ISSN: 2666-2795 Vol. 6 No. 2 December, 2021

*International Journal of Applied Engineering Research*

# Data Notarization without Blockchain

**Francesco Buccafurri, Vincenzo De Angelis, andGianluca Lax**
University of Reggio Calabria, via Università 25, 89122, Reggio Calabria, Italy
bucca@unirc.it, vincenzo.deangelis@unirc.it, lax@unirc.it

***Date of Submission: 07th May 2021 Revised: 20th June 2021 Accepted: 24th June 2021***

***How to Cite:*** **Buccafurri, F., De Angelis, V., and Lax, G. (2021). Data Notarization without Blockchain. International Journal of Applied Engineering Research, 6(2)**

***Abstract*** **- The process of data notarization plays an important role in multiple application contexts. Data notarization is typically implemented by leveraging blockchain technologies. In certain cases, the adoption of blockchain in real-life applications could be not adequate, due to the cost of transactions, in the case of public blockchains, or the complexity of the solution, in the case of permissioned blockchains. Therefore, it is interesting to explore possible alternatives to blockchain. This paper proposes a notarization scheme leveraging existing social networks. The scheme allows multiple notarizations on the same data and, due to the social-network paradigm which it relies on, it is strongly oriented to cooperative work and information-sharing-based applications.**

*Index Terms* - Dematerialization, E-government,Public Verifiability, Social Networks

## INTRODUCTION

In various application contexts, digital data notarization plays an important role. Apart from the classical concept of document notarization, which regards the action of a Public Notary aimed to assure that a document is authentic and can be trusted, there are many other cases in which data should be notarized with public verifiability. This is the case of clinical data, biomedical databases, genomic data, social media data, data archiving, online digital contents, news, digital identity management, digital evidences for forensics purposes, workflow traces for accountability, and so on [1], [2], [3], [4], [5].

The problem of data notarization has been deeply addressed in the last years thanks to the spread of blockchain technologies [6], [7]. Indeed, blockchain natively supports immutability and data sharing and, thus, can be profitably used for notarization [8], [9].

Depending on certain aspects (for example, the amount of data to notarize), the use of public blockchains could not be adequate, due to the required economic cost for each notarization. On the other hand, permissioned blockchains are complex (multi-organization) ecosystems, in which we cannot guarantee data immutability because the controlling authorities can become flawed by an attacker, then establish a coup of flawed validators, and then cancel or create arbitrary transactions [2]. Indeed, to have a robust notarization system, the use of public blockchain should be anyway combined with permissioned blockchains [10].

This paper tries to give an answer to the following question. Can we implement data notarization not relying on blockchain?

We find the response to the above question in the domain of social networks, being aware that they can be used for various applications that go much beyond sociality goals [11], [12], [13], [14].

The basic idea underlying our solution is to leverage both (1) the power of online social network to share information among people, (2) the plausibility of certain assumptions concerning the dependability of services provided by online social networks, (3) the trustworthiness of social network providers, and (4) the level of assurance of the accountability associated with the interactions between the notarization entities and their social network profile. The notarization functions are spread out over social-network profiles by giving posting and searching operations a central role in both notarization generation and notarization verification processes in a very usable and scalable fashion. This aspect makes our notarization scheme, similarly to blockchain-based approaches, strongly oriented to cooperative work, information-sharing-based applications, and also suitable to notarize streams of data, thus applicable to any application scenario. In other words, our social-network-based approach does not lose the innovation given by the application of blockchain to data notarization with respect to previous (standard) digital-signature-based approaches [15], [16], [17], [18], [19].

The paper is organized as follows. Section II is devoted to the related literature. In Section III, we present our social network-based notarization solution. The security analysis and the security properties required for the proposed notarization protocol are reported in Section IV. Finally, in

Section VI, we discuss some further details of our proposal, sketch possible future work and draw our conclusions.

## RELATED WORK

In this section, we survey the most relevant proposals related to digital notarization. As mentioned in the introduction, before applying blockchain [6], [7], researchers focused their attention mainly on the problem of document notarization (thus in a restricted application domain), by applying PKI-based digital signatures schemes. Among these approaches, we cite here the proposal of [18], who focus on technology and trust issues related to the long-term validation of a digital signature and propose a mechanism for cumulative data notarization that results in a successive trust transition towards new entities, modern technologies, and refreshed data. The proposed framework uses recursive XML elements so that a notarization token structure encapsulates an identical data structure containing a previous notarization token. However, only through the application of blockchain, the paradigm of data notarization assumed a new form, becoming oriented to multiple application domains and to open, cooperative, and sharing environments. Our paper is then related to this type of data-notarization approaches.

The use of blockchain to notarize social media contents is proposed in [3]. The paper focuses on how to assure that input data are not forged before inserted into a block, and the proposed solution exploits an official service provider that generates digital signatures of uploaded contents. A public key infrastructure protocol is used to verify whether content is forged or not.

A blockchain-based notarization service that uses smart digital contracts to seal a biomedical database query and the respective results are proposed in [20]. The goal is to ensure that retrieved data cannot be modified after retrieval and that the database cannot validly deny that some data have been provided as a result of a specific query. An improvement of this solution, which supports data versioning, is presented in [21].

Still, in the field of medical data, the authors of [2] show the benefits of using blockchain technology as a notary service in the network sharing of clinical data. Among others, an interesting perspective is the use of blockchain for clinical trials in genomic data management. In this approach, versioning of documents and their notarization assume a crucial role.

In [22], the authors propose a blockchain-based data-sharing framework that guarantees the authenticity of shared data in real-time and provides transactional privacy. A prototype of the proposed framework is presented to prove that privacy, integrity, and fine-grained access control over shared data are provided. This proposal aims to reduce the turnaround time for data sharing, to improve the decision-making process, and to reduce the overall cost.

A protocol to notarize files over blockchain is proposed in [23]. The protocol guarantees the communication between two sub-systems: the Blockchain and a centralized archiving solution. The major offered services are document archiving, document retrieving, and document proof existence. The blockchain is used to trace transactions related to archived documents.

In [5], a solution to notarize online digital content such as books, music, and movies is proposed. The solution utilizes the Inter Planetary File System (IPFS) and blockchain, relying also on smart contracts to comprise actions on documents. IPFS is used to store digital contents and blockchainto notarize document storage and other actions.

The authors of [10] present a digital notary scheme with high security and low cost. A client of the service receives evidence in three steps. In the first step, the evidence is received almost immediately, but a lot of trust is required. In the second step, less trust is required, but the evidences received seconds later. Finally, in the third step evidence is received within minutes via the interaction between two blockchains, one of which is public.

A specific application of blockchain for data notarization regards the domain of digital identity management. [24],describes the provisions of services to citizens based on blockchain, such as e-residency, a program launched in Estonia[25], [4] in which the blockchain notary service allows-residents, regardless of where they live or do business, to notarize their marriages, birth certificates, business contracts on the blockchain.

The authors of [26] propose a notarized federated identity management model supporting user authentication when providers do not know each other. In the paper, a notary service is introduced, owned by a TTP, aimed to dynamically notarize assertions produces by identity providers.

