

# Zero Knowledge Succinct Noninteractive ARguments of Knowledge

Saravanan Vijayakumaran  
sarva@ee.iitb.ac.in

Department of Electrical Engineering  
Indian Institute of Technology Bombay

October 15, 2019

# zkSNARKs

- Arguments
  - ZK proofs where soundness guarantee is required only against PPT provers
- Noninteractive
  - Proof consists of a single message from prover to verifier
- Succinct
  - Proof size is  $\mathcal{O}(1)$
  - Requires a trusted setup to generate a common reference string
  - CRS size is linear in size of assertion being proved

# Bilinear Pairings

- Let  $G$  and  $G_T$  be two cyclic groups of prime order  $p$
- In practice,  $G$  is an elliptic curve group and  $G_T$  is subgroup of  $\mathbb{F}_{r^n}^*$  where  $r$  is a prime
- Let  $G = \langle g \rangle$ , i.e.  $G = \{g^\alpha \mid \alpha \in \mathbb{Z}_p\}$
- A symmetric **pairing** is a efficient map  $e : G \times G \mapsto G_T$  satisfying
  1. **Bilinearity**:  $\forall \alpha, \beta \in \mathbb{Z}_p$ , we have  $e(g^\alpha, g^\beta) = e(g, g)^{\alpha\beta}$
  2. **Non-degeneracy**:  $e(g, g)$  is not the identity in  $G_T$
- Finding discrete logs is assumed to be difficult in both groups
- Pairings enable multiplication of secrets

# Computational Diffie-Hellman Problem

- **The CDH experiment  $\text{CDH}_{\mathcal{A}, \mathcal{G}}(n)$ :**
  1. Run  $\mathcal{G}(1^n)$  to obtain  $(G, q, g)$  where  $G$  is a cyclic group of order  $q$  (with  $\|q\| = n$ ), and a generator  $g \in G$ .
  2. Choose a uniform  $x_1, x_2 \in \mathbb{Z}_q$  and compute  $h_1 = g^{x_1}, h_2 = g^{x_2}$ .
  3.  $\mathcal{A}$  is given  $G, q, g, h_1, h_2$  and it outputs  $h \in \mathbb{Z}_q$ .
  4. Experiment output is 1 if  $h = g^{x_1 \cdot x_2}$  and 0 otherwise.
- **Definition:** We say that **the CDH problem is hard relative to  $\mathcal{G}$**  if for every PPT adversary  $\mathcal{A}$  there is a negligible function  $\text{negl}$  such that

$$\Pr[\text{CDH}_{\mathcal{A}, \mathcal{G}}(n) = 1] \leq \text{negl}(n).$$

# Decisional Diffie-Hellman Problem

- **The DDH experiment**  $\text{DDH}_{\mathcal{A}, \mathcal{G}}(n)$ :

1. Run  $\mathcal{G}(1^n)$  to obtain  $(G, q, g)$  where  $G$  is a cyclic group of order  $q$  (with  $\|q\| = n$ ), and a generator  $g \in G$ .
2. Choose a uniform  $x, y, z \in \mathbb{Z}_q$  and compute  $u = g^x, v = g^y$
3. Choose a bit  $b \xleftarrow{\$} \{0, 1\}$  and compute  $w = g^{bz + (1-b)xy}$
4. Give the triple  $u, v, w$  to the adversary  $\mathcal{A}$
5.  $\mathcal{A}$  outputs a bit  $b' = \mathcal{A}(G, q, g, u, v, w)$

- **Definition:** We say that **the DDH problem is hard relative to  $\mathcal{G}$**  if for all PPT adversaries  $\mathcal{A}$  there is a negligible function  $\text{negl}$  such that

$$|\Pr[\mathcal{A}(G, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(G, q, g, g^x, g^y, g^{xy}) = 1]| \leq \text{negl}(n)$$

