1 Introduction

Identity in internet was build around the ideas of openness and trust. In it's early days, when new peers wanted to join a network, the earliest mechanism to identify peers on the network by simply asking another peer for a hosts file containing all known entities in the network, where every party was effectively unauthenticated and anonymous. With the advent of large economic and governmental activities in the Internet, identity was and is a necessity today, therefore has been continuously added in different forms to the online world. Today's identity in the internet is facilitated by a dynamic collection of parties, ranging from cookie providers over companies like Google and Facebook to the organizations maintaining critical infrastructure, including DNS records and Certificate chains. The unsystematic structure of identity and the resulting concerns about lack of legitimacy, privacy, accountability and transparency of the involved parties is a hurdle for large scale deployment of digital identity, therefore inhibiting the growth and efficiency gains a more concise environment for identity could facilitate.

Anonymous credentials as first introduced by Chaum[], are a missing piece in the collection of possibilities for peers to identity themselves and assert attributes of each other. Essentially, anonymous credentials allow entities, organization or institutions to securely associate certain attributes to each other while preserving their privacy. Intuitively, an entity that issues a certain claim (e.g. x is human, x is older than 21years, x has a German passport) about another entity is not able to gather information about the usage of this claim afterwards. At the same time, an entity that verifies a claim learns nothing from verifying the claim nor the user the claim is about, besides the fact, whether the claim is true or false.

While current proposals for anonymous credentials have become efficient, they lack the incentive structure of traditional identity schemes due to the missing monetization strategies of user data. While being technologically superior and qualified for deployment in many critical applications, it is not clear if they will be adopted without a clear monetization strategy. Additionally, the underlying cryptographic schemes assume mutually authenticated parties in a permissionless peer-to-peer environment. As identity is and has been a primacy of governments and institutions, this assumption seems impractical and needs further clarification.

Our contribution is a cryptoeconomic framework for secure, decentralized identity. We will describe recent cryptographic results and economic mechanisms for a network of entities which want to exchange identity information. The network we describe should allow entities in the network to permissionlessly and securely issue, prove and verify anonymous credentials linked to pseudonyms under their respective control.

On authentication -¿ Appendix Authentication, as in establishing an initial link between a physical entity and its digital representation is an atomic action in digitization and the bootstrapping process of identity. Previous work left this problem mostly aside, as it is in many cases non-technical and context or application specific and often done by proxy (verifying a sensors key fingerprint, verifying a users drivers licence, asserting control over admin@domain.com). In the context of decentralization and privacy, the

complexity further increases as permissionless networks which require a government id or certificate from an authority to sign up lead into regression. To our knowledge, essentially two opposing models for authentication are present; Authoritative and the web of trust. With more recent approaches like keybase[] or uport, the trust root is still authoritative, but the diversification drastically improves security and trust. On the other hand, a web of trust where peers authenticate each other in person/physically is - in theory - more decentralised and secure.

We want to state our notion of authentication at least metaphorically to provide some vague guideline:

> "A person is a person through other persons." - Proverb, Ubuntu philosophy [15]

Cryptography in particular and computer science has an inherent drive towards decentralisation, as removing single points of failure is simply good engineering practice. While deduction from metaphors is considered harmful, for identity, there's a social correspondence for inherent decentralisation. Social science often does not consider access or privileges granted by other entities - notions that capture the larger part of technical concepts of identity. As a social construct, identity is partially described by roles in communities or networks of humans, however, most characteristics constituting identity, e.g. (subjective) sex, family or self-esteem can not be thought without privacy.

> [quote random identity definition from social science here]

Adoption of decentralised identity is unlikely due governments primacy on identity, e.g. we believe only a model encapsulating this primacy may be able to overcome and replace it. Our work therefore aims for compliance and compatibility and the ability to continuously decentralize itself.

2 Overview of cryptographic scheme

The network we describe should allow entities in the network to permissionlessly issue, prove and verify attributes about each other.

