located in KRDS already and $n_{\mathrm{KR}}^{(\mathrm{KRDS})}$ is the number of $\mathrm{KR}^{(\mathrm{KRDS})}$, $\mathrm{fr}_i\in \mathrm{KR}^{(2)}$ for some $1\le i\le n_{\mathrm{KR}}^{(2)}$, $\mathrm{fr}_j\in \mathrm{KR}^{(\mathrm{KRDS})}$ for all $1\le j\le n_{\mathrm{KR}}^{(\mathrm{KRDS})}$.

Then operating system code of fr_j ($\mathrm{fr}_{j,\mathrm{oscode}}$) should be changed to WINCOMMON, if it is not set as such, in order to indicate that the file can be run on several Windows operating systems. Finally, we obtain $\mathrm{KR}^{(3)}$.

4. KRDS construction

4.1. *KRDS data format and software*

KRDS data formats should be based primarily on those of NSRL RDS since it constitutes a basis for KRDS. Therefore, KRDS data formats should be the same as or a superset of NSRL RDS. KRDS is composed of four types of records: Product Record, Operating System Record, Manufacturer Record, and File Record. Here, File Record is designed to contain additional SHA-256.

For efficient construction of KRDS, proprietary software ("KRDS Manager") was developed. This software was

Table 1 – Number of useful hash data in NSRL RDS.

ISO file contains hash data	Total ($n_{FR}^{(0)}$)	Effective ($n_{FR}^{(1)}$)	Ineffective	
			Operating system code mismatch	Product code mismatch
Non-English software	12,253,935	11,983,710	270,225	204
Operating system	4,053,716	3,238,424	815,292	83
Application software	21,700,239	19,869,329	1,830,910	2919
Image & graphics software	5,095,602	4,978,200	117,402	177
Sum	43,103,492	40,069,663	3,033,829	3383

Table 2 – Number of useful metadata in NSRL RDS.

Metadata		Total	Effective	Ineffective
PR	Total	17,504 ($n_{PR}^{(0)}$)	7090 ($n_{PR}^{(1)}$)	10,414
	Windows XP	61	24	37
	Language (Korean)	99	40	59
Operating system		340 ($n_{OSR}^{(0)}$)	340 ($n_{OSR}^{(1)}$)	0
Manufacturer		1277 ($n_{MR}^{(0)}$)	1277 ($n_{MR}^{(1)}$)	0

developed using Visual Studio 2005 C++ and MS-SQL 2005 Standard Edition was used as a database engine.

4.2. *Analysis and import of NSRL RDS*

Using the effectiveness analysis method described in Section 3.2, the NSRL RDS can be analyzed and imported to form the base KRDS.

4.2.1. *Effective hash data*

Table 1 shows the results of applying the first step of the effectiveness analysis method described in Section 3.2. According to the table, among about 43 million hash data, about 7% and 0.008% lack the operating system code and product code, respectively.

It implies that 7% of the hash data cannot be imported because it has no associated operating system information. Most of the operating system codes in the hash data were "UNKNOWN", "UNK", or "GEN". This means that the operating system on which the file is installed is indeterminable. It is critical because NSRL RDS contains information on software running on DOS, Windows (spans from 3.1× to Vista), OS2, MAC OS X, PocketPC2002, Linux and Unix. It was also discovered that records that lack a product code always lack an operating system code, but not vice versa.

4.2.2. *Effective metadata*

In Table 2, analysis of Product Record $PR^{(0)}$ using $FR^{(1)}$ reveals that about 59.5% have no relevant hash values. This table also shows that metadata on operating system and manufacturer are all effective without loss.

Among Product Records, there are as many as 61 metadata on Windows XP but 60.6% of these metadata have no associated hash data. Among 99 metadata on products supporting Korean, 59.5% have no associated hash data as well. These imply that about 60% of the metadata pertaining to Windows XP and software supporting Korean are obsolete. Due to this problem, it is very difficult to extract a subset of RDS by metadata.

Table 3 – Number of exclusive records in NSRL RDS.