In [1], the role of blockchain notarization has been highlighted also in the field of genomics services. In this paper, the author present is *Genesy*, an innovative blockchain platform that notarizes genomic data, thus facilitating the safe interaction with all the interested parties (such as research centers, pharmaceutical companies, hospitals, and geneticists).

The aim of our paper is then to explore an alternate way to obtain data notarization without using blockchain. Evidently, the proposal should not be view in competition with blockchain-based notarization, but only a different perspective, applicable when, as highlighted in the introduction, the adoption of blockchain is not feasible.

## THE SOCIAL-NETWORK-BASED NOTARIZATION SOLUTION

In this section, we present our solution for data notarization. For the sake of presentation, we refer, as the subject of notarization, to a *record*. However, by the term

record, we generically mean some data in any format and any size.

The model which we refer to is composed of:

**The Actors:** The entities involved in the notarization process: The *notarization master* $M$, the *notarization entity* $S$, and the *social network* $T$.

**A Posting Integrity Mechanism:** A mechanism provided to the notarization entities and the notarization master to prevent the corruption of their posting timeline.

**A Registration Protocol:** The initial phase, where the real-life identity of notarization entities is associated with a profile in the social network $T$ and authenticated to thenotarization master. A secure sharing of this information isestablished.

**A Notarization Generation Protocol:** The protocol followed by notarization entities to notarize a record.

**A Revocation Protocol:** The protocol aimed at inhibiting a given notarization entity to validly run the notarization protocol from the moment of revocation on.

**A Notarization Verification Protocol:** The protocol followed by any party to verify the validity of the notarization of the record.

**The Security Parameters:** The parameters concurring at establishing the level of security of the whole notarization process.

Now, we describe in detail each of the above components.

**Actors.** Any instance of the model consists of one notarization master$M$, a set of notarization entities (that canbe dynamically updated), and one social network $T$. Only$T$is assumed to be trusted. Both notarization entities and notarization master have a profile in $T$. We assume that thissocial network supports (at least) (1) a friendship relation,(2) the possibility for notarization entities to post a textmessage equipped with a timestamp in a public section of their profile, (3) the possibility of searching for a message posted in a public section by means of usual text-search tools. For each notarization entity $S$, a friendship with thenotarization master is established in the social network $T$.We require that in $T$, automatically, any message postedin his public section by a friend $S$ of $M$ is conditionally (according to a given condition set by $M$) replicated by $M$in its public section, by including the same information, insuch a way that the deletion of the original message does not propagate to the replicated messages. This step is performed in the notarization generation protocol.

**Posting Integrity Mechanism.** Each notarization entity $S$ is provided with a second pre image resistant trap-door function$f: \mathcal{D} \rightarrow \mathcal{D}$, where $\mathcal{D}$is a finite set. Moreover, $S$owns a secret information $p$ such that, for a given output $y$ of the function $f$, finding $x$ such that, $y = f(x)$ is feasible only for$S$ (who owns$p$). Each post (belonging to the notarization protocol) of a notarization entity S is associated with a value$y$ such that the next post of the same notarization entity isassociated with a value $x$ such that $y = f(x)$. Consider thatonly $S$ is able to generate a legal new post because he is theonly party able to compute the value to associate with the post. As shown in Section IV, this measure prevents attacks based on the posting-timeline corruption.

**Registration Protocol.** All (real-life) notarization entities are originally identified (and assigned to a social network profile) by the notarization master. At the end of this phase, the registration protocol starts.

The first step is the publication done by the notarization entity $S$ in the public section of his profile of the first message$\langle y_0, I, ID_S \rangle$, where$y_0$is a random value of $\mathcal{D}$,$I$ is used to associate $S$ with the notarization master $M$,and $ID_S$is the identifier of $S$ in the social network $T$.

Moreover, for each notarization entity $S$, the associationbetween a real-life identity of $S$ and his social network profileis published as a welcome message for $S\langle y_0, J, ID_S \rangle$, in the public section of the notarization master profile, where$y_0$and $ID_S$ are the same as the registration message, and$J$ is the real-life identity of $S$.

**Notarization Generation Protocol.** Let $D_i$ be the $i$-th (i> 0) record (in order of time) being notarized by thenotarization entity $S$ with identifier $ID_S$ and $h$ be a $k$-bit cryptographic hash function. We define the notarization message as$\langle y_i, h(D_i), ID_S \rangle$, where$y_i \in \mathcal{D}$.

A notarization message$\langle y_i, h(D_i), ID_S \rangle$ is said linked (in $S$) if (recursively)$f(y_i) = y_{i-1}$ and either $\langle y_{i-1}, h(D_{i-1}), ID_S \rangle$ is the latest linked notarization message associated with $ID_S$occurring in $T$ or $\langle y_{i-1}, I, ID_S \rangle$is the first message postedby $S$ at the registration phase (i.e., $y_{i-1} = y_0$).

The notarization generation protocol consists of the publication done by $S$ in the public section of his profile of a notarization message$\langle y_i, h(D_i), ID_S \rangle$, which enforces that $M$ either:

1) replicates this message in its public section (thus producing a message also called confirmation message),if it is linked in S and there not exists another message of the form $\langle y_i, h(D^*), ID_S \rangle$in $S$ or $M$ with$h(D_i) \neq h(D^*)$, or
2) publishes an aborting message (for $y_i$)$\langle y_i, h(D_i), ID_S \rangle$ (meaning that the notarization procedure fails), if the notarization message is linked in $S$and there exists another message $\langle y_i, h(D^*), ID_S \rangle$ in$S$ with $h(D_i) \neq h(D^*)$, or

3) does nothing, otherwise

Moreover, the protocol enforces that the maximum time between the publication of the message by $S$ and thepublication of the corresponding message by $M$ is notgreater than $\Delta t$ which we call reaction time constraint.The security mechanism based on the trap-door results in a chain of notarization messages starting from the original registration message $\langle y_0, I, ID_S \rangle$ that only the notarizationentity is able to generate. This chain is also published bythe notarization master M due to the replication mechanismdescribed above. According to the protocol, the chain isreplicated by $M$ for all linked notarization messages through confirmation/aborting messages. However, if a notarization message is not linked, no message is published by $M$. Inother words, no confirmation message for a notarizationmessage $\langle y_i, h(D_i), ID_S \rangle$ such that $f(y_i) \neq y_{i-1}$may occurin the posting timeline of $M$ or such that the iterated self-composition of $f$ to $y_i$ does not allow us to reach $y_0$ after$i$ steps, where $\langle y_0, I, ID_S \rangle$is the first message posted by $S$at the registration phase.

Algorithm 1 schematizes the above protocol.

If multiple notarization entities occur, each one performs the Notarization Generation Protocol to notarize the same report i.e., each of them publishes a notarization message including the same digest. For each notarization message published by the notarization entities, the master publishes a confirmation message, or an aborting message, or does nothing according to the protocol defined above.

**Revocation protocol.** The revocation of a notarization entity S is done by the notarization master by simply removing the friendship of $S$ and by publishing a revocation messageidentifying the time and reasons of revocation. If a revocationmessage with timestamp $t_r$ occurs for $S$, we say that$S$ is revoked at each instant greater than $t_r$.

