Verifiable Labels for Digital Services: A Practical Approach

Maël Gassmann
School of Engineering and Computer Science
Bern University of Applied Sciences
Biel/Bienne, Switzerland
email: mael.gassmann@bfh.ch

Annett Laube-Rosenpflanzer
School of Engineering and Computer Science
Bern University of Applied Sciences
Biel/Bienne, Switzerland
email: annett.laube@bfh.ch

***Abstract*—Users often feel unsafe and unsecure when they use digital services. For normal users without technical backgrounds, it is difficult to recognize if a website is genuine. This makes them vulnerable to phishing attacks. In order to solve this issue, many organizations use corporate designs or logos to guide users through their websites. However, all this can be easily copied. More technical means are also advertised as solutions, like trusted Transport Layer Security (TLS) certificates or Extended Validation (EV) certificates, but they are too complicated for non-technical users and barely make any difference. Right now, users lack a way to easily verify that they are using the intended digital service. A pure visual indication, e.g., with simple graphic files or technical means users do not understand, is not sufficient. Using the TLS Public Key Infrastructure (PKI), verifiable labels will use these certificates to bind an entity's label to the certificate's key pair. Instead of trying to provide automated trust, verifiable labels acknowledge the presence of ill-intentioned entities. In order to differentiate them from trustworthy actors, cryptography is used to define facts, which allows a user client to form easily understandable recommendations and analyze a certain actor's reputation. Thus, allowing users to naturally develop an opinion and make an educated guess as to whether an entity is worthy of their trust or not. The end goal would be that most business websites that ask for some level of trust would use verifiable labels; this way, websites with bad or no labels would start to stand out.**

***Index Terms*—Trust; Anti-Phishing; Digital Label.**

I. Introduction

Nowadays, if website owners want to try and certify an accordance to a label, one sole option is at their disposal: The usage of copyable and thus untrustworthy digital representations, such as pictures or electronic documents. Without having to make any distinction between true and false claims, it can already be deduced that it has as much value as a self-proclamation and is at least hard and inconvenient, if not impossible, to verify. This is leading naïve Internet users to give their trust to a service unworthy of any. Moreover, it is far from affecting only a limited number of people, as since 2020, phishing attacks have become by far the most common type of attacks performed by cybercriminals [1]; 41% of security incidents begin with the initial access gained by a phishing attack [2]; approximately 1.385 million new phishing web pages are set up each month [3]; and overall, phishing is in the top three cybersecurity threat trends [4].

The real problem is there; a verifiable label would truly add value to anybody's Internet experience by directly reducing the impact of phishing. One standalone example of such a label is the 'Digital Trust Label' [5]. However, it is very limited in its range of action. Verifiable labels strive to establish a distributed framework for the development of labels in general and enhance user friendliness.

The rest of the paper is structured as follows: Section II analyses the current state of Internet related technologies; Section III describes the concept of verifiable labels, its underlying infrastructure and protocols; Section IV explains how the concept was adapted to a working prototype; finally, the work is concluded in Section V.

II. State of the Art

A. TLS Certificates

Based on Public Key Infrastructure (PKI) to establish chains of trust and using X.509 certificates to bind web-servers to key pairs and domain names, Transport Layer Security (TLS) certificates are nowadays widely used to encrypt communications on the Internet [6]–[8]. These so-called chains of trust are all built upon an entrusted third party—a root of trust—that certifies the trustworthiness of other entities, which in turn are sometimes allowed to do the same. Such entrusted third parties are called Certificate Authorities (CA), as shown in Figure 1.

Furthermore, the X.509 certificate itself can contain a variety of different claims. For instance, one way to bind a certificate to a server is to include its specific domain inside.

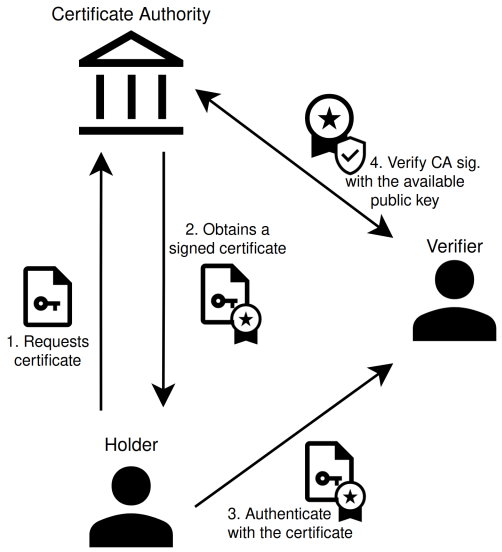

Figure 1. Minimalistic representation of a PKI.