In the following section, we give some intuition for the underlying cryptographic schemes and the properties they yield. However, we suppress any details for briefness. In a later section, we will describe cryptoeconmic mechanisms to incentive network beneficial behaviour.

2.1 Design goals and building blocks

Some properties and notions of the underlying cryptographic schemes [https://eprint.iacr.org/2001/019.pdf,

Some properties and notions of the underlying cryptographic schemes [https://eprint.iacr.org/2001/019.pdf, https://eprint.iacr.org/2006/454.pdf]: https://eprint.iacr.org/2006/454.pdf]:

Consistency of credentials. It should not be possible for colluding users, issuers or verifiers to obtain a credential from an issuer where the issuance is not intended.

User privacy. If users show their credentials, this does not yield information to a verfifier or a potential eavesdropper besides the fact if a user posseses a given credential or not. In particular, if a user controls different pseudonyms holding different credentials, they can not be linked.

All-or-nothing non-transferability. Entities should be disincentivised to share their credentials. This is realized in the protocol, where sharing one pseudonym or credential is equivalent to sharing all secret keys within the system.

Attributes. An attribute is a claim that is made by an entity about itself or another entity, e.g. $Alice.age >= 18$. In particular, this includes public key bindings, e.g. $alice@foo.org.pub\_key = 0xabc123$ and $fingerprint.cert.foo.eth. = D2 : 9D : ... : CD : F5$. Attributes which have been authenticated by third parties may function like certificates. However, they will not be directly presented to third parties but will rather be reasoned about in zero knowledge.

2.2 Summary of the underlying schemes:

Credentials with encoded attributes. A user may wish to only prove certain statements about itself without reveling the attribute itself. In general, identity attributes can be encoded in credentials as booleans or via multi message proofs. Together with range proofs, this allows a user to prove a variety of statements about itself without revealing any additional information.

Non-transferable Anonymous Credentials. Non-transferable Anonymous Credentials is a scheme based on the strong RSA assumption, where users controlling a master secret key can derive different pseudonyms with different issuers. For a particular pseudonym, an issuer may then issue a credential, that the user later may prove possession of to a verifier via a proof of knowledge. The scheme has some extensions, namely All-or-nothing Non-transferability and optional local and global revocation. All-or-nothing Non-transferability means in practice, that if a user decides to share a credential with another peer, it can only do so by sharing all credentials under her master secret key.

n-Times Anonymous Authentication Scheme n-times Anonymous Authentication is a credential system where an entity can authenticate at most n-times with a credential in a given time frame. Initially, a user runs an interactive protocol (Obtain Dispenser) with an issuer, to obtain a token dispenser, where a token represents an arbitrary credential, encapsulating claims about the attributes of the authenticating entity. The user may then use the tokens in a zero-knowledge proof of knowledge (Show) to prove the issuer's claim about itself to a verifier. The verifier publishes the token to a public append only ledger and invalidates it.
This dispenser allows the entity to generate up to n fresh tokens per time step. At the end of a time period, the dispenser refills and new tokens can be used. Given honest verifiers, this construction yields the following core properties:

1. Anonymity of honest users: The Obtain Dispenser is facilitated with a Pedersen commitment scheme, hiding the commited values. Therefore if a token from the dispenser is used from a user to authenticate with a verifier, an issuer is not able to trace the usage of the tokens, even if issuer and verifier collude.

2. Identification of Violators: If a token is used multiple times, the verifier can deanonymize the violator. The violators dispenser can then be revoked.

Public append only ledger. Our design requires a secure, public, scalable, high integrity ledger with low transaction cost and low latency (e.g. ETH 2.0). If a user authenticates anonymously, the verfier has to publish the used token to the ledger, s.t. other verifiers can detect if a user tries to reuse the same token. With state of the art secure ledgers as eth 1.0, publishing used tokens could be facilitated as well via batching techniques or plasma based schemes operated by verifiers.

