Watermarking Public-key Cryptographic functions

Aikaterini Samari*

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications

ksamari@di.uoa.gr

Abstract. This PhD thesis studies the notion of watermarking for the case of *public key* cryptographic functionalities, such as public key encryption and digital signatures. Watermarking a public key cryptographic functionality enables the embedding of a mark in the instance of the secret-key algorithm such that the functionality of the original scheme is maintained, while it is infeasible for an adversary to remove the mark (unremovability) or mark a fresh object without the marking key (unforgeability). A number of works have appeared in the literature proposing different definitional frameworks and schemes secure under a wide range of assumptions. In this thesis, we approach this problem by proposing a new definitional framework that distinguishes between watermarking cryptographic functionalities and implementations (e.g. ElGamal encryption scheme as an implementation of the encryption functionality). Our definitional framework provides a meaningful relaxation comparing to other works that have appeared in the literature (e.g. [12,18]). Taking advantage of our new formulation, we present various constructions for watermarking public key encryption and digital signatures functionalities and implementations which are proven secure based on *standard* hardness assumptions. The security analysis of our constructions is performed along the lines of the definitional framework which is proposed in this thesis.
Furthermore, this PhD thesis studies the problem of the design of a Public Key Infrastructure (PKI), a system which binds public keys to real-world identities. In practice, PKIs are handled by trusted entities called Certification Authorities (CAs). In this thesis, focusing on providing a provably-secure construction, we present a generic construction for a PKI, which, depending on how it will be instantiated, may avoid the centralized nature of PKIs, such as CA-based PKIs.

Keywords: watermarking · cryptographic functionality · security model · public key encryption · digital signatures · public key infrastructure.

1 Introduction

In the digital era, watermarking has been a powerful tool, widely used in practice, to secure copyrighted material. Watermarking digital objects like pictures, video

* *Dissertation Advisor*: Aggelos Kiayias, Associate Professor

or software is usually achieved by embedding a special piece of information, the *mark*, into the object so that it is difficult for an adversary to remove it without damaging the object itself (unremovability). At the same time, the embedding of the mark should not result to a significantly different object, or an object with different functionality. A plethora of watermarking schemes exists in the literature (e.g. [4,13,26,27]), most of them focusing on watermarking "static" or else "perceptual" objects (e.g. images) and formal security definitions for watermarking "perceptual" objects have been given by Hopper et al. [19].

Besides perceptual objects, there has been a recent focus on *software watermarking* and precisely on watermarking cryptographic functionalities. Watermarking cryptographic functions has various real-life applications. Consider for instance the case of VPN clients. An organization might wish to distribute VPN clients to its employees where every employee has a public/secret-key pair. Watermarking the VPN client restricts the employees from sharing their clients since, due to the unremovability and unforgeability property, given any client one could detect to whom does this client belongs to (assuming the ID of the user is embedded in the watermark). The first rigorous definitions for software watermarking were given by Barak et al. [8,9]. Informally speaking, according to [8,9], a marking algorithm, marks a program/circuit C by producing a new circuit $\widetilde{C}$, which does not alter the functionality of C. Then, a detection algorithm can deduce whether a circuit is marked or not, for *any* circuit which is given as input. The basic security requirement that should be satisfied by a watermarking scheme is *fragility/unremovability* which requires that no polynomial time adversary should be able to remove the "mark" from a marked circuit $\widetilde{C}$, unless it substantially changes its functionality. Having set their definitional framework, Barak et al. [8,9] explore the relation of software watermarking with the notion of indistinguishability obfuscation (iO) [8,9,17] and provide an impossibility relation between iO and this specific definition of watermarking.

Following the work of Barak et al. [9], a number of results have appeared in the literature, e.g. [23,12,21,7,28,18], focusing on watermarking cryptographic programs/functions, proposing different definitional frameworks and watermarking schemes which are secure under a wide range of assumptions. More specifically, a number of results focus on watermarking classes of *pseudorandom functions*, such as [12,21,10,32,28,33], while a number of works address the problem of watermarking public-key cryptographic functionalities, i.e. [12,32,18,7,6].