In the case of TLS certificates, there are three major types of X.509 certificates that are used.

a) Domain Validated (DV) Certificate: These are the most basic types of certificates. The CA will only verify that the applicant has control over the requested domain name; this is typically done through email validation. More recently, the Automatic Certificate Management Environment (ACME) protocol allowed CAs to issue DV certificates without any intervention from their side [9]. When the ACME protocol is used, the certificate can be obtained free of charge.

b) Organisation Validated (OV) Certificate: Not only is the domain ownership verified, but also the legal existence and physical location of the applicant. Automation is, of course, out of the question. Such a certificate can be obtained for a range of 200 to +1000 USD per year [10].

c) Extended Validation (EV) Certificate: EV certificates undergo the most rigorous validation process; this includes all steps taken for OV certificates, including legal status, operational existence, and telephone verification [11]. The price range goes from 400 to +1700 USD per year [10].

OV and EV certificates were advertised as a way to prevent the customers' users from being prone to phishing, as the web browser, recognizing an EV certificate, used to display a green indicator containing the entity's legal name. Thus, users who knew of that distinction would change their behavior according to the level of certification displayed. However, studies showed that user behavior did not alter, and polls showed that the padlock's meaning was not understood correctly. Worse even, security researchers were able to prove that some EV certificates could be gotten with colliding organization names, which could be quite misleading as the domain would be hidden by the legal name in some browsers. That is why, in September 2019, most browsers stopped displaying any direct visual distinctions between DV, OV, or EV certificates, which invalidates the main selling point of these products [12].

Moreover, because CAs are private companies, the regulations are not always followed with the same rigor, as not all validation processes can be automated. A PKI infrastructure is always very sensitive to mistakes, and the verification process has proven to not be enough [13]. However, one thing is sure: TLS certificates do a good job of binding a domain name to its corresponding server, which holds the key pair. Especially with the help of the ACME protocol.

B. Decentralised Identifiers

In opposition to the traditional central authoritative system that PKI represents, Decentralized IDentifiers (DID), an open standard in active development, is part of a broader movement that strives towards decentralized identity. A DID resolves to a DID document—typically hosted on a decentralized network or infrastructure, e.g., a block chain or a distributed ledger—which contains a set of public keys, authentication methods, service endpoints, a time-stamp to keep an audit history, and a signature for its integrity [14].

A Verifiable Credential (VC) is a claim created from the key pair of a DID (the issuer) and is issued to a holder's wallet by using a holder proof. This holder proof varies greatly between implementations, and efforts are being made to standardize it. Self-Sovereign Identity (SSI) solutions strive to provide a way to assert, present, and verify claims in a decentralized manner [15].

The current limitations are that not everything is yet standardized. For instance, linking an existing TLS certificate key pair to a DID misses specifications. More than that, VCs need a wallet with a functional, universal holder proof. In the end, this solution is not yet defined and widely used enough to be applied in this specific use case.

III. Concept

A. Different Perspective

The root of the verifiable label concept lies in a shift of perspective on what trust is and how can it be made identifiable to an end-user. As TLS EV certificates proved, a seemingly good concept will still need to be understood by anyone who uses the Internet in order to have any impact, especially by those who do not have any technical background. First, one must understand how trust is perceived as a concept alone; for this, a philosophical definition of trust is adequate.

'Trust is important, but it is also dangerous. It is important because it allows us to depend on others—for love, for advice, for help with our plumbing, or what have you—especially when we know that no outside force compels them to give us these things. But trust ***also involves the risk*** *that people we trust will not pull through for us, for if there were some guarantee they would pull through, then we would have no need to trust them.* ***Trust is therefore dangerous****. What we risk while trusting is the loss of valuable things that we entrust to others, . . . '* [16]

That is, when a person *decides* to place their trust in someone else, they know about the risks—risks that can be clearly identified as they are based on facts.

