Linked Closed Data Using PKI: A Case Study on Publishing and Consuming data in a Forensic Process

Tamer Fares Gayed, Hakim Lounis
Dépt. d'Informatique
Université du Québec à Montréal
Succursale Centre-ville, H3C 3P8,
Montréal, Canada
gayed.tamer@courrier.uqam.ca lounis.hakim@uqam.ca

Moncef Bari
Dépt. de Didactique
Université du Québec à Montréal
Succursale Centre-ville, H3C 3P8,
Montréal, Canada
bari.moncef@uqam.ca

Abstract—**The main aim of the Linked Open Data (LOD) project is to publish data publicly without access restriction in order to be consumed upon Unified Resource Identifier (URI) resolution. The latter provides more description about the resources being represented through the resolvability and discoverability of more others resources. Sometimes, data/resources need to undergo an access restriction to be consumed only on a small scale for keeping its confidentiality. However, while the power of the LOD resides in the resolvability of more URIs related to the resources in hand, a curious question imposes itself: how can we achieve a compromise between URI resolvability and access restriction? This paper discusses how the represented data cam be secured. It illustrates how the Public Key Infrastructure (PKI) can be applied to restrict the access to confidential resources of represented data being published using the Linked Data Principles (LDP), while maintaining the resolvability of such restricted resources. This brings out a new era of research related to the counter part of LOD, a research topic called the Linked Closed Data (LCD). A good example to elaborate this compromise question is a case study retrieved from the Cyber Forensics (*CF*) field where the tangible Chain of Custody (*CoC*) is represented using the LDP to exploit the resolvability feature of such principles on different resources of the Electronic-*CoC* (*e-CoC*). The latter should also obey an access restriction in order to be shared only between role players who published the data and juries who are going to consume it.**

Keywords-Linked Open Data; Linked Data Principles; Linked Closed Data; Public Key Infrastructure; Digital Certificates, Cyber Forensics, Chain of Custody.

I. INTRODUCTION

The classical way for publishing and accessing documents in the World Wide Web (WWW) [12] is through hypertext links, which allow users to navigate over the Hyper Text Markup Language (HTML) documents using browsers and search engines [1].

Today, the WWW has radically altered the way to share information [15]. The interrelation is not just between documents but it has evolved to also link the data within these documents (i.e., Linked Data-LD), using the same web aspects (URI [13], Hyper Text Transfer Protocol-HTTP [2]). Thus, the HTTP URIs are used not only to identify web documents but also real objects and abstract concepts in the world, the fact that allows the latter to be dereferenceable/resolvable (i.e., it means that HTTP clients can look up the URI using the HTTP and retrieve a description of the resource that is identified by this URI).

While the primary unit of the hypertext web are the HTML connected by untyped hyperlinks, the LD uses the Resources Description Framework (RDF) [3] to link such data using typed statements allowing arbitrary link of things (i.e., resources) in the WWW. The web aspects are then called the technology stack or LDP, which encompasses three components: URI, HTTP, and RDF [14]. The most visible project using this technology stack is LOD [20][4]. This project and its derivative [27] attracted the interest of many researchers of the data cloud to construct several cloud-based LD management systems [24][25][26].

Generally, the LOD aims to bootstrap the web of data by identifying existing data sets that are available under open licenses [17] (i.e., converting them to RDF according to the LDP, and publishing them publicly on the Web). The openness (i.e., no license and no access restrictions) and resolvability of resources are two likely factors in the success of this project.

The knowledge representation concept has been persistent at the centre of the field of Artificial Intelligence (AI) since its founding conference in the mid 50's. This concept is described by Davis & al. with five distinct roles [43]. The most important role is the definition of knowledge representation as a surrogate for things. In this paper context, the *e-CoC* is constructed through the LDP as a surrogate of the tangible one. Later, the resources of *e-CoC* will be then consumable by humans and machines.

However, several times, URI/URL resources need to obey some access restriction, where a specific set of people are those who are authorized to access such resources. LDP should be bended to realize the adaptation of publishing and consuming the resources on a small scale without loosing the resolvability feature of these resources. Thus, a compromise question arises in this case, how we can realize the access restriction over certain URI/URL resources while keeping the resolvability feature of the same resources. In addition, this question brings out a new era of research called the LCD [20], where the publisher would take step of imposing access restrictions to protect his information [21][7] from anonymous consumption. A very good example to elaborate this idea, is a case study retrieved from the CF field, where the tangible *CoC* is represented using the LDP (i.e., the work

ISBN: 978-1-61208-340-7

in [7], listed all the advantages of using LDP to represent tangible *CoC*). As well, this work explained in a theoretical way, how the represented resources could obey an access restriction using PKI. The framework depicted in [7] provides a PKI layer, which explains how the represented resources can be shared between role players and the juries. Current paper will not only explain how this scenario can be implemented and applied, but it is also considered as a bridge connecting two recent works; the work published in [19] and [21].

The work provided by Rajabi et al. in [19], explained theoretically how PKI is used to achieve the trustworthiness of LD and how different datasets are exchanged in a trusted way. The work provided by M. Cobden et al. in [21], outlined in a vision paper, the need to have an access restriction on the LOD. Each work apart does not provide the complete picture to realize the LCD using PKI. In [19], the work explains how the PKI can be used to secure the resources of LD, but did not put the scope on how such stuffs can be implemented and applied, and how this work can bring out a new era of research related to the counter part of LOD (i.e., LCD). However, in [20] the work outlined the need of the LCD in certain domain (e.g., business and finance), but did not refer to the PKI solution, or how the LCD can be realized. Thus, this paper complements and completes the half picture of both works, by explaining how the PKI and digital certificates are used to restrict the access of resources in the LD cloud while keeping the resolvability of such resources, and then resulting the LCD.

This paper is organized as follows: the next section, discusses the state of the art of URI identifications, the LOD Project, PKI and Digital Certificates, and *CoC* in CF. Section 3 explains in a linked data manner, how PKI is applied to LOD. Section 4 provides methodology and experimentation explaining how the digital certificates can be used to share LD resources between the role player and juries. Finally, last section summarizes and concludes the depicted work.

II. State of the Art

A. *URI Identifications*

URI is a string of characters used to identify a name or a web resource. URI and HTTP are the two essential technologies of the web upon which the LD relies on. As mentioned in the last section, we use URI to identify any entity that exists in the world. On the web, any URI is always accompanied by the HTTP, which makes the entity being represented, deferenceable/resolvable to more resources. Both technologies were integrated with HTML to structure and link web documents. Nowadays, the data presented in these documents are integrated with the RDF to structure and link different data and resources.

An RDF consists of three slots called triple: resources/subject, properties/predicate, and objects. In addition, resources are entities retrieved from the web (e.g., persons, places, web documents, pictures, abstract concepts/resources, etc.). RDF resources are represented using URIs, of which URLs are subset. Resources have properties (attributes) that admit a certain range of values or can be attached to another resource. As well, the object field can be also a literal value or a resource [16].

