Ethereum Transactions and Smart Contracts among Secure Identities

Francesco Buccafurri, Gianluca Lax, Lorenzo Musarella, and Antonia Russo

University Mediterranea of Reggio Calabria, Italy
{bucca,lax,lorenzo.musarella,antonia.russo}@unirc.it

Abstract

One of the limitations of the current Blockchains is that recipients of transactions (originated from both users and smart contracts) must preliminarily sign up the system. In contrast, the nature of Blockchain would allow the implementation of services with a high degree of flexibility and interoperability, once the subjects can be securely identified someway. In this paper, we overcome this limitation by integrating Public Digital Identity with Ethereum via Identity-Based-Encryption (IBE). An important feature of the solution is that it does not require additional trust w.r.t. that necessary for IBE and Public Digital Identity systems.

1 Introduction

The interest towards Blockchain [30] is constantly increasing during the last years, due to its power to enable new business scenarios. Blockchain technologies attract the attention of both industries and researches, in various fields, besides computer science, mainly also economics and law. As a consequence, any aspect regarding those technological features that impact how applications can be designed and used is very relevant.

One of the current limitations of Blockchains (even smart-contract oriented) is that actors of transactions and smart contracts are required to voluntarily subscribe to the services of the application platform implemented over Blockchain. This aspect makes Blockchain platforms little appropriate to those situations in which services may involve dynamically unregistered subjects. The proposal of this paper regards the following situation. Consider a set S of subjects who are identifiable in a certain (secure) way. For the moment, it does not matter how. Suppose in this set there are users who may be involved in a service based on Blockchain (for example, Ethereum), in different moments, depending on dynamic conditions. Thus, for example, `Alice`, who is already registered to the service, has to transfer money (or a given token) to `Bob`. But `Bob`, despite being in S, is not in the service yet. In other words, we would like to enable some *suspended* actions, to avoid to compromise the liveness of the system. Indeed, according to the features currently supported by Ethereum (and the other Blockchains), `Alice`'s action should be denied by the system until `Bob` signs up the system. We obtain a similar use case when the sender is a contract instead of a human user.

Our proposal, contextualized in the Ethereum environment, is aimed to overcome the above limitation, by enabling over Ethereum transactions and contracts among (secure) digital identities, whose existence is independent of the specific application platform. This allows the design of flexible, dynamic and interoperable services, with considerable benefits in many cases, especially in crowd-based or multi-organization domains.

To do this, we faced a number of problems. The first one is which notion of digital identity we may adopt to have a realistic result. One could think of an identity built as a combination of (verified) social network profiles owned by the subject being identified. This could be an option, but we think that whether a Public Digital Identity System exists, like those that are

compliant in EU with the eIDAS regulation [5], this is the best way to follow. Thus, in our paper we refer to SPID [11], which is the Italian System of Public Digital Identity introduced in accordance with the eIDAS initiative.

The second point is how to link in a secure way digital identities and Ethereum addresses. Our solution leverages Identity-Based-Encryption (IBE) [7], which gives a direct role to the notion of identity and then a direct link between Ethereum keys and identity of the user, once she/he is able to provide the PKG (i.e., the party issuing the IBE private key) with the proof of her/his SPID identity. From this point of view, this paper is an evolution of the work presented in [22], in which the idea of integrating IBE and Blockchain is presented for the first time in the simpler context of Bitcoin Blockchain, thus without the possibility of involving unregistered users. We highlight that the role of IBE is crucial in our proposal, because a direct integration of SPID with Blockchain (like in [16]) would require that a Blockchain-side entity (an application or a smart contract) should play as a Service Provider of the public digital identity system. This would require the full trust in this entity, concerning the assessment of identity.

The paper is structured as follow. In Section 2, we introduce the notion of digital identity with the related technologies and Identity-Based Encryption, which is used to bind a digital identity and a public key. In Section 3, we describe our proposal aiming at associating a public digital identity with a blockchain transaction. In Section 4, we provide the technical details about how our solution works. The related work is discussed in Section 5. Finally, in Section 6, we draw our conclusions.

2 Background

In this section, we recall two concepts related to our proposal, which are public digital identity and identity-based encryption.