Instead of distributing a trust people have to blindly believe in, verifiable labels proposes the idea of providing simple facts about Internet entities so that anyone with no technical background can, in a reasonable time, learn how to navigate the Internet with a valid sense of which entity deserves their trust. Trust is, after all, an individual decision, and users must be able to make that decision for themselves and not have to rely on a third-party organization they do not even know exists.

To do this, cryptography is paramount, as it is the sole option available to make any virtual information a tangible fact. The system must be implemented on top of the currently widely used Internet cryptographic technologies (e.g., TLS certificates) in order to have any chance of success, while also striving to be flexible and pushing towards more decentralized technologies (e.g., blockchains) because they provide a non-authoritative infrastructure.

B. Definitions

1) VERIFIABLE LABEL
   A verifiable label is a data structure that is bound to two domain names; the holder's and the issuer's. This is done by signing the label with both cryptographic identifiers

(e.g., TLS certificate). Other attributes will be present in order to allow users to derive a clear reputation for each label declared on a website and thus, the direct trustworthiness of the web entity.

2) VERIFYING USERS
Simple users that visit a website. If a valid label is detected, the user will be able to see it, list facts that concern it, and develop an idea of this label's reputation.
3) LABEL-WORTHY WEB-ENTITY
Such an entity can request a label from its corresponding issuer. If an issuance occurs, they can display their digital label on their website, which is visible and verifiable by anyone. It cannot be copied.
4) ISSUER
The entity that can verify and decide of its own accord who is worthy of being labeled. It will keep a record of who has been issued its label and can confirm it.
5) TIME STAMP AUTHORITY (TSA)
The backbone of the concept is here; this time stamp authority [17] will follow specific automated guidelines. While the automation makes sure that every issuer plays by the same rules, the guidelines aim at enforcing duplicate label prevention. As all issuers need an unexpired certificate, they will have to issue renewal requests, which, built upon one another, start to create a reputation. Every time-stamped issuer certificate will be stored in a publicly readable storage.

C. Protocol

a) Issuance of a label: Figure 2 depicts it. This is the least protocoled part of the system. A website must create a verifiable label and sign it with its TLS certificate. This ensures that the draft label certificate is bound to the domain name and also comes from the stated owner. The incomplete digital label can be sent to the issuer; no channel is specified. If the issuer decides to accept the request, it will sign it with its own TLS certificate, add the new signature to the now complete verifiable label, and send it back. Finally, the issuer save a copy of the signature and requester's domain in the list of its own draft certificate. In order to make a valid issuer certificate out of this draft, the issuer has to request a new time-stamp and signature from the TSA, as explained in Figure 3.

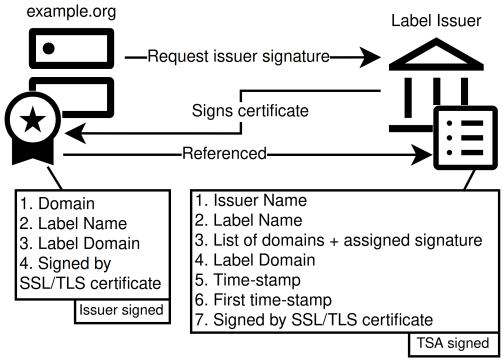

Figure 2. Certificate Issuance

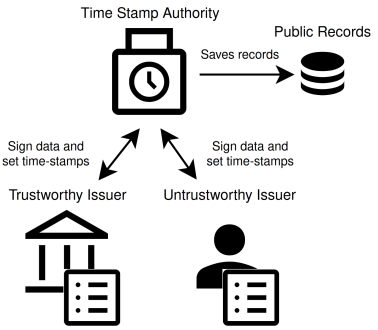

Figure 3. Time Stamp Authority

b) Issuer Certification: As stated before, an issuer's trustworthiness is defined by its own reputation. This reputation is built with time and the help of an automated time-stamp authority. The TSA's role is to reissue new signatures—necessary for the issuer's label to be considered cryptographically valid—and time-stamps to all requesting issuers that are on the brink of expiration. As it does so, it will first store a copy of the renewed certificate in a publicly readable storage and then send it back. However, if the issuer is new, i.e. does not possess a first time-stamp, the TSA will have a look at the requested label name, domain name, issuer name, and all fields that might be prone to confusing a human being if not filled with good intentions. If it is considered not to be confusing as well as not a duplicate of any existing labels, the web entity will receive its first time-stamp and signature, making it an issuer.

