Ethical identity, ring VRFs, and zero-knowledge continuations

Jeffrey Burdges   Handan Kilinc-Alper   Alistair Stewart
Sergey Vasilyev

16-17 May 2024

https://eprint.iacr.org/2022/002
https://github.com/w3f/ring-vrf/

Zero-knowledge in substrate?

Yes WASM works, but 10x slower for some optimized code.

EIP-2537 (BLS12-381) adds 9 precompiles (hostcalls). 👎

Extra complexity means fewer curves
- Addition & hash-to-curve appear unecessary.

Inflexibility yields worse performance
- Groth16 needs 4 pairings
- Batch verifiers
- Large parameters like KZG trusted setups

https://github.com/paritytech/arkworks-extensions

Zero-knowledge in substrate?

Yes WASM works, but 10x slower for some optimized code.

EIP-2537 (BLS12-381) adds 9 precompiles (hostcalls). 👎

Add hostcalls for the "slow parts" but use WASM elsewhere 👍

- Bandersnatch, etc. — single & multi-scalar multiplication
- BLS12-381, BLS12-377, BW6-761, BW6-767, BN254 —
  - Same, plus multi-miller loop & final exponentiation

`https://github.com/paritytech/arkworks-extensions`

Arkworks & others mostly descend from Zcash.

Adapt crates directly, without reimplementation.

```
#[cfg(not(feature = "substrate-curves"))]
mod curves {
    pub use ark_ed_on_bls12_381_bandersnatch as bandersnatch;
    pub use ark_bls12_381 as bls12_381;
}

#[cfg(feature = "substrate-curves")]
mod curves {
    pub use sp_ark_ed_on_bls12_381_bandersnatch as bandersnatch;
    pub use sp_ark_bls12_381 as bls12_381;
}

pub use curves::*;
```

https://github.com/paritytech/arkworks-extensions

Ring signatures prove the actual signer exists in some publicly specified list, known as the ring.

Examples: Some “deniable” key exchanges, Monero, ZCash, etc.

Ring can be a fancy set commitment like ZCash, but membership proofs are always very expensive.

EC VRF

A verifiable random function (VRF) proves evaluation of a pseudo-random function (PRF) determined by a signing key.

$\mathrm{ECVRF.Verify}(\mathrm{msg}, \mathrm{aux}, \mathrm{pk}, (\mathrm{out}, R, R_{\mathrm{msg}}, s))$

$$
\begin{aligned}
\mathrm{inbase} &:= H_{\mathcal{G}}(\mathrm{msg}) \\
c &:= H(\mathrm{msg}, \mathrm{aux}, \mathrm{pk}, \mathrm{out}, R, R_{\mathrm{msg}}) \\
s\,\mathrm{inbase} &== c\,\mathrm{out} + R_{\mathrm{msg}} \\
s\,G &== c\,\mathrm{pk} + R \\
&\texttt{return}\, H(\mathrm{out}, \mathrm{msg})
\end{aligned}
$$

Ring VRF is a ring signature that's also a VRF.

A ring verifiable random function (ring VRF) is a ring signature that proves evaluation of a pseudo-random function (PRF) determined by the actual key pair.

For what do you use a ring VRF?

I can't believe I got away with ______.

The NSA's massive stack of amateur porn.

Best rejected paper award.

Pedersen VRF

We set $\mathsf{compk} := \mathsf{sk}\, G + b\, K$ to be a Pedersen commitment to sk.

$\mathsf{PedersenVRF.Verify}(\mathsf{msg}, \mathsf{aux}, \mathsf{compk}, (\mathsf{out}, R, R_{\mathsf{msg}}, s, t))$

$$\begin{aligned} \mathsf{inbase} &:= H_{\mathcal{G}}(\mathsf{msg}) \\ c &:= H(\mathsf{msg}, \mathsf{aux}, \mathsf{compk}, \mathsf{out}, R, R_{\mathsf{msg}}) \\ s\, \mathsf{inbase} &== c\, \mathsf{out} + R_{\mathsf{msg}} \\ t\, K + s\, G &== c\, \mathsf{compk} + R \\ &\mathtt{return}\, H(\mathsf{out}, \mathsf{msg}) \end{aligned}$$

Just EC VRF except for t, b and pk being compk.