A digital identity is defined as information on an entity used by computer systems to represent an external agent that may be a person, organization, application, or device [4]. Another similar definition given by ISO/IEC 24760-1 reports digital identity as a set of attributes related to an entity [1].

In this paper, we refer to a specific notion of digital identity, the *public digital identity*, which is recognized by law at international level making the basis for non-repudiable accountable applications. A concrete instantiation of this notion in the European Union is based on the Regulation (EU) N. 910/2014 [5] on electronic identification and trust services for electronic transactions in the internal market (eIDAS Regulation), issued on 23 July 2014 and fully effective from 1 July 2016. It has the purpose of providing a normative basis at EU level for fiduciary services and providing the means of Member States' electronic identification: the eIDAS regulation aims to provide a common regulatory basis for secure electronic interactions between citizens, businesses and public administrations and to increase the security and effectiveness of e-business services and e-business and e-commerce transactions in the European Union. Thanks to the principle of mutual recognition and reciprocal acceptance of interoperable electronic identification schemes, eIDAS wants to simplify the use of electronic authentication against public administrations, both by companies and by citizens. Each Member State maintains it own electronic identification systems, which have to be accepted by all other member states. For example, Italy has notified the EU Commission the institution of *SPID*, the Italian public system for the management of the digital identity of citizens and businesses [11]. Thanks to the eIDAS regulation, it is possible for Italian citizens to access the online services of other EU countries (university services, banking, public administration services, other online services) using SPID credentials, and at the same time, European citizens in possession of recognized

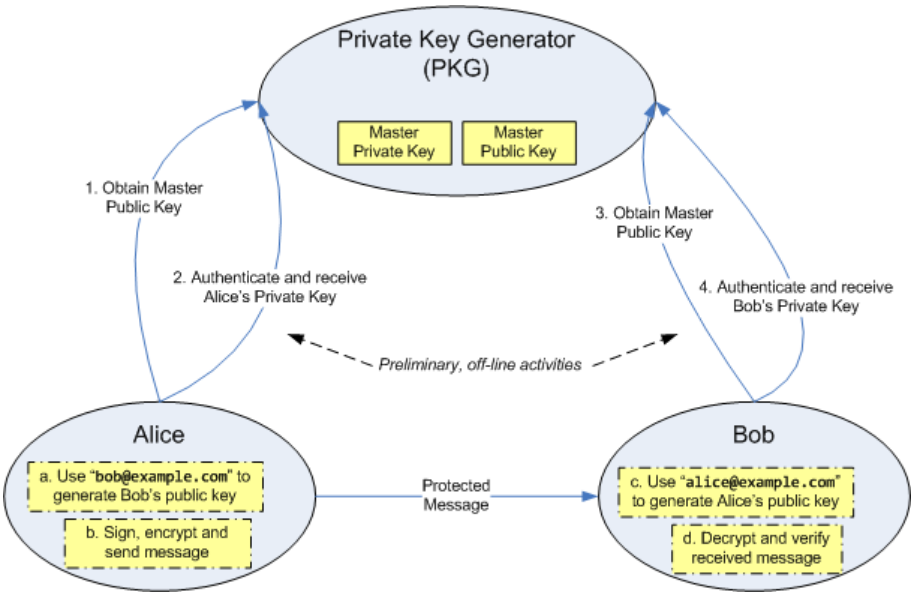

Figure 1: Operations carried out in an IBE scheme.

national digital identities within the eIDAS framework will have access to the services of Italian public administrations.

The second concept we present is Identity-based Encryption. It is known that in asymmetric cryptography, each user owns a public and a private key, and public keys are typically arranged by a Public Key Infrastructure, which binds public keys with the respective identities of entities through a process of registration and issuance of certificates by a certificate authority (CA). Identity-based Encryption is a solution in those cases in which pre-distribution of keys is inconvenient or infeasible due to technical restraints.