In the context of this thesis, as indicated by the title, we focus on watermarking *public-key* cryptographic functions. The use of this term essentially refers to watermarking public key cryptographic functionalities, notable examples of which are public key encryption and digital signatures. At a high level, watermarking a public-key cryptographic functionality enables the embedding of a mark in the instance of the secret-key algorithm such that the functionality of the original scheme is maintained. At the same time, it should be infeasible for an adversary to remove the mark (unremovability) or mark a fresh object without the marking key (unforgeability). We tackle this problem by proposing a new definitional framework for watermarking cryptographic functionalities, and

providing a various watermarking constructions for public key encryption and digital signatures.

Furthermore, motivated by the fact that watermarking in general binds a digital object with some specific information, this thesis also studies a related but different problem, the problem of the design of a Public key Infrastracture (PKI). PKIs are of significant importance in securing communications across the internet, as they aim to ensure that public keys used for communication indeed belong to the claimed entities. In practice, PKIs are handled by trusted entities called Certification Authorities (CAs) which ensure the authenticity of an identity-public key binding, so called a *digital certificate*, by signing the digital certificate with their signing key. A Certification Authority (CA) is responsible for maintaining a database with so called digital certificates, i.e. essentially identity-public key pairs signed by the CA, registering new certificates and revoking old ones. In this thesis, we approach the problem of the design of a PKI by presenting a generic construction which depending on its instantiation may avoid relying on trusted entities such as CAs.

2 Contributions

The contribution of this thesis is two-fold. In the core part of the thesis we study the problem of watermarking cryptographic functionalities, where the focus is turned on watermarking public key encryption and digital signatures functionalities. Then, we study the problem of the design of a Public Key Infrastracture.

On the side of watermarking cryptographic functionalities, the initial motivation behind our work goes back to the impossibility result of Barak et al. [9] and the subsequent work of Cohen et al. [12]. Cohen et al. [12], by modifying the definitions of Barak et al. [12], propose a watermarking scheme for a class of pseudorandom functions (PRFs) called puncturable PRFs. Their watermarking scheme is based on iO and it provably satisfies *unremovability* property. In the same work, the authors define the notions of "Watermarkable Public-key Encryption" and "Watermarkable Signatures" and describe how schemes can be constructed by utilizing iO. However, to the best of our knowledge, all current candidate constructions for iO (e.g. [17]) are based on non-standard multilinear map group assumptions. Therefore, a question that was left open by [12] and which we address in this thesis, is the following:

Can we watermark public key cryptographic functionalities efficiently from standard assumptions?

In this thesis, we tackle this question by rethinking the definitional framework for watermarking public key cryptographic functionalities and answer this question in the affirmative but in a slightly different model than the watermarking model proposed in [12]. Our contributions both at the definitional and the constructional level are summarized as follows:

1. We approach cryptographic watermarking, by making a relaxation and refinement to the model considered in previous works (e.g. [12]), which we argue that is suitable for public key cryptographic primitives and maintains all the relevant to practice features that the previous formulations enjoyed. The basic characteristics of our model are the following: First, a small shared state is allowed between the marking and detection procedures, which is publicly available and can be potentially maintained in a distributed ledger. Second, an authority responsible for watermarking programs/circuits. Instead of embedding a message in a specific circuit which is given as input (e.g. as in [9,23,12]), it samples itself a circuit embedded with a message of the client's choice, or in the simpler case, it just samples a marked circuit. Specifically, the watermarking authority embeds a message in the secret key algorithm for the functionality (e.g. decryption algorithm) and returns it to the client along with the corresponding public key algorithms of the functionality (e.g. encryption algorithm). We note that this formulation is consistent with the specific definitions for "Watermarkable Public Key Encryption" and "Watermarkable Signatures" suggested in [12].

A further refinement of our model is distinguishing between the notions of watermarking cryptographic functionalities and cryptographic implementations. Intuitively, the former notion captures constructions for public key cryptographic primitives that satisfy the basic properties of watermarking. The latter notion, considers as a starting point a specific cryptographic implementation, for example El Gamal encryption scheme [14], and aims to capture what would it mean to watermark it. In plain words, a marking service may want to watermark, say, ElGamal public key encryption scheme because this specific implementation of public key encryption is the one that is standardized, backwards compatible, or sufficiently efficient for the context within which the cryptographic system is used. The watermarking model presented in this thesis is a unified approach of the models considered in the papers [7] and [6].

2. We propose watermarking constructions for public key encryption and digital signatures which we prove as secure based on standard hardness assumptions. We note that in our constructions, one bit of information is embedded in a decryption or signing circuit, in other words, circuits are either *marked* or *unmarked*. Below, we list the constructions that are presented in this thesis.

