The DecCert PKI: A Solution to Decentralized Identity Attestation and Zooko's Triangle

1st Sam A. Markelon
University of Florida
smarkelon@ufl.edu

2nd John True
University of Florida
jtrue15@ufl.edu

***Abstract*—We propose DecCert, a decentralized public key infrastructure designed as a smart contract that solves the problem of identity attestation on public blockchains. Our system allows an individual to bind an identity to a public blockchain address. Once a claim of identity is made by an individual, other users can choose to verify the attested identity based on the evidence presented by an identity claim maker by staking cryptocurrency in the DecCert smart contract. Increasing levels of trust are naturally built based upon the amount staked and the duration the collateral is staked for. This mechanism replaces the usual utilization of digital signatures in a traditional hierarchical certificate authority model or the web of trust model to form a publicly verifiable decentralized stake of trust model. We also present a novel solution to the certificate revocation problem and implement our solution on the Ethereum blockchain. Further, we show that our design solves Zooko's triangle as defined for public key infrastructure deployments.**

***Index Terms*—identity attestation, public key infrastructure, Zooko's triangle, decentralization, blockchain**

I. Introduction

The openness and pseudo-anonymous (and for some deployments actually anonymous) nature of blockchain based systems are an oft desired property of the technology. A user simply needs to generate a public, private key pair (wallet) per the protocol specification to interact with the system, while providing no direct identifying information. However, for many decentralized applications it is essential that the identity of a user can be verified. This could be due to necessity of complying with local regulations revolving around the application (know your customer, age restrictions, etc.), to prevent fraud, or to prevent Sybil attacks (see subsection II-C). For applications deployed on private blockchains the solution to such a problem is trivial. However, on public blockchains the solution requires one to solve a complex decentralized identity attestation problem. This is the problem of verifying a user's claim of identity in a trustless environment.

Applications like Civic [1] use centralized repositories to verify user provided credentials (usually a form of government issued identification) to tie the claimed identity of a user to a wallet address (public key) on the blockchain by stamping a proof of said verification on chain. Not only are there potential privacy issues with such a solution, but government issued identifications are relatively easy to steal or forge in the physical world. This can lead to fraudulent claims of identity being verified. Such an occurrence can be particularly devastating when a user leverages their identity and likeness for value on a particular decentralized application, like minting a non-fungible token collection. Therefore, we propose a solution based on decentralized public key infrastructure. Such a system prevents fraud by proving the claim of identity via links to well known online presences and having users verify said claims by first examining these links, checking their validity, and finally by staking capital in a certificate that binds a wallet address to the claimed identity. In doing so, outside observers can gain trust that a particular wallet actually belongs to a claimed identity by examining the amount staked in a particular certificate.

Decentralized public key infrastructure (PKI) as an area of active research has been motivated by the failings of traditional centralized PKI. Centralized PKI, particularly those using the hierarchical certificate authority model, has suffered from highly damaging attacks that exploit the single point of failure issue with any centralized system. Such examples include the DigiNotar incident [2] and the Comodo hack [3]. Moreover, swift and dependable certificate revocation remains a largely unsolved problem with traditional centralized PKI deployments [4].

The emergence of blockchain technology has lead to a recent flurry of research on decentralized PKI. Blockchain technologies offer an immutable ledger which is decentralized and publicly verifiable, along with a distributed consensus mechanism for appending data to said ledger. The hope is that high performance and secure PKI can be implemented on the blockchains by bringing certificates and PKI functionality on to these trustless decentralized systems. In turn, leading to wide adoption and replacement of the outdated and insecure traditional PKI models.

In this paper we present DecCert, a novel decentralized PKI that allows users to bind identities to wallet addresses in a publicly verifiable way. We make a number of of contributions in this paper: (i) an attestation of identity trust model through staking, (ii) a token based revokation mechanism, (iii) a proposed use case of DecCert for preventing fraud on an NFT marketplace, and (iv) an analysis of our system through the lense of Zooko's triangle. We show that DecCert solves Zooko's triangle as it provides human-meaningful certificates that are both secure and decentralized.

II. Background

We provide background necessary to understand the motivation and design considerations for this work.

A. Public Key Infrastructure, PGP, and the Web of Trust

Public key infrastructure deployments are systems that allow users to create cryptographic certificates binding an identity to keys. PKI is essential for modern web security, allowing for the secure transport of data. Modern web PKI uses centralized hierarchical certificate authorities (CAs) for authenticating the validity of certificates. For a primer we point the reader to [5].

Pretty Good Privacy (PGP) [6], invented by Phil Zimmerman in 1991, is an encryption program that makes use of certificates for authentication of identity. Rather than a CA, PGP introduced a concept known as the *web of trust* for solving identity attestation. The web of trust is a decentralized network of signatures over certificates that function to endorse the claimed identity on a certificate. The more signatures over an unknown certificate and the closeness of these signatures to your own certificate plus your own certificate's signatures in the network graph lead to increase trust in the claim of identity presented in the unknown certificate. However, PGP still requires a centralized key server to store certificates, the signatures over those certificates, and resolve searches for certificates. This is as opposed to the system we propose, where certificate storage and searching for certificates is done entirely by interacting with a trustless blockchain. Moreover, the identity attestation provided by PGP is weak as a claim of identity is only tied to an email address, and there is nothing preventing a user from creating multiple fake certificates.