**Algorithm 1 Notarization Generation Protocol**

**Notation** $S$: the notarization entity
**Notation** $M$: the notarization master
**Notation** $T$: a social network
**Notation** $h$: a $k$-bit cryptographic hash function
**Input** $ID_S$: the identifier of $S$ in $T$
**Input** $D_i$: the $i$-th record being notarized by $S$
**Input** $y_i$: a value in $\mathcal{D}$

```
S publishes ⟨y_i, h(D_i), ID_S⟩
if (⟨y_i, h(D_i), ID_S⟩ is linked in S) then
    if (∄⟨y_i, h(D*), ID_S⟩ in S and ∄⟨y_i, h(D*), ID_S⟩ in M) then
        M publishes ⟨y_i, h(D_i), ID_S⟩ in Δt
    else if ((h(D*) ≠ h(D_i))) then
        M publishes the aborting message ⟨y_i, h(D_i), ID_S⟩ in Δt
    end if
end if
```

**Notarization Verification Protocol**. Concerning the validity of the notarization, our verification procedure, for any detected potential notarization entity, outputs two possible values each equipped with an attribute, resulting in 4 different outcomes `(value, attribute)`:(1) `(valid, safe)`, (2) `(invalid, safe)`, (3) `(valid, unsafe)`, and, finally, (4) `(invalid, unsafe)`. The full meaning of the attributes will be clear in Section IV. Basically, a `safe` outcome means that no anomaly is detected. Conversely, an `unsafe` outcome indicates the occurrence of an anomaly (i.e., attacker failure). Interestingly, the anomaly does not affect the (even legal) effects of the notarization, because the returned value is always `valid` or `invalid`. Therefore, the `attribute` is extra information given as a warning to detect an anomaly (for example, a blocked attempt of attack). As it will be clear in Section IV-B, besides the `attribute`, the notarization verification procedure could return in principle further information related to the origin of the anomaly. For simplicity, we do not consider this feature here.

Let $D$ be the record whose notarization has to be verified.The verification protocol works as follows. First, the digest$h(D)$ is computed. Then, $h(D)$ is searched among the publicinformation posted in the social network $T$.

If $h(D)$ is not found, the verification returns the outcome `(invalid, safe)`, with no other information. In other words, record D is detected as not notarized. Otherwise, weave the following cases:

1) Both the notarization message$\langle y, h(D), ID_S \rangle$ and its corresponding confirmation message exist and $S$ is notrevoked. In this case, the verification procedure returns `(valid, safe)` w.r.t $S$. Observe that, according tothe notarization generation protocol, the presence of the confirmation message ensures that$\langle y, h(D), ID_S \rangle$ is linked in S.
2) A notarization message $\langle y, h(D), ID_S \rangle$ exists either inS or M (not in both).In this case, the verification procedure returns:
   a) `(valid, unsafe)` w.r.t $S$ if either:
      i) the found message $\langle y, h(D), ID_S \rangle$ is linked in$S$, there not exists an aborting message for $y$ in$M$ with time delay w.r.t. the notarization messageless than (or equal to) $\Delta t$ (thus compliant withthe reaction time constraint $\Delta t$), and $S$ is not revoked at the time stamp of the notarization message, or
      ii) The found message $\langle y, h(D), ID_S \rangle$ is linked in S, a message exists (in $S$ or$M$) $\langle y, h(D'), ID_S \rangle$,such that$D \neq D'$, the message with lowest time stamp between $\langle y, h(D), ID_S \rangle$and $\langle y, h(D'), ID_S \rangle$contains$h(D)$, and $S$ is notrevoked at the timestamp of the notarizationmessage.

   b) `(invalid, unsafe)` w.r.t $S$, otherwise.

We observe that in the case of multiple notarization entities, the notarization verification protocol associates each notarization entity with an outcome `(value, attribute)`. The algorithm describing this protocol is shown in Algorithm 2.

**Security Parameters.** The level of assurance of the authentication procedure of notarization entities when logging into their social network profile, say $LoA(Auth)$, is a parameterof the model (the value of this parameter can be assumed belonging to the score given by NIST in [27], thus from1 to 4). According to our model, a notarization solution enforces that all (real-life) notarization entities are originally identified (and assigned to a social network profile) byte notarization master (registration protocol) with a given level of assurance, say $\langle LoA(Id), p\rangle$, which is a pair ofparameters denoting with $LoA(Id)$ the identification levelof assurance adopted in the registration phase (among thelevels given by NIST in [27], thus from 1 to 4), and with $p$if the identification is *in-person* ($p$ = 1) or *remote* ($p$ = 0).

$\langle LoA(Id), p\rangle$ Induces the level of assurance of the *identification* function of the notarization.

$LoA(Auth)$, $\langle LoA(Id), p\rangle$ and the other security featuresof a concrete instantiation of the model induce the level ofassurance of the notarization, thus the security of the non repudiation service implemented by the notarization itself. Other parameters could exist, depending on the concrete instantiation of the model.

**Algorithm 2 Notarization Verification Protocol**

**Notation** $T$: a social network
**Notation** $S$: the notarization entity
**Notation** $M$: the notarization master
**Notation** $h$: a $k$-bit cryptographic hash function
**Notation** $\Delta t$: the reaction time constraint
**Notation** $ID_S$: the identifier of $S$ in $T$
**Notation** $m$: a notarization message $\langle y, h(D), ID_S\rangle$ for the record D
**Notation** $m'$: a notarization message $\langle y, h(D'), ID_S\rangle$ $(D' \neq D)$
**Notation** $\overline{m}$: an aborting message $\langle -y, h(D), ID_S\rangle$ on $y$
**Notation** $t_m$: time delay w.r.t. the notarization message $m$
**Notation** $min(m, m')$: a function returning the message with the lowest timestamp
**Input** $D$: the record to be verified
**Output** $results$: an empty list

1: Compute $h(D)$;
2: Search $\langle h(D)\rangle$ on $T$;
3: **if** ($\nexists \langle h(D)\rangle$) **then**
4: add $\langle$`null`, (`invalid`, `safe`)$\rangle$ to $results$;
5: **return** $results$;
6: **end if**
7: **for each** $ID_S$ found **do**
8: **if** ($\exists$ $m$ in $S$ **and** $\exists$ $m$ in $M$ **and** $S$ is not revoked **then**
9: add $\langle ID_S$, (`valid`, `safe`)$\rangle$ to $results$;
10: **else**
11: **if** ($\exists$ $m$ in $S$ **or** $\exists$ $m$ in $M$) **then**
12: **if** ($m$ is linked in $S$ **and** $\nexists\ \overline{m}$ in $M$ with $t_m \leq \Delta t$ **and** $S$ is not revoked) **then**
13: add $\langle ID_S$, (`valid`, `unsafe`)$\rangle$ to $results$;
14: **else if** ($m$ is linked in $S$ **and** ($\exists$ $m'$ in $S$ **or** $\exists$ $m'$ in $M$) **and** $min(m, m') = m$ **and** $S$ is not revoked) **then**
15: add $\langle ID_S$, (`invalid`, `safe`)$\rangle$ to $results$;
16: **else**
17: add $\langle ID_S$, (`invalid`, `unsafe`)$\rangle$ to $results$;
18: **end if**
19: **else**
20: add $\langle ID_S$, (`invalid`, `unsafe`)$\rangle$ to $results$;
21: **end if**
22: **end if**
23: **end for**
24: **return** $results$;

## SECURITY MODEL

In this section, we state the security properties required forth proposed notarization protocol. We denote by $SM$ thesecurity model so obtained. Such properties are analyzed interms of the possible attacks that an adversary can perform. The adversaries considered are the most powerful, namely, internal or external notarization entities and the notarization master $M$. Hence, a protocol that is secure against suchadversaries is also secure against any other adversary who is an outsider or a normal notarization entity in the system.

