CYBERLAWS 2011

The Second International Conference on Technical and Legal Aspects of the e-Society

February 23-28, 2011 - Gosier

Guadeloupe, France

CYBERLAWS 2011 Editors

Lasse Berntzen, Vestfold University College - Tønsberg, Norway

Adolfo Villafiorita, Fondazione Bruno Kessler/ University of Trento, Italy

Mark Perry, University of Western Ontario/Faculty of Law/ Faculty of Science - London, Canada

CYBERLAWS 2011

Foreword

The Second International Conference on Technical and Legal Aspects of the e-Society [CYBERLAWS 2011], held between February 23-28, 2011 in Gosier, Guadeloupe, France, followed the multiplications of cybercrime acts concerning privacy and anonymity in the information society. In accordance with the principle of freedom of expression and the right to privacy, the use of anonymity is legal. Users can access data and browse anonymously so that their personal details cannot be recorded and used without their knowledge by any other entity, including another user. As there are situations were content/information providers might wish to remain anonymous for legitimate purposes, they should not be required to justify anonymous use. The dangerous side of the legal anonymity is the shadow for illegal, damaging, and not easily to sue individuals and actions. Corporate and individual hassle, corporate and individual frauds, threats, and impersonations are only a few of these actions. While privacy, anonymity and freedom of speech are achieved rights, there is a vacuum on education, punishments, and laws that can be easily applied at the same speed with which a cybercrime propagates. Applying the Civil Court legislation is tedious and naturally, too late to timely repair the damage and prevent its quick propagation. There is a need for special laws to either prevent or quickly reprimand. In this case, the identity must be legally and undoubtedly validated. There is a need of internationally adopted guidelines to be applied by victims. There is a need for harmonization between national laws for a new era of eDemocracy.

The second CYBERLAWS 2011 provided a forum where researchers, government representatives, international bodies, law enforcement organisms and special groups were able to present recent lessons learned, use cases, and set the priorities on problems and directions related to the anonymity, privacy, identity, and laws that should or are governing their legal use.

We take here the opportunity to warmly thank all the members of the CYBERLAWS 2011 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to CYBERLAWS 2011. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the CYBERLAWS 2011 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that CYBERLAWS 2011 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the area of technical and legal aspects of the e-Society.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the beautiful surroundings of Gosier, Guadeloupe, France.

CYBERLAWS 2011 Chairs

Lasse Berntzen, Vestfold University College - Tønsberg, Norway
Mark Perry, University of Western Ontario/Faculty of Law/ Faculty of Science - London, Canada
Adolfo Villafiorita, Fondazione Bruno Kessler/ University of Trento, Italy
Lilian Edwards, University of Sheffield, UK
Claire Chambers, University of West England - Bristol, UK

CYBERLAWS 2011

Committee

CYBERLAWS Advisory Committee

Lasse Berntzen, Vestfold University College - Tønsberg, Norway
Mark Perry, University of Western Ontario/Faculty of Law/ Faculty of Science - London, Canada
Adolfo Villafiorita, Fondazione Bruno Kessler/ University of Trento, Italy
Lilian Edwards, University of Sheffield, UK
Claire Chambers, University of West England - Bristol, UK

CYBERLAWS 2011 Technical Program Committee

Bechara Al Bouna, Antonine University - Baabda, Lebanon
Sylvia Archmann, European Institute of Public Administration (EIPA)- Maastricht, The Netherlands
Ilija Basicevic, University of Novi Sad, Serbia
Farid Enrique Ben Amor, Coro Center for Civic Leadership - Southern California / Cornell University, USA
Lasse Berntzen, Vestfold University College - Tønsberg, Norway
Carlo Blundo, Università di Salerno, Italy
Clare Chambers, UWE Bristol Law School, UK
Clelia Colombo, Government of Catalonia, Spain
Kevin Curran, University of Ulster, UK
Glenn S. Dardick, Longwood University, USA / Edith Cowan University, Australia
Jana Dittmann, Otto-von-Guericke University Magdeburg, Germany
Noella Edelmann, Centre for E-Government, Danube University Krems, Austria
Lilian Edwards, University of Sheffield, UK
Steven Furnell,University of Plymouth, UK
Matjaz Gams, Jozef Stefan Institute-Ljubljana, Slovenia
Huong Ha, TMC Business School / TMC Academy, Singapore
Rajesh Ingle, PICT, Pune, India
Evika Karamagioli, Gov2U - Athens, Greece
Ah-Lian Kor, Leeds Metropolitan University, UK
Rudolf Legat, Umweltbundesamt GmbH, Austria
Diego Liberati, Italian National Research Council, Italy
Ralf Lindner, Fraunhofer Institute for Systems and Innovation Research (ISI)-Karlsruhe, Germany
Edith Maier, FHS St. Gallen (University of Applied Sciences), Switzerland
Thomas Margoni, Faculty of Law - Faculty of Science, University of Western Ontario, Canada
Alok Mishra, Atilim University - Ankara, Turkey
Joon S. Park, Syracuse University, USA
Peter Parycek, Donau-UniversitKrems, Head of Center for E-Government, Austria
Mark Perry, University of Western Ontario/Faculty of Law/ Faculty of Science - London, Canada
Polona Picman Stefancic, Rea IT Research Centre, Slovenia
Tanja Röchert-Voigt, Universität Potsdam, Germany
Lior Rokach, Ben-Gurion University of the Negev, Israel
Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster / Leibniz Universität Hannover / North-German Supercomputing Alliance, Germany

Copyright Information

Table of Contents

Object Security and Verification for Integrated Information and Computing Systems

Claus-Peter Rückemann
Leibniz Universität Hannover (LUH),
Westfälische Wilhelms-Universität Münster (WWU),
North-German Supercomputing Alliance (HLRN), Germany
Email: ruckema@uni-muenster.de

Birgit Frida Stefanie Gersbeck-Schierholz
Leibniz Universität Hannover (LUH),
Certification Authority University of Hannover (UH-CA),
Germany
Email: gersbeck@ca.uni-hannover.de

***Abstract*—This paper presents the results of the development of techniques for implementing communication for complex integrated information and computing systems based on Object Envelopes. This provides means for flexible information exchange, namely objects, in mission critical environments, based on verification methods and cryptography. It covers some challenges of collaborative implementation, legal, and security issues with these processes. A major task is integrating information systems with Distributed and High Performance Computing resources in natural sciences disciplines, like epidemiology information systems, for building integrated public / commercial information system components within the e-Society. The main focus of this paper is on trust in information and how modular system architectures can make use of Object Envelope techniques. It shows that by object envelopes and signing, future security enhanced information and computing systems can be created.**

Keywords–Information Systems; Computing Systems; Security; Verification; Distributed Systems; Public Key Cryptography; High Performance Computing; Object Management.

1. Introduction

Todays information and computing systems are facing challenges from complex environments and heterogeneous content. The associated problems mostly emerge as security and legal issues, resulting in shortcomings for international collaboration management [1]. Over the last years a long-term project, Geo Exploration and Information (GEXI) [2] for analysing national and international case studies, has examined chances to overcome the deficits.

This paper presents the results of these projects using a newly implemented form of envelope regarding content data security for digital objects, Object Envelopes (OEN), in use with integrated information and computing environments in a collaboration framework. These OEN have shown successful content centred solution for various cases integrating the sections High Performance Computing (HPC), Distributed Computing (DC) and services, and natural sciences.

There are numerous situations where the use of information within complex distributed information and computing system environments is subject to security issues and legal regulations, especially if the information is in any way sensitive, highly charged or must be highly reliable [1]. Todays high end resources lack in methods for secure job and object handling. Many environment contexts base of legally binding premises. The information handled within these systems is one of the crucial points of concerns. For "trust in information" and "trust in computing" situations a collaboration framework has been created and tested with various implementation scenarios. The use cases showed two groups of systems, reflected by collaboration matrices [3].

This paper is organised as follows. Section 2 presents the motivation and implementation scenario including the problems with common technology. Sections 3 describes the fundamental implementation architecture for the solution. Sections 4 and 5 explain the requirements for "trust in information" and object signing and verification. Section 6 shows the solution for integrated systems (OEN). Section 7 reports on the evaluation and Sections 8 and 9 summarise the lessons learned and conclusion and outlook on future work.

2. Motivation and implementation scenario

The information and computing system components make use of various technologies, IPC, sandboxing, embedded applications, browser plugins, remote execution, network protocols, computing interfaces as well as public and sensitive data. The major motivation is to create an architecture of system components based on secure, signed, and verified objects in order to press ahead with standardisation for content and object management and reducing complexity. Figure 1 shows some of the basic application scenarios.

There exists a number of scenarios showing how "trust in computing" and "trust in information" can more easily be achieved by reducing complexity for the partners in otherwise very complex systems. The screenshot shows examples of data objects that are subject to protection:

- vector data and multi-dimensional data,
- raster data (aerial, remote sensing, and photographic),
- primary and secondary spatial information,
- calculation, measurement, and processing results,
- meta data and interactive information,
- commercially provided or licensed data, . . .

2.1. State of prominent technology

As an example let us take a look at a method for signing widely used Portable Document Format (PDF) files. Adobe

ISBN: 978-1-61208-122-9

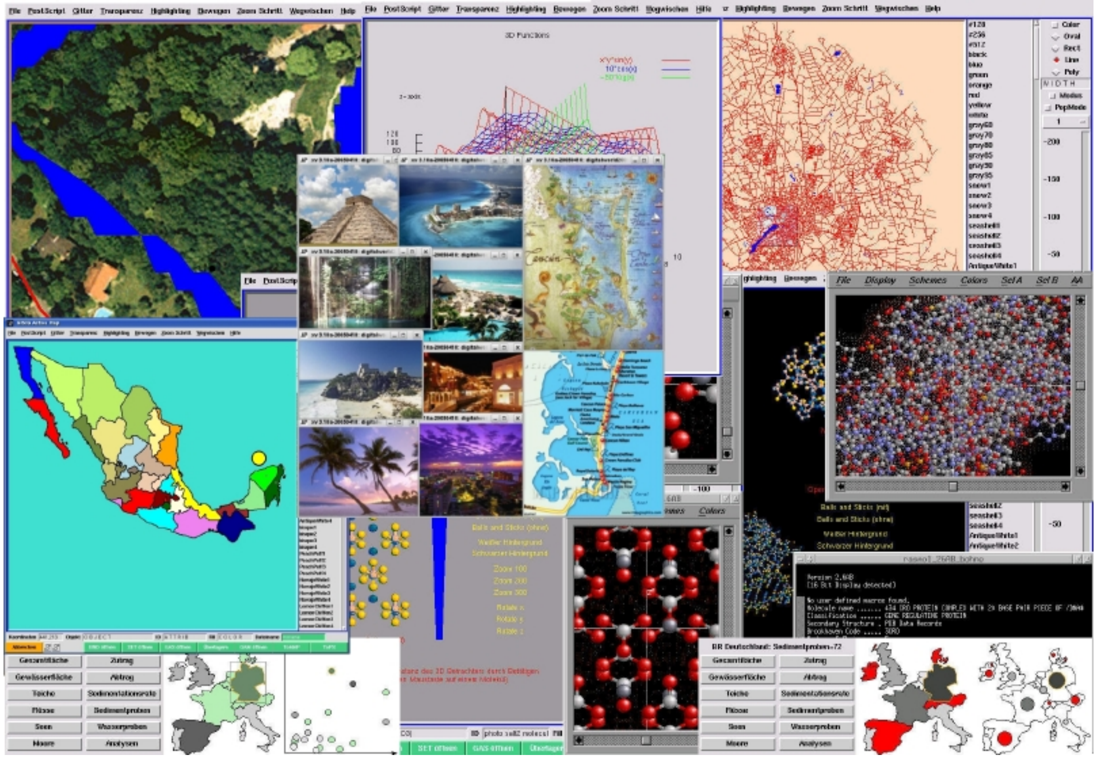


Figure 1. Application scenario from the GEXI case study showing object content scenarios.

uses the Public Key Cryptographic Standard (PKCS) for its proprietary products [4]. A byte stream is built from the PDF document and a digest is calculated. The hash value is encrypted with the private key (signature) and embedded as PKCS#7 into a copy of the document at a defined space. Besides the signed digest the embedded PKCS#7 object includes the full certificate of the signer. Meta data is hold in the signature dictionary. Verification is done using public key and certificate chain using the information from the PKCS#7 object. Possible revocation can be queried via the Online Certificate Status Protocol (OCSP) responder or a Certificate Revocation List (CRL).

2.2. Problems for complex use cases

Although the certificate processing conforms to the X.509-v3 certificate standard (RFC 3280) and standard signature objects are generated as PKCS#7 (RFC 2315), the solution is not appropriate for more complex information system situations. Even the use case portability of this practice does not guarantee for future application. In the case of other file formats the algorithms cannot be implemented due to different properties of “missing” features of these formats. For example a JPEG raster file cannot be signed the way a PDF file can be signed. In some cases the different file formats like JPG, PNG, TIFF or PDF might be embedded into PDF documents but this cannot be implemented for a complex system where hundred thousands of signed objects might have to be embedded into a single context, e.g., into a spatial view. This method based on PDF or other proprietary envelope has been recognised not flexible enough to serve as a generic solution for any complex multi-format information system. The main reasons are, that the algorithm:

- is not portable in between different file formats,
- does not respect meta-data of the information handled,
- does modify the original documents,
- is not intuitively extendable for information systems,
- and there is no flexible and open implementation available, and further on there are
- security issues associated with available products,
- the proprietary solution is not completely transparent,
- the XML has large overhead for huge object collections,
- huge transfer rates for large number of objects, and
- security issues with transfer actions to outer networks.

3. Fundamental implementation architecture

The fundamental architecture is based on a layered concept for the implementation and operation of information and computing systems [1]. The “trust in information” is twofold, regarding the content information domain and the utilisation information domain. It has been possible to transparently separate nearly all of the implementation aspects for the three columns and layers.

The case study showed that for the application within the integrated information and computing system the flexibility of content handling largely profits from object envelopes. Objects have to be signed with digital signature and timestamps of the originating authors and manipulation.

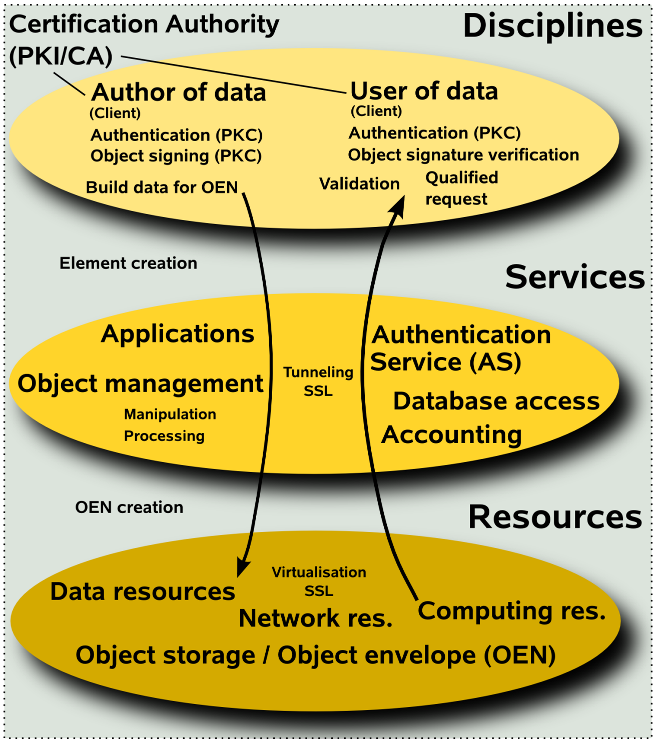


Figure 2. Object handling for integrated systems.

Figure 2 shows the object handling for integrated information and computing systems. The following sections give a detailed description on requirements and the processes depicted in this illustration. For most information systems used in mission critical application environments it is essential to assure accurate data objects all over the life-cycle of objects, thus guaranteeing “trust in information”. Cryptographic techniques specified as public key cryptography in Public Key Infrastructure (PKI) environments [5] provide a framework for addressing important security considerations of authentication and data integrity. In this regard the authority (CA) signs the public user keys in order to maintain the integrity of the public key, expiration information and other important information contained within the user Public Key Certificates (PKC).

4. Requirements for trust in information

Digital signature capabilities allow object authors to set up a secure signing environment and allow the consumer of the data object to validate the object concerning integrity and authentication of the signer. The following passages describe the certificate requirement for trust in information.

4.1. Object cryptography

Asymmetric public key cryptography is based on the use of public / private key pairs [6]. A public key is typically disseminated in the form of a certificate, whereas a private key is a separate and distinct data structure always protected from unauthorised disclosure in transit, use and storage. The keys of a key pair are different, but related according to the circumstances that one has to decrypt what the other encrypts. Given an encryption key it is computationally not feasible to determine the decryption key and vice versa.

4.2. Object PKI

A PKI provides a trusted and authenticated key distribution infrastructure [7], [8], [9]. Its primary purpose is to strongly authenticate the parties communicating with each other, though the use of digital signatures, where the CA is an independent authority that issues PKC for binding the identity of a user to a public key by means of CA’s digital signature. Furthermore, the CA will record and track the issued PKC and will schedule expiry dates for certificates and ensure certificates are revoked. There exist solutions for special use, like the PKCS. These solutions comprise fundamental definitions for special data structures and algorithms, providing the base of common PKI implementations. They define the syntax/format for a digital signature [10] and provide means for distributing certificates and revocation lists. A common trust model in a PKI is a strict hierarchy of CA institutions where all entities in the hierarchy trust the single root CA. The root is the starting point for trust. A certification hierarchy forms the certification path (chain of trust), from the certificate back to the root CA. To verify the trustworthiness of user certificates signed by a CA, the certificate chain of trust will be built by mapping the issuer name of the subordinate certificate to the subject name of the certificate higher up the chain and verifying it is signed with a valid signature, that has not expired. The Object Envelope (OEN) is able to describe any form of object PKI data.

4.3. Meta data

Various meta data is necessary to describe the signed object data. For chronology as well as for plausibility the security of the time and data association is important. Integrated system components as well as interested parties must be able to use these meta data as well as for example must be provided with means to verify that the time stamps associated with an object are authentic and hold integrity. Trusted time stamp authorities are required for this service. This may be a function of the CA, respective a dedicated time server service. The Object Envelope (OEN) is able to describe any form of embedded or referred meta data.

5. Object signing and verification

The following workflow considers the distributed implementation of the respective system components (Figure 2) within the layers.

5.1. The signing process

The following sequence describes the signing process for the implementation in operative context (case study on environmental sciences / epidemiology). The signing process consists of single operation actions requiring some prerequisites:

- **Disciplines layer:** The author of data objects, the originator, e.g., co-worker of a human health organisation, signs the created objects, e.g., disease case numbers, with his private key and according meta data.

The signing requirements for this process are:

- Asymmetric key pair / PKC.
- PKI-enabled application. A special client for communicating with the information system services is desirable.

During signing procedure, the data object is digested with a hash algorithm and then the hash value is encrypted with the signer's private key. If the object changes, the message digest changes. Though, a message digest is simply a unique number created at signing time that identifies the object data that was signed. Containing this information, a signature element for the OEN is generated, including the signer's public key, data content, CA Certificates, and element meta data. The signature object elements are passed on to the service layer, including the object data. Within the disciplines layer object signing requires a client, able to handle the services that the PKI has enabled. Specifically, encryption / decryption and digital signature generation is requested by OEN envelopes. In addition, the services and client software must be able to access the data and key / certificate life-cycle management functions. Software provided with those features, is said to be PKI-enabled. Widely used applications are already PKI-enabled, like Webbrowsers and more popular e-mail clients and electronic forms packages. For future integrated information and computing components, PKI support will be an essential feature.

- **Services layer:** Objects are processed by the services.
- **Resources layer:** Processed objects are stored to the distributed storage.

5.2. The verification process

The following sequence describes the verification process for the implementation in operative context (case study on environmental sciences / epidemiology).

- **Disciplines layer:** An user, e.g., a member of a research team at an university, requests objects.
- **Services layer:** The user is authenticating at the authentication service (AS). The AS requests the objects or service operation.
- **Resources layer:** Objects are collected from the distributed resources.
- **Services layer:** Objects are calculated, validated, accounted, and provisioned via services for the user client.
- **Disciplines layer:** Consumer's client application retrieves the signed data objects and performs the desired validation and verification procedure.

The requirements for the verification process regarding PKI and object infrastructure are:

- PKI-enabled application, a special user client interface is necessary for using the information system services.
- Object Envelope including the signed object, containing signer's PKC and CA certificates.
- OCSP responder or compatible revocation system.

To validate a signature, the consumer's software client first retrieves the author's certificate from the OEN and generates a digest hash of the document using the same hash algorithm the signer used (for example SHA-256). Then the hash value encrypted with the author's private key during signing process is decrypted using the author's public key, if successful, signer's authentication is valid (verifying signer's identity). Then the decrypted hash value is compared to the even locally generated hash value. If they are identical, the integrity check is valid (verifying object's integrity). Furthermore, as well as the signing process, the verification workflow requires a PKI enabled client on the disciplines layer. In particular, encryption/decryption and digital signature verification must be supported. To establish certificate trust, the application builds and validates the certificate chain as described above. To facilitate the verification workflow the CA certificates are embedded in the OEN. After the chain is validated, and the trust anchor is found from the certificate trust list, the client determines whether any of the certificates in the chain have been revoked. The client software looks for a valid revocation response like an OCSP response or an embedded CRL reference for the OEN object.

6. Solution for use with integrated systems

What we needed, was not only a signature standard and an envelope technology but a generic extensible concept for information and computing system components. The benefits for development, configuration, and use of complex information and computing systems are:

- no overhead, minimising communication,
- transparent handling,
- no proprietary algorithms.

