Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 5, May 2016, pg.184 – 190

Verifying Digital Artefacts on the Web

Dr. Shubhangi D.C[1], Soumya.S.Tole[2]

[1]Professor, Dept CSE Visvesvaraya Technological University, Kalaburagi, Karnataka, INDIA

[2]P.G Student, Dept CSE Visvesvaraya Technological University, Kalaburagi, Karnataka, INDIA

Abstract— The present Web has no broad instruments to make computerized relics such as datasets, code, messages, and pictures undeniable and lasting. For advanced antiquities that should be changeless, there no generally acknowledged strategy to authorize this permanence. These deficiencies have a genuine negative effect on the capacity to replicate the consequences of procedures that depend on Web assets, which thus intensely affects ranges, for example, science where reproducibility is imperative. To take care of this issue, we propose trusty URIs containing cryptographic hash values. We demonstrate how trusty URIs can be utilized for the check of advanced ancient rarities, in a way that is free of the serialization design on account of organized information documents, for example, nanopublications. We illustrate how the substance of these records get to be permanent, including conditions to outside computerized ancient rarities and in this manner expanding the reach of certainty to the whole reference tree. Our methodology adheres to the certain standards of the Web, specifically openness and decentralized design, and is completely perfect with existing principles and conventions. Assessment of our reference executions demonstrates that these configuration objectives are in fact fulfilled by our methodology, and that it stays useful notwithstanding for expansive records.

Keywords— "URI", "Hash values", "Nanopublications", "Secret key", "Cryptography"

---

1. INTRODUCTION

In numerous regions and specifically in science, reproducibility is critical. Certain, permanent, and changeless computerized ancient rarities are an essential fixing for making the aftereffects of computerized procedures reproducible, however, the present Web offers no normally acknowledged strategies to guarantee these properties. Tries for example, the Semantic Web to distribute complex information in a machine-interpretable way bother this issue, as computerized calculations working on extensive measures of information can be required to be considerably more helpless than people to controlled or tainted substance. Without fitting counter-measures, malevolent on-screen characters can damage or trap such calculations by including only a few precisely controlled things to huge arrangements of data information. To take care of this issue, we propose

a way to deal with make things on the (Semantic) Web certain, changeless, what's more, perpetual. This methodology incorporates cryptographic hash values in Uniform Resource Identifiers (URIs) and holds fast to the center standards of the Web, to be specific openness what's more, decentralized design. This article is an broadened and reconsidered rendition of a paper.

2. Related Work

To make digital resources on the web verifiable, immutable, and permanent, propose a technique to include cryptographic hash values in URIs[1].call them trusty URIs and they show how they can be used for approaches like nanopublications to make not only specific resources but their entire reference trees verifiable. Digital artefacts can be identified not only on the byte level but on more abstract levels such as RDF graphs, which means that resources keep their hash values even when presented in a different format. approach sticks to the core principles of the web, namely openness and decentralized architecture, is fully compatible with existing standards and protocols, and can therefore be used right away. Evaluation of reference implementations shows that these desired properties are indeed accomplished by its approach, and that it remains practical even for very large file. As the amount of scholarly communication increases, it is increasingly difficult for specific core scientific statements to be found, connected and curated. Additionally, the redundancy of these statements in multiple for a makes it difficult to determine attribution, quality, and provenance. To tackle these challenges, the Concept Web Alliance has promoted the notion of nanopublications (core scientific statements with associated context)[2]. In this document, a model of nanopublications along with a Named Graph/RDF serialization of the model. Importantly, the serialization is defined completely using already existing community developed technologies. Finally, the importance of aggregating nanopublications and the role that the Concept Wiki plays in facilitating it.