Identity-based Encryption (IBE) [7] allows any party to generate a public key from a known identity value (for example, an e-mail address). A trusted third party, called the Private Key Generator (PKG), generates the corresponding private key. To operate, the PKG first publishes a master public key and retains the corresponding master private key (referred to as master key). Given the master public key, any party can compute a public key corresponding to an identity by suitably combining the master public key with the identity value. To obtain a corresponding private key, the party authorized to use the identity ID contacts the PKG, which uses the master private key to generate the private key for the identity ID. The operations carried out in an IBE scheme are summarized in Figure 1.

As a result, parties may encrypt messages (or verify signatures) with no prior distribution of keys between individual participants, once their identity is known and well-defined. However, to decrypt or sign messages, the authorized user must obtain the appropriate private key from the PKG, by proving the possession of the proper identity. The most used IBE systems have been proposed by Boneh-Franklin [19] and by SakaiKasahara [33].

3 Our proposal

The goal of this paper is to allow the association of a digital identity with a blockchain transaction. Among the possible mechanisms to handle digital identity, such as OAuth [8], OpenID [9] Windows CardSpace [13], we refer to the notion of public digital identity, which has been defined by the Regulation (EU) N. 910/2014 [5]. Our choice is motivated by the fact that we expected that, in the next years, public digital identity will involve the most of EU people: for example, on February 2017, Germany notified its national identity which has more than 40 million registered citizens [12].

In our solution, we have the following types of entity:

- a user, a physical or legal person using a digital identity for authentication. Each user can be associated with one or more public digital identities.
- a public identity digital system with identity provider IP, which creates and manages public digital identities. Without loss of generality, we assume it is unique.
- an IBE system with Private Key Generator PKG (see Section 2). It is managed by a public or private organization and provides the mapping between a digital identity and a pair of asymmetric encryption keys (called IBE keys).
- a Distributed Ledger allowing smart contracts (i.e., Blockchain 2.0).

In this scenario, we identify the following types of operation that users carry out.

1. *Digital Identity Registration.* To obtain a digital identity, a user must be registered to the public identity digital system. In this phase, the real identity of the user is verified before issuing the public digital identity and the security credentials.

   A public digital identity is identified by the pair $\langle username, IP \rangle$, where *IP* is the identifier of the identity provider that issued the public digital identity and *username* is a string. For example, the user X registered by the Identity Provider Y is identified by the X@Y. Moreover, any Public Digital Identity System compliant with eIDAS defines also a string *UID* (Universal ID), which is a single numeric identifier independent of the identity provider, in case of multiple identity providers.

2. *IBE private key gathering.* To obtain the IBE private key, a user contacts the Private Key Generator (PKG) of the IBE service to receive the master public key, if it is not already known. Then, the Private Key Generator, by acting as a service provider of the public digital identity system, authenticates the user by an eIDAS-compliant scheme, as illustrated in Figure 2.

   First, the user using a browser (`User Agent`) sends to PKG a request for gathering the IBE private key (Step 1). Then, PKG replies with an authentication request to be forwarded to *Identity Provider* (Step 2). If the received request is valid, *Identity Provider* performs a challenge-response authentication with the user (Steps 3 and 4). In case of successful user authentication, *Identity Provider* prepares the statement of user authentication, which is forwarded to PKG (Step 6). Finally, PKG provides the user with the IBE private key (Step 7).

   Observe that the IBE public key is always calculable from the digital identity of a user, provided that the IBE master public key is known.