Zero-knowledge continuations..

Q: What are the fastest/cheapest SNARK proofs?

A: Ones we reuse without reproving.

Groth16.Verify$(X, (A, B, C))$

$$e(A, B) = e([\alpha]_1, [\beta]_2) \cdot e(X, [\gamma]_2) \cdot e(C, [\delta]_2)$$

Groth16.Verify$(X, (A, B, C))$

$$e(A, B) = e([\alpha]_1, [\beta]_2) \cdot e(X, [\gamma]_2) \cdot e(C, [\delta]_2)$$

$$X = \mathsf{sk}\, G + \mathsf{comring}\, L$$

$$\mathsf{Groth16}\left\{ \mathsf{sk}, \mathsf{comring} \;\middle|\; \begin{array}{l} \exists d \text{ s.t. } \mathsf{pk} \leftarrow \mathsf{Posideon}(sk, d) \\ \exists o \text{ s.t. } \mathsf{pk} \in_o \mathsf{comring} \end{array} \right\}$$

Groth16.Verify$(X, (A, B, C))$

$$e(A, B) = e([\alpha]_1, [\beta]_2) \cdot e(X, [\gamma]_2) \cdot e(C, [\delta]_2)$$

$$X = \mathsf{sk}\ G + \mathsf{comring}\ L$$

$$\text{Groth16} \left\{ \mathsf{sk}, \mathsf{comring} \;\middle|\; \begin{array}{l} \exists d \text{ s.t. } \mathsf{pk} \leftarrow \mathsf{Posideon}(sk, d) \\ \exists o \text{ s.t. } \mathsf{pk} \in_o \mathsf{comring} \end{array} \right\}$$

Special(ized) G(roth16) means inner Groth16 leaks secrets, but..

Groth16.Verify$(X, (A, B, C))$

$$e(A, B) = e([\alpha]_1, [\beta]_2) \cdot e(X, [\gamma]_2) \cdot e(C, [\delta]_2)$$

$$\begin{aligned} X &= \mathrm{sk}\, G + b\, K + \mathrm{comring}\, L \\ &= \quad \mathrm{compk} \quad + \mathrm{comring}\, L \end{aligned}$$

Add $K_\delta := \frac{\gamma}{\delta} K$ to trusted setup

$$\begin{aligned} X' &:= X + b\, K & \qquad B' &:= r_1 B + r_1 r_2 [\delta]_2 \\ A' &:= \frac{1}{r_1} A & \qquad C' &:= C + r_2 A + b\, K_\delta \end{aligned}$$

Marginal signer cost of eight $\mathcal{G}_1$ mults plus two $\mathcal{G}_2$ mults 🎉

Are there other zero-knowledge continuations?

Avoid the Groth16 side channel and use CDH over Posideon..

$$X = \mathsf{sk}\, G + b\, K + J_{\mathsf{pk}}.x\, L_x + J_{\mathsf{pk}}.y\, L_y$$

$$\text{Groth16}\left\{\, \mathsf{sk}_0 + \mathsf{sk}_1 2^{128}, J_{\mathsf{pk}} \;\middle|\; \exists d \text{ s.t. } J_{\mathsf{pk}} = \mathsf{sk}_0 J_0 + \mathsf{sk}_1 J_1 + d J_2 \,\right\}$$

Use with hidden KZG opening of J_{pk} like Caulk/Caulk+

Revokation could be done using a “cuckoo filter” in a KZG.

$$\text{Groth16}\left\{ \mathsf{sk}, \mathsf{pk}, i_1, i_2, i_3, \mathsf{comring} \;\middle|\; \begin{array}{c} \exists d \text{ s.t. } \mathsf{pk} \leftarrow \mathsf{Posideon}(sk, d) \\ \exists o \text{ s.t. } \mathsf{pk} \in_o \mathsf{comring} \\ (i_1, i_2, i_3) \leftarrow \mathsf{Posideon}(pk) \end{array} \right\}$$

Non-revokation proof: $\mathsf{pk} \neq f(i_j)$ for $j = 1, 2, 3$ where f is a KZG
Nolonger like Caulk/Caulk+.

Q: How can identity be safe for online use?

A: By revealing nothing except users' uniqueness.