Future objectives, combined with client components are:

- channels for limiting communication traffic,
- qualified signature services and accounting,
- using signed objects without verification,
- verify signed objects on demand.

The tests done for proof of concept have been in development stage. A more suitable solution has now been created on a generic envelope base. The current solution is based on OEN files (extension used is `.oen`) containing element structures for handling and embedding data and information. Listing 1 shows a small example for a generic OEN file.

```
<ObjectEnvelope><!-- ObjectEnvelope (OEN)-->
<Object>
<Filename>GIS_Case_Study_20090804.jpg</Filename>
<Md5sum>...</Md5sum>
<Sha1sum>...</Sha1sum>
<DateCreated>2010-08-01:221114</DateCreated>
<DateModified>2010-08-01:222029</DateModified>
```

```
<ID>...</ID><CertificateID>...</CertificateID>
<Signature>...</Signature>
<Content><ContentData>...</ContentData></Content>
</Object>
</ObjectEnvelope>
```

Listing 1. Example for an Object Envelope (OEN).

An end-user public client application may be implemented via a browser plugin, based on appropriate services. With OEN instructions embedded in envelopes, for example as XML-based element structure representation, content can be handled as content-stream or as content-reference. The way this will have to be implemented for different use cases depends on the situation, and in many cases on the size and number of data objects. Listing 2 shows a small example for an OEN file using a content DataReference.

```
<ObjectEnvelope><!-- ObjectEnvelope (OEN)-->
<Object>
<Filename>GIS_Case_Study_20090804.jpg</Filename>
<Md5sum>...</Md5sum>
<Sha1sum>...</Sha1sum>
<DateCreated>2010-08-01:221114</DateCreated>
<DateModified>2010-08-01:222029</DateModified>
<ID>...</ID><CertificateID>...</CertificateID>
<Signature>...</Signature>
<Content><DataReference>https://doi...</DataReference></Content>
</Object>
</ObjectEnvelope>
```

Listing 2. OEN referencing signed data.

One benefit of content-reference with high performant distributed or multicore resources is that references can be processed in parallel on these architectures. The number of physical parallel resources and the transfer capacities inside the network are limiting factors. Whereas the XML signature standard (RFC 2807) [11] proclaims the feasibility that XML signatures can be applied to arbitrary digital content via indirections, this only answers the problem of huge data regarding quantity or size theoretically. For practical use in real-life use cases one would prefer solutions matching to the situation, being flexible, transparent, open, portable, and using general modular components. For qualified requests signatures and signature groups can be verified. For non-qualified requests signatures can be ignored. All OEN can be embedded into existing information and computing system components. Listing 3 shows a small example of an OEN embedded into a GISIG Active Source component.

```
#BCMT-----------------------------------------------
###EN \gisigsnip{Object Data: Country Mexico}
#ECMT-----------------------------------------------
proc create_country_mexico {} {
global w
#  Sonora
$w create polygon 0.938583i 0.354331i 2.055118i ...
#BCMT-----------------------------------------------
###EN \gisigsnip{Object Data: Object Envelope (OEN)}
#ECMT-----------------------------------------------
#BOEN <ObjectEnvelope>
##OEN <Object>
##OEN <Filename>mexico_site_name_tulum_temple.jpg</Filename>
##OEN <Md5sum>251b443901d87a28f83f8026a1ac9191 *mexico_site_name_tulum_temple.jpg</Md5sum>
##OEN <Sha1sum>f0eb9d21cfe2c9855c033be5c8ad77710356c1eb *mexico_site_name_tulum_temple.jpg</Sha1sum>
##OEN <DateCreated>2010-08-01:221114</DateCreated>
##OEN <DateModified>2010-08-01:222029</DateModified>
##OEN <ID>...</ID><CertificateID>...</CertificateID>
##OEN <Signature>...</Signature>
##OEN <Content><ContentDataReference>http://.../mexico_site_name_tulum_temple.jpg</ContentReference></Content>
##OEN </Object>
#EOEN </ObjectEnvelope>
...
proc create_country_mexico_autoevents {} {
global w
$w bind legend_infopoint <Any-Enter> {set killatleave [exec ./mexico_legend_infopoint_viewall.sh $op_parallel ] }
$w bind legend_infopoint <Any-Leave> {exec ./mexico_legend_infopoint_kaxv.sh }
$w bind tulum <Any-Enter> {set killatleave [exec $appl_image_viewer -geometry +800+400 ./mexico_site_name_tulum_temple.jpg $op_parallel ] }
$w bind tulum <Any-Leave> {exec kill -9 $killatleave }
}
...
```

Listing 3. OEN embedded with Active Source.

Additionally, algorithms like check sums (MD5, SHA or others) or encryption for content or meta data can be handled very flexible. Common modules for these algorithms are md5sum, sha1sum, sha512sum, gpg, and many other tools supporting functions and features like authentication, integrity, reliability, confidentiality, and authorisation.

7. Evaluation

The primary benefits of the presented solution using OEN with signed objects are that the algorithm is

- portable in between different object and file formats.
- It respects meta-data for the objects.
- Original documents can stay unmodified.
- The solution is most transparent, extendable, flexible, and scalable, for security aspects and modularisation.
- Guaranteed data integrity and authentication derived from the cryptographic strength of current asymmetric algorithms and digital signature processes.
- Flexible meta data association for any object and data type, including check sums and time stamps.

Main drawbacks are:

- Requirements for use outside the case studies: Interoperability between multiple PKIs, a global cryptosystem on the internet (Global PKI), special PKI-enabled software clients to generate, store and manage certificates and associated data is not already implemented.
- Risks: Lost, destroyed, or compromised private keys and loss of primary verification for keyed object data.
- Inconveniences: Authors have to register at a CA and request digital certificates.

With modern information and computing systems object management is a major challenge for software and hardware infrastructure. Resulting from the case studies with information systems and computing resources, signed objects embedded in OEN can provide a flexible solution. PKI technology offers means to attest, identify and manage the exchange of encryption keys and secure transmission between parties. Although PKI technology has not already been broad-based adopted by public and private organisations as it mostly only is supported for optional use with single processes like e-mail communication, it is a valuable support for creating a secure object life-cycle for mission critical high end information systems.

8. Lessons learned

The OEN solution has been found to be a flexible and extensible solution for creating a secure environment for integrated information and computing systems. The case study showed that nearly any data structure can be handled with object envelopes in embedded or referred use. Signatures, check sums, and meta data can be used in various ways for the purpose of the information and computing system. Key loss is not critical for the data itself. Service providers and users can ensure the integrity and re-keying. For scalability, e.g., for different object sizes from some bytes up to several Giga-bytes, it is preferable to have more than one fixed method. Therefore embedded and referred data has to be supported. This leads to the conclusion that in the future of integrated information and computing systems we will need to create means of securely submitting modular application components into the services pipeline.

9. Conclusion and future work

The security and verification of information content is an essential part of the challenge to build future integrated information and computing systems. Object Envelope (OEN) techniques can help to establish a flexible and portable way for using content data. Further on with implementation and legal issues, the security aspect are on the rise for any complex system. Even though PKI technology offers means to attest, identify, manage the exchange of encryption keys and secure transmission between parties, there has not been broad-based adoption of PKI technology by public and private organisation. After all, a significant number of countries recognise digital signatures as legally binding. In case of security enhanced integrated information and computing system components object signing provides a robust solution to facilitate “trust in information” and to overall support “trust in computing”. In order to put this implementation into international public practice there is a need for future PKI development and deployment offering a global public key cryptosystem for the Future Internet. This work showed that it is possible to bring complex information and computing systems to life, being able to create interfaces that can also be interfaces between the logical columns and interest groups.

Acknowledgements

We are grateful to all national and international academic and industry partners in the GEXI cooperations for the innovative constructive work and to the colleagues at the LUH, IRI, HLRN, WWU and the participants of the postgraduate European Legal Informatics Study Programme (EULISP) for prolific discussion of scientific, legal, and technical aspects.

References

[1] C.-P. Rückemann, “Legal Issues Regarding Distributed and High Performance Computing in Geosciences and Exploration,” in *Proceedings of the International Conference on Digital Society (ICDS 2010), The International Conference on Technical and Legal Aspects of the e-Society (CYBERLAWS 2010), February 10–16, 2010, St. Maarten, Netherlands Antilles / DigitalWorld 2010, International Academy, Research, and Industry Association (IARIA)*. IEEE Computer Society Press, IEEE Xplore Digital Library, 2010, pp. 339–344, Berntzen, L., Bodendorf, F., Lawrence, E., Perry, M., Smedberg, Å. (eds.), ISBN: 978-0-7695-3953-9, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5432414 (PDF) [accessed: 2010-03-28], (Best Paper Award).

[2] “Geo Exploration and Information (GEXI),” 1996, 1999, 2010, URL: http://www.user.uni-hannover.de/cpr/x/rprojs/en/index.html#GEXI (Information) [accessed: 2010-05-02].

[3] C.-P. Rückemann, “Legal Base for High End Computing and Information System Collaboration and Security,” *International Journal on Advances in Security*, vol. 3, no. 3&4, 2010, (to appear) ISSN: 1942-2636, URL: http://www.iariajournals.org/security/ [acc.: 2010-08-18].

[4] “Acrobat Digital Signatures, Digital Signature User Guide for Acrobat 9.0 and Adobe Reader 9.0,” Adobe, 2010, URL: http://www.adobe.com/devnet/acrobat/ [accessed: 2010-08-10].

[5] “ITU-T Recommendation X.509 ISO/IEC 9594-8, The Directory: Authentication Framework,” 2000, URL: http://wwwitu.int/itu-t/recommendations [accessed: 2010-05-29].

[6] W. Diffie and M. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, 1976, pp. 644–654, URL: http://www.cs.purdue.edu/homes/ninghui/courses/Fall04/lectures/diffie-hellman.pdf [acc.: 2010-09-18].

[7] C. Adams and S. Lloyd, *Understanding PKI: Concept, Standards, and Deployment Considerations*, 2nd ed. Addison Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2002, 322 pages, ISBN: 0672323915.

[8] A. Nash, W. Duane, C. Joseph, and D. Brink, *PKI: Implementing and Managing E-Security*. McGraw-Hill OsborneMedia, New York, USA, 2001, 513 pages, ISBN: 0072131233.

[9] B. F. S. Gersbeck-Schierholz, “Trustworthy Communication by Means of Public Key Cryptography,” 2010, URL: http://www.rrzn.uni-hannover.de/fileadmin/it_sicherheit/pdf/pki2010_gersbeck.pdf [accessed: 2010-05-29].

[10] B. Kalinski, “PKCS#7: Cryptographic Message Syntax Version 1.5,” March 1998, Internet Request for Comments: RFC 2315, RSA Laboratories, East (ed.).

[11] “XML-Signature standard,” W3, 2010, URL: http://www.w3.org/TR/xmldsig-core/ [accessed: 2010-08-10].

Mobile Phone Anomalous Behaviour Detection for Real-time Information Theft Tracking

Vrizlynn L. L. Thing[1], Perumal P. Subramaniam[2], Flora S. Tsai[2], and Tong-Wei Chua[1]

[1]Cryptography & Security Department
Institute for Infocomm Research, Singapore
{vriz,twchua}@i2r.a-star.edu.sg
[2]School of Electrical and Electronic Engineering
Nanyang Technological University
peru0002@ntu.edu.sg, fst1@columbia.edu

***Abstract*—Due to the prevalence of mobile phones usage and their increasing features and functionalities, the amount of personal and confidential data residing in the phones is becoming substantial. In the event of information theft by applications residing on the phones, the loss of such important data can be damaging to the user's reputation or result in a financial loss. We show in this paper how these applications can appear to be non-malicious but are stealthily retrieving and exporting confidential information without leaving any trace, thus bypassing detections by current state-of-the-art anti-virus solutions. We propose a tool to detect and track the behaviour of these applications in real-time so as to collect evidence. Using this tool, we can successfully monitor the applications non-intrusively, detect the "misbehaving" applications, alert the users, and log the evidence of malicious activities with timestamp information to facilitate forensic investigations and institute accountability.**

Keywords: Mobile device forensics, Android, information theft, anomaly detection, spyware.

I. Introduction

As mobile phones are becoming increasingly prevalent and are constantly evolving into "smarter" devices (i.e. smartphones with higher processing power and enhanced features), it is a common scenario that users are installing applications that appeal to them while at the same time generating and storing more personal information on their phones. In the presence of information theft activities introduced by the installed applications, there is a risk of users losing their personal and sensitive information, or confidential corporate data (e.g. emails).

Due to the potential risks, the capabilities to perform detection of such malicious information theft activities on the phones and the collection of relevant evidence to facilitate forensic investigation become essential. However, current mobile phone forensics are still restricted to the research and analysis of static data on the Subscriber Identity Module (SIM), memory cards and the internal flash memory [1], [2], [4], [5], [8], [10]–[13], [16], [19].

To achieve anti-theft protection and anti-virus scanning on mobile phones and devices, there exist several mobile devices security solutions [6], [7], [9], [15], [18] in the market. The features of these solutions include blocking the phones, deleting the data and finding the phone (via GPS and map display) through remote access by the user, in the event that the user loses his/her phone. They also support the detection and removal of malicious applications from the phones. However, the detection of virus is through a signature based mechanism and do not support anomaly detection or real-time monitoring for information theft protection against malicious applications. More information on the related work is discussed in Section II.

In this paper, we propose a tool to monitor, detect, and track cyber criminal activities relating to information thefts on the mobile phones or devices. The tool intercepts sensitive information access non-intrusively (i.e. without interfering with normal application operations), triggers an alert and performs evidence collection (i.e. logging and timestamping) for mobile devices. In our work, we focus on mobile phones running the Android operating system. The reason is that while there exists the availability of a high number of applications in the Android Market online portal, the market place is not well regulated, unlike the iPhone App Store. Applications written by third party developers are not required to be approved before being available for download by mobile phone users. This scenario presents a potential risk in malicious activities being introduced to the phones. In addition, Android is a new mobile platform but its fast rising popularity among users and the phone manufacturers [14] calls for a need for us to address the security and forensic issues with regards to the information theft problem.

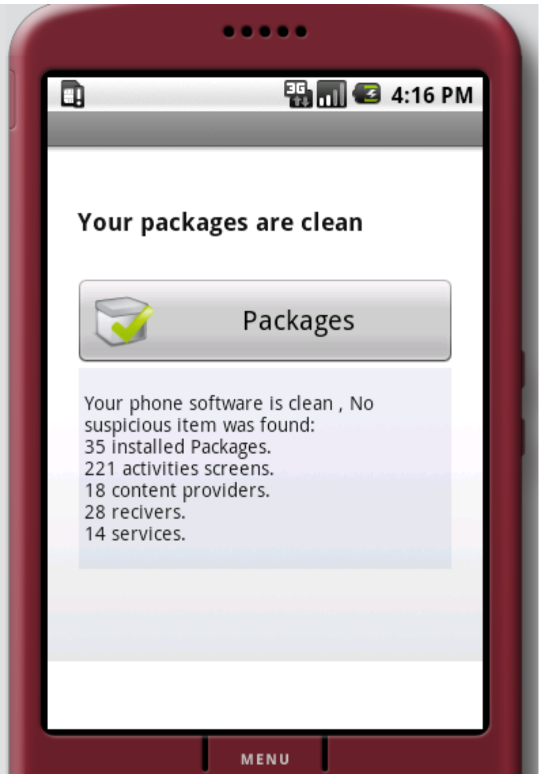
The rest of the paper is organised as follow. In Section II, we present an overview of the related work in the area of mobile phone anti-spyware. We describe the design of our detection and tracking tool in Section III. The experiments and results are presented and discussed in Section IV. Future work is described in Section V and conclusions follow in Section VI.

II. Related Work

In the personal computer terminology, malicious applications or software include viruses, worms and other exploits. Such malicious applications or software can also exist in mobile phones, and can be exploited by criminals to steal sensitive information from the phones' users discreetly. In this section, we discuss the current state-of-the-art mobile security solutions.

There are a number of anti-virus software in the market for mobile devices, such as the Kaspersky Mobile Security [15] and Norton Smartphone Security [7]. They support the Symbian and Windows mobile platforms. Mobile security solutions for Android include the Droid Security [9], SMobile Mobile Security [18] and F-Secure Mobile Security [6]. The Droid Security tool enables the user to remotely block and delete the data on the phone in the event that the phone is lost. Other features include the GPS locator, virus scanning and the identification of dangerous websites. Similar features are supported by the SMobile Mobile Security and the F-Secure Mobile Security solutions.

We tested the freely available trial version of the Droid Security solution to detect the presence of a malicious application on the Android phone. We built the malicious application based on the code snippet of a simple tips calculator [17] by modifying it to access the phone contacts information and exporting it to an external party in the background. Therefore, in the foreground, the application appears to be an innocent looking application performing basic calculator operations but is in fact carrying out information theft activities stealthily.



Your packages are clean

Your phone software is clean , No suspicious item was found:
35 installed Packages.
221 activities screens.
18 content providers.
28 recivers.
14 services.

Fig. 1: Scan results with Droid Security

When using the Droid Security software to scan for malicious applications, it could not detect the tips calculator as potentially harmful to the user. Figure 1 shows the results of scanning the phone with the Droid Security. All the installed applications were found to be clean.

Through this small experiment, we observed that although existing mobile security solutions can provide protection from information theft due to the physical loss of the mobile phones or devices and support virus scanning to detect and remove known malicious software, there is a need for a solution to monitor, detect, alert and collect evidence of information theft (which is conducted stealthily) on the mobile phones in real-time so as to facilitate the forensic investigations of such violations by the law enforcement agencies. Therefore, in this paper, we design a real-time information theft detection and application behaviour tracking tool, and present it in the next section.

III. Information Theft Detection and Tracking

Most of the applications for Android are available in the Android Market which is an online store for Android applications. Users browse and download applications published by third-party developers as an open service and this process is not well regulated [3]. Thus, it is difficult to determine whether a newly available application is malicious or not. To perform a real-time monitoring and detection of the anomalous and suspicious behaviour of installed applications, we propose designing a tool that intercepts selected applications' access to sensitive information and alerts the user of their intentions.

A. API Hooking

The Application Programming Interface (API) is an interface which is used by an application to request services from libraries and operating systems. Hooking is a technique where the normal program flow is diverted. API Hooking is a procedure where the API calls by programs to interact with the kernel are intercepted. Figure 2 shows an overview of API hooking.

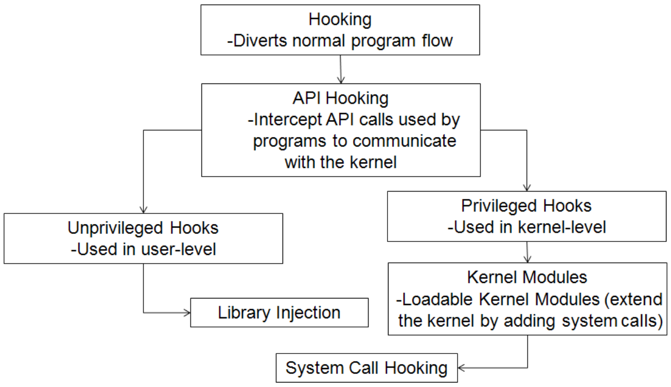

Hooking
-Diverts normal program flow
API Hooking
-Intercept API calls used by programs to communicate with the kernel
Unprivileged Hooks
-Used in user-level
Privileged Hooks
-Used in kernel-level
Library Injection
Kernel Modules
-Loadable Kernel Modules (extend the kernel by adding system calls)
System Call Hooking

Fig. 2: API Hooking

There are two types of hooks, unprivileged and privileged hooks. Unprivileged hooks can only be executed within the user program address space through library injections while privileged hooks can be executed in kernel level to replace or add on to system calls.

B. Tool Design

In Android, the applications access the kernel through "legal entry points", which are known as the system calls. The system calls enable the mobile phone applications to access the kernel while maintaining the system's stability. They provide an interface for a user-space process to request for operating services such as specific accesses to different hardwares or fundamental services (e.g. generic system read and write as shown in Figure 3).

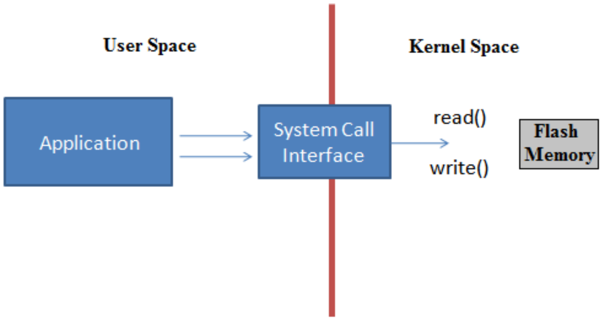


Fig. 3: System Call

System call interception is an example of a privileged hook (3). Even though a malicious application can function discreetly on the phone, its operating system service access can still be intercepted so as to facilitate the monitoring of its actual underlying activities. Therefore, to detect any suspicious theft activities, we build a loadable kernel module to observe the user selected applications and detect any request for services from the operating system to access sensitive and confidential information on the phone. This monitoring module "captures" the system calls to support our tool's analysis of the activities undertaken by the applications.

To perform system calls interception on the phone, we accessed the system call table to locate the system call references. The system call table address at 0xc0539900 was revealed through the "print &sys_call_table" call. After which, we added the functions to intercept specific system calls (i.e. system access, network socket calls, folder deletion, change of ownership rights). The program flow can also be diverted automatically if the calls are found to be malicious (Figure 4). Our tool binds to an application upon selection and non-intrusively monitors its activities and alerts upon access to sensitive information on the phone. In a compromised application, the tool can detect the anomalies even if the application "misbehaves" in the background.

