This is the post-print of the following article: G. Lax and A. Russo, "A Lightweight Scheme Exploiting Social Networks for Data Minimization According to the GDPR," in IEEE Transactions on Computational Social Systems, vol. 8, no. 2, pp. 388-397, April 2021, doi: 10.1109/TCSS.2020.3049009. Article has been published in final form at: ieeexplore.ieee.org/document/9330798
© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

A lightweight scheme exploiting social networks for data minimization according to the GDPR

Gianluca Lax, *Member, IEEE* and Antonia Russo *Member, IEEE Young Professionals Committee*
E-mail: {lax, antonia.russo}@unirc.it

***Abstract*—In many application domains, there is a need to ensure that users satisfy some requirements to use a service: for example, there is a minimum age to buy alcoholic beverages or to watch some videos on YouTube. In these situations, organizations typically collect more personal information than the necessary to provide a better service. The consequence is a personal data leakage that violates the data minimization principle stated by the General Data Protection Regulation 2016/679. This paper proposes a new approach for allowing individuals to maintain control over the disclosure of their data, deciding which information to disclose and for how long. Our approach is based on the use of social networks, and an implementation on Facebook is presented to show that the proposed solution is effective, cheap, friendly, and simple to adopt.**

***Index Terms*—Access control, authentication, privacy, eIDAS Regulation, General Data Protection Regulation.**

I. Introduction

NOWADAYS, the number of accessible online services is growing considerably and users have become the major players in the process of information exchange among parties. Often, when accessing an online service, users must perform authentication to prove their real identities. However, in some cases, the grant of a service could be based on the disclosure of only certain subject's characteristics [1]. Consider an online user who wants to access a media content reserved for subjects in possession of specific attributes, such as to be of age. Commonly, the user must fill in fields that contain also unnecessary information, such as name, surname, and nationality, and this results in a leakage of personal data. Moreover, users often are not sufficiently aware of the treatment of their data and ignore their privacy rights [2, 3].

With reference to the above example, the problem we address in this paper is finding a solution that guarantees two properties: 1) only adult people should access this content (access control) and 2) the service provider should know only that the accessing user is adult and not any further information (minimization). To remark the importance of this problem, we observe that one of the ten principles of *self-sovereign identity* [4] stands for the minimization of disclosed information, that is, the disclosure should involve the minimum and necessary amount of data. Recently, this principle has been claimed with the issuance of the General Data Protection Regulation (GDPR) [5], which makes data minimization a relevant goal in accessing online services. Most of the proposed solutions to this problem are based on blockchain technology, whose main advantage is that saved information is distributed and cannot get lost. However, if private information is lost, data on blockchain cannot be removed, modified, or hidden [6]. As observed in [7], blockchain technology is a good foundation but it is not a necessity: this result suggests us to think of a different approach.

In this paper, we propose a new solution based on the use of a social network in charge of providing users with the means of issuing and verifying claims and credentials. Responding to the logic of selective disclosure of attributes, a user knows and controls the information shared with the social network and with other actors, such as attribute providers or service providers. The primitive operations on which our solution is based are the exclusive-or function, a hash function, and a pseudo-random number generator: using these simple functions and with the support of an attribute provider, a user can generate a credential. A service provider can verify a credential with the help of the social network, which works as a secure and transparent repository of hidden information. Moreover, users are identified and authenticated through an eIDAS authentication [8]. We instantiated the general solution to a real-life scenario using Facebook as a social network and described the detailed data workflow to show the effectiveness of our proposal.

Our solution offers several advantages with respect to the state of the art. The first is related to user-friendliness [9], as people are favorably disposed to work in the environment of social networks, which they well know. The second advantage concerns costs, as social networks can be used typically for free. The third advantage regards scalability and availability: social networks can manage a very high number of users, and redundancy is implemented to ensure service availability. Consequently, the probability of service interruption is very low. A further advantage is about compliance with GDPR. Indeed, art. 83 of GDPR states that the supervisory authority shall impose administrative fines in respect of infringements of privacy rights and fines should be effective, proportionate, and dissuasive. The use of a technique that reduce saved data in relation to the purposes for which they are processed can provide a company with a tool to avoid administrative fines. A final observation is that the technique presented in this paper applies to various contexts for protecting user's privacy in accessing online services, like e-health applications [10, 11] or wireless sensor networks [12, 13].

The paper is organized as follows. In Section II, we define

G. Lax and A. Russo are with the DIIES Dept., University Mediterranea of Reggio Calabria, Via Graziella, Località Feo di Vito, 89122 Reggio Calabria, Italy

Manuscript received April 19, 2005; revised August 26, 2015.

TABLE I
NOTATION.

U	User
SP	Service Provider
A	Attribute
AP	Attribute Provider
IP	Identity Provider
SN	Social Network
C^X	Credential issued by X
P	Pseudo-random number generator
H	Cryptographic hash function
r	Random

the approach proposed to allows users to keep control over the personal data used for accessing services. Section III shows a running example that uses Facebook as a social network. In Section IV, we evaluate the effectiveness of the proposed solution. In Section V, we discuss how our proposal reaches the expected goals and how attacks are contrasted. In Section VI, we discuss related work and provide a comparison with the state of the art. Finally, in Section VII, we draw our conclusions.

II. Proposal description

In this section, we present the approach proposed to allow users to keep control over the personal data used for accessing services on the Web. We start by introducing the scenario and the notation.

A. Preliminaries

The scenario we consider is composed of the following typologies of entities:

- let U be the user, an individual who should own and control the personal data used to access online services;
- let SP be the service provider, a party creating and offering end-user services;
- let A be an attribute regarding a quality, a characteristic, or a competence ascribed to U;
- let AP be the Attribute Provider, an organization responsible for establishing and maintaining attributes of individuals, and issuing attribute credentials;
- let IP be the identity provider, an entity in charge of creating and managing digital identities;
- let SN be a social network.

We introduce the notation used in this paper (Table I):

- given a social network SN, we denote the following functions of SN:
  - $Send(M,U)$, which denotes the sending of the private message M to the user U;
  - $Post(T,h)$, which denotes the posting of the text T indexed by the hashtag h;
  - $Search(h)$, which returns the texts posted with hashtag h.
- we denote by C^X a credential generated by an entity X;
- let P be a pseudo-random number generator;
- let H be a cryptographic hash function;
- r denotes a random bit string.