MetaLex Document Server (MDS)[3], an ongoing project to improve access to legal sources (regulations, court rulings) by means of a generic legal XML syntax (CEN MetaLex) and Linked Data. The MDS defines a generic conversion mechanism from legacy legal XML syntaxes to CEN MetaLex, RDF and Pajek network files, and discloses content by means of HTTP-based content negotiation, a SPARQL endpoint and a basic search interface. MDS combines a transparent (versioned) and opaque (content-based) naming scheme for URIs of parts of legal texts, allowing for tracking of version information at the URI-level, as well as reverse engineering of versioned metadata from sources that provide only partial information, such as many web-based legal content services. The MDS hosts all 28k national regulations of the Netherlands available since May 2011, comprising some 100M triples. An essential aspect of science is a community of scholars cooperating and competing in the pursuit of common goals. A critical component of this community is the common language of and the universal standards for scholarly citation, credit attribution, and the location and retrieval of articles and books. In a similar universal standard for citing quantitative data that retains the advantages of print citations, adds other components made possible by, and needed due to, the digital form and systematic nature of quantitative data sets, and is consistent with most existing subfield-specific approaches[4].Although the digital library field includes numerous creative ideas, limit ourselves to only those elements that appear ready for easy practical use by scientists, journal editors, publishers, librarians, and archivists. The ability to calculate hash values is fundamental for using cryptographic tools, such as digital signatures, with RDF data. Without hashing it is difficult to implement tamper-resistant attribution or provenance tracking, both important for establishing trust with open data. A novel hash function for RDF graphs[5], which does not require altering the contents of the graph, does not need to record additional information, and does not depend on a concrete RDF syntax. We are also presenting a solution to the deterministic blank node labeling problem.

In naming things with hashes[8],defines set of ways for identifying a thing using the output from a hash function, it standardize current uses of hash outputs in URLs and to support new information-centric applications and other uses of hash outputs in protocols. In incremental cryptography: the case of hashing and signing[9] initiates the investigation of new efficiency for cryptographic transformations, once applied the transformation to some document M,the time to update the result on notification of M should be proportional to the amount of modification done to M.

Security considerations for incremental hash functions based on pair block chaining[11],where they showed how collisions can be obtained in such incremental hash functions, more caution is taken into design process.

Signing RDF Graphs[7],assuming P<GI<NP creation and verification of digital signature of an arbitrary RDF graph cannot be done in polynomial time.an arbitrary RDF graph can be nondeterministically pre-canonicalized into graph before signing. the techniques are the key enables for the use of digital signature technology in semantic web. Computing the digest of an RDF graph[10],it is used to assign unique content-dependent identifires and for use in digital signatures allowing a recipient to verify that RDF was generated by algorithms. algorithm allows for incremental updates to the graph, so the time to recomputed the digest to account for additional statements is proportional to the number of new statements being added.

3. System Architecture

It starts with the registration process, data owner to upload the data and the end user to extract the data then login**.** The data owner uploads their data in the Web server. For the security purpose the data owner encrypts the data file and then store in the Web. The Data owner can have capable of manipulating the encrypted data file. The data owner will send Meta data to Audit Web. In audit Web raw or metadata information is available for auditing and data integrity checking purpose. Data owner will create an end user and the data owner can set the access permission (read or write) to user. Data Auditing and Verification**,** the data owner can also audit the data integrity in the corresponding Web for verifying whether the data is safe or not using digital sign and web URL. If the data is not safe then he will delete the data and re upload the data to the corresponding Web server.

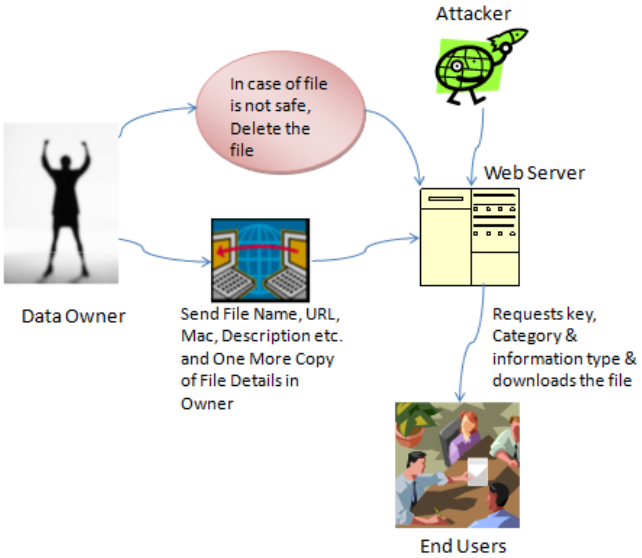

Fig 1: Architecture of Digital Artefacts

Web server is responsible for data storage and file authorization for an end user. The data file will be stored with their tags such as file name, secret key, digital sign, and owner name. The data file will be sending based on the privileges. If the privilege is correct then the data will be sent to the corresponding user and also will check the file name, end user name and secret key. If all are true then it will send to the corresponding user or he will be captured as attacker. The Web server can also act as attacker to modify the data which will be auditing by the audit Web.