B. Keybase

A modern update to the traditional PGP system is the work done by Keybase [7]. The service functions as a centralized key server that maps cryptographic keys to social media presences such that these maps can be publicly audited. Keybase refers to a valid mapping as a proof. We use this terminology in our design, although our mechanism to verify a valid proof is at this stage more rudimentary due to the limitations of smart contracts.

C. Sybil Attacks

Large-scale peer-to-peer systems face security threats from hostile remote computing elements. To resist these threats, many such systems employ redundancy. However, if a single adversarial entity can present multiple identities, it can control a substantial fraction of the system, thereby undermining this redundancy. One approach to preventing these so called "Sybil attack" is to have a trusted agency certify identities. The 2002 work by Douceur shows that without a logically centralized authority, Sybil attacks are always possible except under extreme and unrealistic assumptions of resource parity and coordination among entities [8].

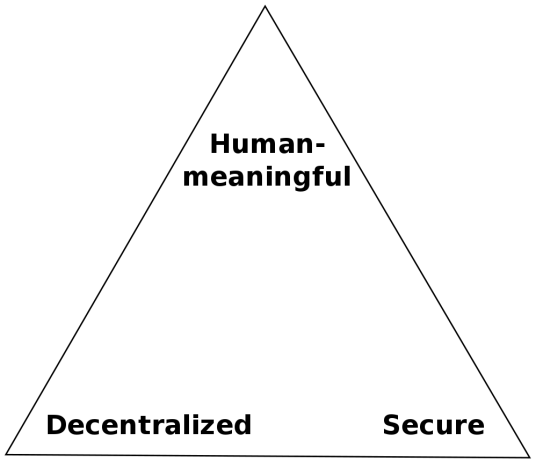

Fig. 1. Zooko's triangle [9]

D. Zooko's Triangle

Zooko's triangle is the conjecture by Zooko Wilcox-O'Hearn that an identifier in a network protocol can achieve only at most two of the three of the following properties: human-meaningful, secure, decentralized [10]. This trilemma is presented graphically in figure 1.

We can expand upon these properties with respect to PKI systems. We then use these definitions to give us a common frame of reference when evaluating various deployments.

- **Human-meaningful**: Certificates can be viewed such that have human identifiable and recognizable names bound to keys.
- **Secure**: We define security as a number of sub-properties. A PKI system must achieve at least the baseline property to be considered at all secure.
    1) **Baseline**: Identifiers (a name, a url, etc.) are bound to (associated with by principle of being on the same certificate) strong modern cryptographic keys such that after receiving a certificate of some entity a user can securely exchange information with said entity. This is is course done by using the key gotten from the certificate to encrypt and authenticate communication between user and entity. Moreover, certificates can be revoked by the owner of the certificate.
    2) **Identity Retention**: Baseline security is achieved with the addendum that identifiers are *uniquely* bound to keys.
    3) **Revocation Under Key Loss**: Certificates can be revoked even in the face of key loss or theft.
    4) **Verifiable Identity Attestation**: Baseline security is achieved with the addendum of publicly verifiable identity attestation. That is to say that an implicit claim of identity (based on the identifier an entity selects when creating a certificate) can be verified to a high degree of certainty to belong to the entity

that owns and is well recognized by said identifier by any outside observer.

- **Decentralized**: Certificates can be indexed and searched by users of the system without the need for a centralized authority.

Zooko's conjecture states that a PKI system could not achieve all three of the above properties simultaneously. There are a number of (decentralized) PKI systems that have demonstrate this conjecture to be false, but our work, DecCert, is the first to achieve this with verifiable identity attestation, optional identity retention, and revokation under key loss.

III. Related Work

We examine a number of other blockchain based decentralized PKIs. In particular we analyze them with respect to the PKI properties of Zooko's triangle. In table I we summarize the properties of each of these works (along with traditional web CA deployments, PGP, and Keybase) in comparison to DecCert.

A. Certcoin

Certcoin, introduced in 2014, was the first fully-fledged PKI system to be implemented on a blockchain [11]. It was built on top of the Namecoin blockchain and guaranteed identity retention. PGP fails at identity retention because anyone can register any multiple times, and traditional CAs fail at identity retention as with multiple CAs existing multiple public keys can be mapped to a single unique identifier. Keybase, on the other hand, does provide identity retention.

While Namecoin does guarantee unique identifiers and Certcoin on top of Namecoin maps these uniquely to a public key, they fail to build trust in claimed identities represented by said identifiers, as it is an open enrollment system. For instance anyone could register the identifier corresponding to "Bill Gates", but there is no certainty if this would actually be the famed businessman and technologist. Moreover, the revocation system put forth in Certcoin fails to work if private keys are totally lost.

B. Bitcoin PGP

In 2015 Wilson and Ateneise tried to replace the traditional web of trust model with a block based model [12] . In this paper the authors propose a system to improve the web of trust mechanism used by PGP by replacing traditional digital signatures with micro-Bitcoin transactions.