c) Validation and interpretation client: As a user with the verifiable label validation and interpretation client installed navigates the Internet, the client will try to detect if a digital label is present on the currently visited website. If it proves to be the case, the certificate validation process will begin, as shown in Figure 4. The first step consists of verifying the label's link with the domain and TLS certificate, that is, making sure the signature is correct and that the domain corresponds to the browser URL. On success, the next step will

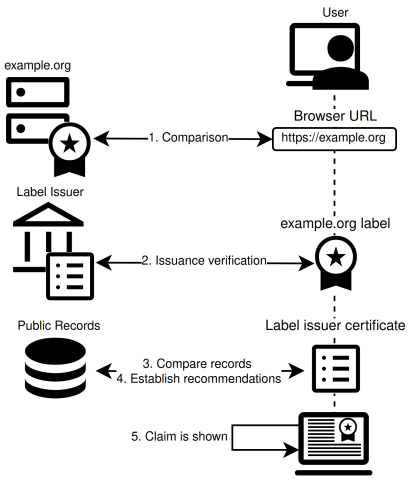

Figure 4. Certificate Validation

be to check the listed label domain for the issuer certificate. Records are to be compared, signatures are to be verified for the same reason as before, and the domain of the label must be found in the list. At last, if everything succeeds again, the client will search the label's public records for what could be considered pertinent to derive a reputation from. This could include refreshment regularity, time of existence, measurement of the movement of certified entities, activity ratio, and number of requests. How these measurements should be interpreted is now unknown; in order to gain more insights, the system would have to be put into place, and practical testing could allow recommendations to be derived. If a big enough dataset of reputation measurement could be collected, a machine learning algorithm could do a good job at identifying untrustworthy labels and trends in how attackers are trying to infiltrate the system. Staying flexible in this interpretation and allowing it to evolve is paramount; one fixed description of what an untrustworthy label is would never be enough; the system needs to be as resilient as the spoofers.

D. Purpose

Accurate protection is possible if we assume that a majority of web entities have adopted this digital label system. Only then would websites with bad or no certifications start to stand out, especially when they require trust, e.g., when they ask for credit card information or propose services.

IV. Implementation

A prototype has been implemented following a minimal working system approach. Furthermore, since different underlying technologies exist, extensibility is a top priority.

A. Verifiable Label Time Stamp Authority (VLTSA)

Starting from the very root of the system, the VLTSA is not a time stamp authority. Preferring a simple method, a library implementing a RFC 3161 [17] client interface was used to interact with an external TSA to provide the time-stamps. A TLS certificate was used as a way to warrant the need for issuers to still directly request the time-stamps from the VLTSA, as a third signature from that certificate is required to make the issuer certificate valid. This server software consists of a simple HTTP API with two paths: the POST method on '/sign' and the GET method on '/get_records'. Meaning it also acts as the publicly readable storage. All of this has been implemented in the most minimalistic way, with abstract interfaces of 'Storage', 'API', and 'Signer'. That is where flexibility is; the logical part of what makes the VLTSA is detached from all other components that could find better long-term alternatives (e.g., more resource-efficient or different time-stamp sources such as a blockchain).

B. Verifiable Label Issuer Client (VLIC)

The simple command-line client has persistent storage and saves all valid given arguments. If provided with a valid request, it will add a domain to its certificate and generate a valid verifiable label certificate (.vlcert), which can be sent back to the holder through any channel. It can issue a signing request to the VLTSA on demand. And, if successful, it will save the verifiable label issuer certificate (.vlicert). This vlicert has to be exposed on the label domain's web-server root as 'cert.vlicert'.

C. Verifiable Label Holder Client (VLHC)

This simple command-line client with no persistent storage can only be used to generate a vlcert without the issuer signature. It has to be manually sent to the issuer. Once a valid vlcert is in the holder's possession, it has to be exposed on its domain's web-server root as 'cert.vlcert'. This prototype thus only allows for one vlcert per holder.

D. Browser Extension Analyzer

A browser extension was a mandatory component of the client, as the active URL has to be accessed to perform the first cryptographic tests. However, the specific environment did not provide any way to download a TLS certificate for a specified domain, which blocked further development. More research showed that by using the native messaging interface, the browser extension can communicate data to an underlying program. Using this method, a cryptographic verifier was developed. It sends back the necessary data to perform a reputation analysis and is then displayed in a panel.

