Motivation

Imagine a zero knowledge airdrop

Or an anon message board with heated back and forth threads

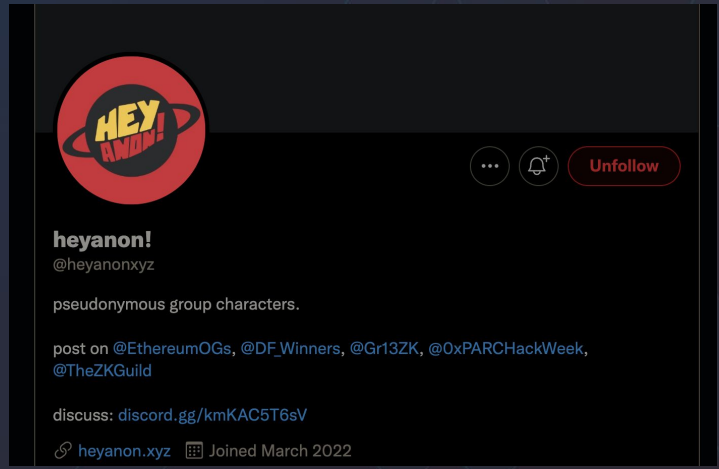
Or anon voting within some anonymity set

... Or any sybil resistant anon application

Anonymous Airdrops using ZK proofs

stealthdrop.xyz is an airdrop utility by 0xPARC (Aayush, Adhyyan, Nalin) that enables ERC20 token airdrops that can be claimed by completely anonymous accounts.

EDIT: @0xB07DAD discovered a vulnerability in this scheme described in more detail here. Fix and more to follow soon!



Motiv

Imagine

Or an ar

Or anon

... Or any

antimatter15 02/15/2022
Yeah but like hash the public key with some random input string

Adhyyan 02/15/2022
then someone could brute force all pub keys and check

antimatter15 02/15/2022
Oh I guess that's not that much entropy right

Adhyyan 02/15/2022
especially since the pub keys are all public in the js code
1

antimatter15 02/15/2022
Hmm yeah it feels like there's probably not a way to derive a nullifier here

Adhyyan 02/15/2022
is there no way to generate and verify a unique signature? (edited)
i remember BTC having a problem with malleable sigs. how did they solve it?

yush 02/15/2022
unless our zk proof subsumes another zk proof that k is properly generated from the private key, signature uniqueness c

@Adhyyan is there no way to generate and verify a unique signature? (edited)
antimatter15 02/15/2022
seems like not https://crypto.stackexchange.com/questions/26974/verifiably-deterministic-ecdsa-signatures

Cryptography Stack Exchange
Verifiably deterministic ECDSA signatures?
ECDSA signatures depend on parameter k that is chosen by the signer. As a result, there are many signatures for the same private key d and message m.
What I want to achieve is a deterministic sign...

yush 02/15/2022
funnily enough eddsa does not require the use of a unique random number for each signature, in which case we'd be go

antimatter15 02/15/2022
but an adversary could still create arbitrarily many signatures with the same key which are valid
seems like the ethereum spec says that signatures are supposed to be made with determinstic ecdsa, but an adversary

yush 02/15/2022
right exactly, thats what we are concerned with for ecdsa -- are you saying thats also true for eddsa?

antimatter15 02/15/2022
oh i misread, but is eddsa relevant?

yush 02/15/2022
oh no, it was just an interesting observation

Adhyyan 02/15/2022
Is there anything about ecdsa which is deterministic? We could use that for nullifiers
for eg. if theres some internal value not related to k, we could we use that

yyan i remember BTC having a problem with malleable sigs. how did they solve it?
ter15 02/15/2022
ility seems to be related but not exactly the same concept. malleability seems to refer to whether someone can alter a signature and create a
e without possessing the private key. it seems like what we need is "signature uniqueness" i.e. only one signature exists for a given secret key
sage.

