Paper notes

arnaucube

Abstract

Notes taken while reading papers. Usually while reading papers I take handwritten notes, this document contains some of them re-written to $LaTeX$.

The notes are not complete, don't include all the steps neither all the proofs.

Contents

1 SnarkPack

Notes taken while reading SnarkPack paper [1].

Groth16 proof aggregation.

i. Simple verification:
Proof: $\pi_i = (A_i, B_i, C_i)$
Verifier checks: $e(A_i, B_i) == e(C_i, D)$
Where D is the CRS.

ii. Batch verification: $r \in^{\$} F_q$
$r^i \cdot e(A_i, B_i) == e(C_i, D)$
$\implies \prod e(A_i, B_i)^{r^i} == \prod e(C_i, D)^{r^i}$
$\implies \prod e(A_i, B_i^{r^i}) == \prod e(C_i^{r^i}, D)$

iii. Snark Aggregation verification:
$z_{AB} = \prod e(A_i, B_i^{r^i})$
$z_C = \prod C_i^{r^i}$
Verification: $z_{AB} == e(z_C, D)$

2 Sonic

Notes taken while reading Sonic paper [2]. Does not include all the steps, neither the proofs.

2.1 Structured Reference String

$\{\{g^{x^i}\}_{i=-d}^{d}, \{g^{\alpha x^i}\}_{i=-d, i\neq 0}^{d}, \{h^{x^i}, h^{\alpha x^i}\}_{i=-d}^{d}, e(g, h^{\alpha})\}$

2.2 System of constraints

Multiplication constraint: $a \cdot b = c$
Q linear constraints:

$$a \cdot u_q + b \cdot v_q + c \cdot w_q = k_q$$

with $u_q, v_q, w_q \in \mathbb{F}^n$, and $k_q \in \mathbb{F}_p$.

Example: $x^2 + y^2 = z$

$$a = (x, y), \qquad b = (x, y), \qquad c = (x^2, y^2)$$

i. $(x, y) \cdot (1, 0) + (x, y) \cdot (-1, 0) + (x^2, y^2) \cdot (0, 0) = 0 \longrightarrow x - x = 0$

ii. $(x, y) \cdot (0, 1) + (x, y) \cdot (0, -1) + (x^2, y^2) \cdot (0, 0) = 0 \longrightarrow y - y = 0$

iii. $(x, y) \cdot (0, 0) + (x, y) \cdot (0, 0) + (x^2, y^2) \cdot (1, 1) = z \longrightarrow x^2 + y^2 = z$

So,

$$u_1 = (1, 0) \quad v_1 = (-1, 0) \quad w_1 = (0, 0) \quad k_1 = 0$$

$$u_2 = (0, 1) \quad v_2 = (0, -1) \quad w_2 = (0, 0) \quad k_2 = 0$$

$$u_3 = (0, 0) \quad v_3 = (0, 0) \quad w_3 = (1, 1) \quad k_2 = z$$

Compress n multiplication constraints into an equation in formal indeterminate Y:

$$\sum_{i=1}^{n} (a_i b_i - c_i) \cdot Y^i = 0$$

encode into negative exponents of Y:

$$\sum_{i=1}^{n}(a_i b_i - c_i) \cdot Y^- i = 0$$

Also, compress the Q linear constraints, scaling by Y^n to preserve linear independence:

$$\sum_{q=1}^{Q}(a \cdot u_q + b \cdot v_q + c \cdot w_q - k_q) \cdot Y^{q+n} = 0$$

Polys:

$$u_i(Y) = \sum_{q=1}^{Q} Y^{q+n} \cdot u_{q,i}$$

$$v_i(Y) = \sum_{q=1}^{Q} Y^{q+n} \cdot v_{q,i}$$

$$w_i(Y) = -Y^i - Y^{-1} + \sum_{q=1}^{Q} Y^{q+n} \cdot w_{q,i}$$

$$k(Y) = \sum_{q=1}^{Q} Y^{q+n} \cdot k_q$$

Combine the multiplicative and linear constraints to:

$$a \cdot u(Y) + b \cdot v(Y) + c \cdot w(Y) + \sum_{i=1}^{n} a_i b_i (Y^i + Y^{-i}) - k(Y) = 0$$

where $a \cdot u(Y) + b \cdot v(Y) + c \cdot w(Y)$ is embeded into the constant term of the polynomial $t(X, Y)$.

Define $r(X, Y)$ s.t. $r(X, Y) = r(XY, 1)$.

$$\implies r(X, Y) = \sum_{i=1}^{n}(a_i X^i Y^i + b_i X^{-i} Y^{-i} + c_i X^{-i-n} Y^{-i-n})$$

$$s(X, Y) = \sum_{i=1}^{n}(u_i(Y) X^{-i} + v_i(Y) X^i + w_i(Y) X^{i+n})$$

$$r'(X, Y) = r(X, Y) + s(X, Y)$$

$$t(X, Y) = r(X, Y) + r'(X, Y) - k(Y)$$

The coefficient of X^0 in $t(X, Y)$ is the left-hand side of the equation.

Sonic demonstrates that the constant term of $t(X, Y)$ is zero, thus demonstrating that our constraint system is satisfied.