Total ($n_{FR}^{(1)}$)	Effective	Ineffective	
	Exclusive ($n_{FR}^{(2)}$)	Identical	Major field overlapped
40,069,663	34,483,804	2,383,340	3,202,519

4.2.3. *Duplicated hash data*

Based on the analysis of duplicated records in $FR^{(1)}$, Table 3 shows that about 13.9% of $FR^{(1)}$ are duplicated. Among duplicated ones, about 5.9% are identical and about 8% are overlapped in major fields: sha-1, file name, product code, and operating system code. Each overlapped record is assumed to have the same hash value because they are logically considered to be the same file. This result does not matter if an investigator only uses hash values. However, to create and maintain duplicate hash data is undesirable when considering that RDS are continuously accumulated.

4.2.4. *Import of NSRL RDS*

Resulting from the aforementioned analyses, we come to the conclusion that only about 45.5% of metadata and about 80% of hash data are useful for our purpose. Based on our constructed model and the NSRL RDS analysis, effectively 8707 out of 19,121 metadata and 34,483,804 out of 43,103,492 hash data have imported from NSRL RDS to form a base KRDS.

4.3. *Adding Korea-specific information*

In order to add Korea-specific information into the base KRDS, 83 software packages were chosen. Each package was

Table 4 – Number of effective hash data records for Korea-specific information.

Software	Total ($n_{CF}^{(2)}$)	Effective ($n_{KR}^{(2)}$)	Ineffective
Operating system	251,454	180,079	71,375
Commercial software	168,988	156,077	12,911
Freeware/shareware	25,829	25,263	566
Sum	446,271	361,419	84,852

Table 5 – Number of effective hash data records for Windows Vista.

Operating system	Total ($n^{(2)}_{CF}$)	Effective ($n^{(2)}_{KR}$)	Ineffective
Windows Vista home basic	36,274	25,282	10,992
Windows Vista business	38,769	27,057	11,712

Table 6 – Number of exclusive records in Korea-specific hash data set.

Software	Total ($n^{(2)}_{KR}$)	Effective ($n^{(3)}_{KR}$)	Ineffective
Operating system	180,079	180,079	0
Commercial software	156,077	70,426	85,651
Freeware/shareware	25,263	10,384	14,879
Sum	361,419	260,889	100,530

installed one-by-one, and a number of files were selected to compute hash values in accordance with the algorithm described in Section 3.3.

4.3.1. *Candidate software*

For Korea-specific data sets, we chose software which constitutes the most common computing environment used in Korea. They consist of 15 MS Windows operating systems that support Korean, 19 commercial software, and 49 freeware or shareware packages top-ranked in popular software download sites.

4.3.2. *Filtering meaningless files and duplicated records*

The total number of files extracted from 83 target software packages was 446,271 excluding the number of files that cannot be installed on specific operating system versions. All operating systems described above were installed and examined to acquire the entire fingerprint. However, commercial software, freeware, and shareware were installed and checked on Windows 2000 Professional Service Pack 4, Windows XP Professional Service Pack 2, and Windows Vista Basic Home respectively, which are believed to be the most popular operating system environment in Korea. Table 4 shows that 19% of the installed files proved to be ineffective for insertion into KRDS when applied the filtering mechanism suggested in Section 3.3.

In particular, for the case of Windows Vista, Table 5 shows that more than 30% of their files were discarded.

4.3.3. *Filtering operating system independent records*

For the initial hash record set $KR^{(2)}$ for Korea-specific software, each record has been compared with those that already exist in KRDS using the method described in Section 3.4 (Table 6). Here, about 72.1% of the hash data that proved to be exclusive should be inserted into the KRDS.

4.3.4. *Final shape of KRDS*

Table 7 shows the final KRDS that contains 9053 metadata and 34,744,693 hash data. Among them, 8707 metadata and 34,483,804 hash data were imported from NSRL RDS, and 346 metadata and 260,889 hash data were added for association with Korea-specific information.