Observe that even though most of the anomalies based on software/network (fault-based) failures are subsumed by the considered attacks, the aim of this section is to analyze thesecurity of our protocol, not its dependability in the largest meaning.

### *A. Assumptions, Security Properties, and Attacks*

Our threat model realistically considers the following assumptions:

**A1**Collision, pre image, and second pre image attacks on The cryptographic hash function $h$ are infeasible;
**A2**The social network $T$ acts as a trusted third party.
**A3**The attacker cannot add or compromise information Shown on the social network accounts of the notarization

Master and notarization entities with no legal authentication;

**A4**The authentication on the social network is configured to require a double-factor authentication (level of assurance3 of the NIST model [27]);

**A5**The initial registration is done at the level of assurance 3 in-person of the NIST model [27];

**A6**The occurrence of the initial registration is provable By both parties (the master and the notarization entity), as well as the integrity of the published registration message;

**A7**The exact duration of the notarization entity's status granting the right to use the notarization service can be Proven by means of secure information external to the System (e.g., records kept by the master);

**A8**The information enabling the trap-door function of the Posting integrity mechanism is kept secret;

**A9** No collusion may occur between the master and the Notarization entity.

Assumption **A8** means that the adopted solution is able to protect the secret against at least a keylogger-based attack on the client device. Now, we are ready to state the security properties of our protocol. Thanks to Assumption**A9**, we consider only the cases where the attacker is either a notarization entity or the notarization master.

**Security Property 1 (SP1) - *Notarization Origin.*** SP1 is defined as follows: *A notarized record should be always verified as notarized by the real notarization entity*.

The attack model we consider to describe how this property can be threatened is the following:

**Attack AA1:** An adversary tries to use a fake social network profile to jeopardize the proof of origin of a notarization.

**Attack AA2**: An adversary attempts to impersonate another identity by deceiving the registration phase.

**Attack AA3:** An adversary attempts to impersonate another identity by stealing everything the legal notarization entity needs to notarize a record (by social engineering, interception, observation, endpoint compromise, guessing, of notarization creation data).

**Attack AA4**: The attacker substitutes the original notarization by a notarization generated by himself on the same record, compromising the proof of origin.

**Attack AA5:** The attacker tries to attribute the notarization of a record to a victim notarization entity.

**Attack AA6:** The master, playing the role of the attacker, tries to simulate the notarization of a record by one of its employees.

**Security Property 2 (SP2)- *Record Integrity.*** SP2 is defined as follows: *The binary content of a notarized record should be always verified as exactly equal to the original record.*

The attack model we consider to describe how this property can be threatened is the following:

**Attack AI1**: An adversary tries to forge a valid notarization starting from a notarization of the victim (i.e., selective forgery attack).

**Attack AI2**: An adversary tries to forge a new record and, accordingly, the associated notarization of a victim notarization entity (i.e., existential forgery attack).

**Security Property 3 (SP3) - *Notarization Verification Dependability***. SP3 is defined as follows: *The notarization validity verification should be always dependable*.

The attack model we consider to describe how this property can be threatened is the following:

**Attack AD1**: An adversary tries to repudiate its social network account or its use, thus invalidating the notarization validity verification.

**Attack AD2**: An adversary tries to repudiate a notarization by illegal operations (w.r.t. the notarization generation protocol)on his timeline.

**Attack AD3:** An adversary tries to repudiate a notarization by jeopardizing the revocation system.

**Attack AD4:** An external adversary (different from the notarization entity) tries to make a valid notarization generated by a certain notarization entity be verified as invalid.

**Attack AD5:** An adversary tries to make a notarization generated over a fraudulently modified record be verified as valid.

**Attack AD6:** The master, playing as an adversary, tries to make invalid valid notarization verification by illegal operations (w.r.t. the notarization generation protocol) on his timeline.

*B. Security Analysis*

In this section, we analyze the practical security of our notarization system. We consider separately all the security properties we have to guarantee, which is notarization origin, record integrity, and notarization verification dependability, as stated in the security model presented in the previous section. Even though multiple notarization entities of a given record might occur, our security analysis

focuses only on the case of a single notarization entity, without loss of generality. Indeed, it is easy to realize that the collusion of notarization entities cannot give any advantage. The following theorem states that the protocol satisfies the property SP1 of the security model $SM$.

*Theorem 4.1 (SP1)*: The protocol is secure against attacksAA1, AA2, AA3, AA4, AA5, and AA6.

**Proof.**

*Resistance to Attack AA1*. Recall that AA1 occurs whenever an adversary tries to use a fake social network profile to jeopardize the proof of origin of a notarization of a legal notarization entity. In particular, the attacker creates profile with the victim's real-life information and, then, tries to impersonate the victim. Let $\langle y, h(D), ID_S \rangle$ be afake notarization message whose aim is to attribute thenotarization of the record $D$ to the notarization entity $S$. Thisattack is vanished by the fact that the notarization generationprotocol requires that the notarization message is linked, to respond the value `valid`. But, thanks to Assumptions **A6**and **A8**, this cannot happen.

*Resistance to Attack AA2*. Recall that AA2 occurs whenever an adversary attempts to impersonate another identity by deceiving the registration phase. Since the social network is trusted (Assumption **A2**) and the master does not collude with any notarization entity (Assumption **A9**), due to Assumption **A5**, the attack cannot be performed.

*Resistance to Attack AA3*. Recall that AA3 occurs when ever an adversary attempts to impersonate another identity by stealing everything the legal notarization entity needs to notarize a record (by social engineering, interception, observation, endpoint compromise, guessing of notarization creation data).The security of our protocol is given by the procedure of authentication which offers the level of assurance 4 of the NIST model (Assumption **A4**).

*Resistance to Attack AA4.* Recall that this attack occurs whenever the attacker substitutes the original notarization by a notarization generated by himself on the same record, trying to compromise the proof of origin. The only way to accomplish this for the attacker is to remove the original notarization message from the victim's timeline and the confirmation message from the master timeline and to perform anew notarization on his social network profile. However, our system is resistant to this attack, according to Assumptions**A3** and **A4**.