- If  $G$  has a pairing, then DDH problem is easy in  $G$

## Some Exercises on Pairings

- A symmetric **pairing** is a efficient map  $e : G \times G \mapsto G_T \subset F_{r^n}^*$  satisfying
  1. **Bilinearity**:  $\forall \alpha, \beta \in \mathbb{Z}_p$ , we have  $e(g^\alpha, g^\beta) = e(g, g)^{\alpha\beta}$
  2. **Non-degeneracy**:  $e(g, g)$  is not the identity in  $G_T$
- Reduce the following expressions
  - $e(g^a, g) e(g, g^b)$
  - $e(g, g^a) e(g^b, g)$
  - $e(g^a, g^{-b}) e(u, v) e(g, g)^c$
  - $\prod_{i=1}^m e(g, g^{a_i})^{b_i}$
- Show that if  $e(u, v) = 1$  then  $u = 1$  or  $v = 1$

# Applications of Pairings

- Three-party Diffie Hellman key agreement
  - Three parties Alice, Bob, Carol have private-public key pairs  $(a, g^a), (b, g^b), (c, g^c)$  where  $G = \langle g \rangle$
  - Alice sends  $g^a$  to the other two
  - Bob sends  $g^b$  to the other two
  - Carol sends  $g^c$  to the other two
  - Each party can compute common key
$$K = e(g, g)^{abc} = e(g^b, g^c)^a = e(g^a, g^c)^b = e(g^a, g^b)^c$$
- BLS Signature Scheme
  - Suppose  $H : \{0, 1\}^* \mapsto G$  is a hash function
  - Let  $(x, g^x)$  be a private-public key pair
  - BLS signature on message  $m$  is  $\sigma = (H(m))^x$
  - Verifier checks that  $e(g, \sigma) = e(g^x, H(m))$

# Knowledge of Exponent Assumptions

- **Knowledge of Exponent Assumption (KEA)**

- Let  $G$  be a cyclic group of prime order  $p$  with generator  $g$  and let  $\alpha \in \mathbb{Z}_p$
- Given  $g, g^\alpha$ , suppose a PPT adversary can output  $c, \hat{c}$  such that  $\hat{c} = c^\alpha$
- The only way he can do so is by choosing some  $\beta \in \mathbb{Z}_p$  and setting  $c = g^\beta$  and  $\hat{c} = (g^\alpha)^\beta$

- **$q$ -Power Knowledge of Exponent ( $q$ -PKE) Assumption**

- Let  $G$  be a cyclic group of prime order  $p$  with a pairing  $e : G \times G \mapsto G_T$
- Let  $G = \langle g \rangle$  and  $\alpha, s$  be randomly chosen from  $\mathbb{Z}_p^*$
- Given  $g, g^s, g^{s^2}, \dots, g^{s^q}, g^\alpha, g^{\alpha s}, g^{\alpha s^2}, \dots, g^{\alpha s^q}$ , suppose a PPT adversary can output  $c, \hat{c}$  such that  $\hat{c} = c^\alpha$
- The only way he can do so is by choosing some  $a_0, a_1, \dots, a_q \in \mathbb{Z}_p$  and setting  $c = \prod_{i=0}^q (g^{s^i})^{a_i}$  and  $\hat{c} = \prod_{i=0}^q (g^{\alpha s^i})^{a_i}$



# Checking Polynomial Evaluation

- Prover knows a polynomial  $p(x) \in \mathbb{F}_p[x]$  of degree  $d$
- Verifier wants to check that prover computes  $g^{p(s)}$  for some randomly chosen  $s \in \mathbb{F}_p$
- Verifier does not care which  $p(x)$  is used but cares about the evaluation point  $s$
- Verifier sends  $g^{s^i}, i = 0, 1, 2, \dots, d$  to prover
- If  $p(x) = \sum_{i=0}^d p_i x^i$ , prover can compute  $g^{p(s)}$  as

$$g^{p(s)} = \prod_{i=0}^d \left(g^{s^i}\right)^{p_i}$$

- But prover could have computed  $g^{p(t)}$  for some  $t \neq s$
- Verifier also sends  $g^{\alpha s^i}, i = 0, 1, 2, \dots, d$  for some randomly chosen  $\alpha \in \mathbb{F}_p^*$
- Prover can now compute  $g^{\alpha p(s)}$
- Anyone can check that  $e(g^\alpha, g^{p(s)}) = e(g^{\alpha p(s)}, g)$
- But why can't the prover cheat by returning  $g^{p(t)}$  and  $g^{\alpha p(t)}$  ?