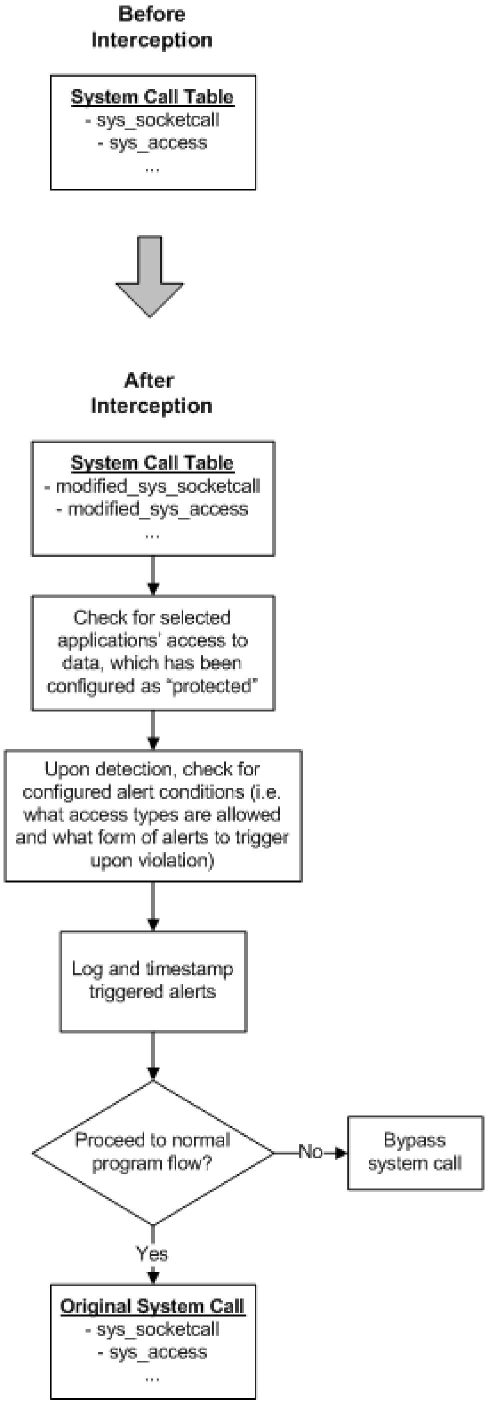

Before Interception
System Call Table
- sys_socketcall
- sys_access
...
After Interception
System Call Table
- modified_sys_socketcall
- modified_sys_access
...
Check for selected applications' access to data, which has been configured as "protected"
Upon detection, check for configured alert conditions (i.e. what access types are allowed and what form of alerts to trigger upon violation)
Log and timestamp triggered alerts
Proceed to normal program flow?
No
Bypass system call
Yes
Original System Call
- sys_socketcall
- sys_access
...

Fig. 4: Tool Design

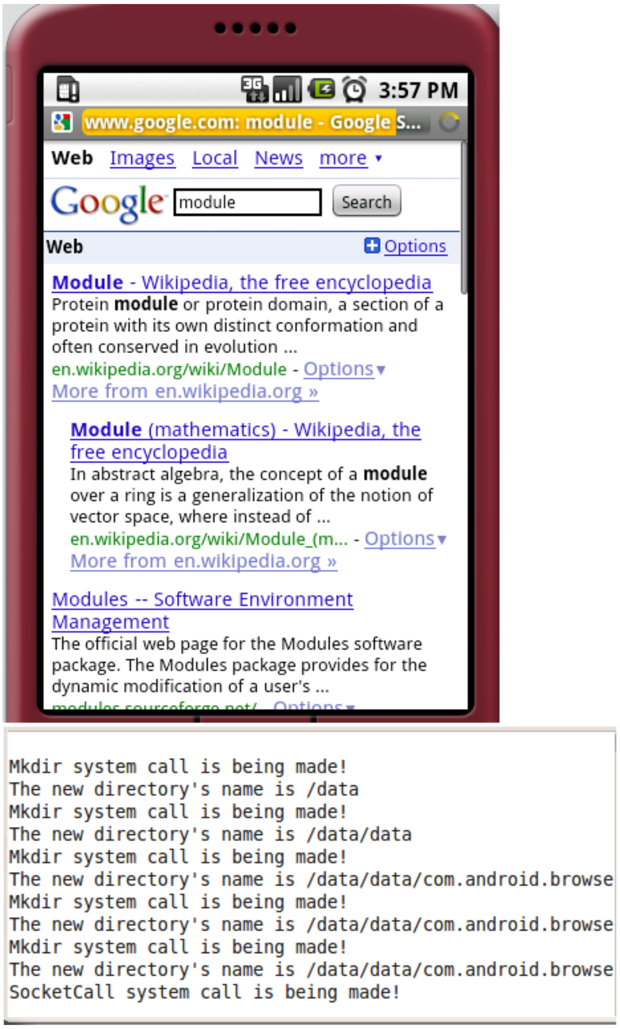
IV. Experiments and Discussions

For our experiments, we configured our tool as follow.

1) Observe and log all important system calls (i.e. system access, network socket calls, folder deletion, change of ownership rights)
2) Specify the alert conditions (e.g. read/write access to contacts information) to detect suspicious activities and possible information theft, and to inform users in real-time (i.e. display floating messages over the applications)
3) Log suspicious activities with timestamp information in the background for evidence collection and forensic investigations

We conducted two experiments. The first was an experiment to track the behaviour of an Internet browser application on the Android phone. The second experiment was conducted on the "misbehaving" tips calculator mentioned in Section II.

In the first experiment, we selected the default Internet browser on the phone to be started. Our tool will advise the user to load the interception module so as to bind to the application. The user then started using the browser as usual while the behaviour tracking was conducted in the background (refer to Figure 5). As the application was not accessing any



3:57 PM

www.google.com: module - Google S...

module

Module - Wikipedia, the free encyclopedia
Protein **module** or protein domain, a section of a protein with its own distinct conformation and often conserved in evolution ...
en.wikipedia.org/wiki/Module - Options▾
More from en.wikipedia.org »

Module (mathematics) - Wikipedia, the free encyclopedia
In abstract algebra, the concept of a **module** over a ring is a generalization of the notion of vector space, where instead of ...
en.wikipedia.org/wiki/Module_(m... - Options▾
More from en.wikipedia.org »

Modules -- Software Environment Management
The official web page for the Modules software package. The Modules package provides for the dynamic modification of a user's ...
modules.sourceforge.net/ - Options▾

```
Mkdir system call is being made!
The new directory's name is /data
Mkdir system call is being made!
The new directory's name is /data/data
Mkdir system call is being made!
The new directory's name is /data/data/com.android.browse
Mkdir system call is being made!
The new directory's name is /data/data/com.android.browse
Mkdir system call is being made!
The new directory's name is /data/data/com.android.browse
SocketCall system call is being made!
```

Fig. 5: Behaviour Tracking on Phone

private user information, no alert was triggered in this case.

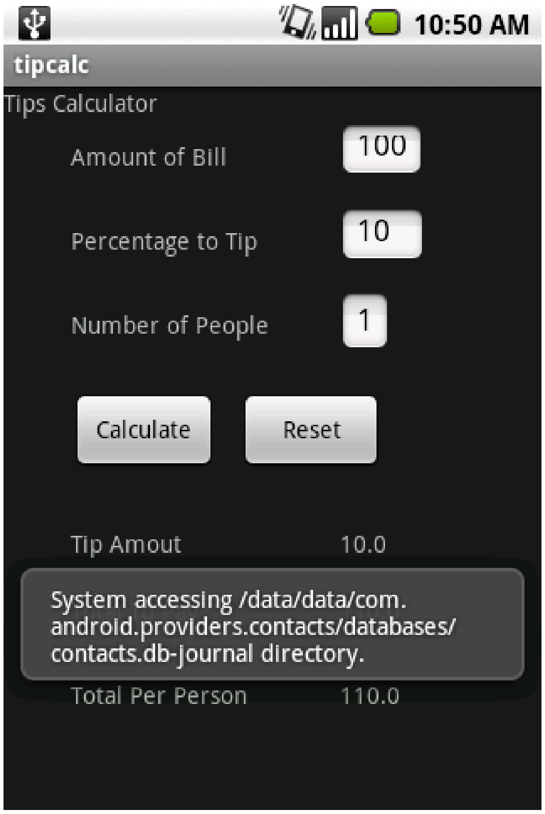
In the second experiment, we first run the tips calculator application without activating our tool. After the application was started, it accessed the phone contacts information stealthily in the background while the user was using the application. The contacts information on the Android phone was successfully sent to the external party phone through SMSes. We performed a manual investigation of the "sent messages" records on the phone and found that no trace of the export could be found.

We closed the application and started it again; this time with our tool activated. In this case, the following alert was triggered.

```
Alert: 12:20 PM April 12, 2010 System
accessing data/data/com.android.
providers.contacts/databases/contacts.
db-journal directory.
```

The log file was updated and the floating message was also displayed over the application to inform the user (refer to Figure 6).

As shown in the experiments, the detection and tracking tool can reliably capture suspicious activities due to the access to secure information, in real-time. These activities are logged and timestamped, and stored in the phone's non-volatile memory to enable further forensic analysis. The user is also alerted of the ongoing underlying suspicious activities carried out by the application through a real-time display of warning messages. This enables the user to be aware of potentially harmful applications even without any prior knowledge that the application is malicious.



10:50 AM

tipcalc

Tips Calculator

Amount of Bill 100

Percentage to Tip 10

Number of People 1

Tip Amount 10.0

System accessing /data/data/com.android.providers.contacts/databases/contacts.db-journal directory.

Total Per Person 110.0

Fig. 6: Alert Message Display

V. Future Work

Our planned future work includes implementing further enhancements to the tool to include the monitoring of relevant Android application interfaces such as the radio interface layer. This would enable the support of a more detailed logging of information such as the phone number to which the stolen data is exported to or which server is accessed by the malicious application. Another enhancement includes the automatic exporting of logged evidence to a server to support further forensic investigation. The displayed alert messages could also be made more understandable (i.e. less technical) to the common users.

VI. Conclusions

As mobile phones become more prevalent with increasing personal sensitive information and possibly confidential corporate data (e.g. accessed through emails) generated or downloaded on to the phones, there exists a potential risk of losing important information due to malicious information theft exploits. With the growing size of the available applications in the Android market, there is also an increased chance of a malicious application being installed successfully by the users unknowingly. The high possibility of potential mobile information theft crimes call for the need to address the security and forensic issues to institute accountability.

Although there exist anti-virus solutions in the market for mobile phones and devices, malicious information theft applications can bypass their detection by remaining stealth at the application level. Since access to the confidential data is an essential step to stealing it, we non-intrusively intercept the system services access, track the behaviour of the applications and alert the user (and record such violations) when the applications "misbehave" by accessing the data they are not supposed to. We implemented our tool for the Android mobile operating system platform due to its rising popularity and its open service online application market. We showed through experiments that even though no SMS record of the stolen data export could be found during a manual investigation, our tool can reliably detect and track the malicious access in real-time. Our tool is also flexible and highly configurable to indicate which system calls to observe and which data access will trigger alerts.

References

[1] Rizwan Ahmed and Rajiv v. Dharaskar. Mobile forensics: an overview, tools, future trends and challenges from law enforcement perspective. *6th International Conference on E-Governance, ICEG, Emerging Technologies in E-Government, M-Government*, pages 312–323, December 2008.

[2] Marwan Al-Zarouni. Mobile handset forensic evidence: a challenge for law enforcement. In *Proceedings of the 4th Australian Digital Forensics Conference*, December 2006.

[3] Android. Android developers. http://developer.android.com/index.html.

[4] Rick Ayers, Wayne Jansen, Ludovic Moenner, and Aurelien Delaitre. Cell phone forensic tools: An overview and analysis update. *National Institute of Standards and Technology, Technical Report 7387*, March 2007.

[5] Fabio Casadei, Antonio Savoldi, and Paolo Gubian. Forensics and SIM cards: an overview. *International Journal of Digital Evidence*, 5(1):1–21, Fall 2006.

[6] F-Secure Corporation. F-secure Mobile Security. http://campaigns.f-secure.com/mobile-security/index.html.

[7] Symantec Corporation. Norton Smartphone Security. http://www.symantec.com/norton/smartphone-security.

[8] Alessandro Distefano and Gianluigi Me. An overall assessment of mobile internal acquisition tool. *Proceedings of the 8th Digital Forensics Research Conference (DFRWS), Digital Investigation*, 5(1):S121–S127, September 2008.

[9] droidSecurity. Droid Security - Anti-virus. http://www.droidsecurity.com/.

[10] Andrew Hoog. Android forensics. *presented at Mobile Forensics World 2009*, May 2009.

[11] Andrew Hoog and Kyle Gaffaney. iPhone forensics. *viaForensics Whitepaper*, June 2009.

[12] Wayne Jansen and Rick Ayers. Forensic software tools for cell phone subscriber identity modules. *Conference on Digital Forensics, Association of Digital Forensics, Security, and Law (ADFSL)*, April 2006.

[13] Wayne Jansen, Aurelien Delaitre, and Ludovic Moenner. Overcoming impediments to cell phone forensics. In *Proceedings of the 41st Hawaii International Conference on System Sciences*, 2008.

[14] Greg Kumparak. Google: Android now shipping on 60,000 handsets per day. http://www.mobilecrunch.com, February 2010.

[15] Kaspersky Lab. Kaspersky Mobile Security. http://www.kaspersky.com/kaspersky_mobile_security.

[16] Pontjho M. Mokhonoana and Martin S. Olivier. Acquisition of a Symbian smart phone's content with an on-phone forensic tool. *Department of Computer Science, University of Pretoria*, 2007.

[17] Data Springs. A brief guide for creating your first android application (tip calculator). http://www.datasprings.com/Resources/ArticlesInformation/AndroidSDKExampleApplicationSampleCode.aspx.

[18] SMobile Systems. SMobile Mobile Security. http://www.smobilesystems.com/.

[19] Svein Willassen. Forensic analysis of mobile phone internal memory. *Advances in Digital Forensics, IFIP International Federation for Information Processing, Springer*, 194:191–204, March 2006.

Clarifying Privacy in the Clouds

Karthick Ramachandran, Thomas Margoni, Mark Perry
Faculties of Science and Law
University of Western Ontario
London, Canada
{*kramach , tmargoni, mperry*}*@uwo.ca*

***Abstract*—Concomitant with the increased market appeal of cloud-based services, there is growing concern over issues of privacy within the architecture. In this paper, we analyze what is meant by the term privacy from a legal perspective, and how the meaning of cloud computing and their operation may be affected in at least one jurisdiction. We also look at some possible solutions to addressing privacy in clouds.**

***Keywords*-Privacy, cloud computing, compliance with legislation**

I. Introduction

Cloud computing represents a relatively recent architecture and business model in the information technology environment. It is a term that describes having data processed, stored or retrieved in a cloud, where 'cloud' means somewhere on the Internet. The Internet is a very generic term, especially when we are interested in knowing the physical location of data or a particular server. Saying it is in the Internet or in a cloud means that, most of the times, we don't know, or don't care, from a computing perspective where it is. A main feature of cloud computing is that for operational purposes the cloud users are not interested in their location. This is extremely advantageous from a technical perspective as from that viewpoint one needs the job to get done without having to worry about availability of resources. However, from a legal perspective it raises many problems, not least that it is increasingly an issue in most jurisdictions that companies address privacy requirements and comply with privacy regulation. Location is a key factor that must be considered. Here, we are addressing the issues of location-independent computing, such as is part of the fundamental design of cloud computing, in terms of privacy legislation, in order to determine high non-compliance situations, and identify some possible approaches to provide solutions, and best practices. We take Canada as the use case for this analysis.

We present an introduction to Privacy in Section II. In Section III, we look at data protection laws, specific to Canada. An introduction to Cloud Computing and its architectural details are described in Section IV. Section V enumerates some threats to data stored in remote servers. We discuss some of the technical approaches to protect user's privacy in a Cloud computing environment in Section VI. And finally we conclude in Section VII.

II. What is Privacy

Early interpretations of legislation in England outlawed eavesdropping and spying on others. The English courts in deciding about the granting of a warrant to "break open doors, locks, and boxes, and to seize a man and all his books" have held that "we can safely say there is no law in this country to justify the defendants in what they have done; if there was, it would destroy all the comforts of society, for papers are often the dearest property any man can have" [1]. Many international treaties, covenants and declarations recognize privacy as a fundamental human right, such as in Article 8 of the 1950 Convention for the Protection of Human Rights and Fundamental Freedoms "(1) Everyone has the right to respect for his private and family life, his home and his correspondence. (2) There shall be no interference by a public authority with the exercise of this right except as in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health of morals, or for the protection of the rights and freedoms of others". A simple condensing of the privacy as an autonomous right was summed up as the "right to be let alone" [2]. However, it must be said that the same authors recognized that technological evolution carries with new threats that were previously protected by other methods. So, for example, the old tort of trespass (vis et Armis, in its origins around 13th century) can be seen and adapted to protect privacy to some extent, until when in order to know what was happening in the seclusion of a house it was necessary to enter the property of the house owner, i.e. to trespass his own property. No doubt that such a remedy's main function was to protect the person and the property of the owner. However, it was also capable of protecting something that was not identified yet.

Advances in technology, in their very nature, enable actions previously not possible, so it is possible to see through walls with infrared cameras and store such images on computers for easy replication and distribution. The so-called 'smart' electricity meters enable the power supply company to know what and when equipment is being turned on and off inside your house, and possibly build a 'power' profile of a customer. No physical invasion of land has occurred. It is also possible to determine whether somebody is growing marijuana plants in his basement without entering the property [3]. Or, listening to private conversations without eavesdropping at the door but by capturing electronic communications [4][5][6]. All those activities that in past were not allowed because the old technology required an action that was considered illegitimate (eg: entering the property without a warrant), have become available because technology now permits to carry out the activity without performing the prohibited action.

Although the concept of privacy and data protection has developed over the centuries to the point where most countries now have legislation regulating these issues, new technologies such as Deep Pocket Inspection, or traffic sniffing, and sophisticated listening and imaging tools, pose an enormous threat on the protection of privacy and personal data.

III. Privacy and Data protection in Canada

Canada does not have a generally accepted tort of invasion of privacy at common law, although in the USA several approaches by the courts have lent strength to privacy protection. However, Canadian legislation address privacy and data protection with regard to different conditions such as the nature of the obliged subject (public or private), and the type of activity carried out (commercial or not), and the sector within which the activity is being carried out (for example the extensive regulation of health services). The Canadian Charter of Rights and Freedoms is entrenched in Canada's constitution. Although it does not specifically give a right to privacy, it does protect citizens from unreasonable search and seizure by the state. This general principle has been interpreted by the Supreme Court of Canada (SCC), which stated that rights should be interpreted in a broad and liberal manner so as to secure the citizen's right to a reasonable expectation of privacy against governmental encroachments and intrusions [7]. The SCC went further by stipulating that privacy should be at the core of modern societies, and referring a previous Canadian Government Study on Privacy and Computers (that dates back to 1972), in a way that seems to suggest the applicability of the concept of privacy to informational aspects as well:

> *"This notion of privacy derives from the assumption that all information about a person is in a fundamental way his own, for him to communicate or retains for himself as he sees fit"*[8]

During the early 90s the SCC took the opportunity to develop the concept of privacy with regard to governmental intrusions, including the use of then new technologies [9] [10] [11]. However, it must be observed that these cases are based on governmental intrusions, mostly prosecutions looking at the extent to which citizens should be protected from unreasonable measures.

Regarding a different but connected area of collection of data by Federal agencies the reference legislation is the Privacy Act of 1982 [12]. Such legislation main aim is the regulation of the collection and use of personal information by the federal government and a number of federal public agencies. Such statute is coupled with another piece of federal legislation that is geared towards the accessibility by citizens of information stored by government agencies [13]. Both the Privacy Act and the Access to Information Act refer to personal data or records retained by the Federal Government, thereby identifying records that can be either under an analogical or a digital expression form. It must be noted how such legislation refers only to federal bodies, and that on a provincial level similar legislation has been enacted (For example in the [14], [15] and [16]).

Looking to the private sector, the most relevant piece of legislation in Canada, and probably the more relevant in light of this study, is without doubt the Personal Information Protection and Electronic Documents Act of 2000, also referred to as PIPEDA [17], that applies to all private sector entities that collect, use, or disclose personal information in the course of commercial activities (with the exception of those provinces that have enacted equivalent legislation). PIPEDA as many other piece of legislation around the world is generally based on the Guidelines Governing the Protection of Privacy and Transborder Flows of Personal Data of 1980 promulgated by the OECD [18]. Those principles, as enacted by PIPEDA may be summarized as follows:

Accountability: the collecting organization is responsible for the collected data

Identifying purpose: the purpose for collecting personal information shall be identified before the information is collected

Consent: individual's consent is required for the collection or disclosure or personal information

Limiting Collection: the collection of data should be limited to those data necessary for the purpose of collection

Limiting Use, Disclosure and Retention: the collected personal information should be used only for the purposes for which it was collected

Accuracy: collected personal information should be accurate and complete. It is a collecting organization duty to maintain such information updated

Safeguards: the personal information collected shall be protected by measures appropriate to the sensibility of the data collected

Openness: the information regarding the organization privacy policies should be readily available to users

Individual Access: individuals shall access the information retained by an organization regarding such individual, and shall also pretend that the information be amended if not correct

Challenging compliant: the individual shall be able to address a challenge regarding the compliance of those principles to the organization designated individual [19].

The implementation of such basic principles that have their roots in the OECD guidelines is particularly important. In fact, jurisdictions such as the European Union, forbid the transmission of personal data when the destination of such flow is a jurisdiction with not acceptable levels of privacy protection, and this has caused some disruption of data trade between the European Union and the United States of America.