The essential thing to publish data is to have a unique domain/namespace minted by a unique URL owned by the publisher [14] (e.g., [38], where "*mydomain*" is a unique namespace in the WWW space) and the URI HTTP are used to relate and identify objects and abstract concepts, thereby maximizing the discoverability of more data/resources. Therefore, a common practice called contents negotiating is used by an HTTP mechanism [2] that sends HTTP headers with each request to indicate what kinds of documents are requested (i.e., is it an HTML or RDF content). The receiver (i.e., the side that receives the HTTP request or Server) can then inspect these headers and select an appropriate representation of resources. The content negotiation uses two different types of URIs [13][44][45]:

- **303 URIs (known as 303 redirect):** the server redirects the client HTTP request to see another URI of a web document, which describes the concept in question. First, HTTP request is triggered for the initial request and the second is triggered when the request is redirected to the retrieval of the appropriate format.
- **Hash URIs:** this type avoids two http requests used by the 303 URIs. Its format contains the base part of the URI and a fragment identifier separated from the base by a hash symbol. When a client requests hash URI, the fragment part is stripped off before requesting the URI from the server. This means that the hash URI does not necessarily identify a web document and can be used to identify real-world objects.

Using the first type of URI, publisher publishes in his own server (i.e., his own domain) the description of any concept using two types of representation: HTML documents containing a human readable representation about a concept, and RDF documents about the same concept. Publisher can also use three different patterns to describe a resource (e.g., resource 'x') [18]:

- URI identifying resource 'x' itself [35].
- URI identifying the serialized RDF document (i.e., RDF/XML [10], Turtle document [11] or N3) describing resource 'x' [36].
- URI identifying the HTML document describing resource 'x' [37].

Using the second type of URI pattern, publisher can define different resources and use then the Hash URI to serve an RDF/XML file containing the definition/terminology for each resource.

B. *Linked Open Data Project*

After the resources are represented and identified using URIs, they will be connected using RDF links, creating a global data graph that spans data sources and enable the resolvability of such resources to a new data source. The LOD cloud project has been constructed upon this basic structure (see Figure 1).

Thus, the LOD is based on the LDP, where URI resources are linked using typed RDF links to other resources within the same or to other data set. Two types of links can be used; links to navigate forward and others to navigate backward between resources.

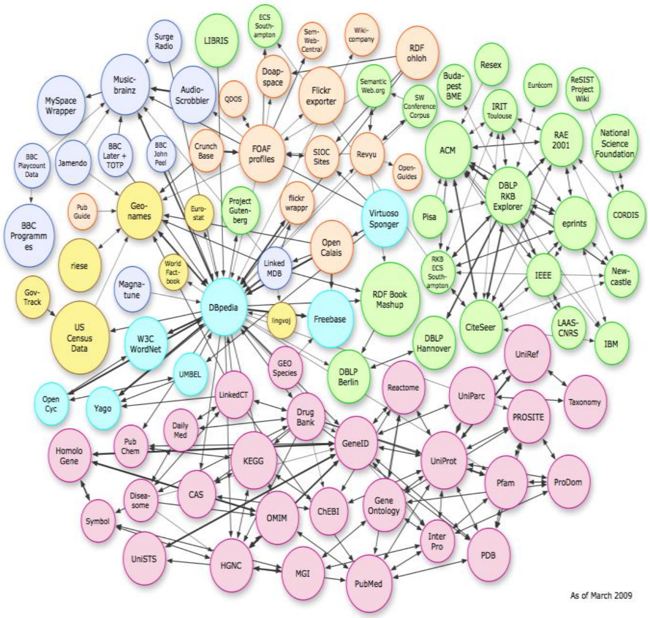


Figure 1. Linking open data cloud diagram

For example, if we have an RDF triple connecting two resources x and y, and we need to move forward from x to y, then this RDF triple should appear in the document describing the resource y. This triple is then called incoming link because it allows to navigate back to resource x. Same case, for the outcoming link, where the RDF triple should appear in the document describing the resource x and allows to navigate forward to resource y [9]. Figure 1 shows the LOD cloud diagram, where each links exists between items in the two connected data sets. Some data sets are connected together using whether, the outcoming links, the incoming links, or both.

C. PKI and Digital Certificates

PKI is a combination of softwares and procedures providing a mean to create, manage, use, distribute, store, and revoke digital certificates [47][48][49][50][51][52]. PKI called Public Key because it works with a key pair: the public key and the private key

A digital certificate is a piece of information (e.g., like a passport) that provides a recognized proof of a person/entity identity. It uses the key pair managed by the PKI to exchange securely the information in order to create trustworthiness between data provider and data consumer in a network environment [5] (i.e., trustworthiness occurs when receiver ensures the identity of the sender. This is called non-repudiation).

Any certificate contains (see Figure 2) the identity of the certificate owner, such as distinguisher's name, and information about the CA (issuer of certification), such as CA's signature of that certificate, and general information about the expiration and the issue date of that certificate [6]. Digital certificate alone can never be a proof of anyone's identity.

A third trusted party is needed to confirm and sign the validity and authority of each certificate and share securely the cryptographic key pair. This party is called Certification Authority (CA).

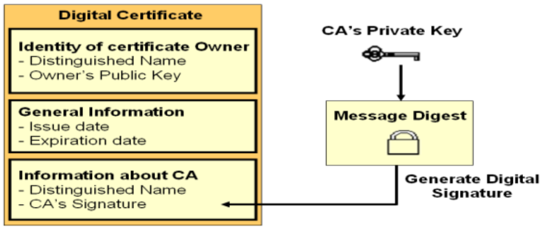


Figure 2. Digital certificate

Since a CA (e.g., VeriSign Inc., Entrust Inc., Enterprise Java Bean Certificate Authority-EJBCA, etc.) relies on public trust, it will not put its reputation on the line by signing a certificate unless it is sure of its validity, the fact that makes them acceptable in the business environment. All digital certificates provide the same level of security, whether they are created by a well-known issuer, or by unknown one. Usually, the information providers request their certificates from well-known parties when they provide services and information with large segment in society. In this paper, the authors imitate the issuer party and create CA certificate instead of buying it from well-known trusted party.

Before going further in how to adapt the digital certificates to the LD, this section should simply underline some important points related to the digital certificates:

1) Purposes

A digital certificate has various security purposes and can be used to [47]:

- Allow only the authorized participant (sender/receiver) to decrypt the encrypted transmitted information (i.e., encryption).
- Verify the identity of either sender or recipient (i.e., Authentication).
- Keep the privacy of transmitted information only to the intended audience (i.e., privacy/Confidentiality).
- Sign different information in order to ensure the integrity of information and confirms the identity of the signer of such information (i.e., digital signatures). Digital signatures also solve the non-repudiation problem by not allowing the sender to dispute that he was the originator of the sent message.

