Thanks to Andy and Jay for all the help during this project

Anonymous Adhaar

The adhaar program is among the largest digital identity scheme in the world. It feats an enrollment of 1.2 billion people, accounting for around 90% of India's population.

Adhaar cards carry both demographic and biometric data, including the holder's date of birth and its fingerprint. They are used in a variety of contexts such as loan agreements or housing applications.

We present how to leverage zero-knowledge to verify an adhaar's card validity. This achievement has a variety of potential applications. Leveraging an existing public key infrastructure allows us to cheaply provide a robust proof of identity. Also, the zero-knowledge property of our verification provides a privacy-preserving way of conducting identity verification. Using the groth16 proving scheme opens up the ability to port valid adhaar card proof holders data onchain.

We developed an example webapp which allows any adhaar card holder to generate a proof of a valid adhaar card. We are open sourcing it today for anyone to use, fork or build on top of it. If you are interested to develop apps using what we have built or to implement a similar setup for other identity schemes, don't hesitate to get in touch.

Zero-knowledge setup

For verifying the adhaar's card validity, the circuit setup is straightforward. We wish to check that the signature provided by the user corresponds to one of a valid adhaar card. Our definition of validity will require that the provided message corresponds to an input signature and public key. We consider checking that a public key corresponds to a particular entity as "business logic". Hence, we will leave such a check to the proof verifying entity - such as a KYC provider backend or a smart contract.

Our circuit will perform two checks:

1. The RSA signature is correct. We raise the signature to the public exponent power, modulus the public key and obtain the provided document hash.
2. The SHA1 padding is correct. We check that the section 9.2 of RFC 8017 is followed when the provided message is raised to the public exponent power, modulus the public key.

Our circuit consists of four inputs:

signal input sign[nb]; // Signature; private signal input hashed[hashLen]; // Adhaar card's hash; private signal public input exp[nb]; // RSA public exponent signal public input modulus[nb]; // RSA modulus

First, we check that the provided RSA signature when raised to the public exponent, corresponds to the input hash. We re-used an implementation found here. Then, we ensure that the decrypted message padding, that is the padded hash, is correct. Adobe's pdf signing process follows the EMSA-PKCS1-v1_5

rules, detailed in this RFC. Adhaar cards use SHA1 as the hashing function before signing. Thus, we had to tweak our reference circuit, initially wrote for SHA256, to verify the padding's correctness. We provide a modified version ofthis circuit.

Although it may introduce limitations, we wish to keep the signature and the document's hash private. We only divulge what is required for any public validation procedure: the public exponent and the public key. In the case of verifying a proof of a valid Adhaar card, this would make an onchain contract able to require that the signature has been performed using a key emanating from the indian government.

Applications

Verifiable yet anonymous identity schemes enable interesting constructions.

First, it could provide an interesting avenue to re-think data hungry KYC procedures. Although we are conscious that a proof of a valid adhaar card alone can not constitute a sufficient piece of information for sensitive applications, it can still act as a component of a more complete KYC privacy-respecting process.

Another interesting implication regards verifiable speech. Over the last few months, different protocols and apps, such as Semaphore or HeyAnoun have brought forward the ability of zero-knowledge proof schemes to prove belonging to a group with verifiable properties, while not divulging any users sensitive information. In that vein, proving an Adhaar card's validity can act as an element of a verifiable yet anonymous voting system. One concrete instance of this could be sybil resistance for quadratic voting in the context of India's public goods projects.

Finally, using the groth16 proving scheme makes it possible to implement such ideas using a decentralized backend, such as Ethereum. One could imagine a registry contract, storing which addresses have posted valid adhaar cards proofs. This would allow for composability to kick-in, making it possible for the adhaar card pki to be leveraged for DeFi protocols or social apps.

Future directions and Applications Areas

An important limitation remains the ability to scale our circuit to large inputs. SHA1 remains a non-zk friendly hash, incurring important performance costs. A typical Adhaar pdf card size hovers around 650Kb, a size beyond what today’s circuits are capable of. Still, being able to perform hashing of such a document would reveal interesting, as it would allow to not allow prove a card’s validity but also its content. Folding schemes like Nova are prime candidates for exploring such an option.

More generally, proof time remains a bottleneck to a seamless UX. On a 8Gb RAM and 2.3GHz 2017 Macbook Pro, an above-average machine compared to everyday devices used in India, generating a proof entails a 10 minutes wait time. Here also, leveraging a different zero-knowledge proving backend, such as halo2, or scheme, as Nova, could provide improved performance metrics.

On a Dapp level, if we were to require from users to use their adhaar proof only once, it would imply tying a card to an adress. Such a requirement is not uncommon, decentralized apps often look for sybil resistant mechanisms in order to protect themselves from spam. If we were to link an adhaar card to a single address, a nullifier construction will be required. However, this would entail the ability for an agent having access to a complete adhaar database to detect which individuals have verified their adhaar card on chain, hence breaking anonymity. This may carry risks for users in a context of regulatory uncertainty.