IV. Cloud Computing - Introduction

Cloud computing is the style of computing in which the users can rent infrastructure, platform or software services from other vendors without requiring the physical access to them. It divides the responsibilities of managing technologies between two different stakeholders who can be geographically situated in different corners of the world. Owing to this advantage, the cloud computing has been widely adopted. MarketsandMarkets estimates [20] Cloud Computing market will increase from $37.8 billion in 2010 to $121.1 billion in 2015 at a compound annual growth rate of 26.2 percent.

Figure 1 presents the evolution of cloud computing. Early on in the development of the internet there were computers that connected to the internet using dial-up, ISDN, T1 or T3 lines. They were then replaced by powerful servers at the (TCP) Internet access points. A single server was then replaced by a rack of servers for power hungry applications. Later the same rack of servers were shared between two or three applications and users, to optimize its usage of services. Moreover, a new paradigm of software as a service evolved where standalone desktop applications were slowly moved to powerful servers for ubiquity and more reliability. Cloud computing evolved out of this stage, where multiple vendors can dynamically provision resources based on their requirements and the resources allocated to them can grow or shrink like an elastic.

Cloud computing (access) can be implemented in three different models (Figure 2); Infrastructure as a Service, Platform as a Service and Software as a Service.

In Infrastructure as a Service, the users can rent the physical/virtual machines from the cloud computing vendor and the user installs the basic software in the machines. The cloud service provider (CSP) can also expose some of the machine renting capabilities as an public API, which can be utilized by the users for dynamic provisioning [21]. Amazon, Rackspace and Slicehost are some of the popular providers of Infrastructure as a Service.

In Platform as a Service, the provider encourages the users to develop their application using the platform provided by the CSP (eg: Google App Engine, Microsoft Azure). The users, while developing their applications using the platform, need to only worry about the expression of their application through the platform. The provider optimizes their infrastructure for the platform and the application once installed in the platform will seamlessly scale and the scalability will be the responsibility of the CSP.

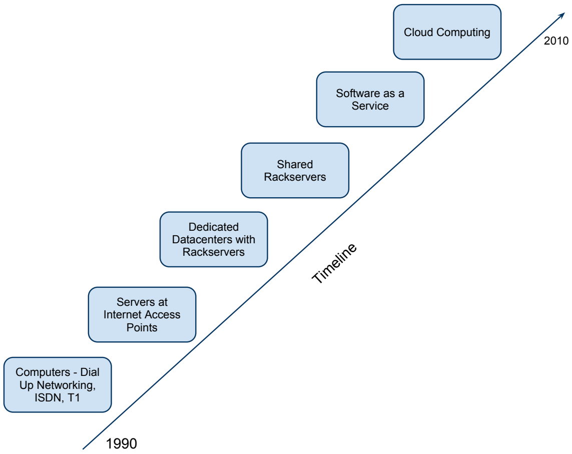

Cloud Computing
2010
Software as a Service
Shared Rackservers
Dedicated Datacenters with Rackservers
Timeline
Servers at Internet Access Points
Computers - Dial Up Networking, ISDN, T1
1990

Figure 1. Evolution of Cloud Computing

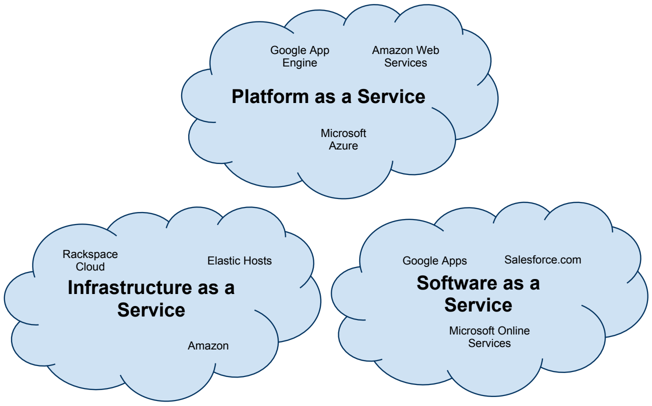


Figure 2. Cloud Architectures

In Software as a Service, the provider implements the software for the client and then provides a virtual container to host the client specific data in the software (eg: Google Docs, Salesforce etc). In this case, the client needs neither to have the technical expertise to host the application or scale nor the expertise to develop the application. The CSP uses its infrastructure to provide the services to the client.

In all the above models, the CSP is responsible for hosting the users data. The user loses the control of the data once it reaches the CSPs data grid. The user is entrusting the provider with data, because either the user has no infrastructure to host the data by themselves or assumes the data will be reliably stored in the cloud providers infrastructure as the cloud provider is trusted to have the necessary expertise to reliably store the data.

This exposes one of the major issues with Cloud computing. Cloud computing paradigm requires disturbing levels of trust by users in the servers that hold their information. Unless there is some means of totally obfuscating the data, the user needs to trust that the data stored by the CSP will be used by them only for the purposes for which it is intended to be used.

V. Threats to Data stored in CSP

There are variety of ways the datas privacy or security can be compromised in a cloud computing environment [22]. Some of them are the following:

A. *Sharing of data with an unauthorized party*

Cloud provider could compromise the confidentiality of the data by sharing the data stored in the system to unauthorized parties. This can go against the terms and conditions of the service and will qualify as the breach of security and contract. The end user could never be aware of such a breach, even if it happened.

B. *Corruption of data stored*

As the cloud computing provider has root access in the physical machine, they will have rights to modify/delete the data. Cloud provider can tamper with the data making the data non-usable or modify the data in a way that system cannot detect the modification. This poses serious threat to the correctness of the application.

C. *Malicious Internal Users*

The employee of a cloud computing provider who has root access to these physical machines, can easily access the data and use it for their advantage.

D. *Data Loss or Leakage*

When a virtual machine is used in an infrastructure, it poses a variety of security issues [23] which could lead to a compromise. Moreover, when the facility which hosts the user's data is subjected to a natural calamity, that would risk the loss of the user's data.

E. *Account or Service Hijacking*

Another risk for the cloud computing provider is, if the service is hijacked, or the computer is hacked by a hacker, the hacker will have full access to the data. As the cloud infrastructure is not under the client's control, it could be more prone to attack as the risk profile of the infrastructure will be unknown to the client.

To summarize storing the data in the cloud, can increase the privacy risks for the following stakeholders:

1) Cloud Computing User
2) Organization using the Cloud Service
3) Implementors of Cloud Platforms
4) Providers of application on top of cloud platforms
5) For the data subject

VI. Approaches to Addressing Privacy Issues in Cloud Computing

There are variety of ways in which the user can ensure that data is protected from the cloud computing provider or the cloud computing provider is made accountable for the data stored. Privacy Enhancing Technologies (PET) can be used by the developers of the application to enhance the privacy of individuals in an application development environment.

Some of PET include:

1) Privacy management tools that enable inspection of server side policies about handling of personal data
2) Secure online access mechanisms to enable individuals to check and update the accuracy of their personal data
3) Anonymizer tools which will help users from revealing their true identity by not revealing the PII (Privately Identifiable Information) to the CSP.

A. *Privacy By Encryption*

Privacy can be enforced by encrypting all the data that is stored in the CSP. The main issue with that architecture is that the cloud provider can be only used for storage of the data. As the data will be unrecognizable for CSP, it will not be possible for CSP to process the data or perform some number crunching tasks on it.

Searchable encryption employs an algorithm that allows users to encrypt the data and then provide the server with trapdoor information [24], so that the server can search for a given string through searchable encryption algorithm. Public Key Encryption with Keyword Search (PEKS) [24] is one of the seminal works in the area of making encrypted data searchable. The authors of PEKS propose to encrypt the message using the Public-Private key infrastructure. Along with this cipher text a Public-Key Encryption with Keyword Search (PEKS) of each keyword is append to the final message. The PEKS has the trapdoor information, which is the extra information sent to the server along with the encrypted keyword for the server to test for the existence of a keyword. Searchable encryption research is at its nascent stage and it is limited only to exact word searches for now.

B. *Privacy By Secure Computation*

Another way to perform computation in the server in a secure way is using secure computation algorithms. The secure computation algorithms enables users to compute use the infrastructure from a insecure environment for computation without revealing the exact input for the computation. Yaos protocol [25] provides some of the basic techniques to perform a computation in a secure way without revealing the inputs. Yaos protocol forces the expression of a computation problem in terms of logical circuit using gates. The input of each gate is randomly encrypted and then then final resulting output is decrypted to get the exact answer of the computation. The encryption and the decryption is done at the clients end. The expression of a simple problem using

Yaos protocol is found to be complex. Hence it still resides in the theoretical realm.

C. Privacy By Using Secure Coprocessors

Secure coprocessors are currently the only realistic way to perform general-computing even when the adversary had direct physical access to the device (in our case adversary can be the CSP itself). It is a very limited computer with its ROM, RAM and battery backup for persistent storage and an ethernet card. When installed in a computer, they can be seen a secure area inside a computer that even the main processor cannot access. Privacy as a Service [26] recognizes these factors and proposes a system architecture in which a coprocessor is installed in every cloud computing system. The data loaded into the cloud is classified based on its significance and security by the cloud user (No Privacy, Privacy with Trusted Provider, Privacy with Non-Trusted Provider). The data tagged with Privacy with Non-Trusted Provider level is processed by the secure coprocessor.

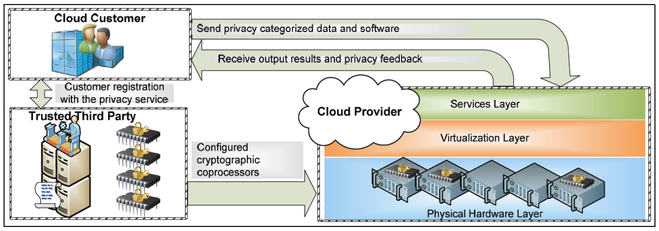

Cloud Customer
Send privacy categorized data and software
Receive output results and privacy feedback
Customer registration with the privacy service
Trusted Third Party
Cloud Provider
Services Layer
Virtualization Layer
Configured cryptographic coprocessors
Physical Hardware Layer

Figure 3. System Model for Privacy by Secure Coprocessors [26]

Figure 3 [26] is an example of a system built using secure coprocessors. Cloud customers, Trusted Third Party and the Cloud Provider are the three main stakeholders of this system. The coprocessor is signed by secret keys by the trusted third party and then is supplied to cloud provider. When a new customer registers with the cloud provider, they share the secret keys with the trusted third party. The co-processors can directly contact the trusted third party for the keys to encrypt the secret data within the coprocessor. The data channel between the co-processor and the trusted third party is secured using a mutually agreed upon public/private key pair during the initial time of supply of co-processors to CSP by trusted third party.

One secure coprocessors cost in the order of hundred thousands, even though PaaS provides some reasoning for the economics behind using them in CSP's machines, for now it looks highly unrealistic to use a coprocessor in the server infrastructure.

We discussed the technical options available to protect user's privacy by having minimum or no trust with the cloud service provider. In all the solutions we noted down the inability of these models to be used in the current cloud environment.

There is a pressing need for the law to provide legal protection to the cloud clients, as they need to trust the cloud provider with their confidential data.

VII. Conclusion

We have discussed some of the issues that confront cloud providers and users, in particular when facing the growing requirement for privacy of data in a growing number of jurisdictions. Although some partial privacy solutions have been suggested, it is unlikely that any of these can be adopted by the providers in the current cloud environment. We are working to develop other approaches to securing privacy for users of clouds, and ensuring that the dangers presented in the clouds are transparent to such users.

VIII. Thanks

The authors thank the Social Science and Humanities Research Council of Canada and the IBM Center for Advanced Studies for their support.

References

[1] *Entick v. Carrington*. 1558-1774 All E.R. Rep. 45.

[2] S.D. Warren and L.D. Brandeis. The right to privacy. *Harvard Law Review*, pages 193–220, 1890.

[3] Kyllo v. United States, 533 U.S. 27 (2001).

[4] T Nabbali and M Perry. Going for the throat: Carnivore in an echelon world. part 1. *Computer Law and Security Report*, 19(6):456–467, 2003.

[5] T Nabbali and M Perry. Going for the throat: Carnivore in an echelon world. part 2. *Computer Law and Security Report*, 20(2):84–97, 2004.

[6] T Nabbali and M Perry. Introducing carnivore: Going for the throat with precision surveillance, TLF v3 0031 (2004).

[7] R. v. Dyment, [1988] 2 S.C.R. 417.

[8] Department of Communications and Department of Justice. Privacy and computers: A report of a task force. Information Canada, Ottawa, 1972.

[9] R. v. Wise, (1992), 70 C.C.C. (3d) 193 (S.C.C.).

[10] R. v. Wong, (1990), 60 C.C.C. (3d) 460 (S.C.C.).

[11] R. v. Duarte, (1990), 53 C.C.C. (3d) 1 (S.C.C.).

[12] Privacy act, (R.S., 1985, c. P-21).

[13] Access to information act, Act R.S.C. 1985, c. A-1.

[14] Ontario province: Freedom of information and protection of privacy act, (1988).

[15] Municipal freedom of information and protection of privacy act, (1991).

[16] Personal health information protection act, (2004).

[17] Personal information protection and electronic documents act, (2000, c. 5).

[18] The Organization for Economic Co-Operation and Development. Guidelines on the protection of privacy and transborder flows of personal data.

[19] Takach G. *Computer Law*, pages 329–330. Irwin Law, 2nd edition (July 2003).

[20] MarketsandMarkets.com. Cloud computing market - global forecast (2010 -2015).

[21] Amazon.com. Amazon ec2 webservices.

[22] Cloud Security Alliance. Top threats to cloud computing v1.0.

[23] T. Garfinkel and M. Rosenblum. When virtual is harder than real: Security challenges in virtual machine based computing environments. In *Proceedings of the 10th conference on Hot Topics in Operating Systems-Volume 10*, page 20. USENIX Association, 2005.

[24] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology-Eurocrypt 2004*, pages 506–522. Springer, 2004.

[25] A.C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160–164. Citeseer, 1982.

[26] W. Itani, A. Kayssi, and A. Chehab. Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures. In *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 711–716. IEEE, 2009.

Legal Consequences of Packet Inspection

Mark Perry
Dept. of Computer Science - Faculty of Law
University of Western Ontario
London, Ontario
mperry@uwo.ca

Thomas Margoni
Dept. of Computer Science - Faculty of Law
University of Western Ontario
London, Ontario
tmargoni@uwo.ca

***Abstract*—Sophisticated network management is now very common. However, the legal consequences in terms of the liabilities, whether civil or criminal, of the Service Provider in connection with the type of management used have been poorly explored. In this work in progress, we identify the research questions, the methodology and work hypotheses of our future research.**

***Keywords*-Deep Packet Inspection; Telecommunication Intermediaries; ISP Liability; Safe Harbours.**

I. Introduction

Access to the internet is seen by most as a fundamental right [1][2]. It is not just about leisure, email, tweeting, accessing Facebook or Google maps, but rather access to the information that has become a prerequisite to freedom of expression in the modern world. It is about fundamental rights connecting the services for citizens from governmental bodies, such as obtaining a birth certificate, a temporary working permit or to e-vote (where applicable). In the 21st century, the internet has become the means to achieve a deep realisation of fundamental rights such as freedom of association, of thought, of pluralism, of communication, of realisation of one own happiness [3][4].

Most importantly, the Internet itself is not about commerce. This is a key point. It does not mean that you cannot commercialise your products or services on line. On the contrary, the creation of new business methods based on the virtualisation of value has been, is, and will be of fundamental importance for the development of economies especially during the current harsh financial times. Nonetheless, the nature of Internet is not to be a network of businesses. It is to be a network of people, who might want to do business, to form a Facebook friend (or to unfriend) somebody, or to elect their representatives. The internet is a platform that has become a social paradigm of our time and of our anthropological evolution as human beings [5].

We are living during a revolution that is much more pervasive than what the Industrial revolution has been some 250 years ago. The economic, social cultural, legal and anthropological modifications that happened then are still under analysis, though nobody doubts that it has been a major cornerstone in human evolution. It has also been said that for the success of the industrial revolution more fundamental than the invention of the steam engine has been the legal invention of the limited liability for incorporations [6]. Through this legal tool, the allocation of risk and benefits changed the old paradigm: it allowed, fostered, and offered the fundamental incentive to the accumulation of capital necessary for risky enterprises that otherwise would have not been undertaken.

The digital revolution is happening simultaneously almost wherever in the world, and in just a fraction of the time it took for the Industrial one. Let us take the example of Blu-Ray. On a single Blu-Ray disk we can store many times more information than that of a new desktop computer of five years ago, *i.e.*, comparing the five dollar disk to the drive of a 3,000 dollar computer. However, the Blu-Ray system will not be the commercial success if its predecessor – the DVD. This is despite it winning the battle against the competitor standard, the High-Definition DVD, HD-DVD [7] – resembling the Betamax versus VHS battle of a few decades ago [8]. In five years, or maybe 5 months, there will be no need for support any more. More and more the latest cutting-edge devices we can buy – or helplessly admire on the shelves of computer stores – come without an optical reader. No DVD, no Blue Ray, no ComboDrive. Did anybody noticed the progressive disappearance of the floppy disk? Although geeks, such as the authors, may keep on our desktop a five and a half inch floppy disk as an archaeological relic, as it was the leading technology of but a few years ago, Moore was right [9].

Information and knowledge will need no physical support any more in order to circulate. And every day somebody reminds us that we are living in a knowledge society or that now the businesses are based on information assets. Expressions such as Software as a Service, Cloud Computing, Web2.0, or their business implementations, GoogleDocs, OviMaps, EC2, etc. are nothing more than a confirmation that everything is translated into information. A lot of information is sent over fibre-optic cables or 3G or 4G networks. Physical support is becoming too slow, and too costly, and do not offer the same level of control that streaming and packet sniffing permits. Everything will be sent over the internet, such as money and knowledge, and furthermore relationships formed.

In terms of economic analysis of the law, to allocate upon users, or even worst, telecommunication intermediaries, the liability for what is transmitted over the Internet (such as that which may violate someone else's intellectual property or privacy, etc.) can be analogised to corporations having to pay for the their debts with the personal assets of the shareholders, beyond the face value of their shares. No limited liability any more. However, whereas governments and policy drafters have never put industrial revolution legal key concept under debate, the same does apparently not hold true for the digital revolution key concept. To charge Intermediaries operating as mere conduits with the legal liability of the potentially infringing content transmitted on their wires would stop the digital revolution, it would stifle innovation, it would disrupt new business methods in favour of the rent-seeker positions of those who have based their success on the old business paradigm. Not differently from those farmers that many years ago started suing the first commercial flights for trespassing the air over their fields, just because Blackstone Commentaries reported that property is a right that extends over the land and up to the stars [10].

In light of this futurist scenario, some legal amendments such as the ”three strikes and out” provision of the HADOPI legislation in France [13], or proposals that at regular time intervals pop up internationally, to modify the liability profile of internet intermediaries, such as ISPs, are particularly threatening. In particular, the former states that if somebody is allegedly illegally downloading copyrighted material three times, her Internet connection will be cut. No more downloads. No more Facebook friendships. No more birth certificates. No more e-voting (where applicable). Whereas the protection of the legitimate interest of the copyright holders is out of question here, and it is widely agreed that measures to foster their business methods are necessary, the guise which many times these reactions take, as in the HADOPI legislation, are the worst we could image: the declared and legally sanctioned statement that the a few bucks of royalties are more important than constitutionally recognised rights. It is surprising and frightening that a country such as France (that has spread the light of Enlightenment over most of the world only a few centuries ago) falls back to such an obscurantist vision of the future.

For these reasons we aim to analyse the current situation in terms of the transmission of information over the internet. We look to information flows in a switched packet network, how it can be identified by the likes of deep packet inspection (DPI), the legal consequences of such identification (ISP liabilities), and which are the best policies that should be implemented.

II. How information is sent over the Internet

The default for the internet (TCP/IP) is based on sending pieces of data over the net as fast as possible. Communications are chunked into packets that are sent over the network toward their common destination. Packets of the same communication may take different routes to get to destination in the most fast, efficient and non-congested way. So, packets of different kind and of different communications travel together around the network. The way in which they are delivered, the general rule, is first-in first-out. This kind of design implies that there is not packet discrimination connected with the source, destination, content, type, carrier, etc. Every packet is treated equally. For example every packet suffers the same way and amount of latency, even regardless whether the packet is of a kind that is time-sensible or not (audio-video packets are treated like http packets, even though they are differently affected by delays in delivery). For this very reason it is argued that the internet, beside the fact that TCP/IP is open and publicly available, and it follows an end-to-end design, has grown so fast [14][15].

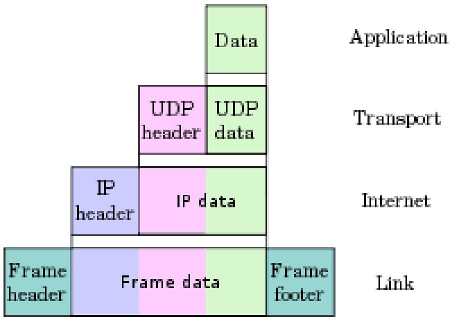


Figure 1: Encapsulation of Internet communications [11]

In such a scenario, no prioritisation of packets (*i.e.*, of type of communications) is envisioned. Some have argued that discrimination of packets might increase network efficiency. It is indeed true that, over an always more congested network, it might be efficient to prioritise those packets that are time sensitive. If a Web-page is visualised on the client browser top-down or bottom-up, it makes little difference for the end user. Contrast this with the increasing use of specific on-demand services. If the end user is visualizing a video or audio (or both) streaming, a delay of a few milliseconds might create a de-synchronization of the images and the audio. This would be noticed and not appreciated by the end user. In the case of a VoIP communication, it could render the communication useless. However, this type of packet discrimination, based on purely technical grounds, is not usually under debate. Also those who strongly advocate against packet discrimination, or in other words, Network Neutrality, do not argue on this aspect, and there are already implementations of communication techniques that try to limit packet latency of time sensitive data flow. The point of Network Neutrality has been already exposed elsewhere

[16].