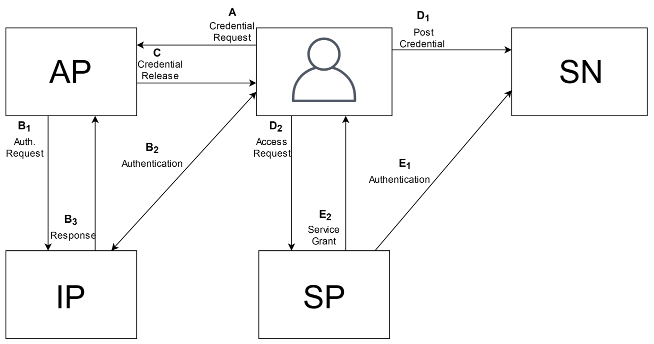


Fig. 1. Conceptual overview of our solution.

B. Proposal definition

The basic idea underlying our solution is that after the identity provider authenticates the user, the user's credential needed to access a service is composed of two parts: the attribute provider publishes the former, the user shares the latter via the social network. The two credential parts are built in such a way that each single component does not disclose any useful information (it appears like a random bit string), but it is possible to reconstruct the credential only by knowing both the parts. The attribute provider and the user can stop sharing the handled part, and this results in the revocation of the credential. Figure 1 depicts a conceptual overview of our approach.

Our proposal is built on a social network offering the three functions described above (i.e., $Send$, $Post$, and $Search(h)$). Once the social network `SN` is chosen, its reference is communicated to all users and attribute providers. We expect that both users and attribute providers have a profile in `SN`. Thus, any new user or attribute provider has to register a profile in `SN`. As this profile is created specifically to access services by revealing minimal personal data, the user does not register (true) personal data and does not give any link to his/her real identity (this is done by setting the highest privacy degree). The connection to the profile of attribute providers in `SN` is publicly available.

There is a preliminary step used to initialize the environment, where two functions are defined:

- $H(x)$ is a one-way hash function that receives an input x and returns a bit string with a fixed length. In words, we require that given a value y, it should be difficult to find any message x such that $h(x) = y$. A cryptographic hash function is an excellent candidate to implement H (for example, SHA-1, SHA-256, RIPEMD-160).
- $P(x)$ is a function that receives an input x and, depending on x, generates a sequence of bits having the same properties as a series of random numbers. A Pseudo-Random-Number generator can implement this function.

These two functions are publicly available and known by all actors.

Now, we describe the procedures carried out by the different actors to implement our approach.

A. *Credential request.* This step is performed every time a user needs to certify the possession of some attribute. Let

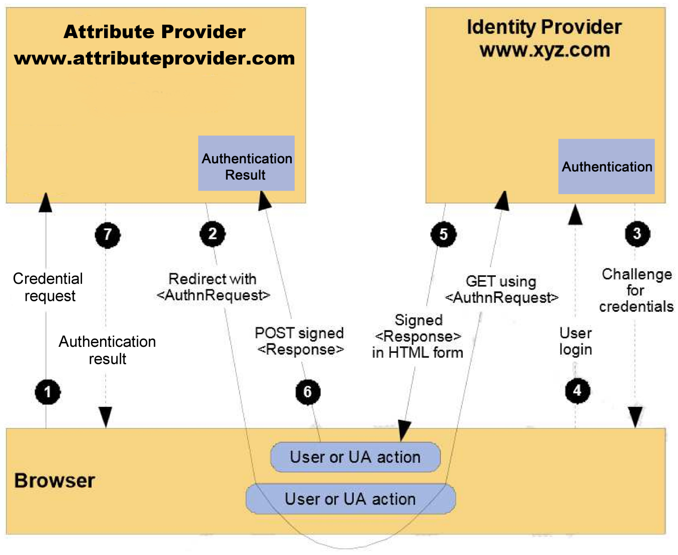


Fig. 2. Authentication process.

A be such an attribute and AP be an attribute provider in charge of certifying A. For the sake of presentation, here we assume that the credential contains only one attribute. However, in the case of more attributes, this procedure is repeated for each attribute.

The credential request is sent by a private message from the social profile of the user to that of the attribute provider (Step A of Figure 1). Specifically, the user generates a random r_1^u, calculates $r_2^u = H(r_1^u)$ where H is the cryptographic hash function, and creates the request containing:

a) the attribute of the user to be certified (i.e., A);
b) the value r_2^u.

This request is sent to AP by calling the social network function $Send(\langle A, r_2^u \rangle, AP)$.

B. *User identification.* After receiving the request, AP needs to identify the user: for this purpose, an eIDAS authentication is performed by using the eID scheme requested by the user (Step B_1). Observe that this authentication can be done by any eID scheme compliant with eIDAS: for the sake of completeness, we provide detail of this authentication process, which is schematized in Figure 2. First, the user sends the credential request to AP by using a browser named User Agent (UA) in Figure 2 (Step 1). Then, AP replies with an authentication request to be forwarded to the eIDAS identity provider declared by the user (Step 2). Now, the identity provider performs the authentication of the user by a challenge-response procedure (Steps 3 and 4), in which (typically) the user is requested to authenticate by login and password or similar mechanisms (for example, by sending an SMS to the phone number declared by the registering user and by asking to send back the text inside the SMS). In case of successful user authentication, the identity provider generates the response with the result of user authentication (Step 5), which is forwarded to AP (Step 6). Finally, AP notifies the result of the authentication procedure (Step 7), and only in case of success, the next phase is carried on. The whole authentication is summed up by the steps B_1, B_2, and B_3 of Figure 1.

C. *Credential generation (*AP *side).* The attribute credential is generated by the cooperation between AP and the user. After identifying the user, AP verifies that the requested attribute A is own by the user. If this is the case, AP prepares a credential C containing the certified characteristics, the temporal validity of the credential, and the reference to the user profile in SN (Step C).
Then, AP generates a random r^{AP} and calculates $C^{AP} = C \oplus P(r_2^u) \oplus P(r^{AP})$, where $\oplus$ is the exclusive-or (XOR) function and P is the function defined in the preliminary step. In other words, C^{AP} can be seen as C encrypted by two keys: r_2^u is chosen by the user, r^{AP} is selected by the attribute provider.
Finally, AP posts C^{AP} on its profile in SN, using $H(r_2^u)$ as an index of this post, thus calling the social network function $Post(C^{AP}, H(r_2^u))$.

