Owner-Centric Networking (OCN): Toward A Data Pollution-Free Internet.

Claude Castelluccia
INRIA Rhone-Alpes
ccastel@inrialpes.fr

Mohamed Ali Kaafar
INRIA Rhone-Alpes
kaafar@inrialpes.fr

Abstract

In this paper, we propose OCN (Owner-Centric Networking), a new Internet architecture that provides individuals control over their contents. We argue that this architecture would considerably improve privacy on the Internet by limiting data pollution.

1 Introduction

In his essay [3], Bruce Schneier famously said "*data is the pollution of the Information Age*". Internet contents (documents, emails, chats, images, videos, etc.) that are posted on the Internet are often disseminated and replicated on different peers or servers, generating what we refer to as *Data Pollution*. As a result, users loose the control and ownership of their contents as soon as they release them.

The crux of the problem is that the Internet simply never forgets and information that are posted linger virtually forever. Furthermore, the current Internet, as it has been designed so far, provides no limit to data diffusion, neither any right to an individual to modify or even remove what he wrote on a forum chat, or on a famous social network's walls.

Similarly to, for example, commercial flyers that gets lost and pollutes the environment, non-controlled Internet contents pollutes the Internet. This data pollution creates many privacy concerns since these lost contents can be used to collect information about users without their consent. For example, there have been several recent cases of employers using social networks (such as facebook) to spy on their employees

As argued by Lessig in [2], "*this lost control over our data threatens our privacy and endanger our intellectual properties. With privacy and copyright, there is a bit of our data that we lost control over. In the case of copyright, it is the data constituting a copy of our copyrighted work; in the case of privacy, it is the data representing some fact about us. In both cases, the Internet has produced this loss of control: with copyright, because the technology enables perfect and free copies of content; with privacy because the technology enables perpetual and cheap monitoring of behavior*".

Unfortunately, Privacy does not occur naturally online, it must be deliberately architected. An architecture for the future Internet should then consider data pollution as one of the most relevant problems to solve.

In this paper, we argue that the Future Internet should provide data control and anti data-pollution mechanisms. More specifically, the Future Internet should support the following features:

- Individuals should have control over their data. In other words, users should be able to retrieve their previously posted contents in order to withdraw or modify them. Put simply, the Internet of tomorrow should enforce the "right to forget", which is a constitutional law in several countries.
- Individuals should be able to claim the ownership of their contents.
- The Future Internet should limit data collection.
- The Future Internet should limit data pollution, by for example, limiting uncontrolled content replication. Contents should only be 'physically' present at a few locations in the Internet, and should not be easily copied and duplicated.

Most of future Internet architecture proposals (Trilogy [6], CCN [1], etc.) seem to have ignored the issues of data control and pollution so far. For example, the content-centric networking (CCN) architecture [1], which proposes to shift the focus from transmitting data by geographic location to disseminating it via named content, actually increases data pollution. In CCN, content are not only hosted by servers but are also diffused from where they are created to where the consumers are [4]. As a result, individuals

completely lose control over their contents as they get distributed (lost) on the Internet without their consent or even knowledge. That said, we believe that content-centric networking is still a very attractive solution, if it evolves towards an owner centric architecture (OCN) that considers contents ownership as bedrock.

Note that some Internet services already provide some OCN-like mechanisms. For example, the Skype chat tool allows a user to retroactively modify its part of a chat. The view of the chat history on both users synchronizes with the new version of the conversation when both users are online. This is a very good example of the type of control that a user should have on its data. We propose to extend this concept to all services and integrate it into the Internet architecture. For example, in an OCN architecture, emails that are exchanged between users would not contain the actual content of the message, but a link to a server(s) that hosts the contents. The recipient of an email that does get a physical copy of the content but gets a link to a server where he can read it. As a result, the sender keeps the control over its content. For example, if, by mistake, he sends a confidential information to an wrong destination, he can, at anytime and hopefully before the recipient reads it, remove it. Furthermore, if the message needs to be modified, this can be done without sending additional emails.

2 Toward an OCN Architecture

In the following, we describe our proposal of an owner-centric networking, that considers content's ownership as a way to design an Internet architecture with the features we mentioned previously.

2.1 Architecture Overview

The proposed architecture is displayed on Figure 1.