*Resistance to Attack AA5.* Recall that this attack occurs whenever the attacker tries to attribute the notarization of a record to a victim notarization entity B. This attack can be performed in three modes:

1) In the first mode, the master $M$ plays the role ofthe attacker. In this case, the master forges a message$\langle y, h(D), ID_S \rangle$, thus trying to attribute the notarization of $D$ to$S$. The notarization verification protocol falls into the case 2(a) i, since $\langle y, h(D), ID_S \rangle$cannot befound in $S$, but it exists in$M$. However, to return a valid response, it requires that the message is linked in $S$.Since $y$ must be the preimage of $y_i$(where $y_i$ belongsto the latest notarization message of$S$) of the trapdoorfunction whose secret is kept only by$S$, accordingto Assumption **A8**, our protocol is secure against this attack.
2) Also in the second mode, the master plays the role of the attacker. Suppose that a notarization entity $A$ generatesa linked notarization message $\langle y, h(D), ID_A \rangle$ tonotarize the record$D$. According to the notarization generation protocol (see item 1), the master should replicate this message in its timeline. Instead, the master posts the message$\langle y, h(D), ID_B \rangle$, thus tryingto attribute the notarization of $D$ to $B$. The notarizationverification protocol falls into the case 2, since$\langle y, h(D), ID_B \rangle$cannot be found in $B$. But, to return a valid response, it requires that the message is linked in$B$. This cannot happen because $y$ belongs to the chainof$A$, thus it is linked in $A$ but not in $B$. Therefore,the notarization verification protocol falls (1) into thecase 2b, thus returning `(invalid, unsafe)` w.r.t.$B$, and (2) into the case 2(a)i,thus returning `(valid,unsafe)` w.r.t. $A$.
3) In the third mode, the attacker is a notarization entity$A$. He just posts a linked (in $A$) notarization message$\langle y, h(D), ID_B \rangle$thus trying to attribute the notarizationof $D$ to $B$. According to the notarization generationprotocol, the master does not publish the confirmationmessage because the notarization message is linked in$A$ but not in $B$. Moreover, the notarization verificationprotocol falls into the case 2b, because the notarizationmessage is not linked in $B$ thus returning `(invalid, unsafe)` w.r.t. $B$. The attack is then contrasted.

*Resistance to Attack AA6.* Recall that this attack occurswhenever the master (playing here the role of the attacker)tries to simulate the notarization of a record by one of itsemployees. To do this, the master can proceed in two modes(naive and enhanced).

1) In the naive mode, the master posts in its timeline afresh message $\langle y, h(D), ID_s \rangle$such that it is linked in $S$. If the master is able to do this, the attack succeeds becausethe notarization verification protocol would return `(valid,unsafe)`, according to item 2(a)i (i.e., simulating that$S$ has deleted the notarization message to repudiate thenotarization itself). However, this attack would require theknowledge of the secret owned by $S$, in

order to compute $y$,i.e., the preimage of $y_i$, (where $y_i$belongs to the latest notarizationmessage of $S$) of the trap-door function. Accordingto Assumption **A8**, our protocol is secure against this attack.

2) In the enhanced mode, the master deletes (or does notpublish) a confirmation message, say$\langle y, h(D), ID_S\rangle$and forges a fake confirmation message, say$\langle y, h(D'), ID_B\rangle$, where $D' \neq D$ to falsely attribute thenotarization of the record $D'$ to $S$. Indeed, in this case,$\langle y, h(D'), ID_S\rangle$, is linked in $S$. However, the timestamp ofthe forged confirmation message is more recent than themessage $\langle y, h(D), ID_S\rangle$, occurring in the timeline of S. Thus, according toitem 2(a)ii of the notarization verification protocol, theverification procedure outputs `(valid, unsafe)` on $D$and `(invalid, unsafe)` on $D'$. Thus, the attack failson both records.

Now, by means of the following theorem, we state thatthe protocol satisfies the property SP2 of the security model$SM$.

*Theorem 4.2 (Record Integrity):*The protocol is secure against attacks AI1 and AI2.

**Proof**.

*Resistance to Attack AI1.* Recall that this attackoccurs whenever an adversary tries to forge a valid notarizationstarting from a notarization of the victim (i.e.,selective forgery attack). It is easy to see that, thanks toAssumption **A6**, only the master could try this attack becausea valid notarization message is always linked in the legalnotarization entity. Suppose now the attacker is the master.In this case, it tries to forge a linked notarization message$\langle y', h(D'), ID_S\rangle$, starting from a linked notarization message$\langle y, h(D), ID_S\rangle$ published by the victim $S$. The aim of theattack is to attribute to $S$ the notarization of the record $D'$.Due to Assumption **A8**, such an attack is impossible.

*Resistance to Attack AI2.* Recall that this attack occurswhenever an adversary tries to forge a new record and, accordingly,the associated notarization of a victim notarizationentity (i.e., existential forgery attack). As the binary outputof a notarization creation is just the digest of the record,an existential forgery attack in a strict sense can be alwayssuccessfully done. However, this digest plays the role ofnotarization only if it is issued as a notarization messageby the notarization entity (and the rest of the protocol istriggered). As a consequence, the practical security againstthis attack is guaranteed by Theorem 4.1.

The following theorem states that the protocol satisfiesthe property SP3 of the security model $SM$.

*Theorem 4.3 (Notarization Verification Dependability*):The protocol is secure against attacks AD1, AD2, AD3,AD4, AD5, and AD6.

**Proof.**

*Resistance to Attack AD1.* Recall that this attackoccurs whenever an adversary tries to repudiate its socialnetwork account or its use, thus invalidating the notarizationvalidity verification. The repudiation of the social networkaccount can be done only by doubting about the securityof the registration phase. According to Assumption **A5**, oursystem is secure against this attack. The repudiation of thesocial-network-account use can be done only by claimingthe violation of the account. However, due to Assumptions**A3** and **A4** we can conclude that our system is secure againstthis attack.

*Resistance to Attack AD2.* This attack may be performed inthree different modes.

1) The first mode occurs whenever an adversary tries torepudiate a notarization by deleting the corresponding notarizationmessage by his timeline. According to our protocol,whenever the notarization entity issues a notarization messagein his social network account, the master generates theconfirmation message on his social network page providedthat the notarization message is linked, by including thesame information as the notarization message. Accordingto Assumptions **A3** and **A9**, the notarization verificationprotocol will output `(valid, unsafe)` because we fall into the case 2(a)i.Therefore, our system is secure against this attack.
2) The second mode occurs whenever an adversary tries torepudiate a notarization by generating a notarization messagenot linked, that is a message $\langle y_i, h(D_i), ID_S\rangle$such that the iterated self-composition of $f$ to $y_i$ does not allow us to reach$y_0$, where $\langle y_0, I, ID_S\rangle$ is the first message posted by $S$ atthe registration phase. The aim of the attacker is to triggerthe confirmation message publication by the master. Indeed,according to 1 of the notarization verification protocol, in this case, the output of the verification protocol would be`(valid, safe)`, so a possible obligation about the record$D_i$ would seem correctly satisfied to any recipient. In asecond step, the plan of the attacker is to delete the message$\langle y_i, h(D_i), ID_S\rangle$from his timeline, in order to enforcethe verification protocol to check whether the confirmationmessage $\langle y_i, h(D_i), ID_S\rangle$is linked or not in $S$ (see 2 ofthe notarization verification protocol) and thus to respond`(invalid, unsafe)`. This way, the repudiation of thenotarization on $D_i$ would succeed. However, the notarizationgeneration protocol enforces the master to verify that themessage $\langle y_i, h(D_i), ID_S\rangle$is linked in $S$, before confirmingit (see 1 of the notarization generation protocol). Thus,thanks to Assumption