D. *Credential use.* Suppose now that the user needs to access a service requiring the possession of the attribute certified by C and that this service is supplied by the service provider SP.
First, the user generates $C^U = P(r^{AP}) \oplus P(r_1^u)$ and posts C^U on his/her profile in SN (Step D_1 of Figure 1), using $H(r_1^u)$ as an index of this post (i.e., by exploiting the function $Post(C^U, H(r_1^u))$). In words, C^U can be seen as the key of the attribute provider used to encrypt C (i.e., $P(r^{AP})$) encrypted by $P(r_1^u)$.
Then, the user opens the web page of the site of SP and logins to the service by the social network account to prove to be the owner of this profile (Step D_2). Then, the user discloses r_1^u and r^{AP} to SP, which is enough to show the possession of the requested attribute, as discussed below.

E. *Credential verification.* SP needs to verify the correctness and validity of the proof presented by the user (Step E_1). Thus, it calculates $r_2^u = H(r_1^u)$ and searches for $\overline{C}^U$ and $\overline{C}^{AP}$ (they are the posts indexed by $H(r_1^u)$ and $H(r_2^u)$, respectively). This is done by calling the social network functions $\overline{C}^U = Search(H(r_1^u))$ and $\overline{C}^{AP} = Search(H(r_2^u))$. If $\overline{C}^U$ is not found in the profile of the user or $\overline{C}^{AP}$ is not found in the profile of an attribute provider, then the credential verification fails, and the service is not granted. Otherwise, SP computes $\overline{C}^U \oplus P(r_1^u) \oplus P(r^{AP})$ and verifies that this is equal to zero. In this case, SP calculates $\overline{C} = \overline{C}^U \oplus \overline{C}^{AP} \oplus P(H(r_1^u)) \oplus P(r_1^u)$ and tests that the result is a valid credential. Specifically, SP extracts from the credential the profile of the user and checks that it is equal to the profile of the user who authenticates. Then, SP extracts from the credential the certified attribute and checks that it is sufficient to access the service (Step E_2). In the positive case, the service is granted to the user. If any of the previous checks fails, the procedure is stopped, and the service access is denied.

F. *Revocation (AP side).* An important aspect concerns the credential revocation. AP should revoke a credential when

a user loses the possession of an attribute, to make unusable a credential (for example, in case of withdrawal of the driving license of a user). Another reason for credential revocation is when the secrecy of the user's password to access the social network profile is compromised.
In this case, AP can make unusable a user's credential C by merely removing the post C^{AP} from its social network profile. Clearly, if the user tries to use C, the service provider cannot find the deleted post in the profile of the attribute provider so that the verification procedure fails.

G. *Revocation (user side).* Also the user can wish to remove personal information after it has been used to access some service. Consider a patient with a disease credential, who wants to hide this information after the use. The patient can reach this goal by merely removing C^U from the profile in such a way that the credential C cannot be recovered. Clearly, excluding the case in which C^U has been copied while it is published in the profile (in this case, no solution to this problem exists), C cannot be recovered without the knowledge of C^U.

We will discuss the advantages and improvements of our solution with respect to the state of the art in Section VI.

C. Social network choice

As seen in the previous section, the proposed approach exploits some features provided by social networks, and in this sense, our solution is orthogonal to the underlying social network. Consequently, the aspect regarding how to implement our approach can be adequately addressed once the social network SN is given.

We analyzed the most known social networks concerning the requested features, which are sending a private message M to a user U (i.e., $Send(M,U)$); posting of the text T indexed by the hashtag h (i.e., $Post(T,h)$) and searching an indexed text with hashtag h (i.e., $Search(h)$).

The results of this analysis are reported in Table II, in which the social networks are classified into three categories. The first category lists some social networks that are fully compliant with our proposal because they provide the requested features. We observed that most of the social networks offer such features. For instance, Instagram users (1) can send direct messages to other users, (2) can post textual information, named *post*, and they must be authenticated to post anything; (3) can use the hashtag symbol # to categorize their post by keywords, (4) can search for posts indexed by a given hashtag typically to find a conversation about a particular topic. Observe that on Instagram, posts have to be composed of at least an image or a video.

The second category (with Twitter, Weibo, Snapchat) is composed of social networks that have some limitations in the post function. Specifically, they have a limit in the length of the text to post (e.g., this limitation is 280 characters in Twitter), and, thus, in the credential size. As a consequence, they could be used only with a limited number of certified attributes or by applying compression in the represented text.

Finally, the last category shows social networks that do not support our solution.

TABLE II
SOCIAL NETWORKS VERSUS REQUESTED FEATURES.

	$Send(M,U)$	$Post(T,h)$	$Search(h)$
Instagram	Yes	Yes	Yes
Facebook	Yes	Yes	Yes
Linkedin	Yes	Yes	Yes
VKontakte	Yes	Yes	Yes
Twitter	Yes	Yes (limited)	Yes
Weibo	Yes	Yes (limited)	Yes
Snapchat	Yes	Yes (limited)	Yes
YouTube	No	Yes	Yes
Whatsapp	Yes	No	No
WeChat	Yes	No	No
Skype	Yes	No	No
Viber	Yes	No	No
WeChat	Yes	No	No

III. RUNNING EXAMPLE

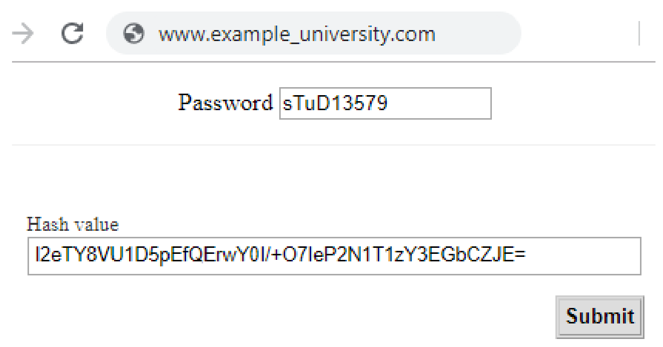
In this section, we show the application of our proposal to a real-life case in which the students of a university U play the role of users, an e-learning site that offers for free some lectures only to such students is the service provider, whereas the attribute provider is done by the university U. Our solution aims to guarantee the following two requirements: 1) the service provider should know whether a given user is a student of U (access-control requirement) and 2) the service provider does not have to know the user identity (privacy requirement).