The work runs along the same vein as ours by using blockchain mechanisms to increase trust of claimed identity, however it lacks some of the fundamental economic reasoning central to the problem. For instance it is not clear why a non-staked Bitcoin transaction is any more trustworthy than a digital signature, considering the endorser in the proposed system is rewarded monetarily for sending a transaction to any Bitcoin PGP certificate, rather than facing repercussions for promoting a potentially bad actor. Moreover, any user can register any identity so no claim of identity can be trustlessly validated.

C. Blockstack

Blockstack [13] is based on lessons learned from actual blockchain PKI deployments. The authors point out that Namecoin, the blockchain used by Certcoin, is vulnerable to a 51% attack as there exists a single miner entity. This, plus network reliability issues on the Namecoin blockchain, prompted the authors to implemented a PKI on the Bitcoin blockchain.

The authors claim that both Namecoin and Blockstack solve Zooko's triangle for PKI systems. The human-meaningful identifiers and decentralization of both systems are obvious. However, the authors only solve the trilemma for the baseline definition of security. Moreover, certificate revocation remains a largely unaddressed problem in the system.

D. SCPKI and DBPKI

SCPKI [14] by Al-Bassam takes the same approach as ours by implementing a PKI as a smart contract on the Ethereum blockchain. The system acts as alternative to traditional web CAs. Like other related works the system fails to account for the verification of claims of identity and does not address the revocation problem in the face of compromised private keys.

Toorani and Gehrmann extended this body of work with DBPKI [15]. Their system touts efficient lookup of verification of identities and revoked certificates by using a distributed accumulator approach. While the system offers performance improvements over prior work, like the others they fail to address question of identity verification. In addition they fail to provide clear deployment or use case details.

IV. DecCert

DecCert at its core acts as a decentralized key index when compared to a traditional key server used in systems like PGP. Under the hood it is implemented as a smart contract that maps identities to blockchain wallet addresses, publicly viewable online presences, stake information, fraud tags, and crucially a revocation token and revoked field. The general scheme described below can be implemented and deployed on any blockchain that supports smart contracts.

The entire state of the DecCert smart contract is publicly viewable on the underlying blockchain that the contract is deployed on. In this way by either viewing the raw data or using an indexer built on top of the smart contract, the system will allow individuals to view certificates and gain a level of trust on the claim of identity based on the proofs of identity, stake information, and fraud tags.

A. A DecCert Certificate

While stored as separate maps of *(address, certificate attribute)* pairs at the DecCert level we can deserialize all of entries in these maps belonging to a single address as a certificate belonging to that address along with the stake information and fraud tags associated with said certificate. At this level of abstraction a DecCert certificate (including the stake information plus fraud tags) consists of the following fields which are provided to (or initialized by) the contract at

PKI System	Human Meaningful	Baseline Security	Identity Retention	Key Loss Revocation	Identity Attestation	Decentralized
Traditional Web CA	✓	✓	✗	✗	✓	✗
PGP	✓	✓	✗	✗	✓	✗
Keybase	✓	✓	✓	✗	✓	✗
Certcoin	✓	✓	✓	✗	✗	✓
Bitcoin PGP	✓	✓	✗	✗	✗	✓
Blockstack	✓	✓	✗	✗	✗	✓
SCPKI	✓	✓	✗	✗	✗	✓
DBPKI	✓	✓	✗	✗	✗	✓
DecCert	✓	✓	✓*	✓	✓	✓

TABLE I

A COMPARISON OF THE PROPERTIES OF VARIOUS PKI SYSTEMS AS DEFINED BY ZOOKO'S TRIANGLE. *NOTE THAT DECCERT PROVIDES OPTIONAL IDENTITY RETENTION. FOR REASONS ADDRESSED LATER IT IS OFTEN NOT AN ACTUALLY DESIRED PROPERTY FOR MANY PKI DEPLOYMENT SCENARIOS.

the time of creation of a new certificate.

DecCert Cert

1) Public Wallet Address: a
2) Identity (name): n
3) Five or less publicly viewable online presence identity proofs: $p_1, p_2, \cdots, p_5$
4) Max stake amount: μ
5) Current amount staked: γ
6) Map of staker addresses to stake amount: $\overrightarrow{S}$
7) Set of fraud tags: $\mathcal{F}$
8) Revocation token: r
9) Revoked boolean value: x

In the following subsections we explain the function of each piece of data as well as present pseudocode algorithms for all of the functionality of the system. In further sections we give an overview and experimental results from an implementation of DecCert, an illustrative use case of our system in a hypothetical fully decentralized non-fungible token marketplace, and provide remarks about how DecCert solves Zooko's triangle for blockchain PKI systems.

B. Public Wallet Address

A user's public wallet address a is the wallet address (public key) on the blockchain that DecCert is deployed upon in which they want to transact under a known identity. The wallet address is also how certificates are indexed in our system as they are guaranteed to be unique.

C. Identity

This simple string field n is a name that ties an identity to a blockchain wallet address. An example could be a person's government name or a well-known and established online handle. This field ensures that participants in a blockchain protocol will have human-meaningful identifiers, the attestation of which can be verified.

