Certonymity: private and regulatable digital identity

Liqun Chen[1], Mark D. Ryan[2],
William A. Wayman[3], and Christopher Williamson[3]

[1] University of Surrey [2] University of Birmingham [3] SW7 Group

Abstract. Digital identity systems face a fundamental tension between individual privacy and regulatory oversight. Existing solutions often provide one or the other, but not both. We present a framework for digital identity that bridges this gap. Our system is built on three key principles: avoiding unnecessary privacy impact, guaranteeing transparency of oversight, and minimising trust in any semi-trusted parties. Our approach supports gradated levels of investigation through four distinct query types, allowing authorities to investigate patterns or links between transactions without necessarily revealing the full identities of the users involved. We achieve this through a combination of threshold cryptography, zero-knowledge proofs, and a public append-only ledger. We articulate our motivations, sketch the technical construction of our system, and present preliminary implementation results to demonstrate the practicality of our approach.

1 A societal dilemma

Cryptography is a social and political subject as well as a mathematical and technical one. A societal dilemma has beset cryptography since its inception:

> How should the requirement to enforce laws and pursue and catch criminals be reconciled with the requirement of individuals to have privacy of their communications and their financial affairs?

Let us consider privacy first. The ability of an individual to keep information private is fundamental to a free and open society. It allows an individual to unrestrictedly consider their options about what to say and how to behave, without undue interference. This promotes a thoughtful and creative population; society as a whole benefits. Ideas can flourish, without fear of repression. Societies that promote privacy have a track record of innovation and economic success.

But a civilised society also requires the rule of law. When people do wrong, there must be the means to identify them and hold them accountable for their actions. A society in which laws are not enforced is unfair and discriminatory against its poor and weak members. Laws are constraints on freedom, but good laws ultimately engender greater freedom by giving citizens the confidence to take economic risks in the

expectation of reward. Naturally, identifying criminals may require access to data (for example, data from payment systems).
The dilemma can clearly be seen by comparing the traditional financial world of bank accounts and the cryptocurrency world. Traditional finance is heavily focussed on rules and regulations, requiring full identification of both the payer and payee. As a result it has very little privacy. In contrast, cryptocurrency is motivated by privacy and freedom. By design, there are no restrictions on who can pay what to whom, and the records of payments do not identify the individuals behind a transaction. However, the difficulty in identifying participants in cryptocurrency systems has led to fraud and money-laundering.

Solution shape. We seek a solution that gives privacy to most people most of the time, but also has a limited ability for law enforcement to carry out investigations. The investigations should be proportionate and transparent. We therefore require a solution that satisfies these three principles:

- **Deanonymisation avoidance** Carrying out an investigation should not necessarily require full deanonymisation of an individual. There may be weaker (more proportionate) queries that can help eliminate lines of enquiry without the need to uncover the identity of any individual.
- **Authority transparency** Citizens should be able to see if queries have been made about them, and hold relying parties to account for the level of queries they make.
- **Trusted party obliviousness** Distributed trusted parties may be required, but they should remain ignorant about the data they manipulate. They should not know the subject or the result of an investigation. This property is important partly for privacy, but also to ensure that the trusted parties have insufficient information to defect from the protocol. Trusted party actions are deterministic and fully automatable.

2 Existing approaches

Self-Sovereign Identity (SSI) and Decentralized Identity (DID). SSI is a model for managing digital identities that allows an identity holder to control how to use their identity. DID, developed by the World Wide Web Consortium (W3C) [13], is an example of SSI. A DID includes a public and private key pair and the public key acts as the holder's DID. When a holder requests to bind their DID with an attribute (such as a university degree), an authorised issuer (in this case, the university) generates a Verifiable Credential (VC) [14] after verifying that the holder holds such an attribute. The VC is a digital signature signed under the issuer's key. After receiving a VC, the DID holder can prove ownership of the VC to a verifier, and the process of this proof is called a Verifiable Presentation (VP). The VP is another signature under the private key of the DID.