For the implementation of our solution, among all the social networks that could be used according to the analysis reported in Section II-C, we selected Facebook because (1) it is one of the most famous one offering the required features and (2) it has been widely used for developing applications in the research context [14]. Indeed, Facebook allows the exchange of private messages between two users with an encrypted connection, the publication of stories and posts that can be indexed by hashtags, and the search for stories by a given hashtag.

The main idea underlying our approach can be summarized as follows: first, the university authenticates the user and publishes on its Facebook profile the first part of the user's credential to access the e-learning site. When the user needs to access the site, she/he publishes on Facebook the second part of the credential. Clearly, each part of the credential appears randomly generated and does not disclose any private information of the user: however, the combination of the two parts allows the e-learning site to verify whether the user is authorized to access the service. The university can revoke the user's access by removing the published part of the credential. Moreover, the user can remove her/his Facebook post after accessing the service to delete the credential.

We describe how the steps of our solution are implemented.

1) *Setup*. For the sake of presentation and without loss of generality, we implement the two functions H and P by exploiting only the SHA-256 function [15], a well-known cryptographic hash function designed by the United States National Security Agency (NSA), widely used for its robustness against attacks. Thus, we define:


www.example_university.com

Password sTuD13579

Hash value

I2eTY8VU1D5pEfQErwY0I/+O7IeP2N1T1zY3EGbCZJE=

Fig. 3. Web page used by students to send r_2^u.

- $H(x)$ =SHA-256(x);
- $P(x) = a_1, \dots, a_n$, where a_1 =SHA-256(x) and a_j =SHA-256(a_{j-1}), with $2 \leq j \leq n$.

These definitions are made available to all the students, the university, and the e-learning site.

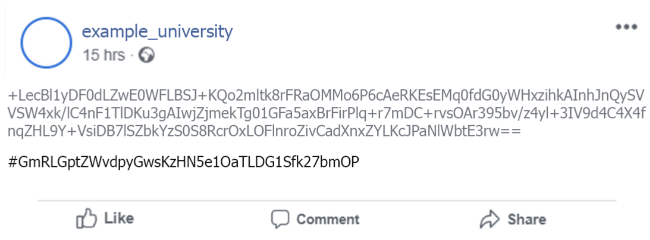
2) *OSN Registration*. In this step, the university creates and makes public its Facebook profile, which we assume to be example_university. Every student registers a Facebook profile, using information (e.g., screen-name, name, photos) unlinkable to the real identity. Moreover, the student sets the highest privacy degree, in such a way that no information is disclosed.
3) *Credential request.* The student needs to prove to be a member of the university. Consequently, the credential request is sent to the university. When the student submits the request for a credential, the university starts a Facebook login procedure. The student proves the possession of the Facebook profile, say example_student, using a single sign-on operation. After successful authentication, the student can type in a password (`sTuD13579` in our example), which will be used as the random r_1^u. Then, a javascript calculates r_2^u =SHA-256(r_1^u), which is I2eTY8VU1D5pEfQErwY0I/+O7IeP2N1T1zY3EGbCZJE= by the base64 representation [16], used to convert a bit sequence into a text (to be included in a post). This value is sent to the university server: observe that only r_2^u, which is generated by a javascript, is sent to the server, not the password, which is not included in the HTML form. This step is depicted in Figure 3.
4) *User identification.* In this step, the user is requested to prove to be a student. First, the user performs the eID-based identification. Then, he/she uses the login and password of the university site to prove to be a student of that university.
5) *Credential generation (*`AP` *side).* In case of successful authentication, the university prepares a credential C containing the certified attribute, the temporal validity of the credential, and the reference to the user profile in `SN`. There exist several ways to represent a credential, for example, by the standard SAML [17] (as done in eIDAS-compliant eIDs).
In Figure 4, we show an example of a credential represented by JSON, which is very easy to understand. The credential describes the type of attribute proved (to be a student), the name of the university, the expiration date, and the screen name of the user.
Then, the university generates a random password r^{AP} (`uNiV2468` in our example) and calculates $C^{AP} = C \oplus P(r_2^u) \oplus P(r^{AP})$, which is posted in the Facebook profile of the university and indexed by SHA-256(r_2^u) = GmRLGptZWvdpyGwsKzHN5e1OaTLDG1Sfk27bmOPMAdI=.
We depict this post in Figure 5. Observe that the password's digest is hashtagged (see the symbol # at the beginning of the second line): this post will be returned when searching for this hashtag.
6) *Credential use.* When the student needs to access the e-learning site and to prove to be a member of the university, then the student calculates and posts in the Facebook profile $C^U = P(r^{AP}) \oplus P(r_1^u)$ using $H(r_1^u)$ as a hashtag (see Figure 6).
Then, the student goes to the e-learning site and logins by Facebook (as done in Figure 3). Then, the user fills in the password used earlier (i.e., `sTuD13579`) to give the service provider the possibility to restore the credential.
7) *Credential verification.*
The service provider calculates r_2^u =SHA-256(r_1^u) and searches for the posts C_1 and C_2 indexed by I2eTY8VU1D5pEfQErwY0I/+O7IeP2N1T1zY3EGbCZJE= and GmRLGptZWvdpyGwsKzHN5e1OaTLDG1Sfk27bmOPMAdI=, respectively.
Now, the service provider calculates $C_1 \oplus C_2 \oplus P(r_2^u) \oplus P(r_1^u)$, thus obtaining the initial credential C (reported in Figure 4). Since this credential satisfies all the checks, the student receives the grant to access the e-learning course for free.
8) *Credential revocation.* When the student wants to hide the possession of this attribute, it is sufficient to remove the generated post so that the credential C cannot be recovered.
On the other hand, also the university can revoke the attribute possession, for example, in case the student leaves the university before the credential expiration time. Again, removing the post associated with this credential is sufficient to make the verification procedure fail.