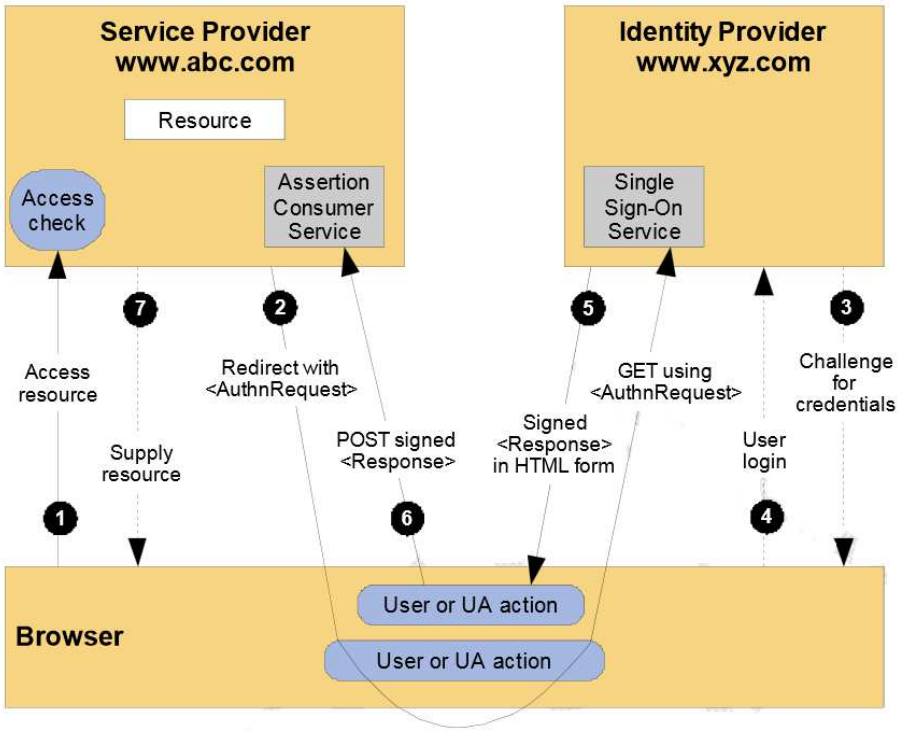

Figure 2: Data flow in an authentication process.

3. *Blockchain Binding.* By this operation, a user associates his IBE public key IBE_p^K with his Blockchain address A. First, the user generates a pair of private and public blockchain keys, and, then, the blockchain address A of the user is computed as the cryptographic hash of the public key. Then, the user generates a transaction from A to A on the blockchain, having in *data* field $\langle UID, E(A)\rangle$, where UID is the universal ID of the public digital identity of the user, and $E(A)$ is the encryption of the user's blockchain address by the user's IBE secret key. This transaction is called *binding transaction*. By this transaction, the user links her/his public digital identity to the blockchain address A: indeed, by computing $E(A)$, the user proves the knowledge of the IBE secret key associated with this UID.

4. *Transaction.* Suppose a user S (sender) wants to send to a user R (receiver) a transaction and let v be the value of the transaction (i.e., the amount of virtual money to transfer). In this case, the following operations are done. First, S obtains the universal ID of R, say UID_r and searches for the most recent binding transaction having UID_r in the payload: this search can be successful or not. If a transaction $T = \langle UID_r, E(A_r)\rangle$ of this type is found, then:

   (a) S deciphers $E(A_r)$ by using the IBE public key calculated from UID_r, to verify that the authenticity of the signature (observe that $E(A_r)$ works as a signature to prove that the right user has generated the binding transaction). If this check fails, T is ignored and another search is carried out.

(b) After deciphering $E(A_r)$, the blockchain address of UID_r is obtained.

(c) S generates a blockchain transaction from his blockchain address A_s to A_r, with value v.

Consider now the case in which no transaction of this type is found, which is the most interesting case. This means that the user R exists but has not yet joined the blockchain. In this case, S generates a blockchain transaction from his blockchain address A_s to the blockchain address of a specific smart contract, say A_{sc}, specifying both UID_r and v. This smart contract stores the information that there is a *sleeping* transaction to UID_r, from the sender A_s and value v.

5. *Cashing.* Suppose that a user R, after registering to the blockchain, wants to receive the *sleeping* transactions sent to him before his registration (i.e., those transactions sent to the smart contract sm and intended for him). Then, he generates a blockchain transaction, named *cashing transaction*, from his blockchain address A_r to the blockchain address A_{sc} (i.e., the same smart contract referred above), having his UID (i.e., UID_r) in the payload. Now, the smart contract searches for the most recent binding transaction T sent from A_{sc} and computes the IBE public key IBE_r calculated from UID_r. Then, it extracts $E(A)$ from the payload of T and deciphers $E(A)$, verifying that UID_r is obtained. Finally, it extract from the stored sleeping transactions those sent to UID_r (if any): for each transaction found, a new transaction to A_r is generated, with the same value as the found transaction.

In the next section, we show how to implement this solution in Ethereum.