2) Protocols

In the field of Information and Communication Technology (ICT), the digital certificate is called SSL/TLS certificate because it uses two essential protocols; the SSL and the TLS [22]. The Former is the short version of the Secure Socket Layer. This protocol is used to describe a security protocol underlying a secure communication between a server and a client. After upgrading this protocol with some encryption standards, the protocol got another acronym called TLS, which is standing for Transport Layer Security. Both protocols are based on the public key cryptography [7]. They are used to establish a secure connection over the HTTP. Classically, the HTTP establishes an unencrypted connection without using the SSL and TLS (i.e., if there is some intruder around monitoring the communication between server and client, he can come with all plain data packages of such transferred data). HTTP is then extended to HTTPS to secure the connection and encrypt all the transferred data with the SSL (i.e., HTTP + SSL/TLS = HTTPS) [46].

3) *Creation Phases*

The creation of a digital certificate passes by four phases (see Figure 3) using the OpenSSL tool [8]. First step, the requester (client/server/CA) generates his own pair of keys (i.e., key file), then he creates a request (i.e., req or csr format file) to the trusted party to issue for him/her a certification (i.e., crt format file). The trusted party (i.e., CA) signs the request and issues the certificate using his own private key (i.e., when the CA is the requester of the certificate, then this certificate is considered a self-signed certificate/root certificate). The created certificate is then transformed to an exportable format (i.e., p12 format) for sending it to the requester.

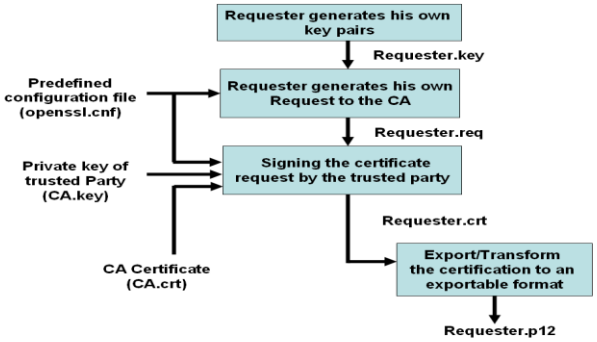

Requester generates his own key pairs
Requester.key
Predefined configuration file (openssl.cnf)
Requester generates his own Request to the CA
Requester.req
Private key of trusted Party (CA.key)
Signing the certificate request by the trusted party
Requester.crt
CA Certificate (CA.crt)
Export/Transform the certification to an exportable format
Requester.p12

Figure 3. Procedures for creating a digital certificate using openSSL tool

4) *Types and Exchange*

There exist three types of digital certificates. Figure 4 presents an abstract scenario where Alice and Bob want to share information over a secure connection (i.e., HTTPS).

Firstly, Alice and Bob should determine a third trusted party called the CA. The latter is responsible to issue SSL/TLS certificates for both of them in order that each can identify himself/herself to the other. CA issues two types of certificates.

- **Server certificate***:* this certificate is issued by the CA and it is used by Alice (i.e., suppose that she is the owner of the information) to identify herself to her authorized clients, like Bob. When Bob tries to access this server, he will be sure that he accessed the right server. Otherwise, Bob will not trust Alice information.

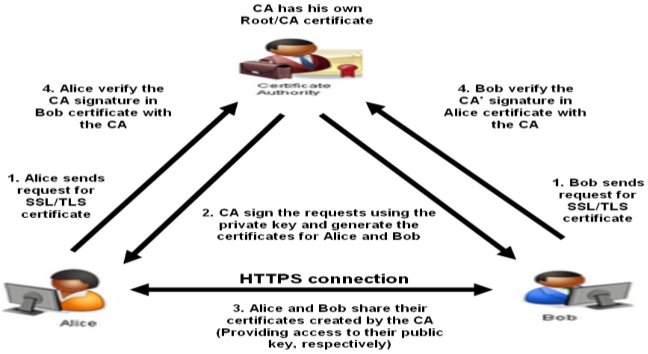

CA has his own Root/CA certificate
Certificate Authority
4. Alice verify the CA signature in Bob certificate with the CA
4. Bob verify the CA' signature in Alice certificate with the CA
1. Alice sends request for SSL/TLS certificate
1. Bob sends request for SSL/TLS certificate
2. CA sign the requests using the private key and generate the certificates for Alice and Bob
HTTPS connection
Alice
Bob
3. Alice and Bob share their certificates created by the CA (Providing access to their public key, respectively)

Figure 4. Sharing SSL/TLS certificates

- **Client certificate**: the CA issues this certificate, and it is used by Bob (i.e., suppose he is the consumer of Alice' information) to identify himself to Alice Alice will not allow any one to access her information unless he has a certificate known by her.
- **CA certificate:** CA also has the own certificate to sign the certificate requests received from the clients and servers. In addition, this type of certificate answers the question of how Alice and Bob ensure the identities of each others. Alice would know that Bob is the right person by verifying that his certificate is signed by the common trusted part authority (CA), as well as for Bob. Both know each others through the CA certificates.

From the definitions mentioned above, we notice that there is no distinguishable difference between the server certificate and the client certificate; both use the certificates to identify themselves to the other. However, the only difference that distinguishes both is about who is providing the information and who will go to consume it.

D. *Chain of Custody in Cyber Forensics*

Digital forensic is a technique for acquiring, preserving, examining, analyzing, and presenting digital evidences to the court of law. *CoC* is a chronological document that accompanies these evidences along the investigation process in order to avoid later allegations or any tampering attempt in such evidences. It provides useful information by answering 5 Ws and 1 H questions. The 5 Ws are the When, Who, Where, Why, What and the 1 H is the How.

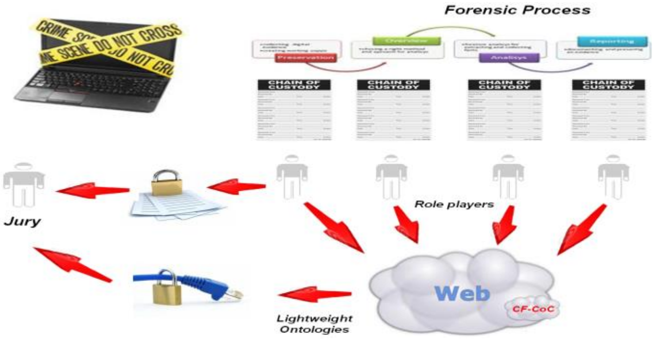


Figure 5. Abstract scenario of tangible and electronic *CoC*

According to the literature, CF includes different forensic models [19][21][26][28][32][33], each model containing a set of forensics phases, where each phase is accompanying by a tangible *CoC* describing all forensic tasks (see Figure 5).