Anonymous Credentials (AC). Two extended features for SSI and DID are relevant to our work. First, VPs can support anonymity. Second, an identity can bind with multiple attributes within a single credential; for example, a credential includes name, date of birth, place of birth, nationality, sex, address and DID. When buying alcohol, the holder wants to prove that they are an adult but does not want to reveal any other attributes or their DID. In the literature, the technology supporting these two features is called Anonymous Credentials (AC). The W3C working group recommends using randomisable signatures to generate VCs and zero-knowledge proofs to achieve anonymous VPs. Potential candidates for randomisable signatures are schemes of Camenisch-Lysyanskaya (CL) [2], Pointcheval-Sanders (PS) [11], or BBS+ [7]. To make an anonymous VP to a VC, the DID holder first randomises the VC and then creates the VP using the DID private key. This VP can be verified by using the issuer's public key rather than the holder's DID.

Revocable anonymity. To prevent the identity and attribute holder from abusing anonymity in ACs, a technique called traceability has been developed to revoke anonymity using a trusted 'tracer', e.g., [3, 9]. When proving the possession of a credential, the holder includes evidence to show that, given this proof, the tracer can find the holder's identity. A particularly notable example in this direction is the identity management system of the Concordium blockchain [4], which has very similar objectives to ours. Their goal is to maintain privacy in financial systems, while still allowing compliance with the requirement of *know your customer* (KYC) and the need of *anti-money-laundering* (AML) rules.

Why these systems don't satisfy our purpose. The systems mentioned above do not satisfy the principle of *deanonymisation avoidance*: any queries made fully deanonymise the user. They don't satisfy our principles of *authority transparency* or *trusted party obliviousness*.

3 Certonyms

A *certonym* ('certified pseudonym') is a digital identity under the user's control, which (when there is probable cause or a legitimate legal basis) allows a relying party to make certain queries that can link it to other certonyms or to individuals. This linking aspect is to allow the enforcement of regulations. Crucially, the linking aspects are only possible in certain circumstances, and only in a way that unavoidably leaves evidence of the linking (**authority transparency**). An individual acquires and uses certonyms as follows:

1. After registering with an Issuer, an individual can create certonyms (say on an app on their phone or PC). Similarly to cryptocurrency addresses, a given certonym is intended to be used only for one or very few transactions; an individual should generate new certonyms regularly.
2. Individuals can use their certonyms to sign data, such as financial transactions. Anyone can verify the signature, and see that the data

was signed by a well-formed certonym that was produced as a result of the user's prior onboarding with a legitimate Issuer.
3. The certonyms held by an individual and the signatures made by them cannot initially be linked to each other or to the individual.

Queries that link certonyms. Certain queries which link certonyms to other certonyms or to individuals are possible, but, as mentioned, such queries can be fulfilled only if certain circumstances hold, and only in a way that produces unremovable evidence. The idea of the queries is to allow relying parties to proportionately investigate patterns in financial transactions. Such relying parties may be operators of platforms that accept certonyms as a form of identity or law enforcement entities. The queries are:

1. `Same_user`: given two certonyms, determine whether they have the same ground identity without revealing that ground identity.
2. `Blind_regroup`: given a certonym, find the other certonyms that have the same ground identity, without revealing the ground identity.
3. `Find_user`: given a certonym, find the ground identity of the user.
4. `User_lookup`: given a ground identity, find the certonyms associated with it.

Note that the queries are defined to permit investigations that don't need to uncover the user's ground identity (**deanonymisation avoidance**).

4 Construction sketch

A certonym takes the form of a tuple: $(vk, C_{id}, \mathcal{H}, E_{br}, \mathcal{G}, \pi)$, where vk is a verification key, C_{id} and E_{br} are ciphertexts, $\mathcal{H}$ and $\mathcal{G}$ are hash values, and π is a zero-knowledge proof. Encryptions are with respect to a threshold public key.