**A6** (ensuring that a fake chain sourcecannot be simulated by the notarization entity), this attackis not possible.

3) The third mode leverages the latency of the confirmationmessage. Let $\langle y, h(D), ID_S\rangle$be the message correspondingto the notarization the attacker tries to repudiate and supposethat this message is linked in $S$. To do this, immediately afterthe previous message, the attacker publishes a fake message$\langle y, h(D'), ID_S\rangle$. When she/he wants to repudiate the notarizationon $D$, she/he deletes the message $\langle y, h(D'), ID_S\rangle$.Observe that the attack succeeds only if the notarizationon $D$ appears valid until it is repudiated (thus apparentlysatisfying the related obligation). The aim of the attacker isto jeopardize the recipient-side check based on timestampsdescribed in item 2(a)ii of the notarization verification protocol,by trying to obtain that the timestamp of the confirmationmessage associated with $\langle y, h(D), ID_S\rangle$is more recentthan the timestamp of $\langle y, h(D'), ID_S\rangle$. Indeed, in this case,the check described in 2(a)ii of the notarization verificationprotocol would output `(valid, unsafe)` on $D'$ (since thenotarization message is linked) and `(invalid, unsafe)` on $D$,thus allowing the repudiation of the notarization on $D$.However, the above situation cannot hold. Indeed, threecases may occur. The first case is that the attacker is able topublish the second message $\langle y, h(D'), ID_S\rangle$within $\Delta t$ fromthe timestamp of the message$\langle y, h(D), ID_S\rangle$ (recall that $\Delta t$is the reaction time constraint defined in Section III). In thiscase, according to option 2 of the notarization generationprotocol, the master publishes an aborting message for $y$.Therefore, according to item 2(a)i, the notarization verificationprotocol returns `(invalid, unsafe)` for both records (sincethey are both inside the interval $\Delta t$), thus inhibiting thenotarization on $D$ and thus vanishing the plan of the attacker.The second case is that the attacker publishes the secondmessage $\langle y, h(D'), ID_S\rangle$after $\Delta t$ from the timestamp of themessage $\langle y, h(D), ID_S\rangle$. In this case, according to item 1 ofthe notarization generation protocol, a confirmation message$\langle y, h(D), ID_S\rangle$ has been published by $M$ before$\langle y, h(D'), ID_S\rangle$, due to the reaction time constraint. Consequently,the notarization verification protocol will return `(valid, unsafe)`, as explained in item 2(a)ii. The repudiationcannot be done. The third case is that the scheduleof the posting-reaction sequence is as follows. First, theattacker publishes the message $\langle y, h(D), ID_S\rangle$ Within $\Delta t$,the master generates the confirmation for this message and,in the meanwhile, the attacker publishes the second message$\langle y, h(D'), ID_S\rangle$. This would trigger the publication by themaster of an aborting message for $y$ thus falling into thefirst case above.

*Resistance to Attack AD3.* Recall that this attack occurswhenever an adversary tries to repudiate a notarization byjeopardizing the revocation system. Specifically, the adversary,after notarizing a record, performs a revocation requestto claim that the revocation occurred before the notarization.As both notarization, confirmation, and revocation messagesinclude a trusted issuing time, the security against this attackis guaranteed by Assumption **A3**.

*Resistance to Attack AD4.* Recall that this attack occurswhenever an external adversary (different from the notarizationentity) tries to make a valid notarization generated bya certain notarization entity be verified as invalid. To dothis, the attacker should remove the notarization messagefrom the victim's social network page and the confirmationmessage from the master social network page. According toAssumptions **A3** and **A4**, our system is secure against thisattack.

*Resistance to Attack AD5.* Recall that this attack occurswhenever an adversary tries to make a notarization generatedover a fraudulently modified record be verified as valid. Todo this, the attacker should inject a notarization messageinto the victim's social network page or the correspondingconfirmation message into the master social network page.
According to Assumptions **A3** and **A4**, our system is secureagainst this attack.

*Resistance to Attack AD6.* Recall that this attack occurswhenever the master, playing as an adversary, tries to makeinvalid a valid notarization verification by illegal operations(w.r.t. the notarization generation protocol) on his timeline.This attack can be performed in three modes:

1) The first mode is tried by the master by deleting theconfirmation message associated with a linked notarizationmessage $\langle y, h(D), ID_S\rangle$. This attack fails because the notarization verification protocol returns the output `(valid,unsafe)`, according to item 2(a)i.
2) The second mode is tried by the master by deleting all theconfirmation messages successive to the linked notarizationmessage of the victim $\langle y, h(D), ID_S\rangle$and, then, by postingan aborting message for y. This attack fails because thereaction time constraint will be not satisfied for the abortingmessage, as described in the case 2(a)ii of the notarizationverification protocol. Indeed, the

output returned by theverification protocol is `(valid, unsafe)`.

3) The third mode to be considered is a fake revocationmessage. This attack cannot succeed thanks to Assumption**A7**.

## PERFORMANCE EVALUATION

Through this section, we perform an experimental validation of the proposed solution. In particular, we analyse thetime needed to execute the registration protocol, notarizationgeneration protocol, revocation protocol, and notarizationverification protocol.

To do this, we developed a web application in JAVA thatperforms the steps of the four protocols and interacts withthe popular social network Twitter. The experiments areperformed on a personal computer equipped with 1.8GHzIntel i7-8850 CPU and 16GB of RAM.

To interact with Twitter and to obtain a realistic estimationof the execution time, we rely on the TwitterAccount Activity APIs. These webhook-based APIs send anotification message, each time an event occurs, to a webapplication we have developed. In our setting, the event isthe reception of a Direct Message and the publication ofa Tweet. To receive webhook events, we need to registera public URL of our web application. As suggested bythe Twitter Documentation, to test locally our application,we can use *ngrok*that allows us to create an https tunnel(required by the Twitter Account Activity) and redirectsevery request to the local port where our web applicationis running. This inter-mediation would be not necessary ina real-life implementation of our protocol. Therefore, weexpect that real performances are also better than thoseevaluated in this section.

We start from the registration protocol, it simply consistsin posting two Tweets. The first (registration message) by thenotarization Entity and the second (welcome message) by theNotarization Master. We performed several tests at differenthours of the day and the time to post a Tweet ranging from100 to 200 ms. Therefore, the registration protocol requiresan overall time of 200-400 ms.

Regarding the notarization generation protocol, it consistsin the computation of the digest of a file and the postingof two Tweets (the notarization message and the confirmation/aborting message). Moreover, to notify the notarizationmaster to post the confirmation message, we use a directmessage. As a hash function, we use SHA256 and compute the digest of a file with a size ranging from 10MB to 100MB.The time to compute this hash function varies from 40 ms(10MB) to 400 ms (100MB). Finally, the time to send thedirect message is about 200-300 ms. Then, the overall timeranges from 640 to 1100 ms.

The revocation protocol consists simply in publishing apost with an overall time of 100-200 ms (We neglect thetime to remove the friendship).