- We present a watermarking construction which shows how to watermark *any* public key encryption scheme which is correct and IND-CPA secure. Along the same lines, we provide a watermarking construction which shows how to watermark any digital signature scheme which satisfies unforgeability (in the sense of EU-CMA security) and an additional property called *verification soundness*. We show that verification soundness is satisfied by standard digital signature schemes, such as Schnorr's signatures [29]. The basic characteristics of the above constructions are the following: First, they require a shared state between the marking and detection procedure which is of logarithmic size with respect to the number of issued marked circuits. Second, they support private detection in the sense that only the entity possessing the detection key can decide whether a circuit is marked or not.

– On the side of watermarking public key cryptographic functionalities, we show how to obtain watermarkable public key encryption and watermarkable digital signature schemes from Identity-based encryption (IBE) [30] and Identity-based signatures (IBS) [30] respectively. Those constructions achieve public key detection of the watermark while the size of state remains logarithmic.
– Finally, in a different spirit than previous constructions, we present a watermarking construction for public key encryption functionality which improves upon our previous constructions for public key encryption in terms of efficiency. Specifically, we present a Watermarked Public key Encryption scheme which is based on Paillier's public key encryption scheme [24] and does *not* require a shared state to be maintained between the marking and detection algorithms.

On the side of the design of a Public key Infrastructure, our work is motivated by the recent advances in *blockchain* technologies. Bitcoin [22], a decentralized currency which implements a distributed append-only ledger of transactions, has opened the discussion for the design of various decentralized applications beyond the realization of a digital currency. Blockchain-based solutions to the design of a PKI and identity-management systems in general, have appeared in the literature as alternative approached to centralized solutions, such as CA-based PKIs. In particular, Namecoin [3], is the first blockchain-based Domain Name System (DNS) which associates domains with IP addresses, by supporting registration of domains and updates of IP addresses. Emercoin [1] is similar as a concept with Namecoin with the difference of handling identity, public key mappings. The concept behind Namecoin [3] and Emercoin [1], is to employ the blockchain for storing, retrieving and verifying identity, public key mappings (or other mappings). A similar approach is followed by other systems that have appeared in the literature, such as Blockstack's BNS [5], Ethereum Name Service [2].

In a different spirit, Certcoin [15,16], aiming at resolving the inefficiency previous approaches (e.g. Namecoin, Emercoin), where checking the validity of a mapping would require to traverse the entire blockchain, relies on the following idea: Certcoin utilizes different components for the storage of an identity-public key mapping and the verification of an identity-public key mapping. An authenticated distributed hash table (DHT) is the component that is employed for the storing identity-public key mappings, while the blockchain will be the component utilized for verification of whether a mapping is valid or not. Verification is performed by employing *cryptographic accumulators*, data structures which enable a compact representation of a set of elements (identity-public key mappings in this case) that allows membership and sometimes non-membership tests.

In this thesis, we present a generic construction for a PKI which further explores both the use of cryptographic accumulators and the idea of employing different components for storing and verifying the validity of certificates. Our generic construction, depending on the instantiation of each of its components, may be suitable, but not only, for a realization on top a blockchain system and in particular on top of a smart contract platform. Furthermore, we

focus on providing a formal model for defining security of a decentralized PKI which is missing from Certcoin [16] and Namecoin [3]. We choose to define security of a (decentralized) PKI along the lines of the Universal Composability framework [11]. We note that the scope of this thesis is not to provide a specific realization of a decentralized PKI, but a general methodology which, depending on its instantiation, could be used as an alternative to the CA-based PKIs.

3 Our watermarking model

Notation. By $\lambda \in \mathbb{N}^*$ we will denote the security parameter of a scheme. By C we denote a circuit which is unmarked and by $\widetilde{C}$ we denote a circuit which is marked. The abbreviation PPT stands for Probabilistic Polynomial Time.

At a high level, the notion of a cryptographic functionality aims to capture a cryptographic primitive in an abstract ideal way, focusing on the properties it should satisfy and it is defined by a number of algorithms and the properties that should be satisfied. On the other hand, the notion of a cryptographic implementation intends to describe a specific scheme that satisfies the properties of the functionality.