Obtaining a credential. User Alice interacts with an Issuer, which confirms Alice's legal identity, encrypts it to produce ciphertext C'_{id}, and signs the ciphertext to produce $S_{C'_{id}}$. Alice and the Issuer contribute randomness to derive a random nonce r, which the Issuer blindly signs to produce S_r and encrypts r to produce C_r. The Issuer stores Alice's legal identity in association with C_r and provides to Alice the credential $(C'_{id}, S_{C'_{id}}, S_r)$. Alice is in control: only she can create certonyms from this credential and she can create new and unlinkable certonyms at any time.

Creating a certonym. Alice generates a new signing key pair (sk, vk). She re-randomises C'_{id}, producing a fresh ciphertext C_{id} for the same plaintext that cannot be linked to C'_{id}. Alice chooses random integer ϵ that is at most N (a global parameter) and computes $\mathcal{H} \leftarrow H(r||\epsilon)$, where H is a hash function. Alice chooses a new nonce and encrypts it, deriving ciphertext E_{br}. From this she computes $\mathcal{G}$, which is defined similarly to $\mathcal{H}$ as a structured hash of the nonce encrypted by E_{br}. These values allow linking of temporal generations of certonyms, as needed for `blind_regroup`. Finally, Alice creates a zero-knowledge proof π of correct construction, that: re-randomisation was done correctly and with respect to a ciphertext for which she holds associated signature $S_{C'_{id}}$, she has knowledge of r and ϵ used in $\mathcal{H}$, she holds a signature on r, and $\epsilon \leq N$.

4.1 Query execution

Same_user query. A relying party RP identifies two certonyms of interest and wishes to determine whether the underlying legal identities are the same, without further privacy impact. RP requests a plaintext equality test [8] with respect to the two ciphertexts at the C_{id} position of each certonym. The output is a single bit indicating whether the ciphertexts encode the same identity.

Blind_regroup query. RP identifies a certonym of interest and requests decryption of E_{br}. We omit full details, but based on the decrypted value, RP inspects the $\mathcal{G}$ position of all existing certonyms and recognises those created using the same or a related nonce as is encrypted by E_{br}. This process identifies all certonyms created by the same user, whose identity remains unknown. Newly created certonyms cannot be linked to any previous certonyms unless a subsequent query is performed.

Find_user query. RP requests decryption of C_{id}, revealing the legal identity.

User_lookup query. RP and the Issuer jointly find the C_r value associated with a legal identity of interest; a decryption request for C_r is made and r is revealed. RP computes the set $\{H(r||\epsilon) : \forall \epsilon < N\}$. For each existing certonym, RP checks whether it contains a value in the set: this will be the case for Alice's certonyms (and only hers with overwhelming probability).

4.2 Authority transparency

An important property of certonymity is that no query can be made covertly: queries require a decryption, which is done with a *transparent decryption* scheme (see [12]). Each ciphertext is encrypted using a threshold public key, in which parties called *trustees* hold shares of the decryption key and any threshold number can jointly decrypt. RP publishes query requests on-chain; each trustee only acts on requests that match on-chain data, and in response publishes its partial decryption of the appropriate ciphertext. So that only RP obtains query answers, trustees encrypt responses to the RP, which must combine them to privately compute the plaintext. Trustees perform plaintext equality tests as needed. Trustees never see details of the queries or results (the plaintext is merely key material, unlinkable to anything else — **trusted party obliviousness**).