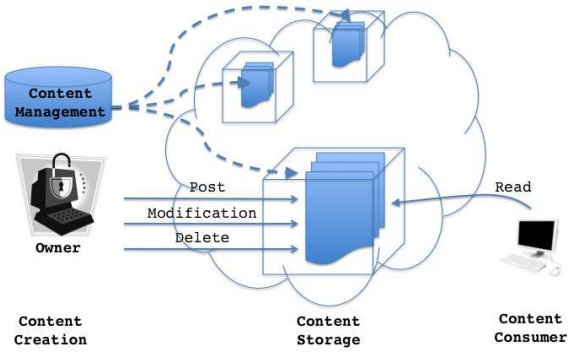

Figure 1. A General Overview of the OCN architecture

It is composed of three main modules. First, the content creation and management module is in charge of managing and controlling all the contents of a user. Second, the content storage module is the component that diffuses the content on the Internet, i.e. the module that defines its actual physical locations. A content storage can be centralized or distributed, and can involve servers, as well as peers participating in P2P networks. Finally, the content exchange and access module defines how contents are exchanged and accessed.

2.2 Content Management

In OCN, a user keeps track of all the contents that he publishes on the Internet. In practice, a user actually stores a list of links of the published contents. Such list can be very large, and more efficient data structures such as chained lists, hash tables, databases or self-balancing binary search trees can be used.

Clearly, the choice of a technique to maintain the data structures needed for keeping the control on published contents, has to consider the performance in terms of content lookups and storage. The most suitable way to implement the content management module is out the scope of this paper. However, in the following, we propose an example of a possible data structure.

In our proposal, each content is indexed by a record that is composed of several attributes. Each of these records are linked using a chained list. A record contains five main attributes as shown in figure 2.

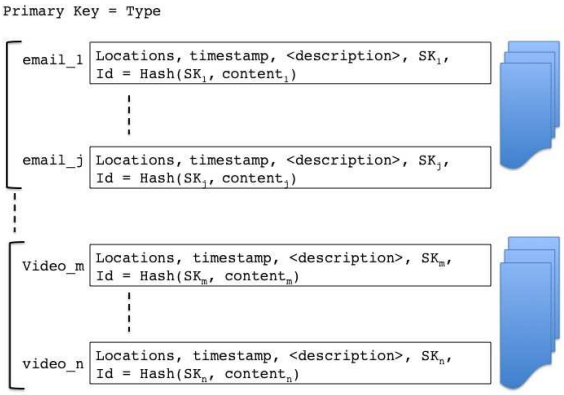

Figure 2. Content Management module format

The first attribute contains the physical locations of the content. The second attribute is a timestamp that attests when the data was created, and optionally keeps track of dates when the content was modified. Basic operations are assumed to be synchronous over all the physical locations.

However, if an owner aims to keep different versions, on several locations, a new content descriptor is added as a new record to the content management module. The third attribute describes the content. It is introduced in order to allow for an easy look up of the contents. Since this description is type-dependant, it is up to the developers to specify which elements the description attribute should contain. As an example, we can specify the destination, subject, and dates for an email-type record. The fourth attribute contains the content's public key and is used to enforce access. In fact for each published content, a user generates a public/private key pair (PK/SK) and embeds the public key inside the content. As we will see later, these keys are used by the user to prove the ownership of its contents. Finally, the last argument is a hashed value of the published content. It is used to verify the integrity of the content.

It is important to mention that the data structure containing the content information is private, and as such, needs to be kept under the owner's control. We can ensure reliability by duplicating the owner's data structure, or by considering a distributed data structure. However, in this paper, we will assume a centralized data structure.

2.3 Content Storage and Control

Content Storage

As in CCN, contents are diffused to where the consumers are. However, as opposed to CCN, they are stored/cached at places *controlled* by owners and not by the network, such that they can easily be retrieved. These places should, of course, be defined according to the users' requests and needs, but under the owner's control. All contents belonging to an individual are somehow linked together such that their owner can easily control them. Conceptually, all contents of an individual are linked with a rope. The data are distributed and stored on the Internet, but at any time, a user can pull on her rope in order to retrieve them. Each time, a new content is added, a new element (a knot) is added to the rope.

Content Control

Once a content has been published, a user might want to modify or withdraw it, i.e. to *control* it. In order to allow these operations, the content creator and content storage must interact according to a protocol.

As depicted in figure 1, the owner's actions on her contents can be summarized by the three following basic operations:

- *Content Publication (post)*: When storing her content on a server, the owner sends along with the content itself, the content identity (ID), her (temporary) public key and the signature of the content with the corresponding private key. Such a signature is intended to be used when the owner aims to access her content as a proof of ownership. This operation ends by adding an entry to the content management module, with the corresponding attributes.

- *Content Modification*: A user can modify its contents on a server. Upon request of a modification message on a content, the server storing the content must verify the ownership of the requested content ID_j. To do so, the server asks the user to sign the message ($nonce$,ID_i), where $nonce$ is a random value generated by the server. By verifying the signature with the public key that is embedded in the content, the server can verify that the user knows the associated secret key and is therefore the owner of the content. Note that, since the public/private key pair is temporary, this mechanism does not authenticate the user and therefore does not reveal the user's identity. It only proves that the user that wants to modify the content ID_j is actually its owner.

- *Content Deletion*: Content removal is similar to the modification procedure with the difference that once the owner succeeds in removing the content from its physical location, the corresponding entry is also removed from the content management module.

2.4 Content Exchange and Access

In the previous section, we explained how to organize the content control, but did not specify how contents should be distributed nor how users would access them. Recall that in order to reduce data-pollution, content access in the future Internet should limit unnecessary data collection, and more importantly unnecessary content diffusion and replication.

We argue that contents should be exchanged via links. Access should only be allowed via links to the content, and not the content itself. Since Internet connectivity will be ubiquitous in the near future, we believe that this approach is reasonable.

As an example, we consider the emailing services. Using an OCN-like email, a user won't directly send a full text email to her corresponding. rather their favorite email server (wherever we call it a pop or an imap server) will send a reformatted email notification to the destinations, including a link the the text or documents that would have been attached. Similarly, a web page would be composed of the set of links, each pointing to one different components.

It is also important to notice that we are not arguing that P2P architectures, as they are being developed today are to

be revisited. We argue that rather than exchanging files between peers, nodes should exchange links on the files (we should peer the links, not the files).

3 Discussion and Conclusion

We believe that our OCN architecture, described above, would considerably reduce pollution on the Internet and improves privacy by giving back to users control over their data.

This project is still in its preliminary phase and we are aware that there are still many open issues to be solved. In particular, our scheme does not solve all privacy issues. For example, it does not prevent a service provider (such as search engines) from collecting information about its users for profiling or business purposes. These issues can be solved by integrating into this new architecture concepts borrowed from anonymizing networks, such as onion routing and hidden services [7]. These mechanisms aim at hiding the source of packets and, therefore, make data collection difficult and irrelevant. These privacy issues can also mitigated by encrypting contents as often as possible. The idea of encrypting the contents instead of encrypting the communication channels seeems very interesting [4]. We also believe that opportunistic encryption proposols than encrypt traffic transparently to the communicating nodes should also be considered [9, 8].

Furthermore, our scheme is only an architecture proposal, and as such, does not prevent malicious users from violating it, by, for instance, copying contents instead of just accessing them remotely. However, we believe that this last issue can be mitigated with the help of security protocols (such as SPKI certificates [5]) and enforced by laws. Note that the situation is very similar than with environmental pollution. Technology can help to reduce environmental pollution but cannot enforce it. For example, no technology can prevent a boat from emptying its fuel tank in the ocean and pollute the water. Only legislation and law enforcement can help here.

In conclusion, we believe that the new paradigms developed in this project deserve more attention and debates within the community. We advocate that Future Internet architecture proposals should consider data pollution from the beginning. Privacy on the Internet is probably as important as security, and should be considered equally.

We are aware, however, that the OCN concept must be accepted by the applications and users to become a successor to today's architecture. Applications can benefit from the concept besides the gain generated by a higher users satisfaction and by limiting copyright losses. We introduce mechanisms to implement ownership awareness for data publication.

References

[1] V. Jacobson, M. Mosko, D. Smetters, and J. J. Garcia-Luna-Aceves, "Content-centric networking", Whitepaper, 2007.

[2] Lawrence Lessig, "Code2.0", `http://pdf.codev2.cc/Lessig-Codev2.pdf`, 2006.

[3] `http://www.schneier.com/essay-253.html`, January/February 2009

[4] `http://mags.acm.org/queue/200901/?pg=3D8` , January 2009.

[5] C. Ellison and al., "SPKI Certificate Theory". `http://www.rfc-editor.org/rfc/rfc2693.txt`

[6] `http://trilogy-project.org/`

[7] R. Dingledine,N. Mathewson and P. Syverson, "Tor: the second-generation onion router", Proceedings of the 13th conference on USENIX Security Symposium, 2004, San Diego, USA.

[8] C. Castelluccia, G.Montenegro, J.Laganier and C. Neuman. "Hindering Eavesdropping via IPv6 Opportunistic Encryption". European Symposium On Research in Computer Security (ESORICS 2004), Sep 2004.

[9] "FreeSwan project". `http://www.freeswan.org/history.html`