D. Public Identity Proofs

Borrowing from the work of Keybase, we utilize identity "proofs" to establish a link between a claim identity on the blockchain n, a wallet address a, and online presences that link to the claimed identity. These online presences can take the form of Twitter, GitHub, Reddit, a YouTube channel, a website, etc. where the individual claiming such a link between n and a have established and well-known presences. We allow an individual to submit up to five proofs of identity that will appear in their certificate.

A proof p_i takes the form of a URL pointing to a location on one of these online presences that contains the following raw underlying proof.

Identify Proof

1) `Begin DecCert proof`
2) Public Wallet Address a
3) $\mathsf{HashFP}(S_k(\text{'DecCert Proof'}||a))$
4) `End DecCert proof`

We take HashFP to be a hash fingerprint, S_k to be a digital signature where k is the private key corresponding to a, and $||$ to be the standard string concatenation function.

We will later describe the process in which stakers utilize these proofs to build a stake of trust around a *proven* claim of identity or expose a claim as fraudulent.

E. Certificate Creation

We provide the pseudocode for the create certificate functionality in algorithm 1. We take $\overrightarrow{C}$ to be a map of addresses to existing certificates in the particular DecCert contract deployment, $\mathcal{N}$ to be the set of all identifiers used in said deployment, and cert to be the new DecCert certificate being created. Further we use the standard $object[field]$ syntax to assign and retrieve values from individual certificates cert and the global collection of certificates $\overrightarrow{C}$. We use this notation and syntax (along with the notation introduced in the DecCert certificate subsection) in subsequent algorithms.

The first line of the algorithm ensures identity retention for the system. However, if not desired this first require statements can be omitted. Identity retention is often not desired for a number of reasons. For instance natural identifier collisions exist among humans (many people share the same name), and

Algorithm 1 Create Certificate
Input: $a, n, \{p_1, \ldots, p_5\}, \mu, r$

Require: $n \notin \mathcal{N}$
Require: $\overrightarrow{C}[a]$ does not exist
Require: $\overrightarrow{C}[a][x] \neq \mathsf{true}$
  $\text{cert}[a] \leftarrow a$
  $\text{cert}[n] \leftarrow n$
  $\text{cert}[p_1] \leftarrow p_1, \ldots \text{cert}[p_5] \leftarrow p_5$
  $\text{cert}[\mu] \leftarrow \mu$
  $\text{cert}[\gamma] \leftarrow 0$
  $\text{cert}[\overrightarrow{S}] \leftarrow$ empty map
  $\text{cert}[\mathcal{F}] \leftarrow \emptyset$
  $\text{cert}[r] \leftarrow r$
  $\text{cert}[x] \leftarrow \mathsf{false}$
  $\overrightarrow{C}[a] \leftarrow \text{cert}$

in our system we can more deeply provide human-meaningful identifiers than a simple name alone by virtue of the identity proofs. Moreover, it may be desirable to have multiple certificates under the same identity with different blockchain addresses. A user may want to have multiple wallets linked to the same identity for both security and functionality reasons.

The second and third require statement is necessary as it ensure that someone is not trying to create a new certificate with an address that has an existing certificate bound to it or has been previously revoked.

F. Staking

Algorithm 2 Stake Certificate
Input: Address a of certificate to be staked, staker address s, the desired stake amount σ

Require: $\sigma > 0$
  $\text{cert} \leftarrow C[a]$
Require: $\text{cert}[x] \neq \mathsf{true}$
Require: $\sigma < \text{cert}[\mu]/40$
Require: $\text{cert}[\overrightarrow{S}][s][v] < \text{cert}[\mu]/40$
Require: $\text{cert}[\gamma] + \sigma < \text{cert}[\mu]$
  $g \leftarrow \text{cert}[\gamma]$
  $\text{cert}[\gamma] \leftarrow g + \sigma$
  $\upsilon \leftarrow \text{cert}[\overrightarrow{S}][s][v]$
  $\text{cert}[\overrightarrow{S}][s][v] \leftarrow \upsilon + \sigma$
  $\text{cert}[\overrightarrow{S}][s][t] \leftarrow$ current time

Algorithm 3 Remove Stake from Certificate
Input: Address a of certificate to be staked, staker address s

  $\text{cert} \leftarrow C[a]$
Require: $\text{cert}[\overrightarrow{S}][s][t] -$ current time > 6 months
  $\nu \leftarrow \text{cert}[\overrightarrow{S}][s][v]$
  $\text{cert}[\overrightarrow{S}][s][v] \leftarrow 0$
  SEND ν TO s

A max stake amount μ is set by the creator of a certificate. While there are no set guidelines for setting this value, a user that requires high trust from the public or will be doing high value transactions leveraging their identity will want this value to be relatively large.

Potential stakers can look for certificates where $\gamma < \mu$. They can then verify a claim of identity by looking at the identity proofs $p_1, \ldots, p_\ell$, making sure that the presence the proof exists on is definitively linked to the claimed identity, and verifying the hash fingerprint of the signature corresponds to a. The more proofs that are linked would lead to an increase of a potential staker's trust in the identity claim.