Classically, any crime scene should obey an investigation process using a forensics model. Role player of each phase prepares his *CoC* describing all investigation tasks performed in this phase. Later, each role player submits the *CoC* securely to jury in a sealed envelope.

The work published in [7] depicts the need to transform such *CoCs* from documents to electronic data. This work proposed a framework to construct a *CF-CoC* web application hosted somewhere on the web cloud (i.e., domain known by role players and jury, e.g., [39]). The role players can use this application [7] to generate lightweight ontologies using RDF schema (RDFS) [42], representing each phase in the forensic model. Each lightweight Ontology (i.e., with big 'O') contains a set of build-in terms (i.e., retrieved from the semantic web) and custom terms (i.e., created by the role players) describing all the tasks and procedures of this phase. As the represented information should not be published publicly, the framework proposes a PKI layer that protect and foster the published information only between the role players (i.e., who published the data) and the juries (i.e., who will go to consume such data). This layer ensures the identities and authorization of all players in a forensic process.

III. ADAPTING PKI TO LOD

In this section, we will discuss how digital certificates can be applied to LOD to publish and consume data on a small scale. In other words, this section describes how digital certificates are used to restrict the access of certain resources and at the same time, such resources will be resolvable to more resources.

Referring to Figure 1 of the linking open data cloud diagram, we find several data sets interrelating using outer and/or inner links. Each data set is published in a unique domain owned only by the publisher of this data set over the WWW space. Each data set contains set of URI resources that are interrelated between each other, within the same data set or to an outer data set.

Now, imagine that the owner of a data set wants to publish resources using the technology stack/LDP of the LD (URI, HTTP, and RDF) and having such resources resolvable within the LOD cloud, but at the same time, he wishes to publish them in a manner that any anonymous parties on the web space cannot access them.

The idea to realize both features at the same time (i.e., resolvability and access restrictions of resources) resides in the digital certificates. The latter can be used to restrict the resolvability of resources in a one-way manner. With other words, the resources are restricted using digital certificates to be forward resolvable, but not backward resolvable unless the owner of such resources specify and list his authorized clients existing outer of his domain to access his resources. Same concepts can be applied between data sets/resources in the LOD cloud, where each data set owns a digital certificate(s). Thus, publisher of the resources can accomplish his publication task through an enhanced technology stack using a secure access protocol (i.e., HTTPS). Therefore, the current technology stack is transformed from (URI, HTTP, and RDF) to (URI, HTTPS, and RDF).

Imagining a scenario will be as follow: assuming that the publisher (server) and consumer (client) of the LD have already a common trusted party to issue their certificates. The publisher has a domain name named by an IP [40] (i.e., for simplicity consider this IP is corresponding to a domain string name in [39]) to publish his resources in the LOD cloud. The publisher of this domain wants only someone called: 'Jean-Pierre' to consume his resources from his domain within the LOD cloud. In this case, the publisher of the data has restricted the access to his resources to a specific consumer, but he is still able to dereference his resources and resolve them to retrieve more resources outside his dataset/domain. Publisher will be also able to move back to his domain using the backward link, because he owns the server certificate for this domain. Any other anonymous party outside this domain will not be able to access the resources of [39]. If the publisher wants someone else rather than 'Jean-Pierre' accesses his resources, this person should have a client certificate signed by the same trusted party.

Talking in a linked data manner, we can not only consider the client side as a person (i.e., as Jean-Pierre to access restricted resources), but the client side can also be a dataset or a resource within a data set that can access other resources in another data set using outer links (i.e., by moving backward to the publisher resources). In addition, another important point should be underlined; Jean-Pierre/dataset/resources can react also as a server side, if we look to the picture from the inverse direction.

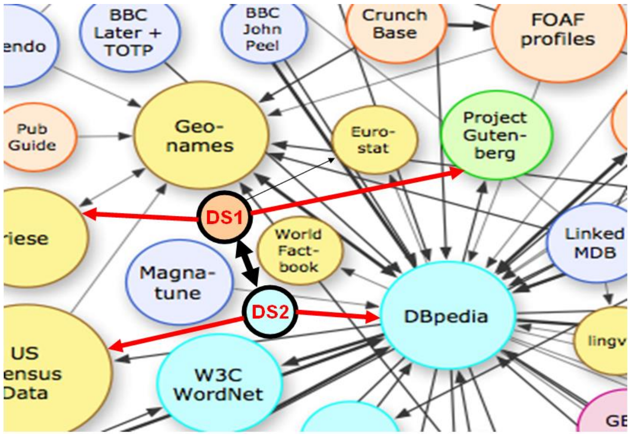


Figure 6. Client/Server certificate between two data sets

Thus, Jean-Pierre/dataset/resource may have also a server certificate for his/its domain and allows the access to only people/dataset/resource that has a client certificate to his/its domain.

To illustrate this idea, Figure 1 of the LOD cloud is zoomed-in, resulting in Figure 6. Let us consider that we have two data sets DS1 and DS2 residing in two different domains. Each domain represents a data set. Both of them are interrelated between each others using inner and outer links. As well, both data sets are related with other data sets in the LOD cloud.

DS1 and DS2 can be client and server at the same time. If we look from the DS1 to DS2, we will see an outer link from DS1 to DS2 and vice-versa. DS1 is considered as a client trying to access the server DS2. Thus, DS1 will have a client certificate for its domain to identify itself to the server certificate installed in the DS2 domain. Now, let us consider if we have the contrary view; DS2 should has then a client certificate to access the server DS1 resources. However, for any other data sets around the scope of DS1 and DS2, they will not be able to resolve their resources with resources from DS1 and DS2 (i.e., at this time, DS1 and DS2 act as servers and requires client certificates from their surrounded data sets). Therefore, the resources of DS1 and DS2 have access restriction while their resources are resolvable with different resources from the LOD cloud, but the latter cannot resolve their resources from the two data sets, DS1 and DS2.

Furthermore, the certificates cannot only used on the level of datasets (i.e., including all resources), but can also be issued on the level of a specific resource within the datasets. This can be realized by issuing the certificate using one of the three URI patterns provided in section 2.

IV. METHODOLOGY

This section explains how the digital certificates are created, installed, and used over a LD set. The experimentation is applied on a scenario between a role player and the jury to share LDP resources. This scenario is explicit. The server part is the side where the *CF-CoC* web application is hosted and owned by jury (i.e., we can see the jury as a provider, because he owns the *CF-CoC* application on his server, at the same time he is a consumer, because he will consume the data that will be published by the role player). The client part is the role player, who will use the *CF-CoC* web application to define, create, and publish the resources over this domain (i.e., we can see the role player as a consumer, because he uses the *CF-CoC* application, at the same time he is a publisher, because he will publish the data to jury using the *CF-CoC*). Role players and juries request digital certificates from the CA. The role player must have a client certificate to identify himself to the server. Jury can be also considered as a client to his server when he will consume such published data. Because jury owns a server certificate, he does not need to have another client certificate to consume the published resources (i.e., the case when the server requires a client certificate takes place when the server acts as a client to access another server. In this case, a client certificate is needed by the first server to access the second server). In our case, we have only one server [39] where the data is published and consumed.