Finally, regarding the Notarization protocol, it consistsagain in computing the digest of the file and searching theTweets of the notarization entities. We considering the timeto search a Tweet for a given hashtag. This time ranges from1400 to 2000 ms. The overall time will be 1440-2400 ms.

Table 1 summarizes the above results.

| Protocol | Operation | Time of Operation | Overall Time |
|---|---|---|---|
| Registration Protocol | Registration Message | 100-200 ms | 200-400 ms |
| | Welcome Message | 100-200 ms | |
| Notarization Generation Protocol | Digest Computation | 40-400 ms | 440-1100 ms |
| | Notarization Message | 100-200 ms | |
| | Confirmation/ Aborting Message | 100-200 ms | |
| | Notification to the Notarization Master | 200-300 ms | |
| Revocation Protocol | Revocation Message | 100-200 ms | 100-200ms |
| Notarization Verification Protocol | Digest Computation | 40-400 ms | 1440-2400 ms |
| | Post retrieval | 1400-2000 ms | |

Table 1. Times overhead of the protocol

## DISCUSSION AND CONCLUSION

The concept of data notarization is more and more importantin a society moving towards a complete dematerializationof documents, transactions, and workflows, bothin the public sector and in business. The protocol we havepresented in this paper is designed by keeping in mindthe above dynamics and starting from the considerationthat the most elective technology used in the last years fornotarization could be not adequate in certain cases.

Therefore, we tried in this paper to address the problemof data notarization without relying on blockchain, with thepurpose of understanding if some realistic alternatives exist.We found an answer in the domain of social networks, withtheir power of sharing information among people. As inother non-native applications of social

networks (which areborn for sociality), we exploit what social networks offerfor free to play strategic functions of organizations and communities in a secure, cheap, and reliable way. This paper, then, follows the path traced by the scientific literature ofthe last years, for which social networks can become partof people workflows, with reciprocal advantages: for peoplehigh usability, low cost, high availability, for social networkproviders a more central role in the society.

In this paper, we tried to apply the above paradigm todata notarization. We addressed the case in which an entityexists able to identify and register notarization entities atthe initial stage, with no limit in terms of its dimension (inprinciple, it could be also an entire Country, if we merge ourprotocol with national digital identity solutions [28], [29]).The scalability of our solutions arises from two facts. Thefirst is that social networks, by definition, provide scalableservices. The second is that our protocol does not makeuse of public-key cryptography, but only cryptographic hashfunctions (which are inherently efficient) and other simpleoperations. The overhead computation for the notarizationmaster is thus feasible. In other words, the pure adoptionof social networks in a data notarization framework usingpublic-key cryptography would be much less realistic thanour solution. Another nice feature of our social-network-baseddata notarization is that even though notarization entitiesmust be previously identified and registered, recipientscan be everywhere and anyone in the world, thanks to thepervasivity of social networks. Indeed, the verification ofnotarizations leverages only public information available ona social network and easily searchable. The solution is alsoprivacy-preserving, in the sense that the universal exposureof notarizations does not affect the privacy of notarizedrecords, as only record digests are published.

Concerning the aspect of privacy, it is worth noting thatwe consider the case in which it is not a privacy threatthat an adversary may discover that a given record has beennotarized by a certain notarization entity [30]. Therefore, thisattack is not contrasted in any case. Consider for examplethe case of records that can be guessed by the adversary.However, the current version of our protocol is thought fortruly sharing environments, or cases in which dictionary-basedattacks are not feasible. If the above conditions do notoccur, we just have to modify our notarization protocol byincluding some salting-based techniques in the computationof digests. In this case, the price we have to pay is that theset of recipients of a notarized information must be prefixed(only those subjects equipped with the information neededto detect the salts). Observe that this is not a limitation. Itis an intrinsic feature because any recipient which is ableto verify a notarized record can perform the attack above,independently of the technique used for notarizations. Alsotiming attacks (based on deadlines, periodic notarizations,etc.) to discover if an even salted record has been notarizedby a certain notarization entity can be contrasted by generatingdummy traffic in the timelines of notarization entities.The implementation of this extension is the subject of ourfuture work.

Our protocol provides notarization entities with anothergood feature. Indeed, our protocol supports implicitly a notarization timestamp service that is secure and enforceableagainst thirdparties, with no extra cost. Another strong pointof our notarization is that multiple notarizations on a singlerecord can be implemented in a very easy and flexible way,with no need for planned exchanges of the record beingnotarized.

The last nice feature we want to highlight is that oursolution is proven to be at least secure in a security model inwhich only realistic assumptions are done and everyone (butthe social network provider) can be the adversary (includingthe notarization master). The complexity of the security analysis(together with the high granularity of our notarizationverification procedure) shows also that the implementationof the apparently trivial starting idea of publishing digests ofnotarized records opens a number of security issues, whichcan be addressed, leading to a definitely non-trivial solution.

Concerning security, it may appear that the assumption oftrustworthiness of the social network provider is too strongand poses our solution in a weaker position with respectto permissioned blockchains (which are another possiblealternative to public blockchain if we want to avoid theeconomic cost of the transactions). Indeed, in the case ofpermissioned blockchains, security relies on the assumptionsthat the validation of blocks is done by different non-colludingindependent parties. This is at the basis of thecomplexity of the organization of a real-life permissioned-blockchain-based solution, not only in the field of datanotarization. However, in our solution, we can easily weakenthe above trustworthiness assumption just by replicatingthe same mechanism in different social networks. In thiscase, the collusion among multiple social network providerswould be necessary to tamper with the notarization process.As a matter of fact, this event can be considered not realistic,also by considering that social network providers are partiesindependent with respect to the parties involved in the datanotarization process and that they do not have any advantageto risk their reputation and also legal consequences.

As a final remark, we highlight that the fact that ourprotocol is stateful cannot be considered a real drawback.Indeed, the state is public and published by the socialnetwork, so efficiently accessible by any party. Interestingly,there is a renewed attention towards stateful notarizationschemes in the recent literature: Indeed, the statefulness of protocols is proven to be a necessary condition for encryption schemes to obtain solution resistant to algorithmsubstitution attacks (ASAs). On the other hand, also theblockchain-based solutions are clearly stateful, and even the standard PKI-based signature process enforces both the signature entity and the recipient to access and check the

status of the certificate via Online Certificate Status Protocol (OCSP) or by downloading CRLs, even though the cipher used to generate and to verify notarizations (e.g., RSA) isstateless.

## ACKNOWLEDGMENT

This paper is partially supported by Project POR FESR/FSE Line B (Action 10.5.12).

## REFERENCES

[1] AF. Carlini, R. Carlini, S. Dalla Palma, R. Pareschi, and. Zappone, "The genesy model for a blockchain-based fair ecosystem of genomic data," Frontiers in Blockchain, vol. 3,p. 57, 2020.

[2] A. Andrianov and B. Kaganov, "Blockchain in clinical trialstheultimate data notary," AppliedClinical Trials, vol. 27, no.7/8, pp. 16–19, 2018.

[3] G. Song, S. Kim, H. Hwang, and K. Lee, "Blockchain-basednotarization for social media," in 2019 IEEE internationalconference on consumer electronics (icce). IEEE, 2019, pp.1–2.

[4] C. Sullivan and E. Burger, "E-residency and blockchain,"computer law & security review, vol. 33, no. 4, pp. 470–481,2017.