While online presences, like a Twitter account, can be compromised and a false proofs presented, it is unlikely that multiple online presences are compromised in unison. Moreover, the greater the duration a proof exists on an online presence gives a staker more confidence in the claim, as false proofs will be rapidly deleted after a compromised online account is recovered.

If desired and if the claim of identity has high likelihood of being true, a staker will then call a stake function and stake up to 2.5% of μ to function as a verification on the claim of identity. The cap of 2.5% is to ensure that we can have higher trust in a verified claim of identity by ensuring multiple parties are staking a claim of identity. While it is possible that an attacker could launch a Sybil attack against the system by creating multiple staker wallets and staking a number of times with these wallets, this is unlikely to occur for a high value identity with large μ due to capital constraints. Even if the expected value derived from the fraud is large enough to warrant such an attack, it is unlikely that an attacker would be able to produce valid online social presence proofs, thus thwarting such an attack.

The *(staker address, stake amount)* pair is then added to $\overrightarrow{S}$ and the amount staked is added to γ after it ensured the new value of γ will not exceed μ. The stake is then locked in the contract for a 6 month period. After this period a staker can removed their stake, freeing up room for other potential stakers, or choose to keep their stake in the contract. We present the pseudocode for staking in algorithm 2. We take v to be the amount staked in the staking map $\overrightarrow{S}$ corresponding to staker s, and t to be the time of the most recent staking for staker s. As seen in algorithm 3 a staker can only remove their stake after the locking period has expired from the time funds were last staked in a particular certificate.

A large quantity of individual stakers, and a high percentage of staked funds γ versus max stake amount μ naturally increases outside observers trust in the claim to identity. Staking can be profitable for the staker if a claim of identity seems valid. This will be illustrated in the NFT marketplace example in section VI.

G. Fraud Tagging

If a potential staker identifies a claim of identity as fraudulent they can call the fraud function in the contract and add their address to the set $\mathcal{F}$. This functionality is presented in algorithm 4. While there is no economic incentive to do so, and the potential staker will have to pay a transaction fee for

Algorithm 4 Fraud Tag

Input: Address a of certificate to be fraud tagged, fraud tagger address f

$\text{cert} \leftarrow C[a]$

Require: $f \notin \text{cert}[\mathcal{F}]$

$\mathcal{F}' \leftarrow \text{cert}[\mathcal{F}]$

$\text{cert}[\mathcal{F}] \leftarrow \mathcal{F}' \cup \{f\}$

the execution of the function, it is seen as good will in keeping the system working.

If $|\mathcal{F}|$ is large this will signal to outside observers that a claim to identity is fraudulent. Moreover, having to pay a gas fee to file a fraud tag will discourage would be saboteurs of the system.

H. Revocation

Algorithm 5 Revoke Certificate

Input: Address a of certificate to be revoked, revokation token ψ

$\text{cert} \leftarrow C[a]$

Require: $\text{cert}[x] \neq \mathsf{true}$

Require: $\text{cert}[r] = \text{Hash}(\psi)$

for s in $\text{cert}[\overrightarrow{S}]$ **do**

$\nu \leftarrow \text{cert}[\overrightarrow{S}[s][v]$

$\text{cert}[\overrightarrow{S}[s][v] \leftarrow 0$

SEND ν TO s

end for

$\text{cert}[x] \leftarrow \mathsf{true}$

Certificate revocation is a notoriously difficult and oft studied problem, whether it be for PGP-like certificates or web certificates governed by a certificate authority [16]–[19]. To that end we present a novel solution the certificate revocation problem. If a certificate holder would like to revoke their certificate in the event of key theft or key loss (or any other reason) our system allows them to do this seamlessly. Upon the creation of a certificate a user selects a value called the revocation input ψ and computes a revocation token r. This value is computed as:

$$r = \text{Hash}(\psi)$$

.

We take Hash to be a cryptographic hash function with a significantly large output size (128 bits). In practice ψ should be chosen as a sufficiently long secret random value the certificate creator stores offline. Thus, due to the one-way nature of cryptographic hash functions no entity but the certificate creator can correctly provide the value ψ to the revoke function.

In the event that a revocation of a certificate is desired the owner of the certificate calls the revoke function with input ψ and this is checked for equality against output r. If this checks, then x is set to true, from the default false. All staked value is immediately released to the stakeholders and the certificate is revoked. This functionality is seen in algorithm 5.

Notably, this revocation mechanism works in the case of private key loss of the wallet that is tied to a certificate. Normally, if a wallet were to be compromised the attacker would not only have access to the victim's funds, but also to to the claim of identity made in the DecCert system. However, as our revocation process does not depend on the private key of a user's certificate we side step this problem. That is revokation still works in the case of private key theft or loss. This in itself is a novel system design mechanism for a PKI deployment.

Mechanisms like certificate revocation lists exist for traditional PKI systems to address the problem of revokation under key loss or compromise. A certificate revocation list is compiled by reporting a compromised or lost key to a certificate authority. However, these lists are centralized and bad actors can make false reports to pollute the list. Moreover, the problem of distributing a verified certificate revokation list in a timely manner remains. This can lead to bad actors retaining the use of a certificate far past the time of compromise [20].