No W3C attribute based bullshit!

Attribute credentials signed by authority.

User agent:

- Validates TLS cert of “site.com”
- Gets attribute request: name, age, nationality, employment status
- Asks user to approve sharing those attributes with “site.com”.
  If approved, proves the atttributes

Issues:

- Attributes are unecessarily invasive.
- Attributes leak across domains. Users cannot change attributes.
- Users make mistakes and/or can be forced.

Attribute credentials signed by authority.

User agent:

- Validates TLS cert of "site.com"
- Gets attribute request: name, age, nationality, employment status
- Validates DPA certificate for attributes at "site.com"
- Asks user to approve sharing those attributes with "site.com". If approved, proves the atttributes

Issues:

- Attributes are unecessarily invasive.
- Attributes leak across domains. Users cannot change attributes.
- Users make mistakes and/or can be forced.

Attribute requests need certificate infrastructure.

Ring consists of people, with one key per person,
maybe populated from government identity documents.

User agent:

1st) validates TLS cert of “site.com”, including CT logs.

2nd) sends ring VRF signature with msg = “`site.com`“.

Ring consists of people, with one key per person,
maybe populated from government identity documents.

User agent:

1st) validates TLS cert of "site.com", including CT logs.

2nd) sends ring VRF signature with msg = "`site.com`".

Do we have a "right to be forgotten" at "site.com"?

If so, use msg = "`site.com`" $+\!\!+$ month

Q: Can ring VRFs give us efficent anonymous payments?

A: Not really, but they can give anonymous rate limiting

"No civilization can possibly survive to an interstellar spacefaring phase unless it limits its numbers" (and its consumption)

— Carl Sagan

We're headed for $+4^{\circ}$C by 2100, so uninhabitable tropics and world carrying capacity below 1 billion people (Steffan).

50% odds "of a synchronous crop failure [$>10\%$] across all four [major maize producing] countries during 2040s" (Chatham House)

Anonymous rationing uses msg $=$ "$\texttt{moutarde}$" $+\!\!+$ week $+\!\!+$ counter
And treats outputs as short lived nullifiers.

Also yields free-to-play games, promotional discounts, etc.

As fraudulent TLS and covid certificates are commonplace..

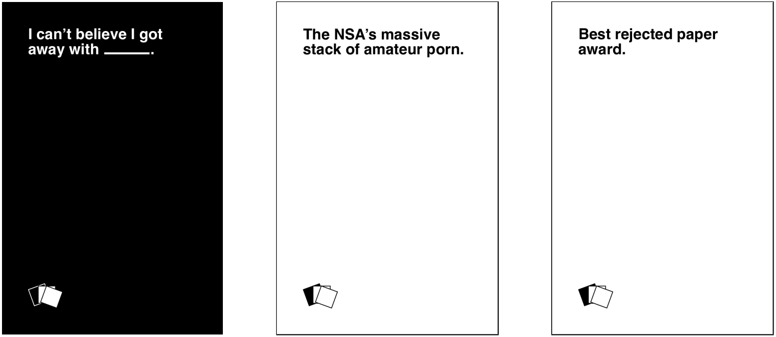
Q: How can ration cards be trusted?

A: By asking users trust a public list of residents, not certificates.

Sassafras

Sassafras: Semi-anonymous sortition of staked assignees
for fixed-time rhythmic assignment of slots

It's a (semi) secret single leader election (semi-SSLE)
by cards against humanity.



Sassafras

Disadvantages:
- Network layer anonymity is weak, but we care little..

Advantages:
- Ouroboros Praos quality randomness
- Vastly more efficient than Boneh's shuffle SSLEs
- Block producers can prove their slot in advance
- Users send tx to upcoming slots via Tor-like .onion circuits.
- Avoids need for memepools, saving bandwidth and CPU.
- Better MEV defenses

Sassafras

Disadvantages:
- Network layer anonymity is weak, but we care little..

Advantages:
- Ouroboros Praos quality randomness
- Vastly more efficient than Boneh's shuffle SSLEs
- Block producers can prove their slot in advance
- Users send tx to upcoming slots via Tor-like .onion circuits.
- Avoids need for memepools, saving bandwidth and CPU.
- Better MEV defenses

Smart contracts, the Ford Pinto of security.