[5] N. Nizamuddin, H. R. Hasan, and K. Salah, "Ipfs-blockchain-basedauthenticity of online publications," in InternationalConference on Blockchain. Springer, 2018, pp. 199–212.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system,"Manubot, Tech. Rep., 2019.

[7] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang,"Blockchain challenges and opportunities: A survey," InternationalJournal of Web and Grid Services, vol. 14, no. 4,pp. 352–375, 2018.

[8] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman et al., "Blockchain technology: Beyond bitcoin," Applied Innovation,vol. 2, no. 6-10, p. 71, 2016.

[9] S. Underwood, "Blockchain beyond bitcoin," Communicationsof the ACM, vol. 59, no. 11, pp. 15–17, 2016.

[10] A. Meneghetti, A. O. Quintavalle, M. Sala, and A. Tomasi, "Two-tier blockchain timestamped notarization with incrementalsecurity," Annals of Emerging Technologies in Computing(AETiC), Print ISSN, pp. 2516–0281, 2019.

[11] V. Balakrishnan, "Using social networks to enhance teachingand learning experiences in higher learning institutions,"Innovations in Education and Teaching International, vol. 51,no. 6, pp. 595–606, 2014.

[12] H. T. Hoi, "Using social networks for english teaching andlearning," in Proceedings of the 2019 2nd Artificial Intelligenceand Cloud Computing Conference, 2019, pp. 173–177.

[13] P. T. Jaeger, B. Shneiderman, K. R. Fleischmann,J. Preece, Y. Qu, and P. Fei Wu, "Communityresponse grids: E-government, social networks, andeffective emergency management," TelecommunicationsPolicy, vol. 31, no. 10, pp. 592–604, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308596107000699

[14] F. Buccafurri, L. Fotia, G. Lax, and V. Saraswat, "Analysis-preservingprotection of user privacy against informationleakage of social-network likes," Information Sciences, vol.328, pp. 340–358, 2016.

[15] J. C. Anderson and M. L. Closen, "Document authenticationin electronic commerce: the misleading notary public analogfor the digital signature certification authority," J. Marshall J.Computer & Info. L., vol. 17, p. 833, 1998.

[16] E. S. Pasqual, J. Dias, and R. F. Cust´odio, "A new methodfor digital time-stamping of electronic document," moment,vol. 9, no. 8, p. 11, 2002.

[17] M. Ruhl, M. Bern, and D. Goldberg, "Secure notarization ofpaper text documents," in Symposium on Discrete Algorithms:Proceedings of the twelfth annual ACM-SIAM symposium onDiscrete algorithms, vol. 7, no. 09, 2001, pp. 437–438.

[18] D. Lekkas and D. Gritzalis, "Cumulative notarization forlong-term preservation of digital signatures," Computers &Security, vol. 23, no. 5, pp. 413–424, 2004.

[19] J. Decker, "The e-notarizationinitiative, pennsylvania, usa,"Digital Evidence&Elec. Signature L. Rev., vol. 5, p. 73,2008.

[20] P. Mytis-Gkometh, G. Drosatos, P. Efraimidis, andE. Kaldoudi, "Notarization of knowledge retrieval frombiomedical repositories using blockchain technology,"in International Conference on Biomedical and HealthInformatics. Springer, 2017, pp. 69–73.

[21] A.-S. Kleinaki, P. Mytis-Gkometh, G. Drosatos, P. S.Efraimidis, and E. Kaldoudi, "A blockchain-based notarizationservice for biomedical knowledge retrieval," Computationaland structural biotechnology journal, vol. 16, pp. 288–297, 2018.

[22] M. J. M. Chowdhury, A. Colman, M. A. Kabir, J. Han, andP. Sarda, "Blockchain as a notarization service for data sharingwith personal data store," in 2018 17th ieee internationalconference on trust, security and privacy in computing andcommunications/12th ieee international conference on bigdata science and engineering (TrustCom/BigDataSE). IEEE,2018, pp. 1330–1335.

[23] H. Magrahi, N. Omrane, O. Senot, and R. Jaziri, "Nfb: Aprotocol for notarizing files over the blockchain," in 2018 9thIFIP International Conference on New Technologies, Mobilityand Security (NTMS). IEEE, 2018, pp. 1–4.

[24] O. Jacobovitz, "Blockchain for identity management," TheLynne and William Frankel Center for Computer ScienceDepartment of Computer Science. Ben-Gurion University,BeerSheva, 2016.

[25] G. Prisco, "Estonian government partners with bitnation tooffer blockchain notarization services to e-residents, bitcoinmagazine. url:https://bitcoinmagazine.com/articles/estoniangovernment-partners-with-bitnation-to-offer-blockchainnotarization-services-to-e-residents-1448915243," 2015.

[26] M. T. Goodrich, R. Tamassia, and D. D. Yao, "Notarizedfederated id management and authentication," Journal ofComputer Security, vol. 16, no. 4, pp. 399–418, 2008.

[27] W. E. Burr, D. F. Dodson, and W. T. Polk, Electronicauthentication guideline. NIST 800-63-2, 2006.

[28] "Electronic identification and trust services (eIDAS),"http://ec.europa.eu/dgs/connect/en/content/electronicidentification-and-trust-services-eidas-regulatoryenvironment-and-beyond.

[29] "Spid-agenzia per l'italia digitale,"http://www.agid.gov.it/sites/default/files/regole tecniche/spidregole tecniche v0 1.pdf, 2016.

[30] B. Al Bouna, E. J. Raad, R. Chbeir, C. Elia, and R. Haraty,"Anonymizing multimedia documents," World Wide Web,vol. 19, no. 1, pp. 135–155, 2016.

## AUTHOR INFORMATION

**Francesco Buccafurri,** Full professor of computer science at the University Mediterranea of Reggio Calabria, Italy. In 1995 he took the PhD degree in computer science at the University of Calabria (Italy). In 1996, he was visitingresearcher at the database and knowledge representation group of Vienna University of Technology.His research interests include information security and privacy,social network analysis, deductive-databases, knowledge representation and non-monotonic reasoning,model checking, data compression, data streams, agents, P2P systems.He has published several papers in top-level international journals and conference proceedings.He serves as a referee for international journals and he is a member of a number of conference PCs.He is Associate Editor of Information Sciences (Elsevier), he is also included in the editorial boardof a number of international journals and played the role of PC chair in a number of international conferences.

**De Angelis Vincenzo,** PhD student in information engineering at the University Mediterranea of Reggio Calabria, Italy. He received the BS degree in information engineering and the Master's degree in telecommunication engineering in 2017 and 2019, respectively. His research interests include information security, blockchain and cloud. He is author of a number of papers published in international journal and conference proceedings.

**Lax Gianluca,** Associate professor of computer science at the University Mediterranea of Reggio Calabria, Italy. He got the habilitation as Full Professor of computer science by the Italian National Scientific Qualification Procedure (2018). His research interests include information security and social network analysis. He is author of more than 100 papers published in top-level international journals and conference proceedings. He serves as a referee for many international journals and is in the program committee of many conferences. He is also included in the editorial board of several international journals and participates in several funded projects.