Data consumer(end user) is nothing but the end user who will request and gets file contents response from the corresponding Web servers. If the file name and secret key, access permission is correct then the end is getting the file response from the Web or else he will be considered as an attacker and also he will be blocked in corresponding Web. If he wants to access the file after blocking he wants to UN block from the Web.

Attacker is one who is integrating the Web file by adding malicious data to the corresponding Web. They may be within a Web or from outside the Web. If attacker is from inside the Web then those attackers are called as internal attackers. If the attacker is from outside the Web then those attackers are called as external attackers. Any can attack for the information so based on URL we will get to know who has hacked, we can maintain the values by secret keys, we can get to know if the attacker has made changes in the file by Url, This is improved. based upon the data downloaded we can get to know which data is good and has more particular information.

4. Methodology

Trusty URI antiques are obvious as in a recovered antique for a given URI can be checked to contain the substance the URI should speak to. It can be distinguished if the ancient rarity got tainted or controlled in transit, expecting that the trusty URI for the required antique is known, e.g. since another ancient rarity contains it as a connection. (Obviously, some individual can give you a controlled antique with an alternate trusty URI.) It specifically takes after that trusty URI ancient rarities are permanent, as any adjustment in the substance additionally changes its URI, consequently making it another antiquity. Once more, you jar of course change your curio and its URI and case that it has dependably been similar to this. You can escape with that in the event that the trusty URI has not yet been grabbed by third parties, i.e. connected by different assets. When this is the case, it can't be changed any longer, since all these connections will even now indicate the old trusty URI and everyone will see that the new antique is an alternate one.

Approach of Trusty URI General structure: Every character is a standard ASCII letter (A-Z and a-z),a digit(0-9),(-) hyphen,(_) underscore called Base 64 character representing in order numbers 0 to 63.every trusty URI ends with atleast 25 Base 64 character, sequence of characters following the last non-Base 64 character is an artefact code. First two characters of artefact is called module identifier where first character represents type of content or the module to choose and second character represents version identifier. sequence of character following the module identifier is an data part, which is identical or which contains hash part. current module generates URIs with exactly 45 Base character. data part has the main content in hash, but can also contains parameters and subtype.

RSA Algorithm:

RSA algorithm are used by modern computers to encrypt and decrypt messages(data).Its an asymmetric cryptographic algorithm. Asymmetric has two different keys.It is also called as public key cryptography as one of them can be given to everyone.

RSA generates public and private keys as a pair of values ie, Let public key be ku=(-,-) and private key be kr=(-,-)

Now find the key value pairs of ku and kr?

Step 1.To choose 2 large prime numbers a & b. usually these are of order 10^100 which makes RSA robust.

Step 2.This is the product of these 2 large prime numbers called as s. s=a*b

Step 3.Choose the variable z such that r is co-prime to s. r=(a-1) * (b-1)

Step 4.Choose the variable c such that e is also prime number and co-prime number. 1<c<z

Step 5.Choose the variable t such that e is also a prime number. (t*c) mod r =1

These are the things by which we can fill private and public keys,so s gets place in both the places and choose c in ku and t in kr(even can keep as vice-versa).

Ku=(c,s) and kr=(t,s)

If we have a message d,we can encrypt it as d^c mod s and called as f.Encrypt f=d^c mod s, where f is the encrypted message which transmits.

In the decrypting end, can Decrypt as f^t mod s=d where m is the original message.