IV. Quantitative Analysis

In this section, we measure some parameters related to our proposal with the aim of showing the effectiveness of our solution.

```
{
    "attribute": "student",
    "university": "example_university",
    "expiration": "30/08/2020",
    "profile": "example_student"
}
```

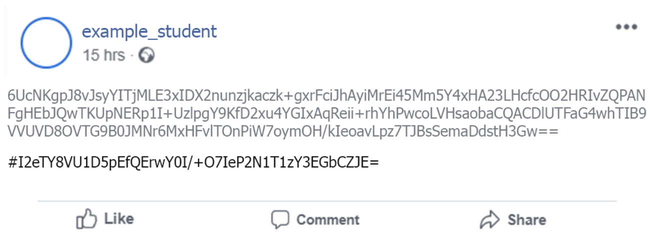
Fig. 4. Example of credential.


example_university
15 hrs ·

+LecBl1yDF0dLZwE0WFLBSJ+KQo2mltk8rFRaOMMo6P6cAeRKEsEMq0fdG0yWHxzihkAInhJnQySV
VSW4xk/lC4nF1TlDKu3gAIwjZjmekTg01GFa5axBrFirPlq+r7mDC+rvsOAr395bv/z4yl+3IV9d4C4X4f
nqZHL9Y+VsiDB7lSZbkYzS0S8RcrOxLOFlnroZivCadXnxZYLKcJPaNlWbtE3rw==

#GmRLGptZWvdpyGwsKzHN5e1OaTLDG1Sfk27bmOP

Fig. 5. Post publishing (university side).


example_student
15 hrs ·

6UcNKgpJ8vJsyYITjMLE3xIDX2nunzjkaczk+gxrFciJhAyiMrEi45Mm5Y4xHA23LHcfcOO2HRIvZQPAN
FgHEbJQwTKUpNERp1I+UzlpgY9KfD2xu4YGIxAqReii+rhYhPwcoLVHsaobaCQACDlUTFaG4whTIB9
VVUVD8OVTG9B0JMNr6MxHFvlTOnPiW7oymOH/kIeoavLpz7TJBsSemaDdstH3Gw==

#I2eTY8VU1D5pEfQErwY0I/+O7IeP2N1T1zY3EGbCZJE=

Fig. 6. Post publishing (student side).

The first parameter is efficiency. We quantify the time costs of the operations of our solution, which are: the number of requests over the network, the number of hash functions executed, and the number of pseudo-random numbers generated. Such costs are measured for every solution's step and are reported in Table III:

- Step A. In this phase, the user generates a random and calculates its hash. Then, she/he generates a connection request to the attribute provider.
- Step B. The attribute provider performs an authentication request and receives a response. Observe that we do not consider the cost of the user's authentication because it depends on the user's speed in the authentication.
- Step C. The attribute provider creates the credential by the generation of three pseudo-random numbers. Then, it calculates one hash function and sends a request to the social network.
- Step D. The user generates the credential by calculating two pseudo-random numbers and applying the hash function to one of them. Then, the user sends a request to the social network and another request to the service provider.
- Step E. The attribute provider calculates one hash function and performs two request connections to the social network. The service provider calculates four pseudo-random numbers and applies one hash function. To verify such that credential, the service provider performs a connection request to the social network. If the verification succeeds, the service provider connects to the user.
- Step F. The attribute provider sends a request to the social network.
- Step G. The user sends a request to the social network.

Observe that Steps A, B, and C refer to operations that are done by users preliminarily (i.e., before they access a service), whereas Steps D and E are performed by a user and a service provider to grant a service. In contrast, Steps F and G refer to an infrequent operation (revocation). Thus, we focus our attention on Steps D and E, which are the most important operations related to service access; moreover, as shown in Table III, their time cost is the highest.

TABLE III
NUMBER OF OPERATIONS FOR EACH SOLUTION STEP.

Operation		**Number of**		
		Requests	**Hashes**	**Randoms**
Preliminaries	Step A	1	1	1
	Step B	2	0	0
	Step C	1	1	3
Service Access	Step D	2	1	2
	Step E	4	2	4
Revocation	Step F	1	0	0
	Step G	1	0	0

TABLE IV
NUMBER OF USER'S OPERATIONS.

Site browsing	Form compilation	Post creation	Hashtag search	Post removing
2	4	1	1	1

We ran some experiments to measure the time required by the different operations and used a 64-bit Windows 10 machine with Intel Core i7-7700K CPU 4.20GHz and 16GB RAM. We measured that the time of 1 million calculations of SHA-256 hashes is about 0.6 seconds and is 10 milliseconds for a single hash. Again, we measured that the time to generate 500 random numbers is about 1 millisecond. Concerning the network request time taken for operations *Send*, *Post*, and *Search*, we measured an average time of 500 milliseconds (the amount of data sent/received is very limited). As we found that the average time of hash computation and random generation is negligible with respect to the time of network request (they differ by 3 orders of magnitude), we omit in our analysis the time of the calculation of hashes and random numbers.

From the analysis of the results reported in Table III, we observe that the time required to conclude the service request/-grant is about 3 seconds, which is usually well accepted by users in a social network context.

The second parameter we consider is related to user-friendliness, which refers to the ability of a service to be used easily by the users and is commonly adopted to evaluate a range of end-user computing technologies. It is well-known that if a social network website does not provide an efficient and user-friendly interface, then its users may be disappointed and switch to another social network [18].

The actions performed by the user are easy: the user visits the website of the service provider and is requested to authenticate by a social-network-based login procedure. After the authentication, the user inserts the password in the website and creates a post containing a simple text with a hashtag. In the occurrence, the user can delete such a post. Which and how many operations a user performs are reported in Table IV. It is evident that these operations are very friendly as social networks are daily used by billions of people. Thus, we conclude that the use of a social network positively affects perceptions of people, who are favorably disposed to work in an environment that they well know.

The third parameter considered is scalability, which refers

TABLE V
NUMBER OF REQUESTS FOR A FULL SERVICE ACCESS.

Service Provider	Identi-fication	SN site	Hashtag search	Post removing
1	5	2	4	2

to the ability of a system to maintain its functionalities despite the scaling of users' requests. This parameter can be calculated by different types of performance measure attributes, such as the number of processed requests and network usage. In Table V, we measure the number of requests for each service access: the number of requests related to the post removing refers to both the user and service provider side. We observe that the measured values are constant with respect to the number of users, which is an indicator of good scalability. Moreover, also the size of exchanged messages is limited and does not depend on the number of users (see Section III). Consequently, we can state that the proposed solution offers high scalability and can support a very high number of users: the upper bound of the number of users is given by the number of users supported by the adopted social network. Consider that several strategies are used to ensure service availability [19] so that the probability of service interruption is negligible. Scalability is also favored because our solution does not require the use of heavy cryptography (hash functions are more efficient than encryption). This point is strongly related to the response time too.