In addition, the server provides to the CA, beside his certificate request, a list containing the names of all role players who are authorized to participate in the current forensic case.

A. *Work Environment*

The operating system used in this experimentation is Windows XP, accompagning with the Internet Information Services (IIS) [34] and the OpenSSL tool [8]. IIS simulate the machine as a server, and the OpenSSL tool is used to create the digital certificates.

1) Internet Information Services (IIS):

It is a group of internet web servers created by Microsoft. It includes two main protocols; the File Transfer Protocol (FTP) and HTTP. When installing the IIS on a machine, the web application on this machine considered as a visual basic application that lives on web server and responds to requests from the browser by processing into HTML interface code result.

2) OpenSSL

This tool implements the SSL v2/v3 and TLS. Both layers are used to create the digital certificates.

B. *Creating Digital Certificates*

This section explains how to create the digital certificate using the four procedures mentioned above (see Figure 3). Before creating the server and client certificate, a CA certificate will be created to sign both client and server requests (i.e., in this scenario, we will create manually a CA instead to buy it from a well-reputated CA). Usually, a well-known CA provider (e.g., VeriSign Inc, Entrust Inc, etc) provides the CA certificate. In this scenario, a CA self-signed certificate is manually created.

1) Self-Signed Certificate:

Before starting, the CA key is generated, *RootCA.key* of length 2048 bits (2 bytes).

```
openssl genrsa  –out RootCA.key 2048
```

The *RootCA.key* is then used to generate the certificate request *RootCA.csr* by providing the country name (i.e., C=CA), the organization name (i.e., O=Cyber Forensics Institution), and the common name of the certificate (i.e., CN=CF-CA)

```
openssl req -new -key RootCA.key -out RootCA.csr -config openssl.cnf -subj "/C=CA/O=Cyber Forensics Institution/CN=CF-CA/"
```

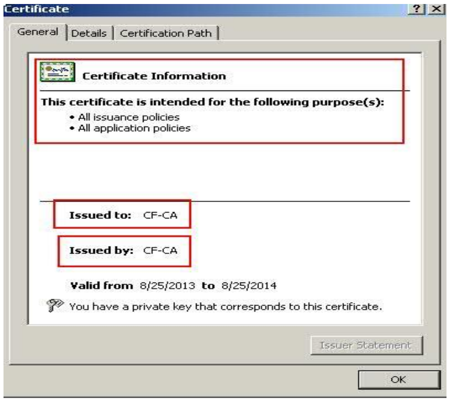
After generating the RootCA.csr, the request is signed using the RootCA.key to generate the requested certificate (*crt* format, *RootCA.crt*), but in this type of certificate, the CA itself will sign the certificate, that’s why it is called self-signed certificate:

```
openssl req -x509 -days 365 –in RootCA.csr -out RootCA.crt -key RootCA.key -config opensslCA.cnf -extensions v3_ca
```

ISBN: 978-1-61208-340-7

Finally, the exportable format p12 is generated to transform the *RootCA.crt* into an exportable format *RootCA.p12*

openssl pkcs12 -export -in RootCA.crt -inkey RootCA.key -certfile RootCA.crt -out RootCA.p12



Certificate Information

This certificate is intended for the following purpose(s):
- All issuance policies
- All application policies

Issued to: CF-CA

Issued by: CF-CA

Valid from 8/25/2013 to 8/25/2014

You have a private key that corresponds to this certificate.

Figure 7. CA self signed certificate

2) Server Certificate:

The server certificate is created for two goals: it lets the role player ensures the identity of the server, as well it is used to check for the client certificate.

As we mentioned in last section, assume that the IP in [40] is corresponding to the server in [39]. This certificate will be issued for the juries to install it on their server. This server will host the *CF-CoC application* [7], which will be used by the role player. Thus, the CA will issue and sign a certificate for this IP name.

First, the *Server.key* is generated using the following command:

openssl genrsa -out Server.key 2048

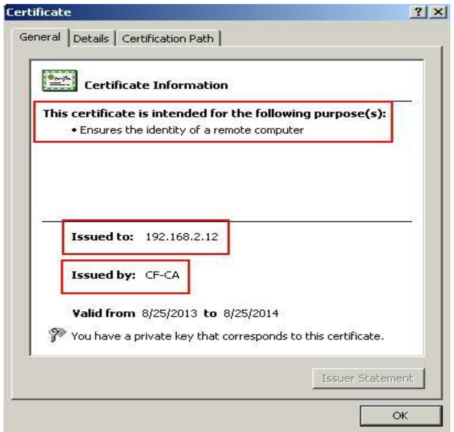
The *Server.key* is then used to generate the certificate request *Server.csr* by providing the country name (i.e., C=CA), the organization name (i.e., O=Cyber Forensics Institution), and the common name of the certificate (i.e., CN=192.168.2.12).

openssl req -new -key Server.key -out Server.csr -config openssl.cnf -subj "/C=CA/O=Cyber Forensics Institution/CN=192.168.2.12/"

After generating the *Server.csr*, the request is signed using the CA certificate *RootCA.crt* and the key *RootCA.key* to generate the requested certificate (i.e., *Server.crt*).

openssl ca -days 365 -in server.csr -cert RootCA.crt –out Server.crt -keyfile RootCA.key -config opensslserver.cnf -extensions server

Because the server certificate is signed by the CA, the *openssl* command uses a build in parameter called '*ca*', to declare that the server certificate will be signed by the CA using its key (*RootCA.key*).



Certificate Information

This certificate is intended for the following purpose(s):
- Ensures the identity of a remote computer

Issued to: 192.168.2.12

Issued by: CF-CA

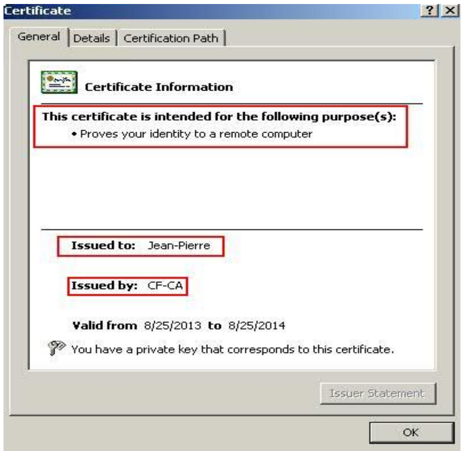
Valid from 8/25/2013 to 8/25/2014

You have a private key that corresponds to this certificate.