# Schwartz-Zippel Lemma

## Lemma

Let  $\mathbb{F}$  be any field. For any nonzero polynomial  $f \in \mathbb{F}[x]$  of degree  $d$  and any finite subset  $S$  of  $\mathbb{F}$ ,

$$\Pr [f(s) = 0] \leq \frac{d}{|S|}$$

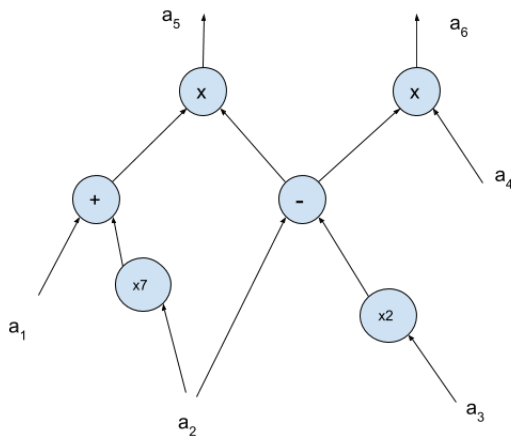
when  $s$  is chosen uniformly from  $S$ .

- Suppose  $\mathbb{F}$  is a finite field of order  $\approx 2^{256}$
- If  $s$  is chosen uniformly from  $\mathbb{F}$ , then it is unlikely to be a root of low-degree polynomials
- Equality of polynomials can be checked by evaluating them at the same random point
- **Application:** Suppose prover wants to prove that he knows a secret polynomial  $p(x)$  which is divisible by another public polynomial  $t(x)$ 
  - Verifier sends  $g^{s^i}, g^{\alpha s^i}, i = 0, 1, 2, \dots, d$  to prover
  - Prover computes  $h(x) = \frac{p(x)}{t(x)} = \sum_{i=0}^d h_i x^i$  and calculates  $g^{h(s)}$  using the coefficients  $h_i$
  - Verifier gets  $g^{p(s)}, g^{h(s)}, g^{\alpha p(s)}, g^{\alpha h(s)}$  and checks

$$e(g, g^{p(s)}) = e(g^{h(s)}, g^{t(s)})$$

$$e(g^\alpha, g^{p(s)}) = e(g^{\alpha p(s)}, g), \quad e(g^\alpha, g^{h(s)}) = e(g^{\alpha h(s)}, g)$$

# Arithmetic Circuits



Circuits consisting of additions and multiplications modulo  $p$

# Quadratic Arithmetic Programs

## Definition

A QAP  $Q$  over a field  $\mathbb{F}$  contains three sets of polynomials  $\mathcal{V} = \{v_k(x)\}$ ,  $\mathcal{W} = \{w_k(x)\}$ ,  $\mathcal{Y} = \{y_k(x)\}$ , for  $k \in \{0, 1, \dots, m\}$ , and a target polynomial  $t(x)$ .

Suppose  $f : \mathbb{F}^n \mapsto \mathbb{F}^{n'}$  having input variables with labels  $1, 2, \dots, n$  and output variables with labels  $n+1, \dots, n+n'$ . We say that  $Q$  computes  $f$  if for  $N = n + n'$ :

$(a_1, a_2, \dots, a_N) \in \mathbb{F}^N$  is a valid assignment of  $f$ 's inputs and outputs, if and only if there exist  $(a_{N+1}, \dots, a_m)$  such that  $t(x)$  divides  $p(x)$  where

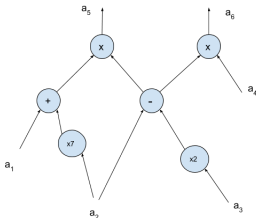
$$p(x) = \left( v_0(x) + \sum_{k=1}^m a_k v_k(x) \right) \cdot \left( w_0(x) + \sum_{k=1}^m a_k w_k(x) \right) - \left( y_0(x) + \sum_{k=1}^m a_k y_k(x) \right).$$

So there must exist polynomial  $h(x)$  such that  $h(x)t(x) = p(x)$ .