4 Implementation

In this section, we instantiate the general approach presented in the previous section to the specific environment of Ethereum: in particular, we show all the operations carried out by two Ethereum users, say Alice and Bob.

1. *Digital Identity Registration.* Both Alice and Bob have a public digital identity: thus, they have been identified by an identity provider, say `example.com`, which gave each of them a public digital identity and a credential for authentication (typically, a password). Now, assume that the username of Alice is `alice` and the username of Bob is `bob`. Thus, the UIDs of Alice and Bob are `alice@example.com` and `bob@example.com`, respectively. Observe that, for the sake of presentation, we used the same identity provider (i.e., `example.com`) for both the users: however, no problem arises in case the public digital identities are issued by different identity providers, because the solution does not depend on the particular UID of the user.

2. *IBE private key gathering.* To obtain the IBE private key, a user connects to the site of the IBE system by the browser (i.e., the user agent) and sends a request for accessing the service. Observe that the IBE system acts as a service provider in this step, because it needs to authenticate the user before issuing the private key. Then, the IBE system replies to the user agent with an authentication request to be forwarded to the identity provider. The identity provider is selected according to the user's UID.

   If the received request is valid, the identity provider performs a challenge-response authentication with the user. In case of successful user authentication, the identity provider

prepares the *assertion* containing the statement of the user authentication for the IBE service provider. The assertion contains the reference to the request message, the authenticated user, the identity provider, the personal information about the authenticated user, the temporal range of validity, and the description of the authentications context. The assertion is signed by the identity provider to guarantee integrity and authenticity.

Now, the assertion returned to the user agent is forwarded via `http POST` Binding to the IBE service provider. The IBE system verifies the assertion and provides the user with her/his IBE private key. We denote by IBE_U^S the IBE private key of the user U.

Concerning the user's IBE public key, they are computed starting from the master public key and the user's UID. We denote by IBE_U^P the IBE public key of the user U.

3. *Blockchain Binding.* Each user needs to have a private and a public blockchain key. The private key is a randomly generated 256-bit string. The public key is generated by the private one by means of a cryptographic function named *elliptic curve point multiplication*. In particular, the used algorithm is Curve Digital Signature Algorithm (ECDSA) and the elliptic curve is `secp256k1` [29].

   The Ethereum address A of a user is computed from the public key K by apply Keccak-256 [10], and finally taking the last 20 bytes of that hash. We denoted by A_U the blockchain address of the user U. Finally, each user generates the binding transaction having as payload $\langle UID, E(A)\rangle$, where UID is the universal ID of the public digital identity of the user, and $E(A)$ is the encryption of the user's Ethereum address done by the user's IBE private key.

4. *Transaction.* Now, both Alice and Bob have their public digital identity associated with a blockchain address. Suppose that Alice has to send some Ether money to Bob, but Bob has not an Ethereum wallet (i.e., he has not an Ethereum address). Clearly, we can image that users run an application (on a PC or a smartphones) to manage transaction generations.

   First, Alice has to know the UID of Bob: the UID of Bob as well as the amount of money to transfer are inserted into the application, which generates a transaction to the smart contract. In Figure 3, we give an implementation of this contract written in Solidity, which is a JavaScript-like language. For the sake of presentation, we do not explain every line of the code: we assume the reader is familiar with Solidity and Oraclize, which is the leading oracle service for smart contracts and blockchain [2] In particular, it is called the function `pay`, using the UID of Bob as parameter (Lines 13-15): this function stores the amount that will be given to Bob when he will register (by `payUID`).

5. *Cashing.* After Bob creates his wallet, he can ask for receiving the amount from the *sleeping* transactions sent to him before his registration. To do this, he generates a cashing transaction to the smart contract illustrated in Figure 3, by calling the function cash. The smart contract first checks if there is some amount for Bob. If any, an oraclize function is used (Line 21), which returns the Ethereum address of Bob by the callback function. Finally, a money transfer to Bob is carried out by the smart contract and the amount to pay to bob is reset (Lines 33-34).