Entities and Syntax. The entities that are involved in a watermarking scheme are a number of users and an entity called "Marking Service" (MS), which is considered trusted. We denote with $\mathcal{T}$ the space of messages to be embedded in a circuit, which we will be also called tags. A (stateful) watermarking scheme for a cryptographic functionality $\mathcal{C}_{\mathcal{F}}$, which is defined by a parameter m and n properties, is comprised by a triplet of algorithms $(\mathsf{WGen}, \mathsf{Mark}, \mathsf{Detect})$ which are defined as follows:

- WGen : On input 1^λ, it outputs public parameters $params$ and a pair of keys (mk, dk). The marking key, mk, is only known to MS, and the detection key dk may be either public or private depending on whether the scheme allows public or private detection. It also initializes a public variable $state$ as empty which can be accessed by both the Marking Service and the clients, but can only be modified by the Marking Service.
- Mark : On input mk, $params$, $\tau \in \mathcal{T}$ (which is chosen by the client) and $state$, it outputs a tuple of circuits $(\widetilde{C}_1, C_2, \ldots, C_{m-1})$ and an efficiently sampleable and representable distribution $\mathcal{D}$ on the inputs of the circuit $\widetilde{C}_1$. It also updates $state$.
- Detect : On input dk, $params$, $state$ and a circuit C^*, it outputs a tag $\tau' \in \mathcal{T}$ or $\mathsf{unmarked}$.

Remark. We note that the variable $state$ should be immutable, in the sense that only the Marking Service is allowed to modify it. In addition, the distribution $\mathcal{D}$ is related to the definitions of "closeness" and "farness" relations between circuits, which are required for defining some of the properties of a watermarking scheme.

Properties. We distinguish the properties that should be satisfied by a watermarking scheme into *correctness properties* and *security properties*. Each property is defined by employing a security game between an adversary and a challenger, where the latter acts on behalf of MS. In all the games, at a first stage, the adversary can obtain some information by issuing queries to the Challenge, Corrupt and Detect oracles. Briefly, those oracles perform as follows: The Challenge oracle on input a tag τ calls the Mark algorithm on input τ and returns the tuple of circuit output by Mark by excluding the first circuit, i.e. the circuit which is considered as marked. The marked circuit can be received by an adversary through a query to the Corrupt oracle. The Detect oracle receives as input a circuit of the adversary's choice and returns either a message τ or $\mathsf{unmarked}$.

The correctness properties of a watermarking scheme are called detection correctness and functionality property-preserving and they are informally defined as follows:

- *Detection correctness*: if a circuit is marked with a specific message τ by the algorithm Mark, then when Detect is invoked on input that circuit, it will return τ with overwhelming probability.
- *Functionality property-preserving*: A watermarking scheme for a cryptographic functionality $\mathcal{C}_{\mathcal{F}}$ should preserve the fundamental properties of the functionality. For example, a watermarking scheme for the Public Key Encryption (PKE) functionality should satisfy correctness and IND-CPA security, where IND-CPA security is one of the standard notions of security of a PKE scheme.

The security properties that should be satisfied by a watermarking scheme are called unremovability and unforgeability. Informally, they are defined as follows:

- *Unremovability*: No PPT adversary $\mathcal{A}$ after querying the Challenge, Corrupt and Detect oracles, should be able to output a circuit C^* which is "close" to any of the received marked circuits, and at the same time it is unmarked or it is marked under a different than the original message.
- *Unforgeability*: No PPT adversary $\mathcal{A}$ should be able to output a marked circuit which is "far" from any of the marked circuits that the adversary has received by issuing queries to the Corrupt oracle.

What remains to be defined are the notions of "closeness" and "farness" between circuits. Assume a circuit $C : \mathcal{X} \times \mathcal{R} \rightarrow \mathcal{Y}$, where $\mathcal{R}$ the randomness space. We denote as $\mathsf{Out}\{C(x)\} = \{C(x;r)\}_{r \in \mathcal{R}}$.

- Let C_1, C_2 be two circuits. We say that C_1 is ρ-close to C_2 with respect to a distribution $\mathcal{D}$ if $\underset{x \leftarrow \mathcal{D}}{\mathsf{Pr}}[C_1(x) \in \mathsf{Out}\{C_2(x)\}] \geqslant \rho$.
- Let C_1, C_2 be two circuits. We say that C_1 is γ-far from C_2 with respect to a distribution $\mathcal{D}$ if $\underset{x \leftarrow \mathcal{D}}{\mathsf{Pr}}[C_1(x) \notin \mathsf{Out}\{C_2(x)\}] \geqslant \gamma$.