The size of  $Q$  is  $m$ , and the degree of  $Q$  is the degree of  $t(x)$ .

- Arithmetic circuits can be mapped to QAPs efficiently

# QAP for an Arithmetic Circuit



- $a_5 = (a_1 + 7a_2)(a_2 - 2a_3)$  and  $a_6 = (a_2 - 2a_3)a_4$
- Choose distinct  $r_5, r_6 \in \mathbb{F}$  and  $t(x) = (x - r_5)(x - r_6)$
- Choose polynomials  $\{v_k(x)\}, \{w_k(x)\}, \{y_k(x)\}, k = 0, 1, \dots, m$  such that

$$\begin{aligned} \sum_{k=0}^6 a_k v_k(r_5) &= a_1 + 7a_2, & \sum_{k=0}^6 a_k w_k(r_5) &= a_2 - 2a_3, & \sum_{k=0}^6 a_k y_k(r_5) &= a_5, \\ \sum_{k=0}^6 a_k v_k(r_6) &= a_2 - 2a_3, & \sum_{k=0}^6 a_k w_k(r_6) &= a_4, & \sum_{k=0}^6 a_k y_k(r_6) &= a_6. \end{aligned}$$

# Pinocchio SNARK from QAP

- Let  $R = \{(u, w)\} \subset \mathbb{F}^n \times \mathbb{F}^{n_1}$  be a relation where  $u \in \mathbb{F}^n$  is the statement and  $w \in \mathbb{F}^{n_1}$  is the witness
- Suppose  $R$  can be verified with an arithmetic circuit, i.e. there is an arithmetic function  $f$  such that  $f(u, w) = 1$  iff  $(u, w) \in R$
- A QAP for  $f$  is derived which has  $N = n + 1$  input-output variables
- Prover has to show he knows  $(a_1, \dots, a_m)$  such that  $t(x)$  divides  $v(x)w(x) - y(x)$  where  $t(x)$  has degree  $d$
- **Common Reference String Generation**
  - Let  $[m] = \{1, 2, \dots, m\}$ . Indices  $\{1, 2, \dots, N\}$  are for IO-related variables while  $\mathcal{I}_{mid} = \{N + 1, \dots, m\}$  are indices of non-IO-related variables
  - Choose  $r_v, r_w, s, \alpha_v, \alpha_w, \alpha_y, \beta, \gamma \xleftarrow{\$} \mathbb{F}^*$  and set  $r_y = r_v r_w$ ,  $g_v = g^{r_v}$ ,  $g_w = g^{r_w}$ , and  $g_y = g^{r_y}$
  - Evaluation key
    - Generate  $\{g_v^{v_k(s)}\}_{k \in \mathcal{I}_{mid}}, \{g_w^{w_k(s)}\}_{k \in \mathcal{I}_{mid}}, \{g_y^{y_k(s)}\}_{k \in \mathcal{I}_{mid}}$
    - Generate  $\{g_v^{\alpha_v v_k(s)}\}_{k \in \mathcal{I}_{mid}}, \{g_w^{\alpha_w w_k(s)}\}_{k \in \mathcal{I}_{mid}}, \{g_y^{\alpha_y y_k(s)}\}_{k \in \mathcal{I}_{mid}}$
    - Generate  $\{g^{s^i}\}_{i \in [d]}, \{g_v^{\beta v_k(s)} g_w^{\beta w_k(s)} g_y^{\beta y_k(s)}\}_{k \in \mathcal{I}_{mid}}$
  - Verification key
    - Generate  $\{g_v^{v_k(s)}\}_{k \in \{0\} \cup [N]}, \{g_w^{w_k(s)}\}_{k \in \{0\} \cup [N]}, \{g_y^{y_k(s)}\}_{k \in \{0\} \cup [N]}$
    - Generate  $g^{\alpha_v}, g^{\alpha_w}, g^{\alpha_y}, g^\gamma, g^{\beta\gamma}, g^{t(s)}$