Figure 8. Server digital certificate

3) Client Certificate:

The role player authenticates himself to the server through the client certificate. Without this certificate, the role player will not be able to access *CF-CoC* application to construct different ontologies for each forensic phase and publish different resources.



Certificate Information

This certificate is intended for the following purpose(s):
- Proves your identity to a remote computer

Issued to: Jean-Pierre

Issued by: CF-CA

Valid from 8/25/2013 to 8/25/2014

You have a private key that corresponds to this certificate.

Figure 9. Client digital certificate

First, the *Client.key* is generated using the following commands:

openssl genrsa –out Client.key 2048

The *Client.key* is then used to generate the certificate request *Client.csr* by providing the country name (i.e., C=CA), the organization name (i.e., O=Cyber Forensics Institution), and the common name of the certificate (i.e., CN=Jean-Pierre).

openssl req -new -key Client.key -out Client.csr -config openssl.cnf -subj "/C=CA/O=Cyber Forensics Institution/CN=Jean-Pierre/"

After generating the *Client.csr*, the request is signed using the CA certificate (*RootCA.crt*) and key (*RootCA.key*) to generate the requested certificate (i.e., *Server.crt*).

openssl ca -days 365 -in Client.csr -cert RootCA.crt -out client.crt -keyfile RootCA.key -config opensslclient.cnf -extensions client

As shown in the last three figures (7, 8, and 9), we noticed that each certificate has its own purpose(s). Purpose(s) of a certificate depends on its type. The type of certificate is defined using the *-extension* in the creation of *crt* certificate. The *–extension* parameter calls the proper module for each certificate type. For example, it calls the *opensslCA.cnf, opensslServer.cnf,* and *opensslClient.cnf* for the CA, server, and client certificates, respectively. However, the *openssl.cnf* contains general configuration of all types of certificates.

C. Installation of Digital Certificates

Before installing the certificate, the CA sends to the jury and the role player their own certificates. Jury installs his certificate on his server and role player installs his certificate on his browser.

1) Self-Signed Certificate:

After creating the CA certificate, the CA sends to the server and client his certificate (i.e., p12 format without the private key of the CA certificate). By clicking on the p12 file (i.e., exportable format), a wizard will be launched to install the CA certificate in the trusted root folder of the current browsers for both server and client. By firstly installing this certificate on the server and client machines, their browsers will automatically identify the issuer of the client and server certificates.

2) Server Certificate:

The CA sends the server certificate to the jury. The latter then starts the installation of the server certificate. Installation of server certificates on Windows XP passes by two phases:

- Running the Microsoft Management Console and follow the steps in [29].
- Installing server certificates using the steps mentioned in [23].

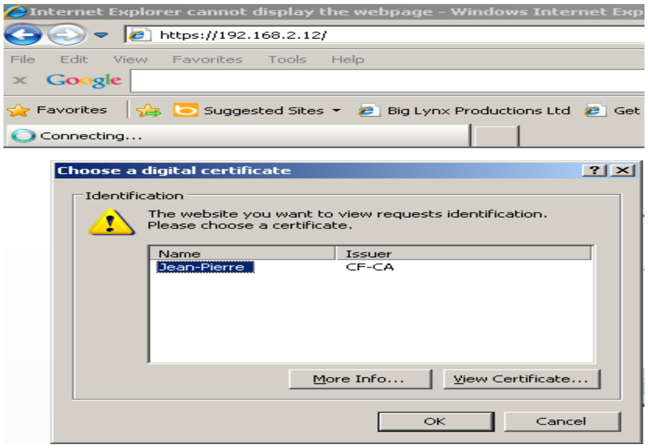
3) Client Certificate:

Installing the client certificate is the same as the CA certificate, but at this time, the wizard installs the certificate in the client/ Personal folder of the browser.

D. Experimentation

This section shows how the scenario is enrolled after the role player and jury install their certificates:

- The client accesses the site by typing the URL of the server 192.168.2.12
- Because the remote server (i.e., where the *CF-CoC* web application is hosted) owns a server certificate, it requires then that his clients also owns a client certificate owned by the same trusted party (In this case, the *CF-CA*), otherwise the browser responded with a blank page (See Figure 10).



Internet Explorer cannot display the webpage - Windows Internet Exp

https://192.168.2.12/

Connecting...

Choose a digital certificate

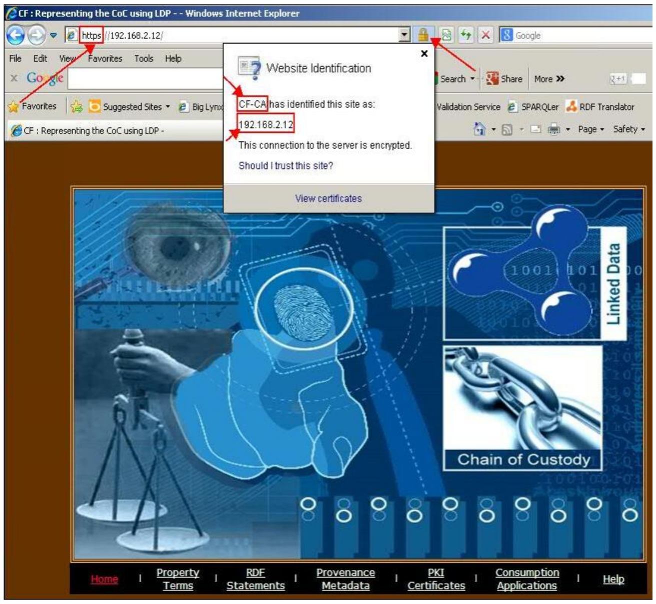
Identification

The website you want to view requests identification. Please choose a certificate.

Name	Issuer
Jean-Pierre	CF-CA

Figure 10. Server requires Client Digital certificate

- Once the server identifies the client certificate, it redirects the client to *CF-CoC* web application (see Figure 11).



CF : Representing the CoC using LDP - - Windows Internet Explorer

https://192.168.2.12/

Website Identification

CF-CA has identified this site as:

192.168.2.12

This connection to the server is encrypted.