Finally, we consider the cost of the solution related to the implementation and set-up costs. The price of our solution is limited because its architecture is mainly based on a social network. Social networks monetize their services with the precious information the users voluntarily reveal in their profiles, in their relationships, in their behavior [20], so that social networks can offer their services to registered users for free (differently from Blockchains). By relying on already existing social network systems, our costs are only those for set-up. Furthermore, it is worth noting that our solution, being compliant with the GDPR principles, can save up to €20M or up to 4% of the total worldwide annual turnover of the preceding financial year, as stated in [5]. On the other hand, solutions based on a blockchain have an increased cost of blockchain development and maintenance that are estimated between 40$-80K$.

V. SECURITY ANALYSIS

In this section, we discuss how our solution reaches the expected goals and how possible attacks are contrasted.

We start by defining the *adversary model*. In our analysis, we assume that identity providers, attribute providers, and social networks are trusted parties, and they run the protocol correctly. Thus, the adversary can be a user, a service provider, or an entity external to the system. In our attack model, the adversary cannot compromise the behavior of the identity provider, the attribute provider, and the social network, and cannot modify the posts published by other users on their social profile. The adversary cannot break the cryptographic primitives (e.g., the adversary cannot generate a message that yields a given hash value) and cannot guess the user's password, secret information, or randomly generated values. Finally, we assume that users and service providers do not collude with each other.

The attacker aims to violate one of the security properties guaranteed by our solution, which are access-control and privacy. We describe how these properties are guaranteed.

In the verification phase, the service provider calculates $C^U \oplus C^{AP} \oplus P(H(r_1^u)) \oplus P(r_1^u)$, and the attribute provider generates C^{AP} after verifying the user owns the attributes to be certified. By construction, we have that $\overline{C}^U = P(r^{AP}) \oplus P(r_1^u)$ and $C^{AP} = C \oplus P(H(r_1^u)) \oplus P(r^{AP})$ so that $C^U \oplus C^{AP} \oplus P(H(r_1^u)) \oplus P(r_1^u) = C$. If C is a valid credential, then the access can be granted because C is obtained starting from C^{AP}, which is generated by the attribute provider. It is worth noting that the XOR function is well-known not only for its efficiency but also for some weaknesses documented in the literature and already exploited in some application contexts, like WEP [21]. The weakness is related to the possibility for a user to generate a new credential C_*^U such that $C_*^U \oplus C_2 = C_*$, where C_* is a fake credential (see Step 7 of Section III). Specifically, to break the access control property, the adversary should be able to create a valid C. The most favorable case is that the user is the attacker so that he/she can generate C^U. In this case, the attacker has to use a suitable C^{AP}, which can be obtained in three ways: 1) by creating a new one, 2) by modifying an existing one, or 3) by using an existing one. Our threat model prevents cases 1 and 2 because the adversary cannot modify information uploaded by the attribute provider so that only the third possibility is available. In this case, given $C_1 = C_1^U \oplus C_1^{AP} \oplus P(H(x_1)) \oplus P(x_1)$ a valid credential, the attacker has to create a new credential $C_2 = X \oplus C_1^{AP} \oplus P(H(x_1)) \oplus P(x_1)$, where C^X is information handled by the attacker. We have that C^X should be equal to $C_2 \oplus C_1^{AP} = C_2 \oplus (C_1 \oplus P(H(r_1)) \oplus P(r^{AP}))$. Now, the attacker has to find x_1 and x_2 such that $P(x_1) \oplus P(x_2) = C^X$ (which is the information the user has to publish). Thus, $P(x_1) \oplus P(x_2) = C_2 \oplus C_1 \oplus P(H(x_1)) \oplus P(x_2)$, then $P(x_1) = C_2 \oplus C_1 \oplus P(H(x_1))$, and $P(H(x_1)) = P(x_1) \oplus C_2 \oplus C_1 = P(x_1) \oplus K$, where K is a constant. Now, the attacker has to find two seeds x_1 and x_2 such that the generated random number sequences are equal up to a constant K: this violates the assumption that P is a random number generator function. Moreover, x_2 should also be the hash value of x_1, and this violates the one-wayness property of the hash function H.

An attack we contrast is the replay attack, which is carried out by taking a credential used by a user to access a service and trying to use it. This attack is contrasted because the credential contains the screen-name of the social network profile so that another account cannot use it.

Concerning privacy, we observe that the credential does not contain any identifying information so that the service provider does not know the identity of the user accessing the service. As described in Section II, the plain-text credential C is encrypted by the one-time pad function, an encryption technique that cannot be cracked, provided that the key used in the XOR function is random. In our case, the key is obtained by the function P, which has been defined to generate pseudo-random numbers. Thus, it is unlikely that an adversary can recover C by knowing C^U or C^{AP} if P is well-implemented (our implementation satisfies this requirement). Another observation is about the use of the randoms r_1 and r_2 in Step A (credential request) of Section II. They are used so that the attribute provider is not aware of when the user exploits the credential (i.e., when the user publishes C^U) or terminates the use of this credential. Indeed, C^U is indexed by $H(r_1)$, and the attribute provider is not aware of r_1 so that it cannot know which hashtag has to be searched. This mechanism reduces the information about the user behavior known by the attribute provider and by the other actors in general.

Finally, we observe that also unlinkability can be achieved. Unlinkability means that the service provider is not able to guess that two different requests come from the same user. To achieve unlinkability, the user must require a new credential for each service request.

VI. Related work

In this section, we review the state of the art starting from Self-Sovereign Identity. A Self-Sovereign Identity must allow ordinary users to make claims, which could include personally identifying information or facts about personal capability or group membership. Ten principles of Self-Sovereign Identity are proposed in [4] A survey of solutions for Self-Sovereign Identity with and without the use of blockchain technology is presented in [7].