2.2.1 The basic Sonic protocol

1. Prover constructs $r(X, Y)$ using their hidden witness
2. Prover commits to $r(X, 1)$, setting the maximum degree to n
3. Verifier sends random challenge y
4. Prover commits to $t(X, y)$. The commitment scheme ensures that $t(X, y)$ has no constant term.
5. Verifier sends random challenge z
6. Prover opens commitments to $r(z, 1), r(z, y), t(z, y)$
7. Verifier calculates $r'(z, y)$, and checks that

$$r(z, y) \cdot r'(z, y) - k(y) == t(z, y)$$

Steps 3 and 5 can be made non-interactive by the Fiat-Shamir transformation.

2.2.2 Polynomial Commitment Scheme

Sonic uses an adaptation of KZG [3], want:

i. *evaluation binding*, i.e. given a commitment F, an adversary cannot open F to two different evaluations v_1 and v_2

ii. *bounded polynomial extractable*, i.e. any algebraic adversary that opens a commitment F knows an opening $f(X)$ with powers $-d \leq i \leq max, i \neq 0$.

PC scheme (adaptation of KZG):

i. Commit(info, $f(X)$) $\longrightarrow F$:

$$F = g^{\alpha \cdot x^{d-max}} \cdot f(x)$$

ii. Open(info, F, z, $f(x)$) $\longrightarrow (f(z), W)$:

$$w(X) = \frac{f(X) - f(z)}{X - z}$$

$$W = g^{w(x)}$$

iii. Verify(info, F, z, (v, W)) $\longrightarrow 0/1$:
Check:

$$e(W, h^{\alpha \cdot x}) \cdot e(g^v W^{-z}, h^{\alpha}) == e(F, h^{x^{-d+max}})$$

2.3 Succint signatures of correct computation

Signature of correct computation to ensure that an element $s = s(z, y)$ for a known polynomial

$$s(X, Y) = \sum_{i,j=-d}^{d} s_{i,j} \cdot X^i \cdot Y^i$$

Use the structure of $s(X, Y)$ to prove its correct calculation using a *permutation argument* $\longrightarrow$ *grand-product argument* inspired by Bayer and Groth, and Bootle et al.

Restrict to constraint systems where $s(X, Y)$ can be expressed as the sum of M polynomials. Where $j - th$ poly is of the form:

$$\Psi_j(X, Y) = \sum_{i=1}^{n} \psi_{j,\sigma_{j,i}} \cdot X^i \cdot Y^{\sigma_{j,i}}$$

where σ_j is the fixed polynomial permutation, and $\phi_{j,i} \in \mathbb{F}$ are the coefficients.

$\boxed{\text{WIP}}$

3 BLS signatures

Notes taken while reading about BLS signatures [4].

Key generation $sk \in \mathbb{Z}_q$, $pk = [sk] \cdot g_1$, where $g_1 \in G_1$, and is the generator.

Signature

$$\sigma = [sk] \cdot H(m)$$

where H is a function that maps to a point in G_2. So $H(m), \sigma \in G_2$.

Verification

$$e(g_1, \sigma) == e(pk, H(m))$$

Unfold:

$$e(pk, H(m)) = e([sk] \cdot g_1, H(m) = e(g_1, H(m))^{sk} = e(g_1, [sk] \cdot H(m)) = e(g_1, \sigma))$$

Aggregation Signatures aggregation:

$$\sigma_{aggr} = \sigma_1 + \sigma_2 + \ldots + \sigma_n$$

where $\sigma_{aggr} \in G_2$, and an aggregated signatures is indistinguishible from a non-aggregated signature.

Public keys aggregation

$$pk_{aggr} = pk_1 + pk_2 + \ldots + pk_n$$

where $pk_{aggr} \in G_1$, and an aggregated public keys is indistinguishible from a non-aggregated public key.

Verification of aggregated signatures Identical to verification of a normal signature as long as we use the same corresponding aggregated public key:

$$e(g_1, \sigma_{aggr}) == e(pk_{aggr}, H(m))$$

Unfold:

$$e(pk_{aggr}, H(m)) = e(pk_1 + pk_2 + \ldots + pk_n, H(m)) =$$

$$= e([sk_1] \cdot g_1 + [sk_2] \cdot g_1 + \ldots + [sk_n] \cdot g_1, H(m)) =$$
$$= e([sk_1 + sk_2 + \ldots + sk_n] \cdot g_1, H(m)) =$$
$$= e(g_1, H(m))^{(sk_1 + sk_2 + \ldots + sk_n)} =$$
$$= e(g_1, [sk_1 + sk_2 + \ldots + sk_n] \cdot H(m)) =$$
$$= e(g_1, [sk_1] \cdot H(m) + [sk_2] \cdot H(m) + \ldots + [sk_n] \cdot H(m)) =$$
$$= e(g_1, \sigma_1 + \sigma_2 + \ldots + \sigma_n) = e(g_1, \sigma_{aggr})$$

References

[1] Nicolas Gailly, Mary Maller, and Anca Nitulescu. Snarkpack: Practical snark aggregation. Cryptology ePrint Archive, Paper 2021/529, 2021. https://eprint.iacr.org/2021/529.

[2] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updateable structured reference strings. Cryptology ePrint Archive, Paper 2019/099, 2019. https://eprint.iacr.org/2019/099.

[3] A. Kate, G. M. Zaverucha, , and I. Goldberg. Constant-size commitments to polynomials and their application, 2010. https://www.iacr.org/archive/asiacrypt2010/6477178/6477178.pdf.

[4] Eth2.0. Eth2.0 book - bls signatures, 2010. https://eth2book.info/altair/part2/building_blocks/signatures.