In the PGP system a revocation certificate exists, which is a self-signature that revokes the certificate corresponding to the signing key [21]. However, this mechanism fails to work if a key is lost, as self-signing would be impossible.

As a backup to this mechanism, if the revocation input is lost, a certificate holder could in practice force an *de facto* revokation a certificate by encouraging users to issue fraud tags against their certificate on the same channels they submitted their identity proofs on.

Operation	Cost in Gas Units	Cost in USD
Deployment	1666538	$427.33
Certificate Creation	202196	$51.84
Staking	69513	$17.82
Fraud Tagging	86902	$22.28
Revokation	46681	$11.07

TABLE II

WE PRESENT THE COST OF THE OPERATIONS OF DECCERT BOTH IN GAS UNITS AND USD BASED ON THE AVERAGE GAS COST AND SPOT PRICE OF ETHEREUM FROM THE LAST YEAR (MAY 22, 2021 TO MAY 22, 2022).

V. IMPLEMENTATION AND TESTING

We implemented a prototype of DecCert for the Ethereum [22] blockchain. The implementation was written in Solidity, the standard programming language for smart contracts deployed on Ethereum. This implementation is publicly available[1]. We tested for correctness of the main functionalities using the Truffle Suite [23] set of testing tools. Through our testing we were able to show that certificate creation, staking, fraud tagging, and revokation work as described in a real deployment scenario.

We also measured the gas cost of each of these functionalities and present this data in table II. We obtained the USD values by taking the average spot price of Ethereum

[1]https://github.com/smarky7CD/DecCert

($3,108.08) and the average gas cost (82.5 gwei) from Etherscan [24] over the last year (May 22, 2021 to May 22, 2022). The values obtained are modest compared to similar projects. For instance certificate creation in DecCert was about half as expensive as registering an Ethereum Name Service domain [25] at the time of testing. Encouragingly, staking, fraud tagging, and revokation are relatively inexpensive. These results show that deploying and using DecCert is economically viable on Ethereum's ecosystem. With the on-going scaling efforts of Ethereum spearheaded by proof-of-stake, sharding, and rollups [26] these costs will become negligible in the near-future.

Further work will be required to make this implementation deployable on the main Ethereum chain. First, it needs to be audited from both a cryptographic perspective as well as a code security perspective. Moreover, the time lock staking functionality needs to be implemented, as well as the logic for stakers to retrieve their funds after the time lock is released and in the event of a certificate revokation. It would also be of interest to develop a graphical user interface for creating certificates, indexing certificates, generating social identity proofs, and generating secure revokation tokens to help non-technical participants interact with DecCert.

VI. NFT Marketplace Use Case

The central question is why would anyone choose to stake a certificate? What value do they derive? We answer these questions here with an illustrative example.

As aforementioned, the pseudo-anonymity a blockchain provides is an oft desired feature of the system. However, for certain applications a verified identity is desired. Take an artist selling art as an non-fungible token (NFT). For readers needing an introduction to NFTs we point them to [27].

The popularity of NFTs has skyrocketed in recent times. NFTs can represent ownership of any underlying unique digital asset. However, digital art is the phenomena that has penetrated the mainstream with the most flare. In fact, as of writing, art NFTs are averaging about $200 million in trading volume per day according to a NFT statistics tracking service [28].

NFTs have allowed artists to connect directly with fans and also make a handsome income. However, the space is rife with fraud. NFT marketplace OpenSea reports that 80% of NFTs minted via their free minting option were fraudulent or spam [29]. The website of Banksy, famed anonymous street artist, was hacked and linked to an NFT project that they were not actually associated with. A buyer was scammed out of over $300,000 for a piece digital art from the sham Banksy collection [30]. Further, rapper Lil Yachty's likeness was used without his consent for an NFT project that netted over $6.5 million dollars [31]. Countless other examples exist. It is clear that the single biggest threat to the current NFT ecosystem is fraud, particularly that targeting the likeness of high profile individuals.

A fully decentralized NFT marketplace could use DecCert as a system for artist identity verification. In return for staking an artist's claim of identity, stakers would make a small percentage of the revenue from an artist's NFT sales. In this way stakers are economically incentivized to stake a valid claim of identity for an artist. Conversely, due to the fact the stake is locked for a time period, they are economically deincentivized from staking fraudulent claims of identity because of the time factor of capital. Due to increased trust that DecCert brings to the marketplace bidders will feel more comfortable buying, thus increasing bidding activity on NFTs, and in-turn augmenting revenue for creators that offsets the small percentage that is paid to certificate stakers.

We provide a diagrammatic view of this use case in figure 2. We will enumerate the flow of integrating DecCert within an NFT marketplace below.

1) A high profile creator makes a certificate linking their wallet address to their identity using DecCert and existing online presences.
2) Stakers view the certificate and outside identity proofs to verify the claim of the creator's identity and if deemed valid stake funds in the certificate.
3) Once a creator's certificate has a significant amount of stakers (enough to create trust among buyers) they can mint their digital work as an NFT on the marketplace.
4) The minted NFT is put up for auction.
5) Potential NFT bidders view the DecCert certificate to gain trust in the artist's claim of identity.
6) Once bidders verify the claim of identity via the evidence presented in the DecCert certificate they bid on the on the NFT.
7) The auction ends after the terminal condition is reached.
8) The winning bidder assumes ownership of the NFT.
9) The creator collects the revenue from the auction minus the staker's share.
10) The stakers are paid out some share (can be set on a per-marketplace basis) of the winning bid equal to the percentage of their stake in the certificate.