By this protocol, we enable on Ethereum the possibility to send money to users without the need to know their blockchain address. The suitable use of the secure digital identity guarantees that only the correct user receives money.

```
pragma solidity ^0.4.25;

import "github.com/oraclize/ethereum-api/oraclizeAPI_0.4.25.sol";
import "github.com/Arachnid/solidity-stringutils/strings.sol";

contract SleepingEther is usingOraclize {
   mapping(bytes32=>string) uidMapping; //mapping between queryID and bool
   mapping(string=>uint) payUid; //mapping between UID and eth value to send
   address public addr;
   using strings for *;
   string pi;

   function pay(string uid) public payable {
        payUid[uid] += msg.value; // add the ether addressed to uid
    }

    function cash (string uid) public payable{
        if(payUid[uid]>0)
            if (oraclize.getPrice("URL") <= address(this).balance) {
                pi = "URL".toSlice().concat(uid.toSlice());
                bytes32 queryId = oraclize_query("URL", pi);
                uidMapping[queryId]=uid;
            }
    }

   function __callback (bytes32 myid, string result, string uid) public {
        if (msg.sender != oraclize_cbAddress())
          revert ();
        bytes memory tempEmptyStringTest = bytes(result);
        if(tempEmptyStringTest.length != 0){
            addr = parseAddr(result);
            uint tot= payUid[uidMapping[myid]];
            addr.transfer(tot);
            payUid[uid]=0;
        }
    }
}
```

Figure 3: The code of the smart contract.

5 Related work

In this section, we survey the approaches present in the literature related to our topic.

Thanks to the assurance of authenticity and uniqueness of transactions, blockchain starts to become the technical core of cryptocurrency, access control systems, asset management, banking, e-voting, etc. [28, 32, 18, 24].

The authors of [31] provide an overview of the blockchain technology and its potential to disrupt the world of banking through facilitating global money remittance, smart contracts, automated banking ledgers and digital assets. In this regard, they provide a brief overview of the core aspects of this technology, as well as the second-generation contract-based developments. From there, their work enforces key issues that must be considered in developing such ledger based technologies in a banking context.

In [14], the authors review applications relying on blockchain by highlighting the potential benefit of such technology in the manufacturing supply chain. Furthermore, they propose a vision for the future blockchain ready manufacturing supply chain.

The paper [17] provides a high-level understanding of how blockchain technology will be a fundamental tool to improve supply chain operations. It illustrates theoretical and conceptual models for use of open blockchain in different supply chain applications with real-life practical use cases as is being developed and deployed in various industries and business functions.

In [26], the authors observe that digital supply chain integration is becoming increasingly dynamic and investigate the requirements and functionalities of supply chain integration, concluding that cloud integration can be expected to offer a cost-effective business model for interoperable digital supply chains. Furthermore, they explain how supply chain integration through the blockchain technology can achieve important transformation in digital supply chains and networks.

The authors of [35] propose an overview of what smart contracts are and what are their main challenges for the future. In particular, they state that smart contracts have three main characteristics: *(i)* autonomy, *(ii)* self-sufficiency and *(iii)* decentralization. Autonomy means that the contracts and the initiating agents do not need to be in further contact. Self-sufficient means that smart contracts are able to raise funds by providing services and spending them when needed. Furthermore, smart contracts are decentralized as they do not are valid on a single centralized server, but they are distributed and self-executed across network nodes. As they can be seen as a distributed application, they have to face almost the well-known challenges of them, such as the reentrancy vulnerability, the privacy issues, how to guarantee the reliability of external information, and so on.

Another topic related to our proposal regards digital identity, in which we can find a rich literature. Bitnation [3] is the world's first Decentralized Borderless Voluntary Nation (DBVN). Bitnation started in July 2014 and hosted the first blockchain for refugee emergency ID, marriage, birth certificate, World Citizenship and more. The website proof-of-concept, including the blockchain ID and Public Notary, is used by tens of thousands of Bitnation Citizens and Embassies around the world.

In [21], the authors focus on the Public Digital Identity System (SPID), the Italian government framework compliant with the eIDAS regulatory environment. They observe that a drawback limiting the real diffusion of this framework is that, despite the fact that identity and service providers might be competitor private companies, SPID authentication results in the information leakage about the customers of identity providers. To overcome this potential limitation, they propose a modification of SPID to allow user authentication by preserving the anonymity of the identity provider that grants the authentication credentials. This way, information leakage about the customers of identity providers is fully prevented.