imatter15 looks like EdDSA still employs a nonce, but it just specifies how that nonce is supposed to be generated (akin to deterministic ECDSA)
/15/2022
u're right; I looked at the full sig construction and it seems EDDSA suffers from the exact same issue
es me think that a unique signature scheme verifiable without a private key is impossible; either 1) it derives only from the public key, in which
can brute force, or 2) utilizes some deterministically seeded randomness, but the only possible deterministic, unknown random seed is the
ey

p 02/15/2022
ight—we should prolly post this lol

02/15/2022
ne realize that it's worth thinking through which class of 'zk-id' applications need nullifers vs. not (edited)

ter15 02/15/2022
ullifiers are less useful in general than one would expect for ECDSA in ZK, such as in the nft gated discord example because eg the owner can
get access and subse
's a sense in which t

02/15/2022
useful for soulbond-

ter15 02/15/2022

02/15/2022
point we may want

t's deterministically
d 'deterministic sigs'

1032 - Edwards-Cur
ds-Curve Digital Signat

n 02/15/2022
e brainstorming for

cious actors signs a
thdrawal we could h
er.

ink this would work

ter15 02/15/2022
ose lines I wonder if
omeone generates a

hat could even be d

February 19, 2022

gubsheep 02/19/2022
Big news yall. Me and @phated just met with the snaps lead and they have wasm working on a branch and are super excited for wasm support, which enables zk/snarkjs <> metamask stuff. Zk airdrop and others could be a reality within the next 6mo cc @lsankar @antimatter15 @yi @nibnalin @vivboop and more (edited)
6

justin 02/19/2022
the sad thing about how this is playing out is that snaps will never be compatible with hardware wallets

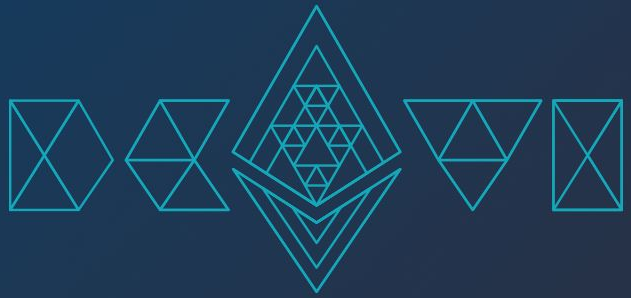
antimatter15 02/19/2022
or new hardware wallets will have to adapt to have beefier compute cores that can do snark stuff

@justin the sad thing about how this is playing out is that snaps will never be compatible with hardware wallets
gubsheep 02/19/2022
I also chatted with EF security team about this

@antimatter15 or new hardware wallets will have to adapt to have beefier compute cores that can do snark stuff
justin 02/19/2022
yep but then the price of secure enclaves will explode

gubsheep 02/19/2022
I want to bring them into the next learning group
1

justin 02/19/2022
so if I'm following correctly, the reason we don't use the circuit based on sigs anymore is that we can't do nullifiers properly + it's much more expensive, right?
or maybe we can could break the circuit in pieces
do the part dealing with the private key on the enclave, and the rest on the host computer
but I don't know much about composing circuits that way



a new zk nullifier scheme for ECDSA

with Aayush Gupta, Kobi Gurkan, Wei Jie Koh, Lakshman Sankar

Personae Labs Geometry Research Personae Labs

inspired by discussions with nalin/adhyyan/gubsheep/uma/boneh/vivek

work originally done at 0xPARC/MIT

Properties we want

Unique

Deterministic

Verifiable without secret key

Noninteractive (unlike tornado.cash or semaphore)

Ideas that will and won't work

deterministic ECDSA signatures

VUFs/unique signatures

hash(message, public key)

hash(message, secret key)

hash(message, public key)$^{\text{secret key}}$ -> DDH-VRF!