5. Conclusion

In this paper, we examined the effectiveness of NSRL RDS and built a Korean RDS based on it. NSRL RDS is the most well-known and widespread used hash set. However, there has been no published research on the analysis of NSRL RDS. We found that only 45.5% of metadata (8707 out of 19,121) and 80% of hash data (34,483,804 out of 43,103,492) are useful in NSRL RDS and imported these effectiveness-proved ones. We then added 346 metadata and 260,889 hash data for Korea-specific information from 83 software packages deemed the most important and popular in Korea. During the process, we discovered that around 30% of the hash records regarding operating system details were duplicated, and as such could be cut to compact RDS. It was also shown that about 55.4% of commercial software and freeware/shareware (100,530 out of 181,340) could be discarded as they are independent on operating system.

Through this study, we were able to understand that NSRL RDS, though it offers massive data sets, needs to be tailored by eliminating the duplicate/obsolete metadata and hash data. We believe that this study is significant as this is the first academic trial to inspect the effectiveness of NSRL RDS and proposes an efficient way of importing and constructing tailored one. Furthermore, it is evident that our strategy and model will be useful for countries using their native languages.

Table 7 – Number of data in KRDS.

Source		Total	NSRL RDS	Korea-specific
Metadata	Product	7486	7090	296
	Operating system	1312	1277	35
	Manufacturer	255	340	15
	Partial sum	9053	8707	346
Hash data	Operating systems	34,744,693	34,483,804	180,079
	Commercial software			70,426
	Freeware/sharewares			10,384
	Partial sum		34,483,804	260,889

REFERENCES

AhnLab Inc., V3 Internet Security 2007, http://kr.ahnlab.com/info/productinfo/html/v3_is2007_pl.html.

ESTsoft Corp., ALTools, http://www.altools.com/.

Farrell Paul, Garfinkel Simson L., White Douglas. Practical applications of bloom filters to the NIST RDS and hard drive Triage. 2008 Annual Computer Security Applications Conference (ACSAC), 2008; pp. 13–22.

Haansoft, Haansoft Office 2007, http://www.haansoft.com/.

Mead Steve. Unique file identification in National Software Reference Library. Digital Investigation 2006;3(3):138–50.

NISTb, Data formats of the NSRL Reference Data Set (RDS) distribution. http://www.nsrl.nist.gov/documents/Data-Formats-of-the-NSRL-Reference-Data-Set-12.pdf.

NISTa, National Software Reference Library, http://www.nsrl.nist.gov.

White Douglas, Ogata Michael. Identification of known files on computer systems, AAFS annual meeting, New Orleans, LA, 2005.

Kibom Kim is a Senior Member of Engineering Staff at the Attached Institute of ETRI, Korea since Aug. 2004. He received B.S.E. degree in information engineering from Cheju National University in 1994, and M.Sc. and Ph.D. degrees in computer science from Korea University in 1996 and 2001, respectively. From Jan. 2001 to July 2004 he was a general manger for development of ECO Inc., Korea. His research interests include digital forensics, computer and network security, and software testing.

Dr. Sangseo Park is a Principal Member of Engineering Staff at the Attached Institute of Electronics and Telecommunications Research Institute (ETRI) in Korea. He previously worked for the Institute for Defense Information Systems (IDIS) and Agency for Defense Development (ADD) as a Senior Researcher before joining current institute. His research interests include information systems security strategy in organizations, information warfare, protection of organizational information, and digital forensics.

Taejoo Chang received B.S.E. degree in electrical engineering from Ulsan University in 1982, and M.S.E. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1990 and 1998, respectively. From Jan. 1982 to Jan. 2000 he was a Member, Senior Member of Technical Staff of Agency for Defense Development (ADD), Korea. In Feb. 2000, he joined the Attached Institute of ETRI, Korea, where he is now a Research Fellow. His research interests include design of cryptographic processors, digital forensics, and statistical signal processing.

Cheolwon Lee received M.Sc. in computer engineering from Chung-Ang University in 1989. He previously worked for the ETRI and KISA as a Senior Researcher. In 2000, he joined the Attached Institute of ETRI, Korea, where he is now a Research Director and Principal Member of Engineering Staff. His research interests include computer and network security, evaluation criteria of security system, and digital forensics.

Sungjai Baik received a BS degree in Economics from Chunbuk National University in 1995. He joined the Supreme Prosecutor's Office in 1995 and has worked as a manager in the Digital Forensics Team. His expertise lies in digital forensic analysis, industrial espionage and white collar crime.