In [25], the authors highlight that, since we are in the Internet era, it is necessary a right blockchain-based identity management, as we have faced identity management challenges since the sunrise of the Internet. In particular, they observe that blockchain technology should offer a way to circumvent this problem by delivering a secure solution without the need for a trusted, central authority. It can be used for creating an identity on the blockchain, making it easier to manage for individuals, giving them greater control over who has their personal information and how they access it. The proposed solution stores the encrypted identity of users, allowing them to share their data with companies and manage it on their own terms.

The paper [34] focuses on pseudonymization, a concept that was only recently formally introduced in the EU regulatory landscape. In particular, it attempts to derive the effects of the introduction of pseudonyms (or pseudonymous credentials) as part of the eIDAS Regulation on electronic identification and trust services and, ultimately, to compare them with the effects of pseudonymization within the meaning of the General Data Protection Regulation (the GDPR). The work examines how eIDAS conceives pseudonymization and explains how this interpretation would translate in practical uses in the context of a pan-European interoperability framework.

In [20], an advanced electronic signature protocol that relies on a public system for the management of the digital identity is proposed by the authors. The work aims at implementing an effective synergy to provide the citizen with a unique, uniform, portable, and effective tool

applicable to both authentication and document signature.

In [15], the authors proposed SCPKI, an alternative PKI system based on a decentralized and transparent design using a web-of-trust model and a smart contract on the Ethereum blockchain, to make it easily possible for rogue certificates to be detected when they are published. The web-of-trust model is designed such that an entity or authority in the system can verify fine-grained attributes of another entity's identity, as an alternative to the centralized certificate authority identity verification model.

The paper [23] argues that existing laws, specifically the federal Electronic Signatures in Global and National Commerce Act ("ESIGN") and state laws modeled on the Uniform Electronic Transaction Act ("UETA"), render blockchain-based smart contracts enforceable and therefore immediately usable.

From the analysis of the state of the art, we conclude that the topic of this paper is surely important. Moreover, to the best of our knowledge, this is the first paper aiming at addressing the important issue of allowing payment towards blockchain users who have not yet joined the blockchain.

6 Conclusion and Future Work

In this paper, we presented a solution to integrate Public Digital Identity with Ethereum to enable transactions (both external and internal) directed to subjects not yet registered to the system. The solution leverages the concept of Identity-Based-Encryption (IBE). The goal is achieved by giving to the Private Key Generator (PKG) of the IBE the role of Service Provider of the Public Digital Identity system. It is worth noting that, in this paper, we only treat the case in which a given amount of cryptocurrency is transferred, but the transfer of tokens with identifier can be easily implemented by using the interface ERC721 [6] and we will address this improvement as a future work. Another direction for the evolution of this work is the complete elimination of centralization introduced by both the IBE and the Public Digital Identity systems. To do this, we can use distributed versions of such systems [27, 25]. Finally, we plan to complete the implementation and to apply it to the industrial projects in which our proposal is contextualized.

Acknowledgment

This paper has been partially supported by the projects "Id Service - Digital Identity and Service Accountability" funded by the Ministry of Economic Development (MISE), project code number F/050238/03/X32, and "SecureOpenNets-Distributed Ledgers for Secure Open Communities", funded by Ministry of Research and Education (MIUR), project id ARS01_00587.

References

[1] ISO/IEC 24760-1:2011 Information technology Security techniques A framework for identity management Part 1: Terminology and concepts (2011), http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

[2] Oraclize (2016), http://www.oraclize.it/

[3] Bitnation Pangea — Your Blockchain Jurisdiction (2018), https://tse.bitnation.co/