3 Micro economic mechanisms

For the following section it is crucial to note, that the amount to be staked by the issuer or payment of the verifier must not be fixed, s.t. an equilibrium between issuers, verifiers and "bounty hunters" can evolve dynamically, considering application specific parameters; user privacy, value at risk, security, economically acceptable fraud rates, etc. While we leave strict classification of the parameter and design space as future work, we describe some theoretical considerations and an agent based simulation of monetary policy in a later section.

Mechanism 1: Issuer's Authentication Staking. When authenticating a user during the Obtain step of the protocol, an issuer stakes a certain value. If another entity manages to obtain a second credential for an existing user (e.g. identity duplication), it can either present it to a resolution mechanism and receive the stake the issuer put on both dispensers, or use it maliciously. The stake should have a value, s.t.:

1. bounty hunters are incentivised to frequently test the issuer for vulnerabilities.
2. if a vulnerability was found, stakes should be payed out automatically by presenting two valid but inconsistent credentials for the same pseudonym to the ledger.
3. the stake should be higher then the cumulative value at risk. Where assigning monetary value at risk is hard (for e.g. private user data), (user feedback driven) heuristics should be applied.
4. the capital cost to become a validator should be minimal, given above constraints.

Mechanism 2: Tokenized credentials. The n-Times Anonymous Authentication Scheme establishes a relation between an issuer and a verifier of a credential via:

1. Show(User(D, $\text{pk}_{Issuer}, t, n), Verifier(pk_{Issuer}, t, n)$) When an issuer creates a token dispenser for the user, it bares the economic risk for the staked amount in case it fails to detect attempts to duplicate the authenticated entity. To compensate for it and to incentivise the issuing of new credentials, there is a certain amount of value attached in the invalidation transaction from the verfier to the issuer after the "Show" step after the credential protocol is completed and the verifier publishes the used credential to the public ledger. If the verifier does not send the transaction invalidating the credential, it can be reused or shared by the user to re authenticate with the same credential multiple times. However, the verifier already learned the required information and does not have a direct incentive to publish the invalidation transaction which requires her to pay the issuer. We address this problem with the next mechanism.

Mechanism 3: Verifier slashing We slightly modify the interactive proof of knowledge in the show step of the credential scheme to yield a verifiable encrypted transcript of the user/verifier interaction. In case a verifier does not invalidate the token via a transaction to the ledger, the user herself can use the transcript to do it on the verfifiers behalf. Similar to authentication staking (mechanism 1), a user that catches a misbehaving verifier receives a bounty from a stake that participating verifiers have to pay before becoming eligible for verifying credentials.

4 Macro economic considerations

To get to the code click here.
We describe shortly several monetary considerations and effects, which we visualize in a simple agent-based simulation model. The main focus is the price development of a single credential (which we denote as "information" in this section) over time.

4.1 Theoretical Considerations

To understand the development of the price for one piece of information, we use the notation:

- c_t cost of information in token,
- e_t cost of information in fiat money,
- N_t number of tokens,
- $E_t = N_t \cdot exchange\_rate = N_t \cdot \frac{E_t}{N_t}$ the networth (value of tokens in fiat).

Assume information i costs c_t at time time t, which is

$$e_t = c_t \cdot \frac{E_t}{N_t}$$

in fiat money. The cost in fiat money follows the dynamic:

$$\frac{de_t}{dt} = \frac{dc_t}{dt} \cdot \frac{E_t}{N_t} + c_t \cdot \frac{d(E_t/N_t)}{dt}$$

We observe two effects. First, a change in the cost in token $\frac{dc_t}{dt}$ changes the price in fiat in the same direction. Here, we assume a decrease as a realistic prediction. As information spreads through the system, more agents can offer it to new demand. It might become invalid or outdated, both leads – inline with standard economics – to lower a price. We call this the *competition* effect.
Somewhat more complex to predict is the *exchange rate* effect – a change in $\frac{d(E_t/N_t)}{dt}$. It depends on the number of tokens N_t and the networth of the system. While the number of tokens is controllable via the Monoid owners and rather a political decision within the system, the networth depends on the value of the service as well as speculative forces. These speculative forces are the main focus of our agent-based simulation model described in section 4.2. A first question is how to set the dynamics of N_t. Shall there be a finite supply increase as in Bitcoin – predefined in the Blockchain – or shall the supply decrease over time – potentially leading to deflation due to scarcity.