Example: 1. a=3, b=11

2. s=3 * 11=33

3. r= (3-1) * (11-1)=20

4. c=7

5. 3 * 7 mod 20 =1,t=3

Ku=(7,33) and kr=(3,33)

If d=2 then, Encrypt, 2^7 mod 33=29

Decrypt , 29^3 mod 33=2 so,d=2(original)

5. Results and Discussion

To test our methodology and to assess its executions, we first took an accumulation of 156,026 nanopublications in TriG organization that we had delivered in past work. We changed these nanopublications into the organizations N-Quads and TriX utilizing existing off-the shelf converters. At that point, we changed these into trusty URI nanopublications utilizing the Java usage. To have the capacity to check not just positive cases (where checking succeeds) additionally negative ones (where checking comes up short), we made duplicates of the subsequent records where we changed an irregular single byte in each of them (just considering letters and numbers, and never supplanting a capitalized letter by its lower-case rendition or the other way around, as a few catchphrases are not case-delicate).

WelCome To Data Owner

SN	File Name	Digital Artifacts	Secret Key	Rank	Date & Time
1	Morning Wishes	-25c65c11a194b4f2cdaa40106a9fe76f5027f8f7	[B@742dce	1	06/04/2016 10:15:01
2	generic program	-25c65c11a194b4f2cdaa40106a9fe76f5027f8f7	[B@19726d9	0	06/04/2016 18:20:16
3	conficpara	-204585521f797f8b3d79e3671d5656d0ce875ae5	[B@c31a61	0	06/04/2016 18:22:41

Fig 2.Files which created Digital Artefacts and Secret key with ranking of file(top n count)

In fig 2,the creation of the digital artefacts and the secret keys are show which has been converted while the data owner will be uploading the file, so that the user can extract the specific files which he will be in need of and without wasting the time just by knowing the secret key of the particular file. It has been shown the ranking of files to make the use more easy to fetch the files which has highest ranking.

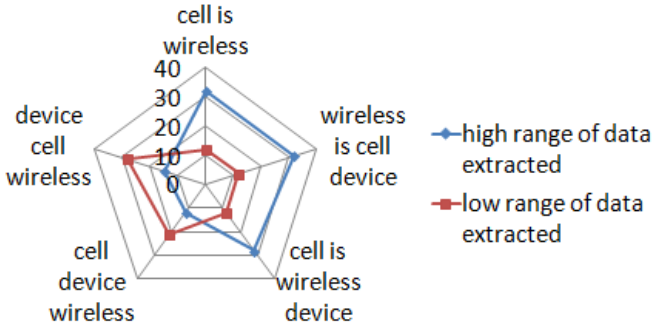

Fig 3.Radar Graph which represent the range of data which is original

In fig 3,Graph representation shows the range of the correct or most similar data and just related data so that the user can easily search the correct data for which he was looking for, as the blue range shows the highly similar data which has been viewed or downloaded many number of times and the red range shows the low level range of data.

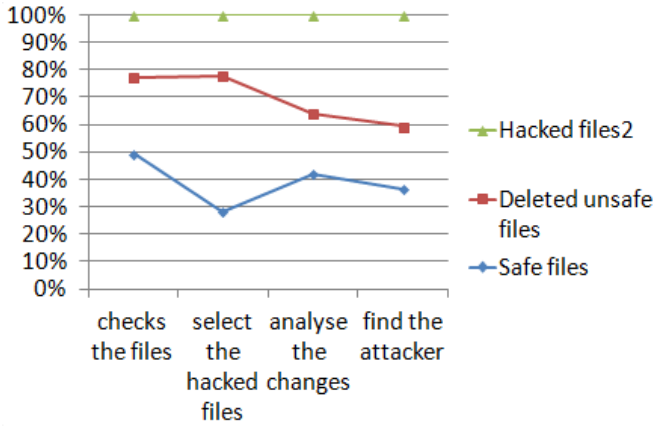

Fig 4.Line Graph which shows the safe and unsafe data

In fig 4,Graph representation shows the safetyness of the files, as the data is uploaded by the data owner, the attacker may hack the file, so it checks the files whether its safe or not, then select the file if its hacked and then find the attacker and the changes done by the attacker and delete the file and reupload the file and maintain safe. the percentage shows the range of safety of the data.