V. Conclusion

A solution that allows Internet entities to create verifiable labels, as well as aiming to reduce fraud was proposed. Based on TLS certificates and time stamp authorities, the current prototype stays flexible and, even if simplistic, already implements all the necessary cryptographic tools.

Future work could investigate the following directions:

- Conduct a field study of a live setup and user experience.
- Study relevant metadata for a good reputation evaluation.
- Provide a comprehensive User Interface (UI) for computers and phones.

References

[1] FBI Internet Crime Complaint Center, "Internet Crime Report 2020," 2020, *Online*. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf [retrieved: 08, 2023].

[2] IBM Security, "X-Force Threat Intelligence Index 2023," 2023, *Online*. Available: https://www.ibm.com/reports/threat-intelligence [retrieved: 08, 2023].

[3] M. Swindells, "How many phishing emails are sent daily in 2023? 11+ statistics," 2023, *Online*. Available: https://earthweb.com/how-many-phishing-emails-are-sent-daily/ [retrieved: 08, 2023].

[4] Cisco, "Cybersecurity threat trends: phishing, crypto top the list," 2021, *Online*. Available: https://umbrella.cisco.com/info/2021-cyber-security-threat-trends-phishing-crypto-top-the-list [retrieved: 08, 2023].

[5] Swiss Digital Initiative, "The Digital Trust Label," Swiss Digital Initiative, 2022, *Online*. Available: https://digitaltrust-label.swiss/ [retrieved: 08, 2023].

[6] T. Dierks and E. Rescorla, "RFC 5246: The Transport Layer Security (TLS) Protocol, Version 1.2," August 2008, *Online*. Available: https://www.rfc-editor.org/rfc/rfc5246 [retrieved: 08, 2023].

[7] IETF Trust, "The Transport Layer Security (TLS) Protocol Version 1.3," August 2018, *Online*. Available: https://datatracker.ietf.org/doc/html/rfc8446 [retrieved: 08, 2023].

 ISBN: 978-1-68558-115-2

[8] R. Housley, W. Polk, W. Ford, and D. Solo, "RFC 3280: Internet X.509 Public Key Infrastructure," April 2002, *Online*. Available: https://www.ietf.org/rfc/rfc3280.txt [retrieved: 08, 2023].

[9] Internet Engineering Task Force (IETF), R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten, "Automatic Certificate Management Environment (ACME)," March 2019, *Online*. Available: https://datatracker.ietf.org/doc/html/rfc8555 [retrieved: 08, 2023].

[10] DigiCert, "Compare TLS/SSL certificates," 2023, *Online*. Available: https://www.digicert.com/tls-ssl/compare-certificates [retrieved: 08, 2023].

[11] CA/Browser Forum, "EV SSL Certificate Guidelines," 2022, *Online*. Available: https://cabforum.org/extended-validation/ [retrieved: 08, 2023].

[12] Chromium Docs, "EV UI Moving to Page Info," 2019, *Online*. Available: https://chromium.googlesource.com/chromium/src/+/HEAD/docs/security/ev-to-page-info.md [retrieved: 08, 2023].

[13] C. Cimpanu, "Extended Validation (EV) Certificates Abused to Create Insanely Believable Phishing Sites," 2017, *Online*. Available: https://www.bleepingcomputer.com/news/security/extended-validation-ev-certificates-abused-to-create-insanely-believable-phishing-sites [retrieved: 08, 2023].

[14] W3C, "Decentralized Identifiers (DIDs) v1.0: Core architecture, data model, and representations," W3C Recommendation 19 July 2022, *Online*. Available: https://www.w3.org/TR/2022/REC-did-core-20220719/ [retrieved: 08, 2023].

[15] W3C, "Verifiable Credentials Data Model v1.1," W3C Recommendation 03 March 2022, *Online*. Available: https://www.w3.org/TR/2022/REC-vc-data-model-20220303/ [retrieved: 08, 2023].

[16] C. McLeod, "'Trust', The Stanford Encyclopedia of Philosophy (Fall 2021 Edition)," 2021, *Online*. Available: https://plato.stanford.edu/archives/fall2021/entries/trust [retrieved: 08, 2023].

[17] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)," RFC 3161, August 2001, *Online*. Available: https://www.rfc-editor.org/rfc/rfc3161 [retrieved: 08, 2023].

 ISBN: 978-1-68558-115-2