Here we briefly recall the main features. In general, the usual arguments made at this regard may be summarised as follows:

A. 1) A usable and healthy Network:

To avoid a too high usage of the bandwidth by a few categories of users and to fix problems of slow and congested networks, bottle- necks, and similar problems (allegedly caused not by low investment in infrastructures, but by high usages of P2P networks). Many counter argue that the easy way to get this is to allocate a limited amount of bandwidth to any user and limit its usage to this given amount. The sum of the total amounts is what a given piece of network is able to carry. However, what usually happens in the wholesale (and partially also retail) market of cable companies and ISPs is quite similar to the behaviour known as overbooking by air companies: since statistically speaking is very unlikely that all the users use all their allocated bandwidth at the same time, it is possible to sell more bandwidth than that available, in a way that increases revenues with a very little probability of vexing users. Sometimes this same argument is sold as a benefit to users, arguing that they get more for their money.

B. 2) Price discrimination:

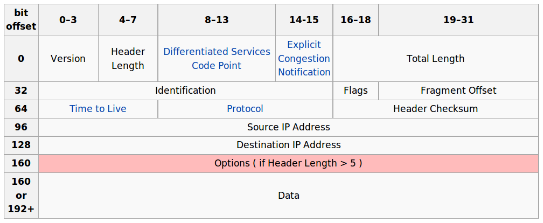
By dividing the market, ISPs can internalise the maximum consumer surplus. If an ISP can determine that some categories of users are interested only in basic services, say surfing and emails, while others need more variegate services, like connecting to Virtual Private Network (VPN) servers and Voice over Internet Protocol (VoIP), and the ISP is further able to accordingly shape/limit connections, then it will be able to sell the basic service to those customers who wouldnt pay a higher fee for extra services, while still charge a higher price to those who need the extra services. In this way, *i.e.*, through market segmentation, ISPs are able to charge the maximum price that each category is willing to pay for a given service and internalise a great share of consumer surplus, raising revenue but disadvantaging consumers. Such a situation is typical of those markets characterised by non perfect competition, *e.g.*, oligopolies.

C. 3) Vertical integration economies:

The same company may own the cable, sell the connectivity, and offer related services (*e.g.*, content purchase, emails, hosting, Television, VoiP, etc). The problem here is that of unfair competition, *i.e.*, if the company is a telephone company it is probably not happy with consumers using VoIP solutions, or at least not third party VoIP services that are sometimes a free of charge. If the ISP is a TV company, then you should rent its films, and not from another online store, or at least if one does it through her ISP store the download speed is faster. This kind of vertical integration represents a typical anticompetitive behaviour.

III. How information is inspected over the Internet

Deep Packet Inspection (DPI) is a set of methodologies used for analysis of data flows on the Internet. It is the intention of this research project to enter in the technical details of this issue [16]. However, it is clear that by using DPI technologies it is possible to know the content of TCP/IP communications. In contrast to other techniques, such as Stateful Packet Inspection where only the headers of the packets are inspected, through DPI the entire content of the packet is inspected and read. We have already indicated that data transferred over the Internet is ”chunked” into small pieces of data (called packets) and those packets are sent out individually over the network, so that they can reach the final destination in the most efficient way. Packets don’t get lost (usually) because the type of information necessary for their correct routing is present in their headers. So when a router receives a packet, the only think the router has to do is to look at the header and identify the information regarding the final definition and forward the packet to that place. When all the packets corresponding to a TCP/IP communication have reached the final destination (usually on a random order, depending of the different latencies, congestions and speeds of the different paths undertaken), the receiving device (application) rebuilds the communication following a specific packet order. Such information (the order in which packets should be ”re-assembled” for a correct representation of the carried information) is once again a type of information contained in the packet header [see Fig. 2]. This is of course an oversimplification of a TCP/IP data transfer. Much more information is contained in the headers, such as for example port numbers, etc. However, the exposed paradigm holds true.



bit offset	0–3	4–7	8–13	14–15	16–18	19–31
0	Version	Header Length	Differentiated Services Code Point	Explicit Congestion Notification	Total Length	
32	Identification				Flags	Fragment Offset
64	Time to Live		Protocol		Header Checksum	
96	Source IP Address					
128	Destination IP Address					
160	Options (if Header Length > 5)					
160 or 192+	Data					

Figure 2: IPv4 Packet header [12]

At this point, is apparent that for a correct routing of TCP/IP packets the content of such packets is completely helpless. What is necessary is the content of the headers. Once source, sequence, destination (and the rest of the identified informations) are read, the data flow can successfully happen. The content of the packet is not necessary for routing purposes.

However, the content of the packets may become interesting for other types of activities. Consider the following scenario: a subject (A) is interested in what another subject (B) is communicating to a third subject (C). If A has enough access/control of the physical Network (say it gains control by breaking into the ISP or Cable Company mainframe, or technically speaking, is the ISP or Cable Company), one of the techniques A could easily use is Stateful Packet Inspection. If A can overlook what communications originate from B and what are received by C, A can easily identify communications from B directed to C. A can also infer additional information from the communication: depending on the time, length, port, it is possible to say, for example, that the communication was a SSH, a VPN, or a P2P. Such type of analysis can provide interesting information to subjects such A that are interested (legitimately or, more commonly, not) in what is sent over the Internet. Nevertheless, following this pattern, it is not possible to identify precisely the content of the information. Imagine the same scenario, but A now uses DPI tools. We said that DPI permits to read the information contained inside TCP/IP packets. Many times this type of intrusion into somebody else communications does not provide the intruder with a clear idea of the content, mainly for the already reported routing pattern of TCP/IP packets. Since they are sent following many different routes, it is not easy to collect enough packets as to rebuild the content of the information. However, if A has the type of control that we said it has in our scenario (A controls everything happens in its Network) then A can easily read the content of any communication that originates and ends inside its network. Not only from A to B or vice versa, but any communication that takes place within the limits of the Network under its control. In fact, if A can sniff all the packets, can read from the headers the source, destination, port, and sequence number, plus can read also the information carried in the body of the packet, A has a total control over the communication. Total control means not only read, but potentially also write privileges.

IV. Future work

Legislation in many jurisdictions regarding ISP liability, or more generally the liability of communication intermediaries, usually has a ”safe harbour' provision', which has its roots in the WIPO Copyright Treaty [17]. The agreed statement in article 8 reads: ”It is understood that the mere provision of physical facilities for enabling or making a communication does not in itself amount to communication within the meaning of this Treaty or the Berne Convention”. As usual in international agreements, the wording is vague and does not provide any hermeneutic tool or policy guidance for the development of national legislation. In future research we intend to address the relationship existing between the usage of packet inspection technologies, especially DPI, and the international and national legal and regulatory situation in terms of privacy protection, consumer protection, IP protection, and the exemptions that service and content providers enjoy.

Acknowledgment

The authors thank the Social Science and Humanities Research Council, IBM Center for Advanced Studies for their support, and colleagues for their useful comments and observations.

References

[1] See French Conseil Constitutionel decision n. 2009-580 of June 10th 2009, available in english http://www.conseil-constitutionnel.fr/conseil-constitutionnel/root/bankmm/anglais/2009580dc.pdf.
All the online resources cited in this work have been last accessed during the month of November, 2010.

[2] See BBC world poll, available at http://news.bbc.co.uk/2/hi/8548190.stm

[3] See the United Nations Universal Declaration of Human Rights, adopted on 10 December 1948, in Paris, especially artt. 18, 19, 20, 26, and 27.

[4] See The Canadian Charter of Rights and Freedoms, in the Constitution Act 1982.

[5] Castells, M., The Rise of the Network Society, The Information Age: Economy, Society and Culture Vol. I. Cambridge,(1996, second edition, 2000) MA; Oxford, UK

[6] Ireland, P., Limited liability, shareholder rights and the problem of corporate irresponsibility, in Camb. J. Econ. (2010) 34 (5): 837-856.

[7] See http://en.wikipedia.org/wiki/HD_DVD

[8] See http://en.wikipedia.org/wiki/Betamax

[9] See the Moore's Law http://en.wikipedia.org/wiki/MooreLaw

[10] See Lessig, L., Free Culture, New York, 2004, p. 3

[11] Source:http://en.wikipedia.org/wiki/File:UDPencapsulation.svg

[12] Source: http://en.wikipedia.org/wiki/IPv4header

[13] Loi favorisant la diffusion et la protection de la cration sur Internet : Loi n2009-669 du 12 juin 2009 parue au JO n135 du 13 juin 2009

[14] Saltzer, J., – Reed, D., – Clark, D.D., End-to-End Arguments in System Design. Second International Conference on Distributed Computing Systems, pages 509-512, April 1981. ACM Transactions on Computer Systems, 2(4), pages 277-288, 1984

[15] Lessig, L., – Lemley, M., The End of End-to-End: Preserving the Architecture of the Internet in the Broadband Era; UC Berkeley Law Econ Research Paper No. 2000-19; Stanford Law Economics Olin Working Paper No. 207; UC Berkeley Public Law Research Paper No. 37

[16] Perry, M., – Margoni, T., Interpreting 'Network Discrimination' in the CRTC and FCC, in Digital Society 2010, 2010, 301.

[17] WIPO Copyright Treaty, 20/12/1996 , adopted in Geneva on December 20, 1996

From e-business to e-laws and e-judgments: 4,000 years of experience

Luigi Logrippo
Département d'informatique et ingénierie
Université du Québec en Outaouais
Gatineau, QC Canada
luigi@uqo.ca

Abstract— **Rapid e-transactions are possible today in many areas of application, which creates a need for rapid resolution of conflictual situations potentially deriving from the performance of these transactions. This will lead to the development of e-laws, e-regulations, e-judgments, and e-enforcement, to be quickly and automatically executed when conflictual situations occur. Examples of possible application of these ideas are found in cloud computing, privacy, security, e-business.It is shown that some principles for the implementation of these ideas can be found in the history of law, starting from very ancient legal systems that looked like sets of logic axioms or computer programs, reflecting the will of the legislator to tightly control the judicial authorities. The role of ontologies for creating complex legal systems, useful to formalize e-laws, is discussed. Principles of consistency and completeness of legal systems are briefly presented. The relevance of Artificial Intelligence methods for e-judgments is briefly evaluated. The principles presented in this work have potential for application in future automated cyberlaws contexts.**

Keywords – cyberlaws; electronic commerce; electronic laws; electronic judgments; electronic courts; electronic enforcement; legal ontologies; completeness of law; consistency of law

I. INTRODUCTION

Rapid e-transactions are possible today in many area of e-business, but there are no mechanisms to quickly address conflictual situations that may derive from them. We conjecture that the need for rapid decision of litigation in contexts of e-transactions will lead to e-laws and e-regulations, to be used by automatic e-courts, leading to e-judgments and e-enforcement, and we present principles that can be used for the development of these concepts.

In Section II we present several examples of situations where these concepts could be useful, in the areas of cloud computing, privacy, security, e-business. The rest of the paper deals with concepts that can be used for the formulation of precise e-laws, e-regulations and e-judgments. In Section III, we leap back thousands of years to show that some structuring ideas that could be used for the formulations of e-laws were known in ancient civilizations. In Section IV we show how ontologies can be used to precisely structure legal systems. In Section V we deal with the problem of completeness and consistency of legal systems. Section VI briefly addresses the use of artificial intelligence methods to arrive at e-judgments. Section VII discusses enforcement and e-penalties.

II. MOTIVATING EXAMPLES

Following are some examples showing the practical usefulness of the concept of e-judgment in the e-business and cyberlaw context. Several other examples can be generated with some imagination. A consequence of this is the usefulness of the concepts of e-laws or e-regulations, which in this paper will be taken to be laws or regulations that can be automatically inferred from, leading to e-judgments. E-courts will be automated mechanisms capable of performing these inferences.

Example 1: Service-Level Agreements (SLAs) for telecom or cloud computing. Subject A leases a line or contracts a cloud computing agreement with operator B. They agree that entity C will arbitrate any disagreements, and they deposit with C an electronic, formalized SLA, specifying certain elements of *QoS* (Quality of Service), as well as penalties for non-compliance. Later A has reason to believe that the promised QoS is not being delivered, and advises B, who disagrees. A then contacts C, who performs some tests or consults existing logs and agrees with A, therefore it sends B an order to pay A a penalty. This is completed within seconds (concepts needed to understand this example are elaborated in [13]).

Example 2: Privacy protection. Suppose that a web query tries to access an external database, but the database access control system denies access on grounds of privacy protection. The requesting agent may have been programmed to appeal this decision by automatically sending a query to an electronic system set up by a body such as a Privacy Commissioner. The latter, after considering the privacy status of the requesting agent and of the data being requested, may prescribe that access should be provided. This e-judgment would be sent to the database access control system, which would allow access.

Example 3: *Security.* This is an area were many types of violations can occur, some of which can be reliably logged. Some of these can be covered by laws or regulations for which the premises can be objectively checked. If an independent, certified log exists that A's machine has tried to snoop in B's, B's machine can automatically request that A by fined, or requested to pay damages by an e-court. Similar examples can be found in the areas of *privacy* and *copyrights*.

ISBN: 978-1-61208-122-9

Example 4: Electronic bidding. A government provides regulations for electronic bidding processes. Bidders deal with individual departments, but a central e-authority has been set up for appeals of contractors against decisions of the departments, regarding compliance with governmental regulations. Departments whose software is not up to date with the current regulations may see their decisions automatically reversed. Suppose that recently the central authority has simplified bidding procedures, but this has not yet been implemented locally.

Example 5: Tax law. Local tax laws may be in contradiction with principles of state or federal law. Or some businesses could charge taxes according to erroneous criteria. Since in many situations taxes are calculated by computer, can these calculations be corrected rapidly by intervention of an e-authority?

Example 6: E-commerce. An online buyer receives goods that do not have the advertised characteristics, or receives them later than promised. Can a quick decision on fair compensation be reached with the help of an e-authority?

Such scenarios are not realistic today because they depend on much relevant information being electronically available, e.g., for Example 1 a precise agreement is needed, together with methods to check whether the terms of the agreement are satisfied, as much as possible independent of human intervention. However, setting up such systems seems to be feasible in many cases.

Once these judicial or quasi-judicial processes are put in place, one can see that in time more areas of application will open up, towards judicial areas that have been traditionally occupied by human courts, especially in situations where decisions can be taken in terms of elementary facts and basic reasoning. The area of commercial law seems to be a prime candidate. In some cases the fact-finding may have to remain in human hands, but still the legal consequences can be automatically derived.

III. Historical precedents

We will show in this section that some structuring concepts that are important for the design of electronic legal system have been known for a very long time, some in fact from the historically recorded beginnings of legislation. The following examples are only a few out of many that could be cited.

A. *Ancient examples of precisely formulated laws*

The first codes that we know are the Sumerian and Babylonian codes of 4000 years ago. These codes were written in a precise, concise and factual style that is familiar in IT today. Here is an article from the Ur-Nammu code, said to be the earliest law code known [18]:

“If a man had let an arable field to a(nother) man for cultivation, but he did not cultivate it, turning it into wasteland, he shall measure out three kur of barley per iku of field.”

We find here the Event-Condition-Action (ECA) style that is familiar in event-driven architectures and active database system [3]. Further, the event consists of three parts: subject, verb, object according to the structure familiar in access control systems, e.g., in the XACML language [12], namely:

Subject: *a man*

Verb: *had let*

Object: *an arable field*

Condition: *but he did not cultivate it, turning it into wasteland*

Action: *he shall measure out three kur of barley per iku of field*

Here the action is a penalty, with a precise method to measure it. In other cases in this code the action is a legal effect, such as the loss of property. There are also articles that do not quite fit this pattern, but will fit other patterns that can be easily formalized. The famous code of Hammurabi of about 300 years later [11] follows the same style, is much more extensive, and is worth reading (although not for people averse to cruel and extreme punishments…).

The Chinese Tang code of year 653 A.D. [4] is another example of a code which is remarkable for its clear style and the intricate decisional procedures it describes. Essentially it is ECA, with few legal concepts. But in terms of Computer Science, one can recognize well-known concepts such as method invocation with parameters, loops with arithmetic, if statements, case statements etc., in the action part, for the calculation of penalties.

Here are two articles from this code:

Ex. 1: “In cases in which someone at first hit a person …, and then snatched his goods, calculate the value of the stolen goods to apply the law on robbery by force. When death resulted, the sentence is exile with labour. When he took the goods by stealth, use the law on robbery by stealth, but increase the penalties one degree. When killing or injuring resulted, apply the laws on intentional battery.”

Ex. 2: “Those who plant public or private land they do not have rights to are liable to a beating of thirty strokes for the first mu or less, increasing one degree for each five mu. After the penalty reaches one hundred strokes, it increases a degree for every ten mu. The maximum penalty is one and a half years penal servitude. The penalty is reduced one degree if the land had been uncultivated. If force was used, the penalty is increased one degree. The crops belong to the government or the owner. ”

These laws show that some legislators in the past have tried to control tightly the work of courts, so that the decisions were almost automatically determined by logical inference once the facts had been established.

In these starkly simple laws we can see the convergence of two conceptual worlds: the real world where situations can take many different aspects, sometimes difficult to classify precisely; and the logical world where a definite, verifiable decision has to be reached by logical deduction. The interface between the two worlds is impersonated by the judge, who has to map the complexity of the reality into a template leading to the decision

B. Precisely regulated legal process: the Roman formula process

The Roman civil law formula procedure [14] is another example of tightly controlled legal procedure of the past, in some cases reducing the final phase of the process to simple fact finding, followed by a logical deduction. Essentially, for each type of litigation there were pre-set formulae consisting of several parts where the main elements of the litigation were expressed in precise, stylized language. In the first phase of this process, the plaintiff approached a magistrate and the magistrate convened the defendant. The three consulted and the magistrate produced, with the agreement of all, a formula and the name of a judge. The judge was essentially an arbitrator, who was responsible for the second phase, where he carried out the instructions of the formula, resulting in a legally binding decision.

The components of the simplest formulae were the *Demonstration*, the *Intention*, the *Adjudication*, the *Condemnation*. The following description of the four basic elements is partly paraphrased from [14].

The principal function of the Demonstration was to indicate the subject matter of dispute (the cause of the action, the title of the plaintiff's right, the origin of his claim), as in the following example: "*Whereas A sold a slave to B"* or, "*Whereas A and B have asked to be assigned a judge for the partition of a farm"*. The Demonstration expressed prerequisites that were uncontested between the parties.

In the Intention, the claim of the plaintiff was expressed in conditional form, thus: *"If it is proved that A ought to convey the sum of … to B*" or: "*If it is proved that the slave in question belongs to A"* or yet: "*If it is proved that A has given silver to B, and A has kept it in bad faith"*.

The Adjudication empowered the judge to transfer the ownership of a thing to one of the litigants, and occurred most commonly in the actions for partitioning an inheritance, for dividing common property between co-partners, and for determining boundaries between neighbouring landholders, e.g.: "*Let the portion of the property that ought to be transferred to A be transferred to him."*

The Condemnation empowered the judge to condemn or absolve the defendant, thus: "*If it proved, condemn A to pay B the sum of … ; if it is not proved, let him be absolved"*.

These components could be varied in several ways, and other components were possible: this type of process was in use for hundreds of years and had to be adapted to many situations. In particular, there were elements by which each party could state other facts and respective rebuttals (*Exceptions*), all to be checked by the judge. This created a nested structure in the formula.

The formula was essentially an instantiation of the law for a specific case. It reduced what could be complex law into a format whose core was essentially ECA, Event-Condition-Action: the Event is specified in the Intention, the Condition in the Demonstration and in the Intention, and the Action in the Adjudication or in the Condemnation. In simple cases, the formula could be set up in such a way that the judge did not need to know the law, and had simply to check facts, i.e., whether the condition in the Intention was true or false (which he could do by using witnesses, inspection, etc.) The Adjudication or Condemnation followed by a simple syllogism [14], i.e., an elementary deduction in predicate calculus.

Reference [7] cites a view by which this procedure was "one whose rapidity, brevity and effectiveness has, perhaps, never been equaled" and it goes on by saying that this view is an understatement.

Today, stylized and agreed formulae are used in legal documents such as land transfer acts, insurance contracts, etc. but not normally in judicial procedures.

C. What can be learned from these precedents

From the Sumerian and Tang codes we can learn that many straightforward laws and regulations can be formulated in ECA style and then easily compiled into software code. A natural choice would be to compile them into a logic-based programming language such as Prolog. Once the facts are determined, decisions are reached automatically. The external interface for a system designed to provide the applicable decisions in real cases could be implemented by clickable boxes. When the facts have been determined and the boxes clicked accordingly by the judge, the sentence is automatic.

But of course modern legal systems are much more complex. The formula system of the Romans provided a step through which everyday legal decision-making was simplified: from it we can learn that, even in complex legal systems, the decision criteria for many legal cases can be expressed in ECA format after instantiation. In Roman times, it was the magistrate who instantiated the law in ECA form. In order to use this method in modern e-business systems, we could perform an analysis and classification of common legal complaints in this environment and then, following the law, set up appropriate formula templates for each of them in a web server. The plaintiff would scan the available formulae to find one that matches her complaint, and would fill it with her parameters. In common text, a formula may run roughly as follows: "*Whereas A has purchased cloud services from B, specifying a minimum QoS and penalties if the QoS is not delivered: A claims that the promised QoS is not being delivered (details: it is too slow, etc.) If A's claim is proved, B must pay A the sum of $X."* An e-entity can be appointed to perform arbitration; the entity will perform tests or consult logs, and may be able to reach a decision within seconds. Alternatively, a remote human arbitrator can be appointed who would examine evidence and fill in templates that would lead to quick automatic decisions according to the pattern pre-set by the formula.