In sum, DecCert can function as a decentralized solution to identity verification in NFT marketplaces. As fraud is the prominent problem in these marketplaces the need for such a solution is obvious. Moreover, as compared to marketplaces verifying identity with systems similar to that of centralized applications like Twitter or Instagram our solution keeps with the decentralized and open spirit of blockchain. Moreover, DecCert can be used as a PKI for any blockchain application that needs to solve attestation of identity in a trustless environment. Much like the above NFT marketplace example, integration could be done such that staker's receive rewards for doing so, therefore economically incentivizing the security of the application in regards to claimed identities.

VII. DecCert and Zooko's Triangle

We will now analyze DecCert with respect to Zooko's triangle. We will go through the three traits of Zooko's triangle in the subsequent subsections and argue how DecCert achieves

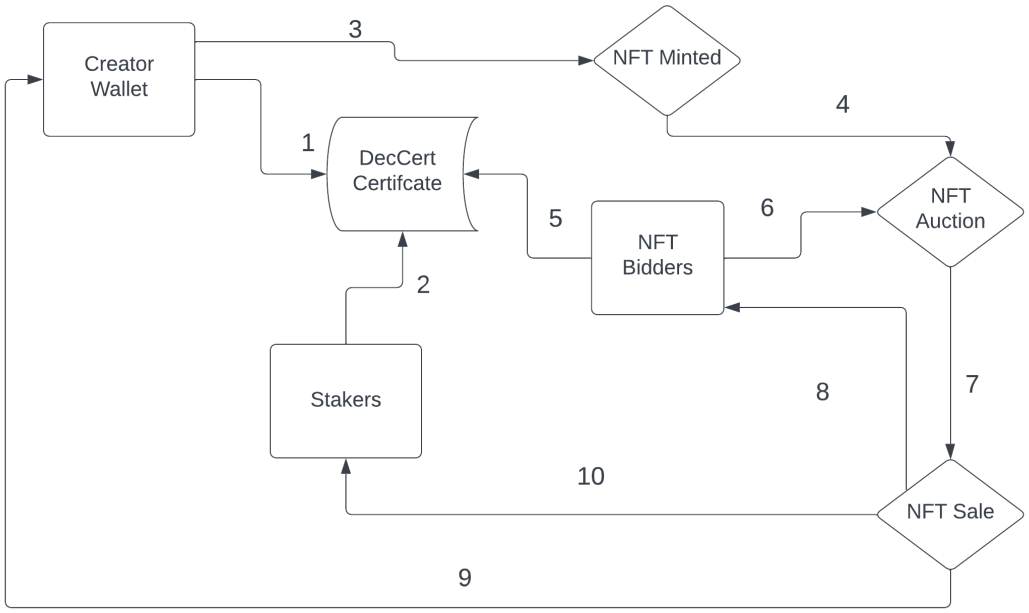

Fig. 2. We illustrate the proposed use case of DecCert for decentralized verification of identity in NFT marketplaces. This is a necessary innovation as currently the space is rife with fraud. In the diagram the blocks represent users and smart contracts while arrows represent users interacting with smart contracts or the results of the respective smart contract's execution. The rounded square blocks represent a user of groups of users, the scroll-shaped block represents the DecCert smart contract, and the rhombus blocks represent functionality of the NFT marketplace's smart contracts.

all of them as a PKI. In doing so we have shown that Zooko's conjecture is false for this type of deployment of PKIs.

A. Human-meaningful

By its construction DecCert allows a user to tie a human-meaningful identifier to their wallet address on the blockchain DecCert is deployed on during certificate creation.

Further, the proofs of identity that link to established online presences not only function to verify a claim of identity, but also provide certificates with further human-meaningful identifiers.

B. Secure

DecCert binds identities to wallet addresses (public), thus forming a certificate and achieving baseline security. As aforementioned, this can be done in such a way that identity retention is achieved if so desired.

DecCert goes further by achieving identity attestation. Through the use of publicly auditable proofs of identity tied to established online presences, the economic incentive to stake valid claims of identity (and conversely the economic disincentive to not stake fraudulent claims), plus the functionality of fraud tagging, DecCert solves the problem of verifying identities in trustless environments. Moreover, by limiting the amount any single account is able to stake and setting the maximum stake amount to be high we defend against Sybil attacks with in our system.

DecCert's revokation functionality allows for instant revokation of a certificate even in the case of private key compromise by using a separate revokation token. Moreover, we provide the back-up option of the flooding of fraud tags on a certificate in the event of revokation token loss. We disincentivize fraudulent fraud tagging due to the cost of executing the functionality on the underlying blockchain.

C. Decentralized

DecCert is implemented as a smart contract that is deployed on a trustless, public blockchain. Certificates, staking information, and fraud tags are all publicly viewable on the blockchain. In this way, DecCert is inherently decentralized, as all the data exists and is verifiable without the presence of a centralized operating authority.