4.2 Simulation Setup

The simulation consists of agents which interact on two markets. First, agents can buy information and hence take the role of verifiers. They ask the system for the information they demand. Each agent possessing the information can offer it for a specified amount of tokens. This makes these agents issuers. The verifier now choose one of the offers. To pay the issuer, the verifier gets active on the second market. Paying in fiat money, it buys the necessary amount of tokens on the exchange market.

In each period, *each* agent has several options to act: (i) buy information, (ii) offer information / respond to a request, (iii) speculate on an the exchange rate market. We distinguish agents by their user type: commercial (i+ii) and speculative (iii).
To analyze the interaction of these two different strategies, we plot the exchange rate and the price of a piece of information in token as well as in fiat money. It shall reveal the strength of the two effects described in section 4.1 – competition vs exchange rate effect.

4.2.1 Technical View

Parameter and Initial Values:

variable	chosen value	explanation
Number of nodes	10	represent agents
Number of tokens	10*10 = 100	each agent starts with 10, might change over time.
Amount of fiat money	10*10 = 100	each agent starts with 10
Time	discrete	
Time horizon	1000	iterations
Number of infos	100	infos are traded on the market
Start price for each info	1	initial price of info
Initial info distribution	?	each agent starts with different initial infos
Price reduce percentage	0.1	agents reduce their offered price in case of failed sale
burn	0.01	in each trade, this amount of token is burned

Timing of events

1. Verifiers decide which info they want to buy.
2. Information Market opens: Verifiers, which want to buy, go to the market and ask for prices.
3. Issuers with the information send their offers.
4. Verifiers choose lowest offer and send a request to the issuer.
5. Speculative agents decide whether and how they want to buy or sell on the Token exchange market.
6. Token Exchange Market opens: Verifiers exchange their fiat money into tokens to pay the issuer. At the same time, speculative agents buy or sell.
7. Issuer receives payment in token.

Exchange rate market:

- how does this market function?

4.3 Quantity Theory of Money

This theory can be summarized in the following equation:

$$M \times V = p \times q \tag{1}$$

with M the amount of tokens, V the velocity of transactions, p the price vector, and q the quantity vector. In our case, we can translate quantities with offers.
From Wikipedia: "The quantity theory postulates that the primary causal effect is an effect of M on P."
However, we are wondering whether we can affect the velocity V by changing M.
Certainly, with constant prices p (and quantities q), only V can react to a change in M.
Then, the equation would predict:

a decrease (increase) in M results in an increase (decrease) in V.

For example, assume M decreases by 10%, then an increase of V by 11% offsets this effect (take 1 over 0.9 to get 1.11). If we assume non-constant prices, the effects are unclear.

4.3.1 Example 1: Proof of group membership credential

If a given a subset of entities forms a group, some policy might be enforced to accept or reject new members. A member might want to proof its membership to some other entity. The target properties require, in this case, that

4.3.2 Example 2: Proof of fulfillment

Some subset of entities might want to close a contract. There might be a set conditions or obligations, that entities are legally obliged to fulfill (e.g. being a member of a certain group).
A contracting party might want to proof the fulfillment of its contractual obligations to an other party.

Soundness: A proof the fulfillment can be generated if and only if that entity has fulfilled its contractual obligations.
Integrity: If an entity receives a proof the fulfillment, it has means to verify if it was tempered with or not.
Authenticity: If any entity receives a proof the fulfillment, it has means to verify that the obligations were full filled by the party it refers to.
Zero Knowledge: An entity learns nothing from a proof the fulfillment, besides the fact if the entity in question has full filled its contractual obligations or not.