Based on the definitions above, unremovability and unforgeability properties are defined with respect to the parameters ρ and γ, respectively. Thus they are formulated as ρ-unremovability and γ-unforgeability.

4 Watermarking constructions

In this section, we provide a general idea about the watermarking constructions that are presented in the thesis. In section 4.1, we describe watermarking constructions for the case of public key encryption and digital signatures with the following features: First, the size of *state* is logarithmic with respect to the number of decryption or signing circuits issued by the Mark algorithm. Second, the running time of the Detect algorithm is linear with respect to the number of marked circuits. Third, one bit of information is embedded into a decryption or signing circuit, in the sense that circuits are considered as marked or unmarked. Then, in section 4.2, we describe a watermarking scheme for the public key encryption functionality which does not require a shared state. In addition, the running time of the Detect algorithm does not depend on the number of marked decryption circuits which are generated by Mark.

4.1 Watermarking constructions with shared state

Assume that $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a PKE scheme which is correct and IND-CPA secure. In Figure 1, we present a watermarking scheme for the PKE scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

Theorem 1. *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an implementation of the public key encryption functionality that has plaintext space* $\mathcal{M}$ *of exponential size (in the security parameter) and satisfies perfect correctness and IND-CPA security. Let* $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^\ell$ *be a pseudorandom function, where* $\mathcal{K}$ *is the key space. Then, the scheme in Figure 1 is a watermarking scheme for the implementation* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*. Namely, it is implementation property-preserving with negligible error, it satisfies detection correctness,* ρ*-unremovability and* γ*-unforgeability, where* $\rho \geqslant \frac{1}{\lambda^c}$*,* $\gamma \geqslant 1 - \frac{\rho}{2}$*, for a constant* $c \geqslant 1$*.*

Along the same lines, we show how to watermark a digital signature scheme comprised by a triplet of algorithms $(\mathsf{SigGen}, \mathsf{Sign}, \mathsf{Verify})$. However, for the case of digital signatures, we additionally require that the original digital signature scheme satisfies an additional property called *verification soundness*. At a high level, verification soundness requires that any PPT adversary which is given a pair of signing-verification keys should not be able to output a valid message-signature pair where the signature is not a possible value of the signing algorithm for this particular message and signing key. We prove that this property is satisfied by well-known signature schemes such as Schnorr's scheme [29].

Watermarking schemes for public key encryption and digital signature functionalities from IBE and IBS respectively. In an IBE or an IBS scheme the public key is an identity of the user's choice, while the secret key (either the decryption or signing key) is computed by an entity called a *Private Key Generator*, based on the identity chosen by the user. Based a similar idea with the construction presented in Figure 1, we show how to watermark the public key encryption and

– WGen: On input 1^λ, it chooses uniformly at random a key K for a pseudo-random function $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^\ell$. It outputs $mk = dk = K$ and initializes the public variable $state \leftarrow 0$ and $i \leftarrow 0$. It chooses a parameter $\rho \geqslant \frac{1}{\lambda^c}$, for a constant $c \geqslant 1$.
– Mark: On input K, $state$, marked, it sets $i \leftarrow state + 1$ and runs $\mathsf{Gen}(1^\lambda)$ with randomness $F(K, i)$. The output is a public-secret key pair (pk_i, sk_i) and the algorithm returns a pair of circuits $(\mathsf{Dec}_{sk_i}, \mathsf{Enc}_{pk_i})$. It sets as $\mathcal{D}_i$ the distribution of the ciphertexts that correspond to plaintexts chosen uniformly from the plaintext space. Then, it updates the value of $state$ by setting $state \leftarrow state + 1$.
– Detect: On input $dk = K$, a circuit C and $state$, it runs as follows.
  For $i = 1$ to $state$:
  - Run $\mathsf{Gen}(1^\lambda)$ with randomness $F(K, i)$ (as the Mark algorithm does) in order to obtain (pk_i, sk_i).
  - Choose $k = 8\lambda/\rho^2$ plaintexts uniformly at random from the plaintext space $\mathcal{M}$, i.e. $m_1, \ldots, m_k$. Then, encrypt them under the public key pk_i by computing the ciphertexts $\{c_j = \mathsf{Enc}(pk_i, m_j)\}_{i=1}^k$.
  - For $j = 1$ to k check whether $C(c_j) = m_j$. If C decrypts correctly at least $6\lambda/\rho$ out of $k = 8\lambda/\rho^2$ ciphertexts $c_1, \ldots, c_k$, return *marked*.