We want a deterministic function of a user's secret key, that can be verified with only their public key, and keeps them anonymous

Solution

If your eth keypair is (sk, pk = g^{sk}) and public message is m

Signature:

public: $hash[m, pk]^{sk}$ <-- nullifier

private: $c = hash2(g, pk, hash[m, pk], hash[m, pk]^{sk}, g^r, hash[m, pk]^r)$

$s = r + sk * c$

$pk = g^{sk}$

g^r [optional output]

$hash[m, pk]^r$ [optional output]

Verifier check in SNARK:

$g^{[r + sk * c]} / (g^{sk})^c = g^r$

$hash[m, g^{sk}]^{[r + sk * c]} / (hash[m, pk]^{sk})^c = hash[m, pk]^r$

$c = hash2(g, g^{sk}, hash[m, g^{sk}], hash[m, pk]^{sk}, g^r, hash[m, pk]^r)$

pk is in anonymity set (merkle proof)

Quantum Secrecy - Interactivity Tradeoff

Noninteractive		Quantum secrecy
ZK Nullifiers	?	Semaphore

Key: If nullifier = any deterministic function(secret key), a future quantum adversary can break *past* anonymity.

But if we source randomness beyond the secret key, the user needs to remember that value (password).

Quantum oracles that can break ECDSA

Between 30-never years. Maybe average somewhere around 2100??

Shors -> prime factorization and discrete log. No hashes.

Requires 2330 signal qubits *plus* noise correction.

As of 2022, we currently have < 20 signal qubits.

1 pager with my mental model here.

Understanding

Paper with proofs: EUF-CMA, secrecy, uniqueness (done)

tl;dr if you can break this, you can break Diffie-Hellman

Blog post (done)

Proof of concept (done)

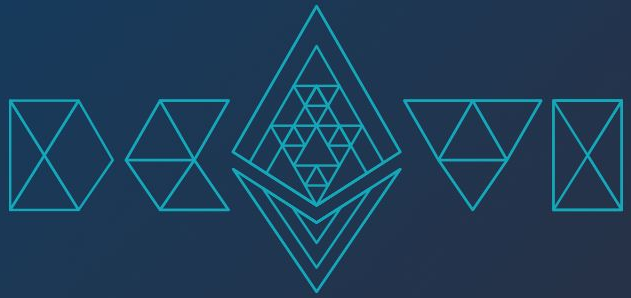
Integrate into wallets: Burner wallet, Metamask in progress

Create an ERC/EIP standard

IETF RFC Appendix Addition

Repo: github.com/zk-nullifier-sig/zk-nullifier-sig

@yush_g @personae_labs



Enter your slide title here.

Your subtitle here.

Your Name

Your title, your organization

Section 1

Section 1 title here.

Section 1 title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- Sollicitudin
- Consectetur
  - Condimentum
    - **Magna**
    - **Ligula**

Section 1 details with an image. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Enter your main point / statement here.

Section 1 details with a main point. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Section 2

Section 2 title here.

Section 2 title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- Sollicitudin
- Consectetur
  - Condimentum
    - **Magna**
    - **Ligula**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- Sollicitudin
- Consectetur
  - Condimentum
    - **Magna**
    - **Ligula**

Section 2 details with an image. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Section 2 details with a main point. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Enter your main point / statement here.

Enter your main point / statement here.

Here’s the timeline.

Event 1

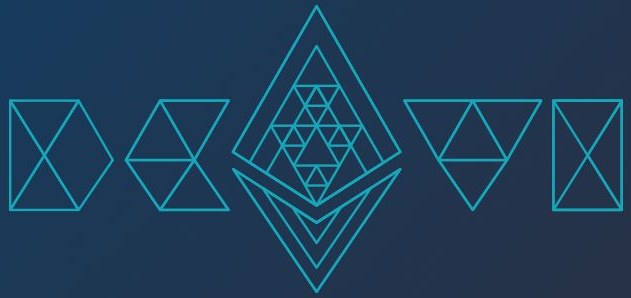
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Event 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Event 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.



Thank you!

Your Name

Your title, your organization

email@emailaddress.com

@twitterhandle