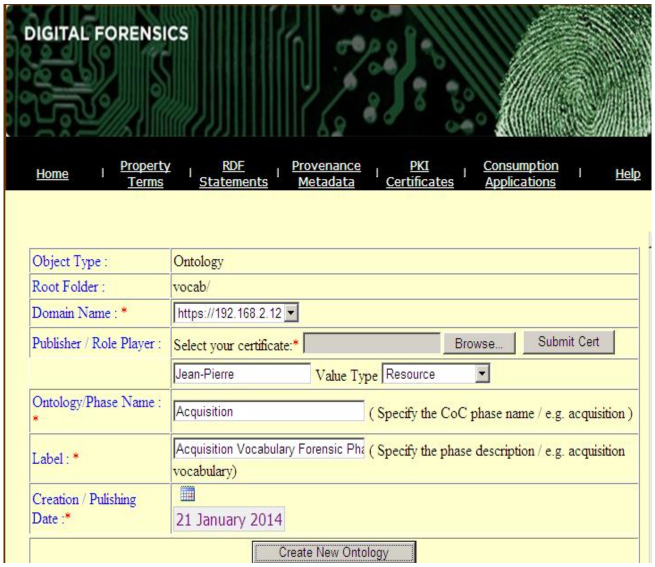
Should I trust this site?

View certificates

Figure 11. Redirection to the Restricted Resources

- Once the role player accesses the application, he starts to publish the ontologies and creates terms describing the forensic phase in hand (See Figure 12, 13).

As we see in Figure 11, the server certificate is installed and shown in the top of the screen as a yellow lock. By clicking on the lock, it will show who issued the certificate (i.e., CA) for this page and to whom it was issued (i.e., [40]).



Object Type :	Ontology
Root Folder :	vocab/
Domain Name : *	https://192.168.2.12
Publisher / Role Player :	Select your certificate:*
	Jean-Pierre Value Type Resource
Ontology/Phase Name : *	Acquisition (Specify the CoC phase name / e.g. acquisition)
Label : *	Acquisition Vocabulary Forensic Pha (Specify the phase description / e.g. acquisition vocabulary)
Creation / Pulishing Date :*	21 January 2014

Figure 12. Create lightweight Ontology phase

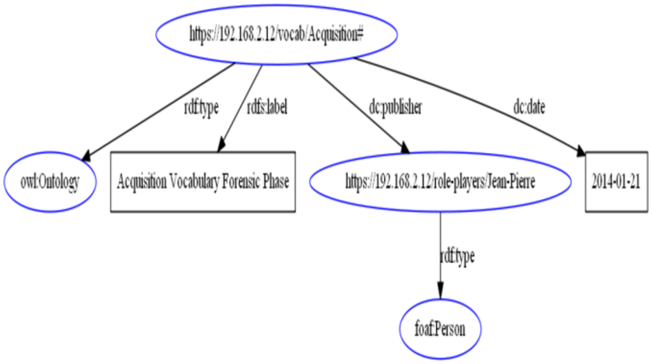

https://192.168.2.12/vocab/Acquisition#
rdf:type
rdfs:label
dc:publisher
dc:date
owl:Ontology
Acquisition Vocabulary Forensic Phase
https://192.168.2.12/role-players/Jean-Pierre
2014-01-21
rdf:type
foaf:Person

Figure 13. Ontology Acquisition phase

Once the role player finishes the publication task, the resources will be available to jury for consumption, as he owns a server certificate of the server, which allows him to view and access such resources published on his server. Resource as Jean-Pierre (see Figure 13) will be resolvable to more extra resources in the same domain [39] or to external domain [41]. However, Jean-Pierre will not be accessible from external resources outer the former domain.

As we mentioned in the last section, a certificate can not be created only for resources on the server but it can be issued for a specific resource on a server. For example, if we imagine that we have a resource 'x' in DS1, and the latter resides in the domain [35], then the field of the certificate called 'issued to' (see Figure 8) will be assigned the complete URL of the resource 'x' (e.g., *CN=192.168.2.12/resources/x).*

V. Conclusion And Future Works

This paper discusses in details how the technology stack/linked data principle of the linked data is adapted to publish data into a small scale while keeping the resolvability of these published resources. The idea is elaborated on a case study retrieved from the CF field, where the tangible *CoC* are represented using LDP. The represented resources are shared in a small scale between the role player and jury through the public key infrastructure approach. This paper opens the door to a new era of research representing the counter part of the LOD, called the LCD, which share all the advantages of the LOD, but with consumption restriction. Therefore, the technology stack (URI, HTTP, and RDF) is enhanced to include the secure access mechanism (URI, HTTPS, and RDF). The work presented in this paper is a bridge connecting dual works; the work proposed in [19] and in [21]. In addition, it underlines that the digital certificates cannot be issued only for datasets, but also for resources within these datasets. Furthermore, the current work provides with technical details the complete scenario of how to use digital certificates to bend resources from LOD to LCD, in order to answer the compromise question between the resolvability of resources and their access restrictions.

According to our knowledge, we are the first who introduced the PKI with the juridical *CoC.* On the other hand, we are implementing the two remaining layers of the framework; provenance layer and consumption layer, and we are working with a cyber criminality laboratory to define different metrics in order to evaluate our *CF-CoC* framework.

References

[1] I. Jacobs and N. Walsh, "Architecture of the World Wide Web," Volume One – W3C Recommendation", http://www.w3.org/TR/webarch/ [retrieved: Mar, 2014].

[2] R. Fielding, "Hypertext Transfer Protocol" -- HTTP/1.1. Request for Comments: 2616, http://www.w3.org/Protocols/rfc2616/rfc2616.html [retrieved: Jan, 2014].

[3] G. Klyne and J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax," - W3C Recommendation, 2009, http://www.w3.org/TR/rdfconcepts/ [retrieved: Jan, 2014].

[4] Linking Open Data, W3C SWEO Community Project, http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData [retrieved: Feb, 2014].

[5] Securing Digital Identities & Information, Entrust, http://www.entrust.com/what-is-pki/#whatis [retrieved: Feb, 2014],

[6] R. Perlman, "An overview of PKI trust models, In IEEE network," vol. 13, pp. 38-43, 1999.

[7] T. F. Gayed, H. Lounis, and M. Bari, "Cyber forensics: Representing and Managing Tangible Chain of Custody Using the Linked Data Principles," The international conference on Advanced Cognitive technologies and Application (IARIA), Valencia, pp. 87-96, 2013.

[8] Official Site of OpenSSL Project, http://www.openssl.org/ [retrieved: Dec, 2013].

[9] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao, "Describing Linked Datasets - On the Design and Usage of VoiD, 'the Vocabulary Of Interlinked Datasets'," WWW 2009 Workshop : Linked Data on the Web LDOW2009, Madrid, Spain, (2009)

[10] D. Beckett, "RDF/XML Syntax Specification (Revised)," - W3C Recommendation.http://www.w3.org/TR/rdf-syntax-grammar/ [retrieved: Jan, 2014].

[11] D. Beckett and T. Berners-Lee, "Turtle - Terse RDF Triple Language," - W3C Team Submission. http://www.w3.org/TeamSubmission/turtle/, 2008 [retrieved: Jan, 2014].

[12] T. Berners-Lee et al., "The World-Wide Web," Communications of the ACM," Vol 37, No. 8, pp. 76-82, 2009.