Such procedures could consist of several steps. For example, if it is impossible to reach a verdict on the base of the available information, there could be formulae to request the parties to make available additional information.

The very existence of such efficient mechanisms may lead to quick agreements between the parties, without even having to use them. Although popular e-business providers such as eBay offer complaint procedures, they are not beyond improvement.

IV. The role of ontologies

A. *Ontologies of subjects and objects*

From a modern point of view, the very ancient codes we have mentioned have the shortcoming of being applicable only in very specific, punctual situations. Modern codes achieve greater generality by the use of structured concepts, called ontologies.

For example, the Ur-Nammu article of law given above can be generalized by introducing a classification of things that can be let, a classification of types of damages, and a classification of possible penalties. Such classifications can be represented in precise form by the use of ontologies.

Trivially, in a situation where there are two things that can be let: fields or houses, and two possible damages, burning or flooding, a norm of the type: *“If a man had let something to another man, but he damaged it, he shall pay the value of the thing to the other man”* can be instantiated in four possible ways:

“If a man had let a field to another man, but he burned it …”

“If a man had let a field to another man, but he flooded it …”

“If a man had let a house to another man but he burned it …”

Etc.

Such ontologies and instantiations are used by lawpeople when they apply the law. They originate from daily life knowledge, specialized knowledge such as engineering, or legal knowledge.

The term ontology has a history in philosophy. It has become a keyword in Computer Science, with a somewhat different meaning, and it is in its second meaning that will be used here. An ontology in this sense is definition of a set of concepts together with their relationships. Various ways of representing ontologies are: sets of logical axioms involving constants, data types, or diagrams (e.g., UML diagrams). Many different ontologies can be present, explicitly or implicitly, in a legal system. For example, inheritance law involves (at least) an ontology describing the structure of a family, an ontology describing rights that the deceased may hold, an ontology describing the objects on which rights can be held, and an ontology of the structure of wills.

By expressing relations in ontologies, powerful generalizations can be obtained. Following are some examples.

Judges and lawyers generalize the application of law by using analogical thinking. But this is based on implicit similarity relationships and assumptions (i.e., ontologies) such as: a norm that applies to x also applies to y if x is similar to y.

The Islamic legal system is one of many legal systems where analogical thinking has a very important role. In the Koran, the use of wine is forbidden because of its intoxicating effects. Islamic tradition then forbids the use of intoxicating drugs. This is an application of the argument *a fortiori* (for stronger reasons). This reasoning can be modeled in logic with the help of an ontology, which in this case is a partial order between intoxicating media, including the fact: *wine < drugs*. Then we need an axiom, e.g:

$x < y \rightarrow (Forbidden(x) \rightarrow Forbidden(y))$

If we wish to model the fact that performing a more serious offence involves a more serious penalty, then we need to add an ontology for penalties, with a partial order among penalties, and a corresponding axiom. For example, in an enterprise there may be an ontology of degrees of confidentiality of the type *UnClassified < Classified < Secret < TopSecret*. There may also be a hierarchy of degrees of protection. Then it is possible to introduce axioms stating that for higher degrees of confidentiality, there must be higher degrees of protection, or of stiffer penalties in case of breaches.

Many types of legal reasoning can be implemented precisely by defining appropriate ontologies. So an e-law should contain not only rules (such as ECA-style rules), but also the appropriate ontologies and axioms needed to define the full extent of the rules.

B. *Ontologies of legal concepts*

Over the millennia, the men and women of law have developed sophisticated ontologies of legal concepts. For example the Roman ‘Law of the XII tables’ (fifth Century BC) said in Table III [17]:

“A person who admits to owing money or has been adjudged to owe money must be given 30 days to pay”.

So here we have the right of the creditor to the money in a specific time span. And:

“After then, the creditor can lay hands on him and haul him to court”.

So here is the power of the creditor to take the debtor to court.

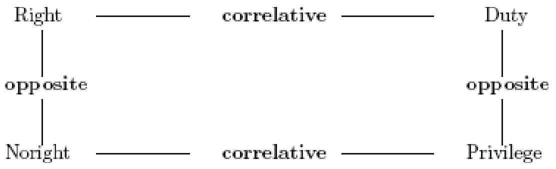
Much of modern western legal theory is constructed in terms of concepts such as these.

The American jurist Wesley Newcomb Hohfeld (1879-1918) developed a well-known ontology of these concepts, as follows:

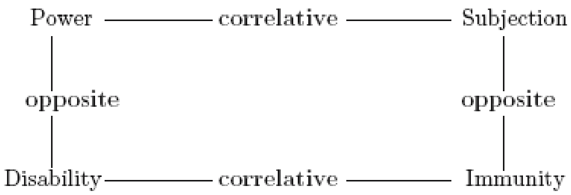
Jural opposites: 1. Right/No-Right 2. Privilege/Duty 3. Power/Disability 4. Immunity/Liability

Jural correlatives: 1. Right/Duty 2. Privilege/No-Right 3. Power/Liability 4. Immunity/Disability.

Reference [16] proposed a representation of this ontology in terms of two conceptual squares: the obligative square and the potestative square. The obligative square is as follows:



While the potestative square is:



The connection between obligative and potestative rights is provided by the fact that one subject's *x* obligative right that another subject *y* does an action is protected through *x*'s potestative right to activate the corresponding sanction against *y*.

These two squares are already implicit in Hohfeld's work. References [15] Ch. 19 and 22 and [16], complete this work by providing formal definitions of the concepts in terms of deontic concepts of obligation and permission.

Many important legal concepts are based on the concepts just mentioned. Hence, the precise formal expression of Hohfeld's ontology continues to be the subject of much interesting research.

Reference [6] presents various ontological networks of legal concepts, not related to Hohfeld's and mostly related to criminal law.

Many legal concepts are fairly precisely specified, but their complete formalization is elusive. It is elusive because they have to remain adaptable to the many situations of real life. And it is elusive because they involve reference to many concepts that one can try to formalize by using complex ontologies, higher order logics, modal logics, etc. Even if complete formalization could be achieved, automatic derivation of consequences from such complex logical systems would be daunting, because of intrinsic computational complexity issues.

V. CHECKING LEGAL SYSTEMS FOR CONSISTENCY AND COMPLETENESS

The matters of legal systems *consistency* and *completeness* were addressed in ref. [1], with citations to earlier work in Philosophy of Law. The remarks of these authors are still valid today. A legal system that is incomplete may not be able to infer a decision for legally relevant situations; a system that is inconsistent may be able to infer more than one decision. The author has presented some considerations on this topic in [7] and we should not repeat what has already been published. We will present here some concepts in order to complete the outlook of this paper. When real-life legal systems are (or appear to be) inconsistent or incomplete, this is taken care of by the (human) judges who use their discretion and knowledge (both of the law and of real life) to interpret the law. Equity and analogy are often used: what is a fair decision in this case? What was the intent of the legislator? Are there similar situations for which there is a known solution? We have seen that such thinking can be represented to some extent by using ontologies, however for complex reasoning we straddle in the area of Artificial Intelligence methods, see below.

From logic we know that first-order theories that include a significant portion of the theory of natural numbers cannot be both consistent and complete. However in practice some consistency and completeness checks can be performed by assuming a small, finite number of elements in the theory. In this case these problems reduce to the problems of consistency and completeness in propositional calculus, and can be addressed by the use of satisfaction algorithms. Although the best known satisfaction algorithms are of exponential complexity, in practice they lead to solutions in reasonable time in most cases [10]. In other words, partial completeness and complexity checks are often feasible.

Consistency. An interesting discussion of the use of 'preferences' to resolve inconsistencies in law and ethics is presented in [15] Ch. 7. This solution is similar to resolution methods already known in computing: sets of rules where inconsistencies can occur are ranked in order of priority and only the highest ranking rule is used in case of conflict.

Completeness. What does it mean for a set of rules to be complete? If complete and finite ontologies exist, it may be possible to check whether all theoretically possible situations have been considered, that a rule exists for each of them (for example for each of the four cases mentioned in Section 4.1). However since many practical situations are legally irrelevant this may lead to many unnecessary questions. The realm of possibilities can often be limited by considering the intent of the law. Suppose that the intent of the law is that no explosive packages should be sent over the mail and suppose that preventing this for different types of explosives should lead to different penalties. Since a single blanket rule is not possible, there will have to be a number of rules, for different types of explosives. Is there a rule for each possible type of explosive? This can be checked if an enumeration (i.e., an ontology) of such type exist. The knowledge that the intent of the law is limited to explosive packages makes it unnecessary to consider what should be the law for other types of packages.

VI. THE ROLE OF AI METHODS, AND HOW FAR SHOULD WE GO?

There is a very considerable research area whose aim is to study the use Artificial Intelligence methods in order to create models of legal thinking (for a brief overview and bibliography, see [2]). This work is very interesting, however often these AI methods do not lead to incontestable results, since they use heuristics that yield ‘acceptable’ solutions of which there can be several.

If AI decision heuristics were used to help deciding legal cases, one can think that different parties could bring different methods to the table, each arriving at a conclusion coherent with the submitting party’s position! This seems to be hardly worth the trouble, since it would complicate the work of the human judge who would still have to decide between the two positions, using human intelligence.

Using such heuristics would be problematic in the case of e-judgment where a single predictable decision must be reached. It appears that in this case it is necessary to use strictly deductive methods based on established facts and precisely, consistently formulated law. On this basis, any judge or any computer should arrive at the same conclusion. The existence of such laws and the possibility of such univocal deductive decisions seem to delimit the area in which the approach we are discussing is feasible.

VII. E-ENFORCEMENT AND E-PENALTIES

How to give teeth to e-laws? This does not seem difficult. One can easily see two types of penalties: monetary (fines, reparation) and exclusion, one leading to the other. So an e-party could be asked to pay a sum of money, to the plaintiff or to the platform provider, and if it does not pay, it could lose its platform or its certificate. Penalties could apply also in cases where a party refuses to collaborate in the e-judicial process, e.g., it does not reply within a specified delay. In many cases this will need no human intervention. Appeals to a human court should be allowed always.

VIII. CONCLUSIONS AND DISCUSSION

We have presented the desirability of developing automated systems of e-laws, e-regulations and e-judgments, in e-business and cyberlaw contexts, based on formally specified laws and logic deduction. Applications were found in several areas. In the initial stages, such systems will not be real legal systems; they will be used mainly in order to attempt quick resolution of complaints. However a time when a legal value will be given to them may not be distant: note that automatically produced tax assessments already have such value. A practical problem concerns how to make available the necessary evidence (e.g. system logs) in normalized electronic forms, however we can expect that in many cases this will be done eventually. Logs are required for other purposes, such as auditing.

The proponents of the use of formal logical deduction in the legal process have pointed out that such use helps towards predictability in the process, which is required for assuring the principle of certainty of law, proposed by Max Weber, among others, as necessary condition for the achievement of economic goals. The results of the legal process are more predictable and uniform if the law is logically clear and consistent and the decisions are reached by formal logical inference from the law and the established facts.

Today, a technological argument for the use of formal logic in the legal process is provided by the fact that information systems are increasingly entrusted roles of legal relevance and the most obvious mechanism for computers to draw legal conclusions is logical deduction. Multi-agent systems are very similar to social systems with their policies, which essentially have the function of laws, and are inferred and enforced automatically. However when human subjects and real-life facts are involved, the decision process may have to be more complicated, possibly requiring human participation.

We have seen that some conceptual base for such systems can be found in very ancient legal systems, and that some of the ideas used in ancient times are still valid today. More recent legal systems have tended to give more importance to the factual and human insight of the courts, something that can’t be handled by automatic systems.

It is an unfortunate habit in IT to start projects without considering what has been done before. It would be really most unfortunate if such an attitude was followed in this area.

Background discussion and references on topics related to this paper can be found in [9].

ACKNOWLEDGMENT

This research was partially funded by a grant of the Natural Sciences and Engineering Research Council of Canada. I am indebted to my students Sofiene Boulares and Waël Hassan for useful ideas.

REFERENCES

[1] C.E. Alchourròn, E. Bulygin, Normative Systems, Springer, 1971.

[2] T.J.M. Bench-Capon, H. Prakken, “Introducing the logic and Law corner,” J. logic and computation, 18(1), 2008, 1-12.

[3] K.R. Dittrich, S. Gatziu, A. Geppert, “The active database management system manifesto: A rulebase of ADBMS features,” Lecture Notes in Computer Science 985, Springer 1995, 3-20.

[4] P.B. Ebrey (Ed.), Chinese Civilization: A Source Book, The Free Press, 1993.

[5] W. Hassan, L. Logrippo, “Requirements and compliance in legal systems: a logic approach,” In: Requirements Engineering and Law, 2008 (RELAW '08), 40-44.

[6] J.C. Joerden, Logik im Recht, 2te Aufl., Springer, 2010

[7] A. Kokourek, “The formula procedure of roman law,” Virginia Law Review, 8 (5) 1922, 337-355.

[8] L. Logrippo, “Normative systems: the meeting point between jurisprudence and information technology?” In: H. Fujita, D. Pisanelli (Eds.): New Trends in Software Methodologies, Tools and Techniques, Proc. of the 6th SoMeTConference, IOS Press, 2007, 343-354.

[9] L. Logrippo, “A formal logic perspective on law, jurisprudence, and information technology in a historical context,” http://www.site.uottawa.ca/~luigi/papers/LegalLogicBlog.htm (Consulted Dec. 2010).

[10] S. Malik, L. Zhang, “Boolean satisfiability – from theoretical hardness to practical success”, Comm. ACM 57 (8), 2009, 76-82.

[11] “Mesopotamia: The Code of Hammurabi,” http://www.wsu.edu/~dee/MESO/CODE.HTM (Consulted Aug. 2010).

[12] OASIS eXtensible Access Control Markup Language (XACML) Technical Committee. http://www.oasis-

open.org/committees/tc_home.php?wg_abbrev=xacml (Consulted Aug. 2010).

[13] P. Patel, A. Ranabahu, A. Sheth, “Service level agreement in cloud computing,” Technical report, Ohio Center of Excellence in Knowledge Enabled Computing (Kno.e.sis), http://knoesis.wright.edu/library/resource.php?id=742 (Consulted Oct. 2010).

[14] E. Poste, Elements of Roman Law by Gaius. Clarendon Press, 1875.

[15] G. Sartor, Legal Reasoning: A Cognitive Approach to the Law. Chapter 5 in: A Treatise of Legal Philosophy and General Jurisprudence, Springer 2005.

[16] G. Sartor, “Fundamental legal concepts: a formal and teleological characterization”, Artificial Intelligence and Law 14 (1), 2006, 710-770.

[17] “Twelve tables”, http://www.fact-index.com/t/tw/twelve_tables.html

[18] “The Ur-Nammu law code”, http://realhistoryww.com/world_history/ancient/Misc/Sumer/ur_nammu_law.htm (Consulted Dec. 2010).

Law Modeling with Ontological Support and BPMN: a Case Study

Aaron Ciaghi, Komminist Weldemariam, Adolfo Villafiorita
Fondazione Bruno Kessler
Via Sommarive 18, Trento 38100, Italy
Email: (ciaghi,sisai,adolfo)@fbk.eu

***Abstract*—Modeling and analysis of legal documents is becoming more widely used in eGovernment practices. To support these activities, various frameworks, standards and ICT-based tools have been developed in the recent years. These approaches are mostly oriented towards defining common standards, managing legal documents and check compliance with current regulations. We have devised a tool-supported methodology that allows to model and analyze laws and procedures within public administrations. The approach used in this paper is based on the Business Process Modeling Notation for the visualization and formalization of business processes. In this paper, we show how our approach can be applied on the part of the Italian Immigration Law[1] concerning Family Reunification as a case study.**

***Keywords*-Laws; Procedures; Modeling; Ontology; Business Processes; Public Administration.**

I. Introduction

Modeling the semantics of laws is gaining attention in the field of legal informatics. Providing a graphical representation of a law can be of great advantage to those who want to understand or analyze it (e.g., citizens or jurists) as well as those who need to implement it. Furthermore, law modeling can play a key role in software engineering (e.g., [1], [2]) for the automation of Public Administration (PA) and the implementation of eGovernment systems.

Legal documents must be made available and accessible in order to facilitate any type of analysis. In order to address these issues, the governments of several countries have adopted XML-based standards for for storing and structuring legal documents (for an overview and a critique of available standards see [3]). The use of XML creates new possibilities of integration of laws with other knowledge management technologies, such as ontology based reasoning techniques and natural language processing [4], [5]. Maat and Winkels [6] also argue that in order to make law sources available to machines, they need to be translated from natural languages to some kind of formal languages.

We have been working on the development of tool-supported methodologies that facilitate modeling with the purpose of analyzing laws that describe PA procedures [7], [8]. We aim at helping the modeling of processes defined by laws, by semi-automatically extracting processes from a legal text marked with special XML tags. In the most recent version of our tool design, we introduced an ontology based intermediate representation of the information contained in laws. Our ontology is written in OWL-DL as an extension of the LKIF core ontology [9].

This paper extends our previous work [8], by presenting the application of the VLPM 2.0 approach to a concrete case study. The case study we consider is the procedure that permits legal immigrants to apply for *Family Reunification*, as defined by the Italian Immigration Law. Requirements for the family reunification request depend on various conditions as detailed in the law (e.g., the availability of suitable housing and sufficient income). We focus on (legal) documents that define, regulate or in some way affect the family reunification procedures. Note that such legal documents should ideally be shown to be contradiction-free both internally and with respect to the governing policies that need to comply with certain regulations. Moreover, there must be a mechanism ensuring that the procedure is respected. In other words, procedures should be modeled and made available for further analysis. In this paper we are only interested in the modeling aspect of the family reunification procedures, and are not concerned with their analysis – possibly formally against legal requirements.

The goal of this paper is to show how our approach can be applied on a real case study. We also intend to discuss the difficulties of applying the current approach as well as its shortcomings. In the next section, we present some background and related works. We present the core concepts of the VLPM 2.0 approach and describe its modeling steps in Section III. In Section IV, we apply these steps onto the family reunification case study. Finally, in Section V, a brief analysis of the current limitations and outline possible future work.

II. Background

Existing technologies and techniques in the legal informatics field include standards for publishing (e.g., AKOMA NTOSO [10]), annotation of laws with context-specific legal ontologies (e.g., Legal Knowledge Interchange Format (LKIF) core ontology [9]) as well as modeling and formally checking laws against legal requirements [11], [12], [13]. Moreover, works that concentrate on the use of visual modeling languages to represent Public Administration procedures as business processes in order to redesign such procedures have been disucussed, e.g., in [14], [15].

[1] D. Lgs. 25 July 1998, n. 286, updated with all amendments up to 15 July, 2009.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-122-9

The approach used in this paper is based on BPMN (Business Process Modeling Notation) for the visualization and formalization of business processes and on OWL-DL for the specification of a business process ontology that extends the LKIF-core [9]. LKIF-core is an ontology designed as part of a generic architecture for legal knowledge systems. It supports concepts like actions, agents (which correspond to UML actors, with the difference that agents must play a role to perform an action) and organizations. However, process related concepts are not as detailed as legal concepts and thus need refinement in order to be used in our methodology.

Our ontology (from now also called VLPM 2.0 ontology) has been developed in order to add semantic information about processes described in legal texts, by extending the concepts of LKIF-core with a business process meta-model that borrows several entities from the BPMN meta-model. The VLPM 2.0 ontology is not a specification of the BPMN meta-model in OWL. Instead, it abstracts the core entities of a business process from the BPMN meta-model in order to obtain a smaller but more generic ontology. In this way, a set of instances of the classes in such ontology can easily be translated to BPMN as well as UML Activity Diagram entities.

The integration of two complementary methodologies was introduced in [17], as it was inspired by the VLPM [7] and Nòmos [18] methodologies. While the latter is a modeling framework that extends a goal-oriented modeling paradigm for arguing about compliance of requirements, the former is a modeling methodology that follows a BPR-based approach with a particular focus on PA processes. The integration of these two approaches suggests a top-down reasoning in which the leaves of a Nomos model are the procedures of a related VLPM model.

The integration of different knowledge management technologies is gaining interest as a tool to aid the introduction of eGovernment solutions. For example, Francesconi et al. [5] introduce an integration of ontologies with law modeling and analysis to help in assessing decisions in software design for public administration applications. Agnoloni et al. [19] discuss the growing interest towards linguistic and semantic technologies due to the need to overcome the problems of access and knowledge of the legal information. These instruments are also a methodological necessity to approach the ever growing problems related to multilingualism in legal text, to the harmonization between EU and National legislation and to the comparative analysis of Law.

Finally, works that describe how to use modeling languages and formal techniques for modeling, specifying, and analyzing business processes and workflows are well described, e.g., in [20], [21]. However, little is usually said on the attempt to model laws and procedures in favor of public administration.

III. From Legal Documents to Models

In this section we describe how our approach can be used to identify information in a set of legal documents that is relevant to our modeling. We divide this into three phases:

1) *Markup*: in this phase we add semantic information to parts of the text that are relevant to the domain that we are going to model. We do so by marking them with tags defined by the AKOMA NTOSO schema. This will be used later to link parts of the text to elements of our model.
2) *Transformation to RDF*: the parts of the text marked as elements of our model are translated to instances of classes of our Business Process Ontology. We obtain a set of RDF statements that represent our model and that are used for traceability.
3) *Conversion to BPMN*: finally, we convert the RDF statements into a BPMN model of the process by using a set of translation rules.