Blockchain technology has the potential to support emerging solutions on the data ownership and governance models. The study done in [22] categorizes these solutions into a taxonomy based on architecture, governance models, and other features.

EverID [23] is a user-centric solution which includes a scalable payment solution (EverChain) with a multi-currency wallet (EverWallet). Through the use of EverID, individuals control their database of identity elements, including their biometrics. This architecture is distributed and lays on an Identity Network (a private Ethereum Blockchain) and a Decentralized App (DApp), a software working on the decentralized network.

A decentralized identity based on hierarchically deterministic keys controlled and generated by the users is proposed in [24]. The architecture is based on a public blockchain in charge of providing a trusted storage layer and a mapping between each decentralized identifier (DID) and the corresponding DID document object (DDO). The lifeID Platform [25] is an identity service based on a permissionless blockchain able to run smart contracts. Users can create their identity by using a biometric-capable smartphone and the lifeID app.

The uPort technology is built on the Ethereum Blockchain [26] uses smart contracts, a mobile application, servers, data, and attestations to be proved. Sovrin is a global public utility for Self-Sovereign Identity and verifiable claims, and implements the concept "identity for all" [33]. It is built on a public blockchain and claims and credentials are represented by Sovrin's tokens which can be used for education, healthcare, and insurance.

A limitation of self-sovereign identity systems is the lack of qualified eID data [27]. For example, in the context of e-Government, employees may need to satisfy additional requirements about their identities that are mapped through qualified electronic identities (eIDs). The main advantage of blockchain-based solutions is that saved information is distributed and cannot get lost. However, if a private key is stolen, it is impossible to hide data. Differently, in our solution, even if the randoms or passwords used to generate the credential are made public, to hide the credential it is enough to remove the corresponding posts. As observed by [7], blockchain technology is an excellent foundation to face the problem, but it is not a necessity. Indeed, there exist other solutions that do not rely on blockchain.

TABLE VI
Comparison with existing solutions.

	[31]	[32]	[30]	[26]	[33]	[23]	This paper
Impl.	No	Yes	Yes	Yes	Yes	Yes	Yes
Eff.	n.a.	High	High	Low	High	Low	High
Usab.	Medium	High	High	Low	Low	Low	High
Scal.	Medium	Medium	Low	Low	Medium	Low	High
Cost	n.a.	Low	Low	High	High	High	Low

An attribute-based credential [28, 29] is a cryptographic container of a few attributes and is signed by an authoritative party. In the IRMA identity platform [30], personal attributes are stored in the user's devices IRMA app: in the case of stolen or lost devices, users have to create all attributes one more time.

The Private Data System (PDS) proposed in [31] enables self-sovereign storage and sharing of private data among online users. This system is composed of spread online nodes, and their role is not based on distributed consensus. ReclaimID is a decentralized service for identity management [32]. The attributes used to access an online service are stored over a name system and under name-spaces of users. Furthermore, attributes are encrypted by the attribute-based encryption (ABE) method.

After describing the related literature, we compare our solution with the state of the art by considering the five aspects presented in Section IV, which are:

1) If the solution has been implemented (*Implementation*).
2) How long the service access takes for the user (*Efficiency*).
3) The degree to which the solution can be used by consumers with satisfaction (*Usability*).
4) How much the solution is scalable (*Scalability*).
5) The price of the solution (*Cost*).

Implementation is a boolean measure, whereas we use the values *low*, *medium*, and *high* for measuring the other parameters. Table VI summarizes the results obtained by comparing the different techniques.

Concerning [31], we note that although the authors provide a workflow of the different phases of the solution, a real use-case implementation is not presented (thus, efficiency cannot be measured). The use of executable choreographies and the division of roles among the nodes result in medium scalability. In the second examined solution [32], the attributes are encrypted, and the system is developed compliant with the OpenID standard, but it is suitable for small or medium applications. It has the same number of operations as our approach and is distributed. The third approach we consider is presented in [30]: this system has low scalability because it is centralized: the Privacy by Design Foundation has published the schema on GitHub for various issuers [34] and these schemas are used by the IRMA app using an auto-update mechanism [35]. Nevertheless, the idea of exploiting the user's device to manage credentials makes this system user-friendly. The solutions [23], [33], and [26] provide users with a wallet to generate transactions over a blockchain network. Since managing blockchain wallets is not easy, their usability is low. The transaction verification relies on the consensus mechanism of the network, which introduces delays at the expense of scalability. Furthermore, these solutions are based on a blockchain, which increases the solution price because each transaction has a fee. The different efficiency of the three solutions is due to the blockchain technology adopted: [23] and [26] exploit the Ethereum blockchain, which is based on the Proof of Work as a consensus algorithm. Instead, [33] implements the Plenum Consensus Protocol, which is considered faster than the Proof of Work. Concerning our approach, according to the results presented in Section IV, it presents the best behavior compared to all the considered metrics.

VII. Conclusion

In this paper, we proposed a new solution for the management of personal data that is based on the use of a social network in charge of providing users with the means of issuing and verifying claims and credentials. Our solution allows a user to control the information shared with other actors by using very efficient operations, which are the exclusive-or function, the hash function, and the pseudo-random number generator. A service provider can verify the attributes of a user by the support of the social network as a secure and transparent repository of the selected and hidden information. The effectiveness of our proposal has been shown in a real-life scenario using Facebook as a social network. We showed that our solution offers several advantages, such as user-friendliness, cheapness, scalability, and availability. To remark the practical significance of the proposed approach, we highlight the importance of our solution concerning the European Union General Data Protection Regulation (GDPR) [5], and in particular, with the data minimization principle. Article 5(1)(c) says that personal data shall be *adequate, relevant, and limited to what is necessary in relation to the purposes for which they are processed*. Although the GDPR does not define these terms because they depend on the specific application, it is important that collected data retain only the minimum amount of information. In the example considered in this paper, the need to known if a user is of age usually results in the knowledge of the user's date of birth, which violates the minimization principle. Our proposal allows a service provider to comply with the GDPR privacy principle, yet keeping the guarantee that the access-control policy is respected.

References

[1] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[2] J. Le, D. Zhang, N. Mu, X. Liao, and F. Yang, "Anonymous privacy preservation based on m-signature and fuzzy processing for real-time data release," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[3] W. Lin, X. Zhang, L. Qi, W. Li, S. Li, V. S. Sheng, and S. Nepal, "Location-aware service recommendations with privacy-preservation in the internet of things," *IEEE Transactions on Computational Social Systems*, 2020.