6. Conclusion and future work

We have exhibited a proposition for unambiguous URI references to make computerized relics on the (Semantic) Web evident, unchanging, and perpetual. On the off chance that embraced, it could considerably affect the structure and working of the Web, could enhance the proficiency furthermore, unwavering quality of instruments utilizing Web assets, and could turned into a vital specialized column for the Semantic Web, specifically for investigative information, where provenance furthermore, evidence are critical. Exploratory information investigations, for instance, may be directed later on in a completely reproducible way inside "information ventures" closely resembling today's product ventures. The conditions in the structure of datasets could be naturally gotten from the Web, like what Apache Maven accomplishes for programming ventures, yet decentralized and verifi- capable.

We have begun to build up a decentralized nanopublication server system. Nanopublications are circulated and

recreated among such servers and recognized by trusty URIs, in this manner guaranteeing that these relics stay accessible regardless of the possibility that individual servers are ended. The current system comprises of four servers in four distinct nations facilitating 5 million nanopublications. What's more, we are chipping away at the idea of nanopublication files that take into consideration the definition and recognizable proof of little or huge arrangements of nanopublications. Such files are nanopublications themselves and, obviously, are distinguished by trusty URIs.data

Acknowledgement

The authors would like to thank a great support of special Officer Dr.Baswaraj.Gadge, Head of the Department Dr. Shubhangi.D.C, Department Professors and lastly to college for constant inspiration and suggestions.

References

[1] T. Kuhn and M. Dumontier, "Trusty URIs: Verifiable, immutable, and permanent digital artefacts for linked data," in Proceedings of the 11th Extended Semantic Web Conference (ESWC 2014). Springer,2014, pp. 395–410.
[2] P. Groth, A. Gibson, and J. Velterop, "The anatomy of a nanopublication,"Information Services and Use, vol. 30, no. 1, pp. 51–56,2010.
[3] R. Hoekstra, "The MetaLex document server," in The Semantic Web— ISWC 2011. Springer, 2011, pp. 128–143.
[4] M. Altman and G. King, "A proposed standard for the scholarly citation of quantitative data," D-lib Magazine, vol. 13, no. 3, p. 5,2007.
[5] E. H¨ofig and I. Schieferdecker, "Hashing of rdf graphs and a solution to the blank node problem," in 10th InternationalWorkshop on Uncertainty Reasoning for the Semantic Web (URSW 2014), 2014,p. 55.
[6] M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon, "XML signature syntax and processing," W3C, Recommendation, June 2008. [Online]. Available: http://www.w3.org/TR/xmldsig-core/
[7] J. Carroll, "Signing RDF graphs," in The Semantic Web — ISWC 2003. Springer, 2003, pp. 369–384.
[8] S. Farrell, D. Kutscher, C. Dannewitz, B. Ohlman, A. Keranen, and P. Hallam-Baker, "Naming things with hashes," Internet Engineering Task Force (IETF), Standards Track RFC 6920, April 2013.
[9] M. Bellare, O. Goldreich, and S. Goldwasser, "Incremental cryptography: The case of hashing and signing," in Advances in Cryptology — CRYPTO'94. Springer, 1994, pp. 216–233.
[10] C. Sayers and A. Karp, "Computing the digest of an RDF graph,"Mobile and Media Systems Laboratory, HP Laboratories, PaloAlto, USA, Tech. Rep. HPL-2003-235(R.1), 2004.
[11] R. Phan and D. Wagner, "Security considerations for incremental hash functions based on pair block chaining," Computers & Security, vol. 25, no. 2, pp. 131–136, 2006.
[12] H. Van de Sompel, R. Sanderson, H. Shankar, and M. Klein,"Persistent identifiers for scholarly assets and the web: The need for an unambiguous mapping," International Journal of Digital Curation, vol. 9, no. 1,pp. 331–342, 2014**.**
[13] J. McCusker, T. Lebo, C. Chang, D. McGuinness, and P. da Silva,"Parallel identities for managing open government data," IEEE Intelligent Systems, vol. 27, no. 3, p. 55, 2012.
[14] J. McCusker, T. Lebo, A. Graves, D. Difranzo, P. Pinheiro, and D. McGuinness, "Functional requirements for information resource provenance on the web," in Provenance and Annotation of Data and Processes. Springer, 2012, pp. 52–66.
[15] R. Gentleman, "Reproducible research: A bioinformatics case study," Statistical applications in genetics and molecular biology,vol. 4, no. 1, 2005.