Certificates can be generated, revoked, and indexed in a decentralized manner. Moreover, the system can be easily integrated into other decentralized applications that need a system for verifiable attestation of identity, as DecCert is implemented as a portable smart contract.

VIII. Conclusion

In this paper we presented DecCert a decentralized blockchain-based PKI deployed as a smart contact. The system has notable advantages over prior work. First, it solves the problem of trustless identity attestation. Moreover, it provides a solution to Zooko's triangle for a robust set of well-defined properties for PKI systems, as it provides human meaningful names to users of the system, it is secured via strong cryptography and economic incentives, and it is decentralized due to its deployment on public blockchains. We also introduced

a novel solution to the certificate revocation problem. Further, we provided an implementation that showcased the feasibility and economic viability of deploying DecCert in the wild. Lastly, we gave an example use case of the system for a decentralized NFT marketplace to solve the problem of fraud in the space.

REFERENCES

[1] Idenity.com. [Online]. Available: https://www.identity.com/home-3/

[2] J. R. Prins and B. U. Cybercrime, "Diginotar certificate authority breach "operation black tulip"," *Fox-IT, November*, p. 18, 2011.

[3] S. B. Roosa and S. Schultze, "Trust darknet: Control and compromise in the internet's certificate authority model," *IEEE Internet Computing*, vol. 17, no. 3, pp. 18–25, 2013.

[4] T. Smith, L. Dickinson, and K. Seamons, "Let's revoke: Scalable global certificate revocation," in *Network and Distributed Systems Security (NDSS) Symposium 2020*, 2020.

[5] C. Adams and S. Lloyd, *Understanding public-key infrastructure: concepts, standards, and deployment considerations*. Sams Publishing, 1999.

[6] S. Garfinkel *et al.*, *PGP: pretty good privacy*. " O'Reilly Media, Inc.", 1995.

[7] Keybase. [Online]. Available: https://keybase.io/

[8] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.

[9] D. Scheirlinck, "Zooko's trianle," 2006. [Online]. Available: https://commons.wikimedia.org/wiki/File:Zooko's_Triangle.svg

[10] Z. Wilcox-O'Hearn, "Names: Distributed, secure, human-readable: Choose two," *http://zooko.com/distnames.html*, 2001.

[11] C. Fromknecht, D. Velicanu, and S. Yakoubov, "A decentralized public key infrastructure with identity retention," *Cryptology ePrint Archive*, 2014.

[12] D. Wilson and G. Ateniese, "From pretty good to great: Enhancing pgp using bitcoin and the blockchain," in *International conference on network and system security*. Springer, 2015, pp. 368–375.

[13] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *2016 USENIX annual technical conference (USENIX ATC 16)*, 2016, pp. 181–194.

[14] M. Al-Bassam, "Scpki: A smart contract-based pki and identity system," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, 2017, pp. 35–40.

[15] M. Toorani and C. Gehrmann, "A decentralized dynamic pki based on blockchain," in *Proceedings Of the 36th Annual ACM Symposium On Applied Computing*, 2021, pp. 1646–1655.

[16] J. K. Millen and R. N. Wright, "Certificate revocation the responsible way," in *Proceedings Computer Security, Dependability, and Assurance: From Needs to Solutions (Cat. No. 98EX358)*. IEEE, 1998, pp. 196–203.

[17] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2003, pp. 272–293.

[18] B. Fox and B. LaMacchia, "Certificate revocation: Mechanics and meaning," in *International Conference on Financial Cryptography*. Springer, 1998, pp. 158–164.

[19] M. Naor and K. Nissim, "Certificate revocation and certificate update," *IEEE Journal on selected areas in communications*, vol. 18, no. 4, pp. 561–570, 2000.

[20] M. Khodaei and P. Papadimitratos, "Efficient, scalable, and resilient vehicle-centric certificate revocation list distribution in vanets," in *Proceedings of the 11th ACM conference on security & privacy in wireless and mobile networks*, 2018, pp. 172–183.

[21] D. Callas *et al.*, "Openpgp message format," 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc4880

[22] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[23] Truffle suite. [Online]. Available: https://trufflesuite.com/

[24] Etherscan. [Online]. Available: https://etherscan.io/

[25] Ethereum name service. [Online]. Available: https://app.ens.domains/

[26] Ethereum scalability upgrade. [Online]. Available: https://ethereum.org/en/upgrades/

[27] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges," *arXiv preprint arXiv:2105.07447*, 2021.

[28] Nft stats. [Online]. Available: https://www.nft-stats.com

[29] Opensea fraud statistics. [Online]. Available: bit.ly/38pg9w0

[30] A. Jordanoska, "The exciting world of nfts: a consideration of regulatory and financial crime risks," *BUTTERWORTHS JOURNAL OF INTERNATIONAL BANKING AND FINANCIAL LAW*, vol. 10, p. 716, 2021.

[31] Lil yachty nft scam. [Online]. Available: https://www.complex.com/music/lil-yachty-sues-nft-seller-using-likeness-raise-millions-without-consent