Trusty URIs: Verifiable, Immutable, and Permanent Digital Artifacts for Linked Data

Tobias Kuhn[1] and Michel Dumontier[2]

[1] Department of Humanities, Social and Political Sciences, ETH Zurich, Switzerland
[2] Stanford Center for Biomedical Informatics Research, Stanford University, USA
tokuhn@ethz.ch, michel.dumontier@stanford.edu

Abstract. To make digital resources on the web verifiable, immutable, and permanent, we propose a technique to include cryptographic hash values in URIs. We call them *trusty URIs* and we show how they can be used for approaches like nanopublications to make not only specific resources but their entire reference trees verifiable. Digital artifacts 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. Our 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 our reference implementations shows that these desired properties are indeed accomplished by our approach, and that it remains practical even for very large files.

Keywords: #eswc2014Kuhn.

1 Introduction

The vision of the semantic web is to make the content of the web machine-interpretable, allowing, among other things, for automated aggregation and sophisticated search procedures over large amounts of linked data. As even human users are sometimes easy to trick by spam and fraudulent content that can be found on the web, we should be even more concerned in the case of automated algorithms that autonomously analyze semantic web content. Without appropriate counter-measures, malicious actors can sabotage or manipulate such algorithms by adding just a few carefully manipulated items to large sets of input data. To solve this problem, we propose an approach to make items on the (semantic) web verifiable, immutable, and permanent. This approach includes cryptographic hash values in Uniform Resource Identifiers (URIs) and adheres to the core principles of the web, namely openness and decentralized architecture.

A cryptographic hash value (sometimes called *cryptographic digest*) is a short random-looking sequence of bytes (or, equivalently, bits) that are calculated in a complicated yet perfectly predictable manner from a digital artifact such as a file. The same input always leads to exactly the same hash value, whereas just a minimally modified input leads to a completely different value. While there is an

V. Presutti et al. (Eds.): ESWC 2014, LNCS 8465, pp. 395–410, 2014.

infinity of possible inputs that lead to a specific given hash value, it is impossible in practice (for strong state-of-the-art hash algorithms) to reconstruct *any* of the possible inputs just from the hash value. This means that if you are given some input and a matching hash value, you can be sure that the hash value was obtained from exactly that input. On this basis, our proposed approach boils down to the idea that references can be made completely unambiguous and verifiable if they contain a hash value of the referenced digital artifact. Our method does not apply to all URIs, of course, but only to those that are meant to represent a specific and immutable digital artifact.

Let us have a look at a concrete example: Nanopublications have been proposed as a new way of scientific publishing [10]. The underlying idea is that scientific results should be published not just as narrative articles but in terms of minimal pieces of computer-interpretable results in a formal semantic notation (i.e. RDF). Nanopublications can cite other nanopublications via their URIs, thereby creating complex citation networks. Published nanopublications are supposed to be immutable, but the current web has no mechanism to enforce this: It is well-known that even artifacts that are supposed to be immutable tend to change over time, while often keeping the same URI reference. For approaches like nanopublications, however, it is important to specify exactly what version of what resource they are based on, and nobody should be given the opportunity to silently modify his or her already published contributions.

With the approach outlined below, nanopublications can be identified with *trusty URIs* that include cryptographic hash values calculated on the RDF content. For example, let us assume that you have a nanopublication with identifier I_1 that cites another nanopublication with identifier I_2. If you want to find the content of I_2, you can simply search for it on the web, not worrying whether the source is trustworthy or not, and once you have found an artifact that claims to be I_2, you only have to check whether the hash value actually matches the content. If it does, you got the right nanopublication, and if not you have to keep searching (this can of course be automated). A trusty URI like I_1 does not only allow you to retrieve its nanopublication in a verifiable way, but in the next step also all nanopublications it cites (such as I_2) and all nanopublications they cite and so on. Any trusty URI in a way "contains" the complete backwards history. In this sense, the "range of verifiability" of a resource with a trusty URI is not just the resource itself, but the complete reference tree obtained by recursively following all contained trusty URIs. This is illustrated in Figure 1.

Another important property of nanopublications is that they are self-contained in the sense that they consist not only of the actual scientific assertions but also of their provenance information and meta-data. This means that nanopublications contain self-references in the form of their own identifying URIs. The calculation of a trusty URI must therefore allow for the resulting URI to be part of the digital artifact it is calculated on (this might sound impossible at first, but we show below how it can be achieved). This leads us to the formulation of the following requirements for our approach: