Master Semester Project

Rémi Coudert

February 7, 2020

1 Introduction

In the current era of digitization, journalism and article publishing in general underwent significant transformations. In the past, a news organization published journals exclusively on paper, using mechanical printers and, later, electronic ones. This meant once a journal was published, it was virtually impossible to edit or remove it.

Most if not all newspapers now support online publishing. The main difference between the two means of publishing lies in news organizations now having total control over their servers and thus over readers' access. In particular, once an online journal has been published, news organization can still edit it or even remove particular articles as opposed to physical journals. Hence, a large number of versions of the same article can exist and typically, only the last version is accessible [1].

This makes it harder for journalists referencing news organizations' articles to keep track of the exact version they can use. The possibility to edit or delete an article combined with articles being numerical also prevents readers to fully trust references.

With the advance of technology and the Internet, the scale and speed of distribution of news completely changed. The number of articles and publishers rose, and the means of access to information as well. In a world of "fake news" and government surveillance as highlighted by cases like Snowden's, the importance of critical and trusted investigative journalism and better technological uses in journalism is as big as ever [2].

This project aims at

1. Making referencing easier and long lasting for journalists citing published articles as well as establish trust of the reference for the reader

2. Protect journalist references from repudiation from news organizations

In the first part, we will define the problems in more detail. In the second part we will devise two specific scenarios and their solutions. In the third part we will discuss the implementation of these solutions. Last, we will discuss conclusions of the project.

2 Problem statements

2.1 Definitions

To avoid confusion due in part to the similarity of actors (e.g. journalists inside or outside of news organizations), we define the following :

- *A News Organization* is an entity publishing *articles* in a *journal*. We will refer to journalists specifically working for these organizations and writing these articles as *newspersons*.
- *A Journalist* is a person interested in writing a piece that references an article from a news organization.
- *A reference* is a citation of an article in a piece written by a journalist. We assume that it takes the form of a link to the article or a footnote citation. Referencing here is the act of using a reference in an article.
- *A reader* is a person interested in reading a journalist piece.

2.2 Easy and trusted referencing of articles

When journalists write a piece referencing an article from an organization, the actual workflow is to research for and write an article and then include references. A reference is the name of the article, its publication date, author, title and a link to it.

Online articles are uploaded on the internet and that makes it harder to keep track of versions and modifications. Readers need to trust that a citation written in a journalist text is indeed coming from the organization they claim it was from.

Thus we identify three problems:

1. Article referencing by journalist is not automated
2. Access to the original text can be impossible (deletion) or inaccurate (edition)
3. Readers have no easy way to check the veracity of citations from journalists

Hence, We want to define a system for efficient, permanent and trusted referencing of articles by journalists with support for versioning even after editing or deleting the original article. We aim to create a trust relation between journalists and readers with regards to journalists' referencing.

More specifically, we define the following goals:

First, we want to provide efficient referencing for journalists when writing pieces. Creating and using a reference in an article should be easy to do and not be a burden on the journalists.

Next, we want to provide permanent versioning of references. Once an article is published by a news organization, journalists should be able to reference this specific version even after edition or deletion of the original online article.

We also want to create trust for readers for journalists' referencing. A journalist should only be able to provide a legit reference to a reader and a reader should be able to easily check references used by journalists in their pieces. This means that journalists should not be able to reference a version of an article that never existed and if a reference is legit it should act as a proof that this specific version was indeed published by the news organizations referenced.

3 Our scheme

3.1 Preliminaries and notations

In the following we will use the following notations:

- sk and pk are shorthands for secret key and private key
- σ_X denotes the signature of object X
- $\mathsf{Sign}(X, sk)$ denotes a signature generation algorithm outputing the signature σ_X for object X with public key pk
- $\mathsf{Verif}(\sigma_X, Y, pk)$ denotes a signature verification algorithm outputing 1 on success, 0 on failure for the verification of signature σ_X against the object Y with public key pk
- $h(X)$ denotes a hash function
- each organization has a unique identifier denoted NOID (News Organization ID)

3.1.1 General scenario

We describe here the overall scenario on which we base the scheme protocol. We introduce three actors.

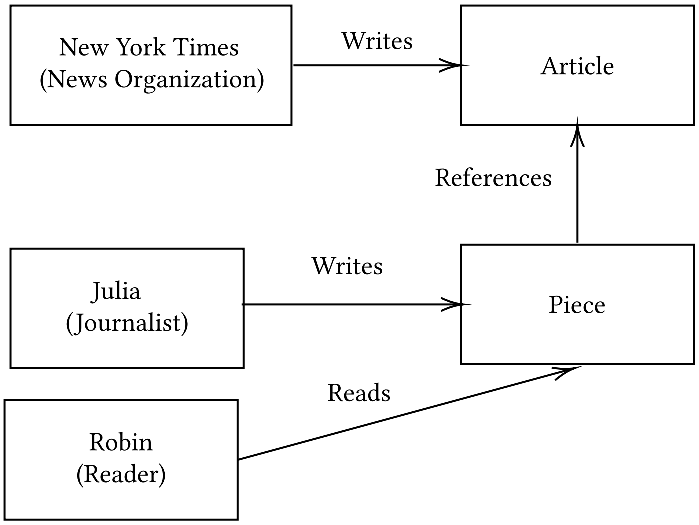

Figure 1: Overall view of scenario

See Figure 1.

- **News organizations (N.O.)**: News organizations' newspersons write articles and publish them online on the organization website. In the examples: The New York Times (NYT)
- **Journalists**: Journalists write their own pieces, based on news organizations' articles and reference their sources to readers. In the examples: Julia
- **Readers**: Readers want to read pieces written by journalists and know for sure that journalists use legit references from the version they claim the article to be. In the examples: Robin

3.2 Protocol

In the following, we assume the existence of a Public Key Infrastructure (denoted PKI) that supports two operations: register a new NOID and its associated public key, and retrieve a public key given a NOID.

3.2.1 Setup

The New York Times first generates its public and secret key pk and sk as well as its NOID and register the public key with the PKI.

This way, Julia and Robin will be able to retrieve the public key by requesting it from the PKI using the New York Times NOID. See Figure 2.

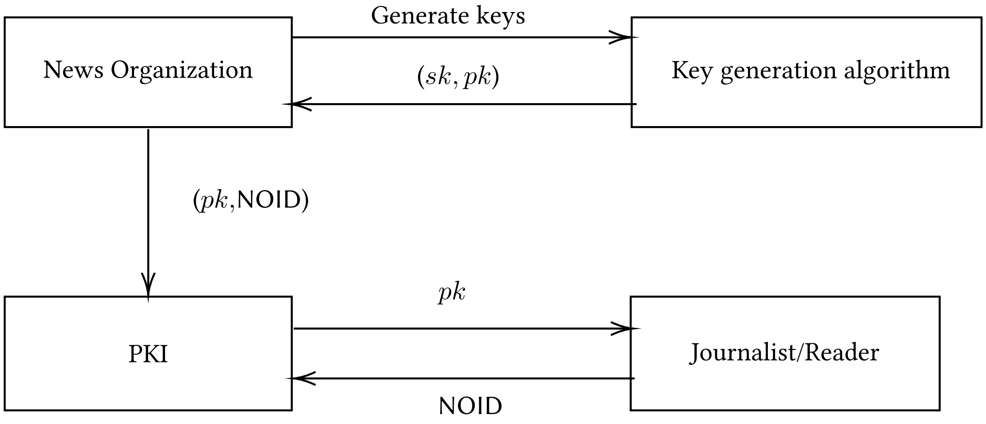

Figure 2: Setup

3.2.2 Records

An article is a piece of text written by a news organization. It has a title, a version, a text and is linked to a specific news organization.

From a specific article we define its corresponding header H and record R as follows:
H = {Title, Version, NOID, h(Text)}
R = {H, Text}.

3.2.3 Article Signing

When the New York Times is about to publish a new article, they create its corresponding header and record H and R and generate a signature $\sigma_R = \mathsf{Sign}(\mathsf{H}, sk)$ from the header using their secret key.

At publishing time online, the NYT attaches this signature to the webpage for everyone to download. See Figure 3.

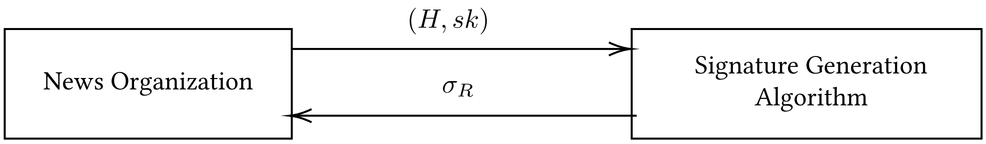

Figure 3: Generation of an article signature from a news organization

3.2.4 Piece Referencing

Julia as an independent journalist writes her own piece and needs to reference the New York Times' article.

To this end, she requests the New York Times' public key from the PKI using their public NOID. Next, she retrieves the record R and header H from the online article content and verifies the signature σ_R available on the webpage against R.

To reference the article, Julia includes in her piece the signature and the record with the link to the online article. See Figure 4.

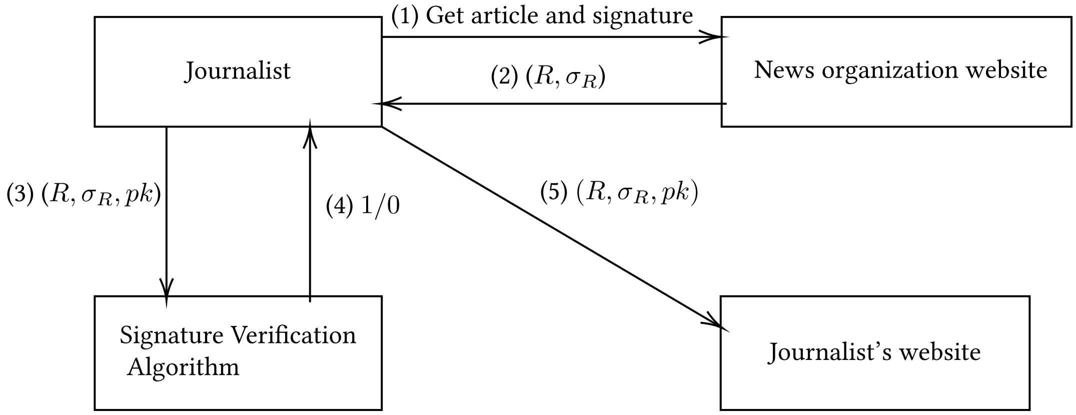

Figure 4: Article referencing by journalists, numbers indicate the order of operations

3.2.5 Piece Verification

When Robin consults Julia's piece, he can automatically retrieve the record and the signature corresponding to the NYT article. Then, he requests the corresponding public key from the PKI.

This way, Robin is able to verify that the reference is legit and that the article has been published by the New York Times with the corresponding text, title and version. See Figure 5.

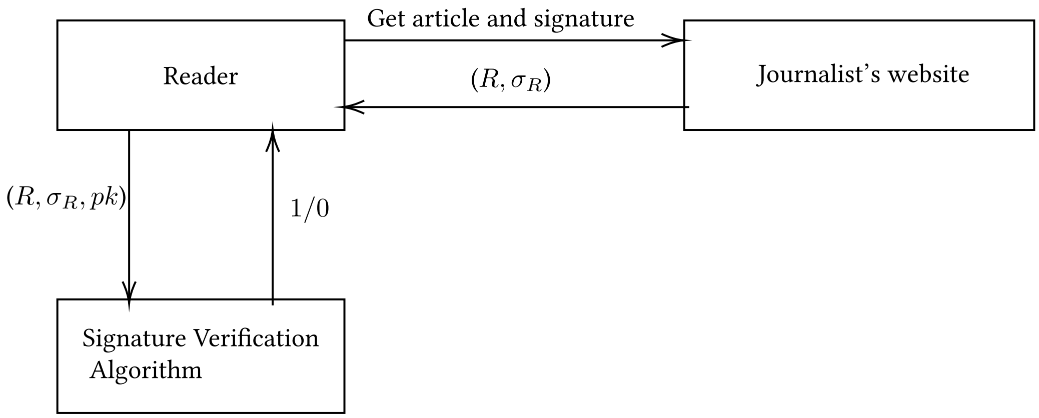

Figure 5: Reference verification

4 About protection against repudiation

The above system only partially protects against malicious news organizations. These organizations cannot deny having signed a published version. However, they can always opt not to sign, thereby making repudiation of those unsigned articles easy

One of the original goals of the project was to implement along with the trusted referencing, a way to protect journalists against repudiation, capture article version history, and force news organizations to update their articles.

Indeed, a news organization could choose to publish some version of an article to a part of the world, and a totally different one elsewhere while still respecting the protocol described. In addition, there is not clear way of knowing what changes happened between versions.

To this end, we though about using a chain-like structure to capture the chronology of articles by linking them and allow user to periodically check the claims of organizations, similarly to ClaimChain [3]. To make sure that organizations cannot lie on their article publication chronology and content, we could have used a protocol similar to Google's certificate transparency [4] to protect the PKI and organizations' claims.

5 Implementation

To implement the described protocol, we created a Go application to generate keys, create signatures and serialize them. We put online a couple of simplistic static pages as examples of articles and we wrote a web extension to verify signatures.

This extension is meant to show how a reader can easily check the validity of the embedded signature. It is compatible with Firefox, Chrome, Opera, and Safari. The implementation was not tested on mobile but according to mozilla documentation, the libraries used are compatible with mobile.

5.1 Cryptographic curves

We chose to use Elliptic-curve Digital Signature Algorithm (ECDSA) with the curve P-384 for compatibility with browsers and the hash function SHA-256.

5.2 Record and signature generation

To implement the record creation and signature generation algorithms we used Go and its package *crypto*. The implementation can generate keys, create records given its content, hash the text, generate the signature and serialize keys and signatures.

This way, a news organization can use this program to generate their keys and create records for every article they want to put online. The record is included in specific HTML <meta> tags in the webpage code.

5.3 Referencing

When a journalist needs to reference an article, they retrieve the record within the page <meta> tags and include it in their online piece when referencing the article.

5.4 Signature verification

When a reader wants to verify that the article they are reading is legit, they can use the web extension we implemented to easily and efficiently check whether the article references are legit.

Since verifications are done within a web extension using JavaScript, we chose to use the *WebCrypto* library and more specifically, *SubtleCrypto*'s verification algorithm and key/signature importation.

For now, we only provide single article referencing. To implement multiple referencing, we would need to make it possible through a slightly different format with more specific tags.

5.5 Serialization

We assumed that the PKI does the key handling, but we still needed to serialize keys and signatures for the actual implementation. To this end, a common choice is the PKCS8 format. It has the advantage of being widely used in linux systems for example and encrypted.

However, WebCrypto lacks the implementation of ECDSA signature importation in PKCS8 format so we had to use plain JSON. See the corresponding bugzilla ticket for more information [5].

5.6 UI

To test the whole pipeline, we put online test articles as static webpages. By using the web extension, we were able to verify automatically if a web page containing an article holds a correct signature to it for the right news organization.

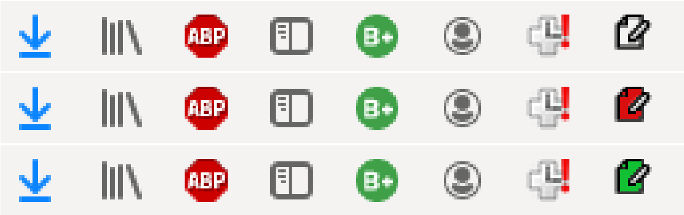

Figure 6: The extension icon (on the right) in firefox toolbar. It becomes green if reference(s) are verified and red otherwise

6 Conclusion

We were able to implement an article record signing and verification protocol that shows that we can automate journalism referencing. The protocols permits to handle deletion of articles by text downloading and edition of article by signature verification. Indeed, if a web page containing an article contains an edited article that without a new signature, the verification algorithm will not succeed and warn readers that the article is not legit.

Moreover, this simple web extension shows that for a minimal amount of work, readers can easily check journalists' citations.

7 Future work and Challenges

The implementation of the easy and trusted referencing of articles is very naive and lacks a few functionalities such as correct and secure handling of keys, a secure and globally reachable public key infrastructure and a real world implementation on articles with different formats.

The non-repudiation of articles is to be implemented as well as discussed in section 5. In addition, the current implementation is a prototype and would need to be integrated with news organization and journalists' workflow, such as in CMS they use.

A concrete implementation for the PKI has to be done as well. One way to do so is to use signatures like internet certificates and protect them with a tool like certificate transparency [4].

One hidden challenge in the implementation is the ability to easily extract text and metadata from articles automatically. This could be done either by using natural language processing and machine learning for example, or by utilizing AFP's NewsML, "an industry-driven format and processing model allowing rich machine-readable representation of news content" [6].

References

[1] Sharon Ringel. *A Public Record at Risk: The Dire State of News Archiving in the Digital Age*. 2019. URL: https://www.cjr.org/tow_center_reports/the-dire-state-of-news-archiving-in-the-digital-age.php.

[2] Susan E McGregor. "Digital Security and Source Protection for Journalists". In: (2014).

[3] Bogdan Kulynych et al. "ClaimChain: improving the security and privacy of in-band key distribution for messaging". In: *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*. ACM. 2018, pp. 86–103.

[4] B. Laurie, A. Langley, and E. Kasper. *Certificate Transparency*. RFC 6962. RFC Editor, June 2013.

[5] Simon Kölsch. *Implement PKCS8 import/export of ECDSA keys for WebCrypto API*. 2017. URL: https://bugzilla.mozilla.org/show_bug.cgi?id=1133698.

[6] *Technical guide to AFP NewsML-G2*. 2012-2019. URL: https://www.afp.com/communication/iris/Guide_to_AFP_NewsML-G2.html.