[13] T. Berners-Lee et al., “Uniform Resource Identifier (URI): Generic Syntax. Request for Comments: 3986,” (2005), http://tools.ietf.org/html/rfc3986 [retrieved: Feb, 2014].

[14] C. Bizer, R. Cyganiak, and T. Heath “How to publish Linked Data on the Web,” 2007 http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/. [retrieved: Feb, 2014].

[15] I. Jacobs and N. Walsh, “Architecture of the World Wide Web,” W3C Recommendation., 2004, Volume One, http://www.w3.org/TR/webarch/ [retrieved: Jan, 2014].

[16] G. Klyne and J. Carroll, (2004), “Resource Description Framework (RDF): Concepts and Abstract Syntax,” - W3C Recommendation, http://www.w3.org/TR/rdfconcepts/, 2004 [retrieved: Feb, 2014].

[17] P. Miller, R. Styles, and T. Heath, “Open Data Commons, a License for Open Data,” Proceedings of the first Workshop about Linked Data on the Web (LDOW), 2008.

[18] L. Sauermann and R. Cyganiak, “Cool URIs for the Semantic Web,” W3C Interest Group, Note 2008, http://www.w3.org/TR/cooluris/ [retrieved: Mar, 2014].

[19] E. Rajabi, M. Kahani, and M. Angel Silicia, “Trustworthiness of Linked Data Using PKI,” World Wide Web Conference (www2012) Lyon, France, 2012.

[20] C. Bizer, T. Heath, and T. Berners-Lee, “Linked Data— The Story So Far,” International. Journal on Semantic Web and Information Systems, Vol 5, No 3, pp. 1-22, 2009.

[21] M. Cobden, J. Black, N. Gibbins, L. Carr, and N. R. Shadbolt, “A Research Agenda for Linked Closed Dataset,” Workshop on consuming Linked Data, Vol 782, ISWC, 2011. [Vision paper]

[22] Extended Validation SSL Certificate: The Next Generation High Assurance SSL Certificate, http://www.evsslcertificate.com/ssl/description-ssl.html [retrieved: Mar, 2014].

[23] Server Certificate Installation Instructions, Microsoft Developer Network: http://msdn.microsoft.com/en-us/library/ms751408.aspx [retrieved: Feb, 2014].

[24] M. Hausenblas, R. M. Grossman, A.Harh, and P. Cudré-Mauroux, “Large-Scale Linked Data Processing – Cloud Computing to the Rescue,” CLOSER, pp. 246-251, Mar 2012.

[25] C. Bizer, A. Jentzsch, and R. Cyganiak, “State of the LOD Cloud,” 2004, http://lod-cloud.net/state/ [retrieved: Jan, 2014].

[26] P. Mika and G. Tummarello, “Web semantics in the clouds,” IEEE Intelligent Systems, Vol 2, pp. 82–87, 2008.

[27] R. L. Grossman et al., “An overview of the open science data cloud,” In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC ’10), ACM. pp. 377-384, 2010.

[28] Technical Working Group for Electronic Crime Scene Investigation, Electronic Crime Scene Investigation, “A Guide for first responders, United States Department of Justine,” 2001, pp.1-81.

[29] IIS Management Microsoft Management Console (MMC), Microsoft, http://support.microsoft.com/kb/892987 [retrieved: Feb, 2014].

[30] E. Casey, “Digital Evidence and Computer Crime - Forensic Science,” Computers and the Internet, 3rd Edition. Academic Press, pp. 1-807, 2011, ISBN: 978-0-12-374268-1.

[31] S.O. Ciardhuain, “An extended model of CC investigations,” International Journal of digital Evidence, Vol. 3, pp. 1-22, 2004.

[32] Y. Yusoff, R. Ismail, and Z. Hassan, “Common Phases of Computer Forensics Investigation Models,” International Journal of computer science and information technology (IJCSIT), Vol. 3, No 3, pp. 17-31, June 2011.

[33] M. D. Köhn, J. H. P. Eloff, and M. S. Olivier, "UML modeling of Digital Forensic Process Models (DFPMs)," in Proceedings of the ISSA 2008 Innovative Minds Conference, Johannesburg, South Africa, pp. 32-36, July 2008.

[34] Internet Information Services, Microsoft, http://www.iis.net/ [retrieved: Mar, 2014]

[35] Virtual Example: http://www.mydomain.com/resource/x.

[36] Virtual Example: http://www.mydomain.com/resource/x.rdf.

[37] Virtual Example: http://www.mydomain.com/resource/x.html.

[38] Virtual Example: http://www.mydomain.com.

[39] Cyber Forensics-Chain of Custody, Tamer Gayed, www.cyberforensics-coc.com [retrieved: Oct, 2013].

[40] Local Network IP, http://192.168.2.12 [retrieved: Jan, 2014].

[41] Friend of a Friend, http://xmlns.com/foaf/0.1/ [retrieved: Jan, 2014].

[42] RDF Schema 1.1, http://www.w3.org/TR/rdf-schema/ [retrieved: Feb, 2014].

[43] R. Davis , H. Shrobe, and P. Szolovits, “What is a knowledge representation?,” AI Magazine, Vol 14(1), pp.17-3, 1993.

[44] L. Sauermann, R. Cyganiak, (2008): Cool URIs for the Semantic Web. W3C Interest Group Note, http://www.w3.org/TR/cooluris/ [retrieved: Jan, 2014].

[45] D. Raggett, A. L. Hors, and Ian Jacobs “Html 4.01 specification - w3c recommendation,” http://www.w3.org/TR/html401/, 1999 [retrieved: Mar, 2014].

[46] Internet X.509 Public Key Infrastructure Certificate Management Protocols: https://tools.ietf.org/html/rfc2510 [retrieved: Mar, 2014].

[47] D. Richard, V. C. Hu, W. Timothy, and S. Chang, “Introduction to Public Key Technology and the Federal PKI Infrastructure,” National Institute of Standards and Technology (NIST), U.S. Government publication, 2011.

[48] T. Moses “Trust Management in the Public Key Infrastructure,” Entrust Technologies”, January 1999. [White paper]

[49] National Institute of Standards and Technology. Public Key Infrastructure Technology, ITL Bulletin :. http://www.nist.gov/itl/lab/bulletns/archives/july97bull.htm [retrieved: Apr, 2014]

[50] J. Davies, “Implementing SSL/TLS Using Cryptography and PKI,” Indianapolis, Indiana: Wiley Publishing Inc, 2011.

[51] E. Barker et al., “Recommendation for Key Management Part 3: Application-Specific Key Management Guidance,” NIST Special Publication 800-57, 2013 Edition.

[52] D. Kuhn, V. Hu, W. Polk, and S. Chang, “Introduction to Public Key Technology and the Federal PKI Infrastructure,” NIST Special Publication 800-32, 2013 Edition.