[4] Christopher Allen, "The Path to Self-Sovereign Identity," http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html, 2016, online; accessed 3-July-2020.

[5] GDPR official website, https://eugdpr.org/, 2020, online; accessed 3-July-2020.

[6] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266–2277, 2019.

[7] D. Van Bokkem, R. Hageman, G. Koning, L. Nguyen, and N. Zarin, "Self-sovereign identity solutions: The necessity of blockchain technology," *arXiv preprint arXiv:1904.12816*, 2019.

[8] eIDAS Regulation, https://ec.europa.eu/futurium/en/content/eidas-regulation-regulation-eu-ndeg9102014, 2019, online; accessed 3-July-2020.

[9] Y. Jiang and J. Jiang, "Diffusion in social networks: A multiagent perspective," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 2, pp. 198–213, 2014.

[10] L. Li, Q. Zhang, X. Wang, J. Zhang, T. Wang, T.-L. Gao, W. Duan, K. K.-f. Tsoi, and F.-Y. Wang, "Characterizing the propagation of situational information in social media during covid-19 epidemic: A case study on weibo," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 556–562, 2020.

[11] W. Wei, B. Zhou, D. Połap, and M. Woźniak, "A regional adaptive variational pde model for computed tomography image reconstruction," *Pattern Recognition*, vol. 92, pp. 64–81, 2019.

[12] W. Wei, H. Song, W. Li, P. Shen, and A. Vasilakos, "Gradient-driven parking navigation using a continuous information potential field based on wireless sensor network," *Information Sciences*, vol. 408, pp. 100–114, 2017.

[13] W. Wei, X. Xia, M. Wozniak, X. Fan, R. Damaševičius, and Y. Li, "Multi-sink distributed power control algorithm for cyber-physical-systems in coal mine tunnels," *Computer Networks*, vol. 161, pp. 210–219, 2019.

[14] Y. Li, P. Luo, and P. Pin, "Utility-based model for characterizing the evolution of social networks," *IEEE Transactions on Systems, Man, and Cybernetics: Sys-*

tems, 2017.

[15] U. NIST, "Descriptions of sha-256, sha-384 and sha-512," 2001.

[16] S. Josefsson, "Rfc3548: The base16, base32, and base64 data encodings," 2003.

[17] J. Hughes and E. Maler, "Security assertion markup language (saml) v2. 0 technical overview," *OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08*, pp. 29–38, 2005.

[18] A. Ellahi and R. H. Bokhari, "Key quality factors affecting users' perception of social networking websites," *Journal of Retailing and Consumer Services*, vol. 20, no. 1, pp. 120–129, 2013.

[19] N. Masinde and K. Graffi, "Peer-to-peer based social networks: A comprehensive survey," *arXiv preprint arXiv:2001.02611*, 2020.

[20] B. Ngonmang, E. Viennet, S. Sean, P. Stepniewski, F. Fogelman-Soulié, and R. Kirche, "Monetization and services on a real online social network using social network analysis," in *2013 ICDM Workshops*. IEEE, 2013, pp. 185–193.

[21] V. Kumkar, A. Tiwari, P. Tiwari, A. Gupta, and S. Shrawne, "Vulnerabilities of wireless security protocols (wep and wpa2)," *IJARCET Journal*, vol. 1, no. 2, pp. 34–38, 2012.

[22] L. Lesavre, P. Varin, P. Mell, M. Davidson, and J. Shook, "A taxonomic approach to understanding emerging blockchain identity management systems (draft)," National Institute of Standards and Technology, Tech. Rep., 2019.

[23] EverID, https://coinosophy.files.wordpress.com/2018/05/everid-whitepaper.pdf, 2018, online; accessed 3-July-2020.

[24] Jolocom, https://jolocom.io/, 2020, online; accessed 3-July-2020.

[25] LifeID, https://lifeid.io/whitepaper.pdf, 2020, online; accessed 3-July-2020.

[26] uPort, https://www.uport.me, 2020, online; accessed 3-July-2020.

[27] S. Ramacher, "Privacy-preserving eid derivation for self-sovereign identity systems," in *ICICS 2019*, vol. 11999. Springer Nature, 2020, p. 307.

[28] B. Hampiholi and G. Alpár, "Privacy-preserving web-shopping with attributes," in *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 2017, pp. 25–36.

[29] L. Cheng, J. Liu, G. Xu, Z. Zhang, H. Wang, H.-N. Dai, Y. Wu, and W. Wang, "Sctsc: A semicentralized traffic signal control mode with attribute-based blockchain in iovs," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1373–1385, 2019.

[30] IRMA-Privacy By Design Foundation, https://privacybydesign.foundation/irma-en/, 2020, online; accessed 3-July-2020.

[31] S. Alboaie and D. Cosovan, "Private data system enabling self-sovereign storage managed by executable choreographies," in *ICDAIS*. Springer, 2017, pp. 83–98.

[32] M. Schanzenbach, G. Bramm, and J. Schütte, "reclaimid: Secure, self-sovereign identities using name systems and attribute-based encryption," in *TrustCom/BigDataSE*. IEEE, 2018, pp. 946–957.

[33] D. Reed, J. Law, and D. Hardman, "The technical foundations of sovrin," *The Technical Foundations of Sovrin*, 2016.

[34] IRMA scheme manager, https://github.com/privacybydesign/pbdf-schememanager, 2020, online; accessed 3-July-2020.

[35] J. C. Nauta and R. Joosten, "Self-sovereign identity: A comparison of irma and sovrin," TNO, Tech. Rep., 2019.

Gianluca Lax is an Associate Professor of Computer Science at the University Mediterranea of Reggio Calabria, Italy. In 2005, he received his PhD in computer science from the University of Calabria. In 2018, he got the habilitation as a Full Professor of Computer Science. His research interests include information security and social network analysis. He is an author of more than 120 papers published in leading international journals and conference proceedings.

Antonia Russo is a PhD Student in Computer Science at the University Mediterranea of Reggio Calabria. She received her MsC Degree in Telecommunication Engineering from the University Mediterranea of Reggio Calabria in July 2018. Her research interests include security, privacy, and social network analysis.