Since interpretation plays a key role in jurisprudence, these phases can hardly be automated. How a PA procedure is implemented is usually not directly described in the text of a law and is thus inferred by expert “users”. For this reason, user interaction is required in order to produce an accurate model. However, in this section we will not discuss an actual implementation of this approach and we thus omit any reference to user intervention. In what follows, we discuss these three phases.

A. Markup

As noted earlier, several formats for legal documents markup are available. Since we intend to make our approach applicable to any legal system and country, we have chosen the AKOMA NTOSO (AN) framework as our input format. AN presents a clean and reiterated structure as well as ontology support (which is a stronger requirement in our case since we intend to use RDF/OWL as interchange format). Furthermore, AN is supported by tools (developed in the context of the Africa i-Parliaments initiative[2] by UNDESA) for authoring and managing legal documents.

AN documents contain a metadata section that contains several subsections to specify identifiers for the documents and information related to the publication of the paper-based version of the document. The `<lifecycle>` and the `<references>` sections are of particular interest to us as the former allows to specify the events in which each document has undergone, while the latter allows to list the entities, individuals, concepts and other documents. All references must thus be explicitly declared in the `<references>` subsection of the `<metadata>` section and are classified by Top Level Classes (TLCs). Each reference has a URI attribute that points to an external resource – in our case, this will be the URI of the instance of a class of our ontology – and an ID attribute that identifies the reference inside the document in which it appears, as illustrated by the following piece of code:

```
<references source="source-link">
   <TLCPerson id="name" href="..." showAs="..."/>
   <TLCRole id="author" href="..." showAs="..."/>
   ...
</references>
```

[2] http://www.parliaments.info/

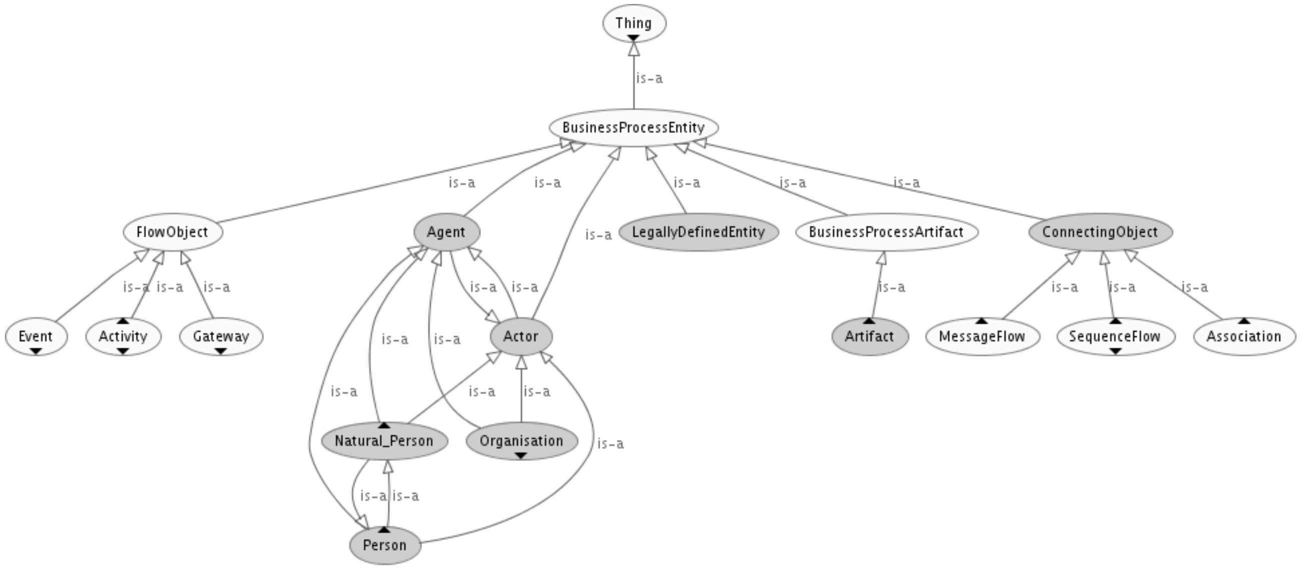


Fig. 1. Core classes of the VLPM 2.0 ontology of business process entities (inferred model). Classes from the LKIF-core ontology are also shown in the diagram (see [22] for a detailed discussion).

Moreover, AN provides several mechanisms to link snippets of the text to declared references. In general, the optional `refersTo` attribute can be used for any block element. However, since we want to keep traceability as fine grained as possible, we consider the use of inline references as a better solution. Therefore, a piece of text should be tagged with a `<span>` tag with the `refersTo` attribute set to the ID of a declared reference, as in the following snippet:

```
<span refersTo="applicant">
   [a] legal alien resident who applies for
   Family Reunification
</span>
must provide proof of availability of
<span refersTo="suitable.housing">
   suitable housing in compliance with
   the sanitary requirements,
</span>
```

B. Transformation to RDF

The next step is mapping textual elements to instances of ontology classes in an RDF Store. Figure 1 shows the core classes of our ontology that represent (business) process entities. The ontology is based on LKIF-core in order to be able to connect business process concepts to legal concepts. This allows us to maintain the models and the laws "synchronized" by relying on a triplestore containing instances of the classes of our ontology. A triplestore is a purpose-built database for the storage and retrieval of RDF meta-data, in this case backed by an OWL ontology. We generally call this an "RDF Store". All the above-mentioned concerns are implementation details and are incorporated in the design of VLPM 2.0. Due to space limitation we are unable to present them; the technical details can be found in [8], [22].

Traceability has to be maintained both from laws to models and from models to laws. AN's ontology and RDF Store support provide us with all the machinery needed to maintain traceability between laws and models by establishing links between fragments of documents and model elements. We achieve this by declaring references to entities in the RDF store in the `<references>` block of an AN document using TLCs. An inline reference points to a TLC reference using its local ID. As said before, each TLC reference has a URI that points to an entity in the RDF store, thereby allowing an inline reference to be connected to such entity. Backwards traceability is achieved by using an object relationship that connects a model element to an object with the same URI of the inline reference. In this way we keep laws and models synchronized, thereby allowing the evaluation of the impact of changes on both sides. A very similar mechanism is used to achieve traceability between the RDF representation and the visual representation.

C. Translation to BPMN

At this point we have instances of the ontology classes represented in RDF Store. Thus, our next step is mapping them into the process model entities. Our ontology of business process entities is designed using principles of UML Activity Diagrams and BPMN. This simplifies the generation of a model in one of these two notations from the contents of the RDF store. It should be noted that this requires to understand the meta-model elements of both the source and target. We devised a translation table (not shown in this paper due to space limitations) that maps ontology classes to AN TLCs, BPMN entities and UML entities. For example, the *Actor* class is mapped to *Person* in AKOMA NTOSO TLC, which is translated to Pool/Swimlane, and Swimlane (AD) and Actor (UCD) in BPMN and UML, respectively. Notice, however, that when mapping to activity diagrams, that automatic extraction of information about the sequentiality of activities is not an easy task. Thus, we must provide a way to personalize the elements in the RDF store by adding relations and properties that are needed to model sequentiality and temporal relationships in general. Finally, we should also mention the difficulties of linking some business process modeling notations to fragment of texts. For example, one of the core elements of business process modeling notations is the Gateway; however, there is no way to link a gateway to a fragment of text. We handle such cases by manually adding the required information when performing the mapping.

IV. Case Study: The Italian Family Reunification Law

This section presents the execution of the steps discussed previously on the Italian Family Reunification regulation case study. Without going into specific details, a legal permanent resident alien in Italy who wishes to apply for family reunification must first obtain a set of documents to prove that he or she will be able to sustain his or her family. Among these documents, one of the hardest to obtain is a certification that the applicant's house can accommodate his or her family. Once all such documents have been obtained, the applicant has to submit the actual application electronically through the website of the Ministry of Internal Affairs. The procedure includes several transactions with different public offices. For this reason, Family Reunification can be an interesting example on which we can test our approach.

We chose the Family Reunification law as case study because it is well supported by a local public organization called CINFORMI[3] whose objective is that of facilitating immigrants' access to public services. We are interested in providing CINFORMI with an automated system to provide information on such services and to automate, where possible, the interaction with its "clients". Therefore, we have taken into account different types of documents (i.e., the text of the Immigration Law and the instructions that CINFORMI provide to immigrants) to represent the whole procedure, including the interaction with CINFORMI. In conducting our case study, we followed the ideal workflow of VLPM 2.0 (cf: Section III).

1) Marking-up: We added AN markup to the relevant parts of the Unified Text and the Implementation Regulation that compose the Italian Immigration Law. We started from the text of the law (originally in PDF format) and we replicated the structure of the law using AN tags.

2) Tagging: We tagged parts of the text that identified entities of the Family Reunification business process. For the sake of example, we translated part of the unified text on Family Reunification[4] as shown in the listing below.

```
<!-- Declaration of references in header -->
<references>
  <TLCPerson name="Legal permanent resident
   alien" id="applicant"
   href="/ontology/person/actor/applicant" />
  <TLCObject name="Housing Suitability
   Certification" id="suitable.housing"
   href="..." />
  <TLCProcess name="Provide housing suitability
   proof" id="provide.housing.suitability"
   href="..." />
  <TLCProcess name="Verify Housing Suitability"
   id="verify.housing.suitability"
   href="..." />
  <TLCOrganisation name="Municipal Office"
   id="municipal.offices" href="..." />
...
</references>
...
<!-- Example statement in body -->
<span refersTo="provide.housing.suitability">
   <span refersTo="applicant">
      [a] legal alien resident who applies for
      Family Reunification
   </span>
   must provide proof of availability of
   <span refersTo="suitable.housing">
      suitable housing in compliance with
      the sanitary requirements,
   </span>
      as
   <span refersTo="verify.suitable.housing">
      verified by
      <span refersTo="municipal.offices">
        the competent municipal offices
      </span>
   </span>
</span>
```

The example specifically shows how actors, tasks and artifacts are tagged. In the example, the statement, despite being relatively vague, gives us information on one of the requirements to obtain clearance for Family Reunification, i.e. obtaining a housing suitability certification. As often happens in laws, information is spread across several documents and still its implementation is mostly left to interpretation. For this reason, in order to build a more accurate model, we had to integrate the contents of our legal sources with non-legal documents such as the instructions published by CINFORMI on their website. We found these kinds of instructions very useful not only because they help for the interpretation of the law also for determining the order of the tasks in the process.

3) Referencing: The references in tagged documents to instances of classes of our ontology allowed us to establish traceability between text and an intermediate RDF representation of the model. This has proven to be extremely time consuming and error prone without software support. However, currently we are investigating the extension of the Bungeni Editor[5] in order to back these issues faced at the moment.

4) Integrating: Finally, by integrating all the information found in the unified text and in the implementation regulation, as well as "unofficial" sources such as CINFORMI's instructions, we have been able to build a BPMN model of the process, focusing on the point of view of the applicant. Figure 2 shows the part of the model related to obtaining the housing suitability certification mentioned above. The diagram is obtained by applying a set translation rules defined in [8] from RDF statements to BPMN constructs. The diagram in figure 2 as proof of concept.

[3] Centro Informativo Per L'immigrazione: http://www.cinformi.it.

[4] Specifically, part of section 29, article 3 ("articolo 29, comma 3" in Italian).

[5] See http://www.bungeni.org (last accessed on December 15, 2010).

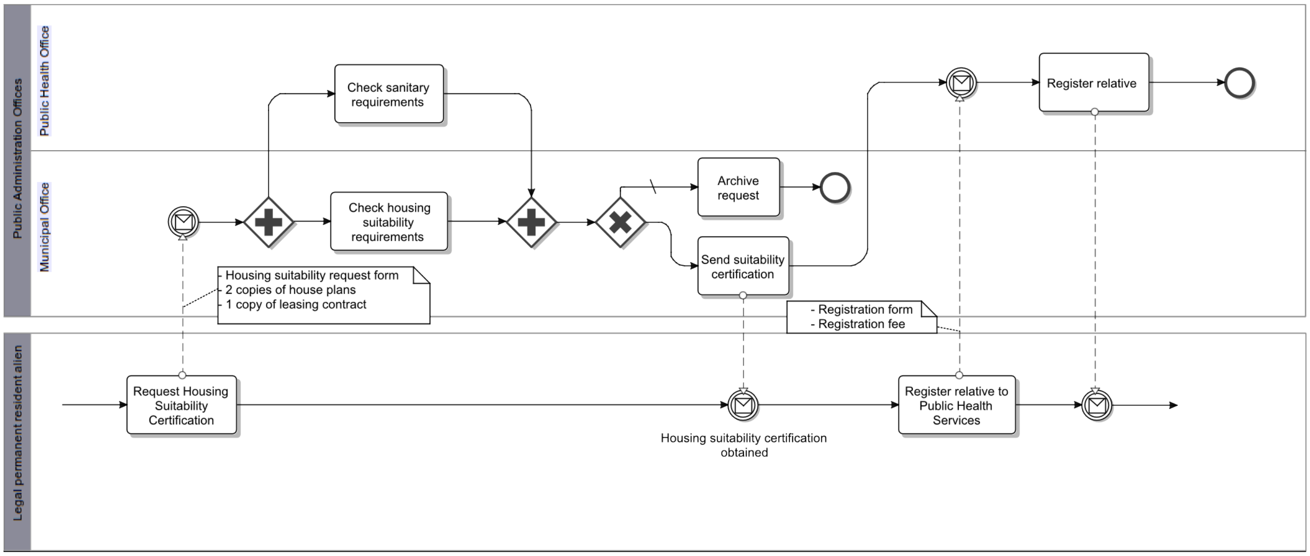

Public Administration Offices
Public Health Office
Municipal Office
Legal permanent resident alien
Check sanitary requirements
Check housing suitability requirements
Archive request
Send suitability certification
Register relative
- Housing suitability request form
- 2 copies of house plans
- 1 copy of leasing contract
- Registration form
- Registration fee
Request Housing Suitability Certification
Housing suitability certification obtained
Register relative to Public Health Services

Fig. 2. An example of the generated Business process model for the part of the Family Reunification application procedure as defined by the Italian Immigration Law.

The model in Figure 2 is just a small part of the complete business process. The complete model includes all the other actors such as the Ministry of Internal Affairs and the procedures to obtain all the other documents required for the application, containing 36 tasks and events.

V. Discussion and Future Work

Public Administrations keep facing issues due to the complexity of legislations and the continuos evolution of the body of laws of their country. Laws are continuously added, amended and repealed, often causing inconsistencies that can go unnoticed even for several decades. With the rise of transnational institutions such as the European Union, this is further complicated.

Several strategies have been proposed in the literature to model laws [6], [11], [14], [15] to assist re-engineering of Public Administration. The main differences with our approach is that in [11] there is no ontology to support model exchange and traceability; instead, a graphical editor for the User Requirements Notation (URN) – called jUCMNav – is used to evaluate the compliance of processes to legal requirements and also by establishing a traceability links between elements of the goal model and the procedure. Similarly, in [15], event-driven process chains are used to translate law paragraphs into process models with the support of semantic process language. The main goal of the authors is that of visualizing and formally model a legally regulated process. The work described in [23] is closer to ours. The authors first transform unstructured legal text into the MetaLex XML interchange format [24]. Secondly, using MetaLex they are able to find and resolve all references in the text and tag these explicitly. This allows them to easily recognize and classify norms in the legal sources [6].

In this paper we have shown how ontology and business process modeling techniques and tools can be applied to model legal documents. In particular, we have used an extension of LKIF-core and BPMN to model the procedure of applying for Family Reunification in Italy. First, we enriched the parts of the law that were interesting for us with semantic information following the AKOMA NTOSO schema. We then produced a representation in RDF of the model using the classes defined in our VLPM 2.0 ontology. This is then used to maintain traceability between the model and the text. Finally, we applied translation rules to convert the RDF statements to a BPMN model.

Considered that we analyze laws from a technological point of view, we should not underestimate the fact that laws are mainly a product of political representatives, who might have an agenda that does not include facilitating understandability. This represents the main obstacle to the introduction of law modeling as a tool to formalize law design. Notice also that the interpretation of a law for a non-jurist remains a bottleneck. This is a time and effort consuming task, usually performed by knowledge engineers with the aid of legal experts. However, some promising approaches (e.g., [6]) can be adopted in our future work, for creating an intermediate model that has an (isomorphic) representation of the structure of the original text before starting the modeling task.

Additionally, at the moment, our approach lacks a software tool to tag the text and to generate the model. Tools to perform the two activities exist, but no integration is currently available. Namely, we do not have a machinery that allows us to automatically reference instances of classes of the ontology in order to establish traceability between text and an intermediate RDF representation of the model. However, the extension to Bungeni Editor that we envisaged in [8] should be able to significantly mitigate these issues.

References

[1] A. Siena, J. Mylopoulos, A. Perini, and A. Susi, “From Laws to Requirements,” in *RELAW ’08: Proceedings of the 2008 Requirements*

Engineering and Law. Washington, DC, USA: IEEE Computer Society, 2008, pp. 6–10.

[2] A. Siena, J. Mylopoulos, A. Susi, and A. Perini, "Designing Law-Compliant Software Requirements," in *ER*, 2009, pp. 472–486.

[3] C. Lupo, F. Vitali, E. Francesconi, M. Palmirani, R. Winkels, E. de Maat, A. Boer, and P. Mascellani, "Deliverable 3.1: General XML format (s) for legal Sources," 2007.

[4] G. Barabucci and F. Vitali, "XDTD as a Simple Validation Language for XML-based Legal Documents," in *JURIX*, 2009, pp. 1–10.

[5] E. Francesconi, S. Montemagni, W. Peters, and D. Tiscornia, "Integrating a Bottom-Up and Top-Down Methodology for Building Semantic Resources for the Multilingual Legal Domain," in *Semantic Processing of Legal Texts*, 2010, pp. 95–121.

[6] E. de Maat and R. Winkels, "Automated Classification of Norms in Sources of Law," in *Semantic Processing of Legal Texts*, 2010, pp. 170–191.

[7] A. Ciaghi, A. Villafiorita, and A. Mattioli, "VLPM: A Tool to Support BPR in Public Administration." in *ICDS*. IEEE Computer Society, 2009, pp. 289–293.

[8] A. Ciaghi, "Towards a Modeling Framework for Law-Making," Master's thesis, University of Trento, Via Sommarive 14, Povo-Trento, 38123, Italy, July 2010.

[9] R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer, "LKIF Core: Principled Ontology Development for the Legal Domain," in *Proceeding of the 2009 conference on Law, Ontologies and the Semantic Web*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2009, pp. 21–52.

[10] F. Vitali and F. Zeni, "Towards a Country-Independent Data format: the Akoma Ntoso Experience." in *Proceeding of the V Legislative XML Workshop*, 2007, pp. 239–252.

[11] G. Mussbacher, S. Ghanavati, and D. Amyot, "Modeling and Analysis of URN Goals and Scenarios with jUCMNav," in *RE*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 383–384.

[12] S. Ghanavati, D. Amyot, and L. Peyton, "Compliance Analysis Based on a Goal-oriented Requirement Language Evaluation Methodology," in *RE*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 133–142.

[13] D. Gorín, S. Mera, and F. Schapachnik, "Model Checking Legal Documents," in *The 23rd International Conference on Legal Knowledge and Information Systems*, 2010.

[14] Paul Alpar, Sebastian Olbrich, "Legal Requirements and Modelling of Processes in e-Government," *Electronic Journal of e-Government*, vol. 3, 2005.

[15] Sebastian Olbrich, Carlo Simon, "Process Modelling Towards e-Government – Visualisation and Semantic Modelling of Legal Regulations as Executable Process Sets," *Electronic Journal of e-Government*, vol. 6, 2008.

[16] G. Booch, J. Rumbaugh, and I. Jacobson, *Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series)*. Addison-Wesley Professional, 2005.

[17] A. Villafiorita, K. Weldemariam, A. Susi, and A. Siena, "Modeling and Analysis of Laws Using BPR and Goal-Oriented Framework," *International Conference on the Digital Society*, vol. 0, pp. 353–358, 2010.

[18] A. Siena, "Engineering Law-Compliant Requirements. The Nòmos Framework." Ph.D. dissertation, University of Trento, Via Sommarive 14, Povo-Trento, 38123, Italy, March 2010.

[19] T. Agnoloni, L. Bacci, E. Francesconi, P. Spinosa, D. Tiscornia, S. Montemagni, and G. Venturi, "Building an Ontological Support for Multilingual Legislative Drafting," in *Proceeding of the 2007 conference on Legal Knowledge and Information Systems*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007, pp. 9–18.

[20] R. Eshuis, "Symbolic Model Checking of UML Activity Diagrams," *ACM Trans. Softw. Eng. Methodol.*, vol. 15, no. 1, pp. 1–38, 2006.

[21] C. E. Gerede and J. Su, "Specification and Verification of Artifact Behaviors in Business Process Models," in *ICSOC*, ser. LNCS, vol. 4749. Springer, 2007, pp. 181–192.

[22] A. Ciaghi and A. Villafiorita, "Improving Public Administrations via Law Modeling and BPR," in *AFRICOMM'10: E-Infrastructure and E-Services On Developing Countries*. LNCS, 2009, pp. 133–142, (To appear).

[23] E. de Maat, R. Winkels, and T. van Engers, "Automated Detection of Reference Structures in Law," in *Proceeding of the 2006 conference on Legal Knowledge and Information Systems*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2006, pp. 41–50.

[24] A. Boer, R. Winkels, and F. Vitali, "Metalex XML and the legal knowledge interchange format," *Computable Models of the Law*, pp. 21–41, 2008.