  Otherwise, return *unmarked*.

Fig. 1: A watermarking scheme $(\mathsf{WGen}, \mathsf{Mark}, \mathsf{Detect})$ for a PKE scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

digital signature functionality based on IBE and IBS respectively. These constructions allow *public detection*, i.e., any entity based only on public parameters can decide whether a circuit is marked or not. We note that for the case of digital signatures, it is required that the IBS scheme, additionally to its standard properties, to satisfy *verification soundness*.

4.2 A watermarking construction without shared state

In the watermarking scheme described in Figure 1, the core idea for deciding whether a (decryption) circuit is marked or not, was to be able to re-generate a number of valid ciphertexts for each public key issued by the Marking Service. Assume for example a public key encryption scheme where one could sample ciphertexts which are valid (or look valid) for any public key and when such a ciphertext is given as input to a (decryption) circuit, some information can be revealed regarding the secret key that has been used to decrypt the ciphertext. Assume that by providing such a ciphertext as input to a circuit C, one could reconstruct a public key indicating that the corresponding secret key was used to decrypt the ciphertext. In that case, deciding whether C is marked or not would require to decide whether the reconstructed public key is one of the keys that have been previously issued by MS. This requires some additional information

to be stored either as part of the detection key or *state*, as for example all the public keys that have been generated so far.

We resolve this issue by following a different approach. The watermarking scheme that we present is based on the PKE scheme proposed by Paillier [24], which exploits the structure of the group $\mathbb{Z}^*_{n^2}$ where $n = pq$ and p, q are prime numbers. In our scheme, p, q are *safe primes*, i.e. they are of the form $p = 2p' + 1$ and $q' = 2q' + 1$, where p', q' primes. Intuitively, our watermarking scheme performs as follows. The Marking Service embeds some authenticated information as part of the secret and public keys, which can only be recovered by using the detection key, which is private and includes the factorization of n. By providing some valid looking ciphertexts as input to a (decryption) circuit C, Detect, by applying some computation on the result returned by C, can recover the authenticated information and reconstruct a public key. Based on the authenticated information that has been extracted, the Marking Service can decide whether the reconstructed public key has been previously issued by her and thus identify the circuit as either marked or unmarked. We prove that this construction is a watermarking scheme for the PKE functionality by relying on the Decisional Composite Residuosity (DCR) assumption [24] and Decisional Diffie Hellman assumption for square n-th residues ($\mathsf{DDH}_{\mathsf{SQNR}}$) [20].

5 A generic construction for a PKI

In this section, we present the main concepts behind the generic construction for a PKI that is presented in this thesis. The results are part of the paper [25].

The main building block of our construction is a *public-state additive universal accumulator*, an accumulator which has the following properties: First, all the operations, besides the key generation algorithm, can be performed based exclusively on public information. This implies that any party can both perform and check operations on her own. Second, the accumulator is *additive*, meaning that it only supports additions of elements and not deletions. Third, the accumulator is *universal*, meaning that one, holding a membership or non-membership witness, can check whether an element belongs to the accumulated set or not, accordingly.

Our construction, which utilizes two public-state additive universal accumulators, supports the following operations: (i) registration of an identity-public key pair, (id, pk), (ii) revocation of an (id, pk) pair, (iii) verification of whether a pair (id, pk) is valid and (iv) retrieval of a public key pk given a specific identity id. The protocol considers two different components which are described as ideal functionalities, along the lines of the UC framework [11]. The first functionality, denoted as $\mathcal{F}_{\mathsf{TP}}$, receives as input a program P and on any input performs any computation of the program P on this input. $\mathcal{F}_{\mathsf{TP}}$ maintains a state which is updated every time a computation is performed. Every computation performed by $\mathcal{F}_{\mathsf{TP}}$, is publicly available as it will be defined by the interaction of $\mathcal{F}_{\mathsf{TP}}$ with an adversary. In our PKI construction, $\mathcal{F}_{\mathsf{TP}}$'s state will consist of the accumulator values and the program P will define how the state will be updated after

any register or revoke operation. Our objective is $\mathcal{F}_{\mathsf{TP}}$ to be realized in practice by any smart contract platform which can run arbitrary Turing complete programs, such as Ethereum platform [31]. A second functionality, called the "unreliable database functionality" and denoted as $\mathcal{F}_{\mathsf{UDB}}$, is employed for the storage of (identity, public key) mappings together with information relevant to our protocol. This functionality is called "unreliable" because an adversary is allowed to modify its contents. Our protocol is *hybrid*, in the sense that parties are making calls to the ideal functionalities $\mathcal{F}_{\mathsf{TP}}$ and $\mathcal{F}_{\mathsf{UDB}}$.