[4] Digital identity (2018), https://en.wikipedia.org/wiki/Digital_identity/

[5] eIDAS - Interoperability Architecture (2018), https://ec.europa.eu/futurium/en/content/eidas-regulation-regulation-eu-ndeg9102014
[6] ERC-721 Token (2018), http://erc721.org/
[7] ID-based Encryption (2018), https://en.wikipedia.org/wiki/ID-based_encryption
[8] OAuth Community Site (2018), https://oauth.net/
[9] OpenID The Internet Identity Layer (2018), https://openid.net/
[10] SHA-3 (2018), https://en.wikipedia.org/wiki/SHA-3
[11] SPID Sistema Pubblico di Identità Digitale (2018), https://www.spid.gov.it/
[12] The eIDAS Regulation in 2017 A pivotal year for digital services in the EU (2018), https://www.gemalto.com/govt/identity/eidas-regulation-in-2017
[13] Windows CardSpace (2018), https://en.wikipedia.org/wiki/Windows_CardSpace
[14] Abeyratne, S.A., Monfared, R.P.: Blockchain ready manufacturing supply chain using distributed ledger (2016)
[15] Al-Bassam, M.: Scpki: A smart contract-based pki and identity system. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. pp. 35–40. ACM (2017)
[16] Angiulli, F., Fassetti, F., Furfaro, A., Piccolo, A., Saccà, D.: Achieving service accountability through blockchain and digital identity. In: International Conference on Advanced Information Systems Engineering. pp. 16–23. Springer (2018)
[17] Banerjee, A.: Blockchain technology: Supply chain insights from erp. Advances in Computers (2018)
[18] Bistarelli, S., Mantilacci, M., Santancini, P., Santini, F.: An end-to-end voting-system based on bitcoin. In: Proceedings of the Symposium on Applied Computing. pp. 1836–1841. ACM (2017)
[19] Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Annual international cryptology conference. pp. 213–229. Springer (2001)
[20] Buccafurri, F., Fotia, L., Lax, G.: Implementing advanced electronic signature by public digital identity system (spid). In: International Conference on Electronic Government and the Information Systems Perspective. pp. 289–303. Springer (2016)
[21] Buccafurri, F., Fotia, L., Lax, G., Mammoliti, R.: Enhancing public digital identity system (spid) to prevent information leakage. In: International Conference on Electronic Government and the Information Systems Perspective. pp. 57–70. Springer (2015)
[22] Buccafurri, F., Lax, G., Russo, A., Zunino, G.: Integrating digital identity and blockchain. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 568–585. Springer (2018)
[23] Cohn, A., West, T., Parker, C.: Smart after all: Blockchain, smart contracts, parametric insurance, and smart energy grids. Georgetown Law Technology Review **1**(2), 273–304 (2017)
[24] Dai, F., Shi, Y., Meng, N., Wei, L., Ye, Z.: From bitcoin to cybersecurity: A comparative study of blockchain application and security issues. In: Systems and Informatics (ICSAI), 2017 4th International Conference on. pp. 975–979. IEEE (2017)
[25] Jacobovitz, O.: Blockchain for identity management. The Lynne and William Frankel Center for Computer Science Department of Computer Science. Ben-Gurion University, Beer Sheva Google Scholar (2016)
[26] Korpela, K., Hallikas, J., Dahlberg, T.: Digital supply chain transformation toward blockchain integration. In: proceedings of the 50th Hawaii international conference on system sciences (2017)
[27] Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 568–588. Springer (2011)
[28] Maesa, D.D.F., Mori, P., Ricci, L.: Blockchain based access control. In: IFIP International Conference on Distributed Applications and Interoperable Systems. pp. 206–220. Springer (2017)
[29] Mayer, H.: Ecdsa security in bitcoin and ethereum: a research survey. CoinFaabrik, June **28**

(2016)

[30] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)

[31] Peters, G.W., Panayi, E.: Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money. In: Banking Beyond Banks and Money, pp. 239–278. Springer (2016)

[32] Pilkington, M.: 11 blockchain technology: principles and applications. Research handbook on digital transformations p. 225 (2016)

[33] Sakai, R., Kasahara, M.: Id based cryptosystems with pairing on elliptic curve. IACR Cryptology ePrint Archive **2003**, 54 (2003)

[34] Tsakalakis, N., Stalla-Bourdillon, S., O'hara, K.: What's in a name: the conflicting views of pseudonymisation under eidas and the general data protection regulation (2016)

[35] Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., Wang, F.Y.: An overview of smart contract: architecture, applications, and future trends. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 108–113. IEEE (2018)