A Novel Rainbow Table Sorting Method

Hwei-Ming Ying, Vrizlynn L. L. Thing

Cryptography & Security Department
Institute for Infocomm Research, Singapore
{hmying,vriz}@i2r.a-star.edu.sg

***Abstract*—As users become increasingly aware of the need to adopt strong password, it also brings challenges to digital forensics investigators due to the password protection of potential evidence data. In this paper, we discuss existing password recovery methods and propose a new password sorting method that aid in improving the performance of the recovery process. This improved method supports a quick binary search instead of the slower linear search as employed in the enhanced rainbow table. We show that this method will result in a 23% reduction in storage requirement, compared to the original rainbow tables, while maintaining the same success rate. It is also an improvement over the enhanced rainbow table as the time taken for the password lookup will be drastically reduced.**

Keywords - Digital forensics; password recovery; search optimization; time-memory tradeoff; cryptanalysis.

I. Introduction

In computer and information security, the use of passwords is essential for users to protect their data and to ensure a secured access to their systems/machines. However, in digital forensics, the use of password protection presents a challenge for investigators while conducting examinations. As mentioned in [1], compelling a suspect to surrender his password would force him to produce evidence that could be used to incriminate him, thereby violating his Fifth Amendment right against self-incrimination. Therefore, this presents a need for the authorities to have the capability to access a suspect's data without expecting his assistance. While there exist methods to decode hashes to reveal passwords used to protect potential evidence, lengthier passwords with larger characters sets have been encouraged to thwart password recovery. Awareness of the need to use stronger passwords and active adoption have rendered many existing password recovery tools inefficient or even ineffective.

The more common methods of password recovery techniques are guessing, dictionary, brute force and more recently, using rainbow tables. The guessing method is attempting to crack passwords by trying "easy-to-remember", common passwords or passwords based on a user's personal information (or a fuzzy index of words on the user's storage media). A statistical analysis of 28,000 passwords recently stolen from a popular U.S. website revealed that 16% of the users took a first name as a password and 14% relied on "easy-to-remember" keyboard combinations [2]. Therefore, the guessing method can be quite effective in some cases where users are willing to compromise security for the sake of convenience.

The dictionary attack method composes of loading a file of dictionary words into a password cracking tool to search for a match of their hash values with the stored one. Examples of password cracking tools include Cain and Abel [3], John the Ripper [4] and LCP [5].

In the brute force cryptanalysis attack, every possible combination of the password characters is attempted to perform a match comparison. It is an extremely time consuming process but the password will be recovered eventually if a long enough time is given. Cain and Abel, John the Ripper as well as LCP are able to conduct brute force attacks.

In [6-9], the authors studied on the recovery of passwords or encryption keys based on the collision of hashes in specific hashing algorithms. These methods are mainly used to research on the weakness of hashing algorithms. They are too high in complexity and time consuming to be used for performing password recovery during forensics investigations. The methods are also applicable to specific hashing algorithms only.

In [10], Hellman introduced a method which involves a trade-off between the computation time and storage space needed to recover the plaintext from its hash value. It can be applied to retrieve Windows login passwords encrypted into LM or NTLM hashes [11], as well as passwords in applications using these hashing algorithms. Passwords encrypted with hashing algorithms such as MD5 [12], SHA-2 [13] and RIPEMD-160 [14] are also susceptible to this recovery method. In addition, this method is applicable to many searching tasks including the knapsack and discrete logarithm problems.

In [15], Oechslin proposed a faster cryptanalytical time-memory trade-off method, which is an improvement over Hellman's method. Since then, this method has been widely used and implemented in many popular password recovery tools. The pre-computed tables that are generated in this method are known as the rainbow tables.

In [16], Narayanan and Shmatikov proposed using standard Markov modeling techniques from natural language processing to reduce the password space to be searched, combined with the application of the time-memory trade-off method to analyse the vulnerability of human-memorable passwords. It was shown that 67.6% of the passwords can be successfully recovered using a 2x10^9 search space. However, the limitation of this method is that the passwords were assumed to be human-memorable character-sequence passwords.

In [17], Thing and Ying proposed a new design of an enhanced rainbow table. Maintaining the core functionality of the rainbow tables, the enhanced rainbow table has an improvement of 13% to 19% over the rainbow tables in terms of success rate or an improvement of 50% in terms of storage space.

In this paper, we present an improvement over the method in [17] by describing a way to overcome its main drawback and show that it outperforms the existing rainbow table and the enhanced rainbow table methods.

The rest of the paper is organized as follow. In Section 2, we present a discussion on the time-memory trade-off password recovery methods and how sorting plays an important role in improving the search time. We then give an overview of the sorting method in Section 3. We describe the design of the sorting method in details in Section 4. Analysis and evaluation are presented in Section 5. Conclusions follow in Section 6.

II. PASSWORD RECOVERY AND SORTING TECHNIQUES

The idea of a general time-memory tradeoff was first proposed by Hellman in 1980 [10]. In the context of password recovery, we describe the Hellman algorithm as follows.

We let X be the plaintext password and Y be the corresponding stored hash value of X. Given Y, we need to find X which satisfies h(X) = Y, where h is a known hash function. However, finding X = h^{-1}(Y) is feasibly impossible since hashes are computed using one-way functions, where the reversal function, h^{-1}, is unknown. Hellman suggested taking the plaintext values and applying alternate hashing and reducing, to generate a pre-computed table.

For example, the corresponding 128-bit hash value for a 7-character password (composed from a character set of English alphabets), is obtained by performing the password hashing function on the password. With a reduction function such as $H \ mod \ 26^7$, where H is the hash value converted to its decimal form, the resulting values are distributed in a best-effort uniform manner. For example, if we start with the initial plaintext value of ”abcdefg” and upon hashing, we get a binary output of 0000000....000010000000....01, which is 64 ‘0’s and a ‘1’ followed by 62 ‘0’s and a ‘1’. $H = 2^{63} + 1$ = 9223372036854775809. The reduction function will then convert this value to ”3665127553” which corresponds to a plaintext representation “lwmkgij”, computed from $(11(26^6) + 22(26^5) + 12(26^4) + 10(25^3) + 6(26^2) + 8(26^1) + 9(26^0)$. After a pre-defined number of rounds of hashing and reducing (making up a chain), only the initial and final plaintext values are stored. Therefore, only the “head” and “tail” of a chain are stored in the table. Using different initial plaintexts, the hashing and reducing operations are repeated, to generate a larger table (of increasing rows or chains). A larger table will theoretically contain more pre-computed values (i.e. disregarding hash collisions), thereby increasing the success rate of password recovery, while taking up more storage space. The pre-defined number of rounds of hashing and reducing will also increase the success rate by increasing the length of the “virtual” chain, while bringing about a higher computational overhead.

To recover a plaintext from a given hash, a reduction operation is performed on the hash and a search for a match of the computed plaintext with the final value in the table is conducted. If a match is not found, the hashing and reducing operations are performed on the computed plaintext to arrive at a new plaintext so that another round of search to be made. The maximum number of rounds of hashing, reducing and searching operations is determined by the chain length. If the hash value is found in a particular chain, the values in the chain are then worked out by performing the hashing and reducing functions to arrive at the plaintext giving the specific hash value. Unfortunately, there is a likelihood that chains with different initial values may merge due to collisions. These merges will reduce the number of distinct hash values in the chains and therefore, diminish the rate of successful recovery. The success rate can be increased by using multiple tables with each table using a different reduction function. If we let P(t) be the success rate of using t tables, then $P(t) = 1 - (1 - P(1))^t$, which is an increasing function of t since P(1) is between 0 and 1. Hence, introducing more tables increase the success rate but also cause an increase in both the computational complexity and storage space.

In [18], Rivest suggested a method of using distinguished points as end points for chains. Distinguished points are keys which satisfy a given criteria, e.g. the first or last q bits are all 0. In this method, the chains are not generated with a fixed length but they terminate upon reaching pre-defined distinguished points. This method decreases the number of memory lookups compared to Hellman’s method and is capable of loop detection. If a distinguished point is not obtained after a large finite number of operations, the chain is suspected to contain a loop and is discarded. Therefore, the generated chains are free of loops. One limitation is that the chains will merge if there is a collision within the same table. The variable lengths of the chains will also result in an increase in the number of false alarms. Additional computations are also required to determine if a false alarm has occurred.

In 2003, Oechslin proposed a new table structure [10] to reduce the probability of merging occurrences. These rainbow chains use multiple reduction functions such that there will only be merges if the collisions occur at the same positions in both chains. An experiment was carried out and presented in Oechslin’s paper. It showed that given a set of parameters which is constant in both scenarios, the measured coverage in a single rainbow table is 78.8% compared to the 75.8% from the classical tables of Hellman with distinguished points. In addition, the number of calculations needed to perform the search is reduced as well.

In all the above methods, the stored passwords can be sorted in their alphabetical order. When a password lookup is performed, the time taken to search for this password can therefore be optimized. Hence, the computational complexity to recover the password is low.

In [17], Thing and Ying proposed a new table structure which has an overall improvement over the existing rainbow tables. Even after taking into consideration the effects of key collisions, it was demonstrated that there was a significant increase (between 13% to 19%) in terms of the success rate of recovery, while maintaining the same storage requirement and computational complexity. The novelty of this method lies in the new chain generation process and the removal of the initial hash storage, which resulted in significant storage space conservation (or successful recovery rate improvement).

The main drawback of method is that each password search will incur a significant amount of time complexity. The reason is that the passwords cannot be sorted in the usual alphabetical order now, since in doing so, the information of its corresponding initial hash value will be lost. The lookup will then have to rely on checking every single stored password in the table.

In the following section, we present our proposed sorting method so that password lookup in the stored tables can be optimized.

III. Sorting Method Overview

Based on the method described in [17], we require sorting of the "tail" passwords to achieve a fast lookup. We introduce special characters that can be found on the keyboard (e.g. *, ', !, @, :, "). There are altogether 32 of such non-alpha-numeric printable characters and we assume for now that they do not form any of the character set of the passwords. We insert a number of these special characters into the passwords that we store. The manner in which these special characters are inserted will provide the information on the position of the passwords after the table has been re-arranged in alphabetical order. The consequence is that this will add more storage space compared to [17] but we will illustrate later that the increase in storage space is minimal and is also lesser than the original rainbow table's storage. The advantage of this sorting method is that the passwords can now be sorted and thus a password lookup can be optimized.

As an example, to recover passwords of length 7 consisting of characters in the alpha-numberic character set, and assuming there are 5700 reduction functions and 6.0 x 10^7 chains, a maximum of only 4 special characters are needed in order to span the entire 6.0 x 10^7 passwords. Since the password length is 7, there are 8 different positions where the special characters can be inserted. Hence, the total number of different values which can be obtained by inserting the special characters $>$ 324 x (8 + 8x7/2! + 8x7x6/3! + 8x7x6x5/4!) $>$ 6.0 x 10^7. Therefore, only a maximum of 4 characters need to be inserted.

IV. Design of the Sorting Method

In this section, we describe the details of computing and assigning the special characters insertion to perform the sorting, and the derivation of the corresponding initial hash value from the sorted passwords.

Let the 32 special characters be x_1, x_2, ..., x_{32}.

The password with no special character in it has an original position at 0.

If the password is xxxxxxx of length 7, we let $\underline{7}$x$\underline{6}$x$\underline{5}$x$\underline{4}$x$\underline{3}$x $\underline{2}$x$\underline{1}$x$\underline{0}$ be the password with the inserted special characters where the underlined numbers represent the positions of the special characters in the password. More than 1 special character can be assigned to each position. In addition, we define $x_i > x_j$ if the character x_i is to the left of the character x_j in the password.

For passwords with exactly one special character x_i, the original position of the password when x_i is at position a is 32a + i.

For passwords with exactly two special characters x_i, x_j where $x_i > x_j$, the original position of the password when x_i and x_j are at positions a and b respectively is 224 + 32i + j + 512a(a+1) + 1024b

For passwords with exactly three special characters x_i, x_j, x_k where $x_i > x_j > x_k$, the original position of the password when x_i, x_j, x_k are at positions a, b and c respectively is 36064 + 1024i + 32j + k + 16384a(a+1)(a+2)/3 + 16384b(b+1) + 32768c

For passwords with exactly four special characters x_i, x_j, x_k, x_l where $x_i > x_j > x_k > x_l$, the original position of the password when x_i, x_j, x_k, x_l are at positions a, b, c and d respectively is 3935456 + 32768i + 1024j + 32k + l + 131072a(a+1)(a+2)(a+3)/3 + 524288b(b+1)(b+2)/3
+ 524288c(c+1) + 1048576d

Note: In the subsequent sections, the same notations as described below will be used.
x_i, x_j, x_k, x_l are the special characters and the values of i, j, k, l range from 1 to 32 inclusive. a, b, c, d are the positions of the special characters and their values ranges from 0 to 7 inclusive.

A. Password Position Assignment

The following describes the procedure of assigning the position of the passwords in the tables to perform sorting.
Step 1: Identify the 32 special characters that do not belong to the character space of the password.

Step 2: Represent each of these 32 characters from x_1 to x_{32}.

Step 3: The first password is left in its original state without any addition of special characters. This will be the password that corresponds to the H value at the start of the chain.

Step 4: The second password will have the character x_1 inserted at the end of the chain. This will be the password that corresponds to the H+1 value at the start of the chain.

Step 5: Subsequent characters are inserted to the passwords such that all the possible characters are inserted to a position. The next higher position will then be allocated for the inclusions of these characters.

Step 6: Once all the positions for 1 character have been filled, 2 characters are used. When they are filled too, 3 characters are used and so on.

Step 7: Continue the assignment of the special characters until all the passwords, excluding the first, have been inserted special characters.

Step 8: These passwords with the addition of special characters can then be sorted in the usual way.

B. Identifying the Positions of Passwords

In the following, we describe the procedure to derive the corresponding initial hash value from the sorted passwords with the inserted special characters.

Step 1: Identify how many special characters are in the password that has been found.

Step 2: Based on the number of special characters and their positions in the password, the corresponding initial hash value is computed as follow.

(a) If there are 0 special characters, then the password corresponds to the initial hash H

(b) If there is 1 special character, then the password corresponds to the initial hash $H + 32a + i$

(c) If there are 2 special characters, then the password corresponds to the initial hash $H + 224 + 32i + j + 512a(a+1) + 1024b$

(d) If there are 3 special characters, then the password corresponds to the initial hash $H + 36064 + 1024i + 32j + k + 16384a(a+1)(a+2)/3 + 16384b(b+1) + 32768c$

(e) If there are 4 special characters, then the password corresponds to the initial hash $H + 3935456 + 32768i + 1024j + 32k + l + 131072a(a+1)(a+2)(a+3)/3 + 524288b(b+1)(b+2)/3 + 524288c(c+1) + 1048576d$

V. ANALYSIS

In this section, we analyse the maximum number of special characters required to sort tables of different sizes and password lengths, as well as demonstrate the storage conservation achieved.

Number of positions that can assigned without using any special character = 1

Number of positions that can assigned using 1 special character = 32 x 8 = 256

Number of positions that can assigned using 2 special characters
= (No. of ways to select 2 special characters) x (No. of ways to place the 2 characters into the 8 positions)
= 32^2 x [8 + 8x7/2] = 36864

In a similar fashion,

Number of positions that can assigned using 3 special characters = 32^3 x [8 + 2x8x7/2 + 8x7x6/3!] = 3932160

Number of positions that can assigned using 4 special characters
= 32^4 x [8 + 3x8x7/2 + 3x8x7x6/3! + 8x7x6x5/4!]
= 346030080

Hence, the total number of positions that can be identified with at most 4 special characters
= 1 + 256 + 36864 + 3932160 + 346030080
= 349999361

Therefore, if less than 350 million passwords are stored, at most 4 special characters are required to identify the position of each password.

Next, we compare the storage requirement between the enhanced rainbow table (incorporating our sorting method) and the original rainbow table by investigating two scenarios.

Scenario 1: 60 million passwords are stored in a rainbow table

Total storage space required for the original table = 1.02 x 10^9 bytes

Total storage space required after the passwords sorting
= 9 x 1 + 10 x 256 + 11 x 36864 + 12 x 3932160 + 13 x 56030719
= 775993340 bytes

Hence, the reduction of storage over the original method
= (1.02 x 10^9 - 775993340) / 1.02 x 10^9
= 0.2392
= 23.92%

Scenario 2: 268 million passwords are stored in a rainbow table

Total storage space required for the original table = 4.556 x 10^9 bytes

Total storage space required after the passwords sorting
= 9 x 1 + 10 x 256 + 11 x 36864 + 12 x 3932160 + 13 x 264030719
= 3479993340 bytes

Hence, the reduction of storage over the original method
= (4.556 x 10^9 - 3479993340) / 4.556 x 10^9
= 0.2362
= 23.62%

We observe that the storage requirement is still significantly reduced even with the inserted characters used in our sorting method.

Tables 1, 2, 3 and 4 show the number of passwords that can be stored in a table with password lengths of 7, 8, 9 and 10 respectively. The values in first row represent the maximum number of special characters added to each password while the values in the second row represent the number of passwords that can be stored.

1	2	3	4
257	37121	3969281	3.50 x 10^8

TABLE 1: PASSWORD LENGTH = 7

1	2	3	4
289	46369	5453089	5.24 x 10^8

TABLE 2: PASSWORD LENGTH = 8

1	2	3	4
321	56320	7208960	7.50 x 10^8

TABLE 3: PASSWORD LENGTH = 9

1	2	3	4
353	67937	9371648	1.05 x 10^9

TABLE 4: PASSWORD LENGTH = 10

We can see that even with only 4 special characters, we are able to store a very large number of passwords in the table. Therefore, a small number of available special characters is sufficient.

Discussions

A shortcoming of this sorting method is the reservation of the special characters, which prevents their usage as password characters. To resolve this, we propose using non-printable characters instead, therefore leaving the printable characters for use as passwords.

VI. Conclusions

This paper describes a sorting mechanism which when applied, has a significant improvement over the orginal rainbow tables. Special characters are added to the storage to allow the sorting of the enhanced rainbow tables so that the password lookup time can be optimized. Even with this insertion of characters to the passwords, the improvement in storage space required to store the same number of passwords is 23% lesser than what is required in the original tables. This is achieved while maintaining the same success rate. Furthermore, it has a speed improvement over the enhanced rainbow tables since it uses a binary search instead of a linear search when performing password lookup. Analysis was also conducted to show that the number of passwords that can be supported by using a small number of special characters, is very large. Therefore, this sorting method can be widely applied. In addition, to circumvent the shortcoming of reserving the printable special characters from being used as passwords, non-printable characters should be chosen for use in this sorting method instead.

References

[1] S. M. Smyth, “Searches of computers and computer data at the United States border: The need for a new framework following United States V. Arnold”, Journal of Law, Technology and Policy, Vol. 2009, No. 1, pp. 69-105, February 2009.

[2] Google News, “Favorite passwords: ‘1234’ and ‘password’”, http://www.google.com/hostednews/afp/article/ALeqM5jeUc6 Bblnd0M19WVQWvjS6D2puvw, [retrieved, December 2009].

[3] Cain and Abel, “Password recovery tool”, http://www.oxid.it, [retrieved, December 2010].

[4] John The Ripper, “Password cracker”, http://www.openwall.com, [retrieved, December 2010].

[5] LCPSoft, “Lcpsoft programs”, http://www.lcpsoft.com, [retrieved, December 2010].

[6] S. Contini, and Y. L. Yin, “Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions”, Annual International Conference on the Theory and Application of Cryptology and Information Security (AsiaCrypt), Lecture Notes in Computer Science, Vol. 4284, pp. 37-53, 2006.

[7] P. A. Fouque, G. Leurent, and P. Q. Nguyen, “Full key-recovery attacks on HMAC/NMAC-MD4 and NMAC-MD5”, Advances in Cryptology, Lecture Notes in Computer Science, Vol. 4622, pp. 13-30, Springer, 2007.

[8] Y. Sasaki, G. Yamamoto, and K. Aoki, “Practical password recovery on an MD5 challenge and response”, Cryptology ePrint Archive, Report 2007/101, April 2008.

[9] Y. Sasaki, L. Wang, K. Ohta, and N. Kunihiro, “Security of MD5 challenge and response: Extension of APOP password recovery attack”, The Cryptographers’ Track at the RSA Conference on Topics in Cryptology, Vol. 4964, pp. 1-18, April 2008.

[10] M. E. Hellman, “A cryptanalytic time-memory trade-off”, IEEE Transactions on Information Theory, Vol. IT-26, No. 4, pp. 401-406, July 1980.

[11] D. Todorov, “Mechanics of user identification and authentication: Fundamentals of identity management”, Auerbach Publications, Taylor and Francis Group, June 2007.

[12] R. Rivest, “The MD5 message-digest algorithm”, IETF RFC 1321, April 1992.

[13] National Institute of Standards and Technology (NIST), “Secure hash standard”, Federal Information Processing Standards Publication 180-2, August 2002.

[14] H. Dobbertin, A. Bosselaers, and B. Preneel, “Ripemd-160: A strengthened version of RIPEMD”, International Workshop on Fast Software Encryption, Lecture Notes in Computer Science,

Vol. 1039, pp. 71-82, Springer, April 1996.

[15] P. Oechslin, “Making a faster cryptanalytic time-memory trade-off”, Annual International Cryptology Conference (CRYPTO), Advances in Cryptography, Lecture Notes in Computer Science, Vol. 279, pp. 617-630, October 2003.

[16] A. Narayanan, and V. Shmatikov, “Fast dictionary attacks on passwords using time-space tradeoff”, ACM Conference on Computer and Communications Security, pp. 364-372, 2005.

[17] V. L. L. Thing, and H. M. Ying, “A novel time-memory trade-off method for password recovery”, Digital Investigation, International Journal of Digital Forensics and Incident Response, Elsevier, Vol. 6, Supplement, pp. S114-S120, September 2009

[18] D. E. R. Denning, “Cryptography and data security”, Addison-Wesley Publication, 1982.