User-generated threshold keys. Standard threshold cryptosystems define a single threshold public key and have fixed parameters for the number of trustees and the decryption threshold. In addition, trustees must all jointly participate in a key generation ceremony to create each threshold public key. Certonymity is compatible with *dynamic threshold schemes*, as introduced in [5]. In such schemes, users generate their own threshold public key and trustee-specific private keys. As applied to certonymity, when a user Alice wishes to create a new certonym, she will generate a fresh threshold public key and private keys. The threshold public key is used to encrypt the ciphertext components C_{id} and E_{br}. Each private key is encrypted to the relevant trustee using a public (non-threshold) key of the trustee. These encryptions are appended to the certonym and a zero-knowledge proof of correct construction is included (see also [10]). To decrypt a ciphertext made with respect to the threshold public key, trustees individually decrypt the appended ciphertext relevant to them to obtain their threshold private key and then apply the same process as in standard threshold cryptosystems to help the relying party decrypt the threshold ciphertext. This mechanism eliminates the requirement that trustees engage in a threshold key generation ceremony and allows user-selected threshold parameters (to the extent deemed appropriate by the relying party).

5 Implementation

A proof-of-concept implementation of a certonymic identity scheme is nearly complete. The key question to be answered by the proof-of-concept relates to the practicality of the required zero-knowledge proofs. In particular, software must be sourced or created to support proof generation and a wide host of cryptographic operations, including proof-friendly hash functions and efficient elliptic curve operations. Proof time must be sufficiently short even on constrained devices. Ideally, cryptographic primitives are supported that allow efficient verification on popular blockchains. We completed this work using the Groth16 ZK-SNARK system [6] and the Gnark software library [1]. We have found the proofs to be practical and the resulting certonyms to be manageable in terms of size, generation time and verification time.

6 Conclusion

Certonymity is an approach to digital identity that extends SSI by allowing queries to be made even when users do not cooperate. An essential aspect of certonymity is that authorities can be held to account for the queries they make.

References

1. Gautam Botrel, Thomas Piellard, Youssef El Housni, Ivo Kubjas, and Arya Tabaie. Consensys/gnark: v0.11.0, September 2024.

2. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO*, pages 56–72, 2004.
3. Jan Camenisch, Ueli Maurer, and Markus Stadler. Digital payment systems with passive anonymity-revoking trustees. *Journal of Computer Security*, 5(1):69–89, 1997.
4. Ivan Damgård, Chaya Ganesh, Hamidreza Khoshakhlagh, Claudio Orlandi, and Luisa Siniscalchi. Balancing privacy and accountability in blockchain identity management. In *Cryptographers' Track at the RSA Conference*, pages 552–576. Springer, 2021.
5. Hossein Ghodosi, Josef Pieprzyk, and Rei Safavi-Naini. Dynamic threshold cryptosystems. In *Proceedings of PRAGOCRYPT*, volume 96, pages 370–379. Citeseer, 1996.
6. Jens Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35*, pages 305–326. Springer, 2016.
7. Au Man Ho, Susilo Willy, and Mu Yi. Constant-size dynamic k-TAA. In *Security and Cryptography for Networks, SCN*, pages 111–125, 2006.
8. Markus Jakobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In *Advances in Cryptology—ASIACRYPT 2000: 6th International Conference on the Theory and Application of Cryptology and Information Security Kyoto, Japan, December 3–7, 2000 Proceedings 6*, pages 162–177. Springer, 2000.
9. Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 571–589. Springer, 2004.
10. Joon Sik Kim, Kwangsu Lee, Jong Hwan Park, and Hyoseung Kim. Dynamic threshold key encapsulation with a transparent setup. *Cryptology ePrint Archive*, 2024.
11. David Pointcheval and Olivier Sanders. Short randomizable signatures. In *CT-RSA*, pages 111–126, 2016.
12. Mark D Ryan. Making decryption accountable. In *Security Protocols XXV: 25th International Workshop, Cambridge, UK, March 20–22, 2017, Revised Selected Papers 25*, pages 93–98. Springer, 2017.
13. W3C. Decentralized identifiers (DIDs) v0.11:data model and syntaxes for decentralized identifiers. https://w3c-ccg.github.io/did-spec/, 2018.
14. W3C. Verifiable credential data model v2.0. Available at https://www.w3.org/TR/vc-data-model-2.0/, 2024.