We model the security of a decentralized PKI in the UC model by defining an ideal functionality $\mathcal{F}_{\mathsf{DPKI}}$. Subsequently, we show that our generic construction is secure by proving that it securely realizes the ideal functionality $\mathcal{F}_{\mathsf{DPKI}}$.

References

1. Emercoin. `https://emercoin.com`.
2. Ethereum name service. `https://ens.domains/`.
3. Namecoin. `https://namecoin.org/`.
4. André Adelsbach, Stefan Katzenbeisser, and Helmut Veith. Watermarking schemes provably secure against copy and ambiguity attacks. In *ACM workshop on Digital rights management*, 2003.
5. M. Ali, J. Nelson, R. Shea, and M.J. Freedman. Blockstack: A global naming and storage system secured by blockchains. In *USENIX Annual Technical Conference (ATC)*, 2016.
6. Foteini Baldimtsi, Aggelos Kiayias, and Katerina Samari. Watermarking probabilistic circuits: The case of digital signatures. *Proceedings of BalkanCryptSec 2018*.
7. Foteini Baldimtsi, Aggelos Kiayias, and Katerina Samari. Watermarking public-key cryptographic functionalities and implementations. In *Information Security - 20th International Conference, ISC*, 2017.
8. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, 2001.
9. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2), 2012.
10. Dan Boneh, Kevin Lewi, and David J. Wu. Constraining pseudorandom functions privately. In *PKC*, 2017.
11. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, 2001.
12. Aloni Cohen, Justin Holmgren, Ryo Nishimaki, Vinod Vaikuntanathan, and Daniel Wichs. Watermarking cryptographic capabilities. In *STOC*, 2016.
13. Ingemar J Cox, Matthew L Miller, Jeffrey Adam Bloom, and Chris Honsinger. *Digital watermarking*, volume 1558607145. Springer, 2002.
14. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, 1985.
15. Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. Certcoin: A namecoin based decentralized authentication system. `https://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf`, 2014.

16. Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. A decentralized public key infrastructure with identity retention. *IACR Cryptology ePrint Archive*, 2014.
17. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, 2013.
18. Rishab Goyal, Sam Kim, Nathan Manohar, Brent Waters, and David J. Wu. Watermarking public-key cryptographic primitives. In *CRYPTO*, 2019.
19. Nicholas Hopper, David Molnar, and David Wagner. From weak to strong watermarking. In *TCC*, 2007.
20. Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Group encryption. In *ASIACRYPT*, 2007.
21. Sam Kim and David J. Wu. Watermarking cryptographic functionalities from standard lattice assumptions. In *CRYPTO*, 2017.
22. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `http://bitcoin.org/bitcoin.pdf`.
23. Ryo Nishimaki. How to watermark cryptographic functions. In *EUROCRYPT*, 2013.
24. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, 1999.
25. Christos Patsonakis, Katerina Samari, Mema Roussopoulos, and Aggelos Kiayias. Towards a smart contract-based, decentralized, public-key infrastructure. In *CANS*, 2017.
26. Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4), 2001.
27. Vidyasagar M Potdar, Song Han, and Elizabeth Chang. A survey of digital image watermarking techniques. In *INDIN*, pages 709–716. IEEE, 2005.
28. Willy Quach, Daniel Wichs, and Giorgos Zirdelis. Watermarking prfs under standard assumptions: Public marking and security with extraction queries. *TCC*, 2018.
29. Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3), 1991.
30. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, 1984.
31. Gavin Wood. Ethereum: A secure decentralized generalized transaction ledger. `http://gavwood.com/paper.pdf`.
32. Rupeng Yang, Man Ho Au, Junzuo Lai, Qiuliang Xu, and Zuoxia Yu. Collusion resistant watermarking schemes for cryptographic functionalities. *IACR Cryptology ePrint Archive*, 2017.
33. Rupeng Yang, Man Ho Au, Junzuo Lai, Qiuliang Xu, and Zuoxia Yu. Unforgeable watermarking schemes with public extraction. *SCN*, 2018.