# Proof Generation for Pinocchio SNARK

- Prover will prove that  $(u, w) \in R$  by showing that  $f(u, w) = 1$
- Prover computes QAP coefficients  $(a_1, \dots, a_m)$  such that

$$h(x)t(x) = (v_0(x) + \sum_{k=1}^m a_k v_k(x)) \cdot (w_0(x) + \sum_{k=1}^m a_k w_k(x)) - (y_0(x) + \sum_{k=1}^m a_k y_k(x)).$$

- For

$$v_{mid}(x) = \sum_{k \in \mathcal{I}_{mid}} a_k v_k(x),$$

$$w_{mid}(x) = \sum_{k \in \mathcal{I}_{mid}} a_k w_k(x),$$

$$y_{mid}(x) = \sum_{k \in \mathcal{I}_{mid}} a_k y_k(x)$$

the prover outputs the proof  $\pi$  as

$$\begin{aligned} &g_v^{v_{mid}(s)}, \quad g_w^{w_{mid}(s)}, \quad g_y^{y_{mid}(s)}, \quad g^h(s), \\ &g_v^{\alpha_v v_{mid}(s)}, \quad g_w^{\alpha_w w_{mid}(s)}, \quad g_y^{\alpha_y y_{mid}(s)} \\ &g_v^{\beta_v v_{mid}(s)} g_w^{\beta_w w_{mid}(s)} g_y^{\beta_y y_{mid}(s)} \end{aligned}$$

- Verifier sees alleged proof as  $g^{V_{mid}}, g^{W_{mid}}, g^{Y_{mid}}, g^H, g^{V'_{mid}}, g^{W'_{mid}}, g^{Y'_{mid}}$ , and  $g^Z$

# Proof Verification for Pinocchio SNARK

- Verification key

- $\{g_v^{v_k(s)}\}_{k \in \{0\} \cup [M]}, \{g_w^{w_k(s)}\}_{k \in \{0\} \cup [M]}, \{g_y^{y_k(s)}\}_{k \in \{0\} \cup [M]}$
  - $g^{\alpha_v}, g^{\alpha_w}, g^{\alpha_y}, g^\gamma, g^{\beta\gamma}, g_y^{t(s)}$

- Verifier computes  $g_v^{v_{io}(s)} = \prod_{k \in [M]} (g_v^{v_k(s)})^{a_k}$  and similarly  $g_w^{w_{io}(s)}, g_y^{y_{io}(s)}$  and checks divisibility

$$e(g_v^{v_0(s)} g_v^{v_{io}(s)} g_v^{V_{mid}}, g_w^{w_0(s)} g_w^{w_{io}(s)} g_w^{W_{mid}}) = e(g_y^{t(s)}, g^H) e(g_y^{y_0(s)} g_y^{y_{io}(s)} g_y^{Y_{mid}}, g)$$

- Verifier checks the  $v_{mid}(s), w_{mid}(s), y_{mid}(s)$  are the correct linear combinations by checking

$$e(g_v^{V'_{mid}}, g) = e(g_v^{V_{mid}}, g^{\alpha_v}), \quad e(g_w^{W'_{mid}}, g) = e(g_w^{W_{mid}}, g^{\alpha_w})$$

$$e(g_y^{Y'_{mid}}, g) = e(g_y^{Y_{mid}}, g^{\alpha_y})$$

- Verifier checks that the same variables  $a_i$  were used in all three linear combinations  $v_{mid}(s), w_{mid}(s), y_{mid}(s)$  by checking

$$e(g^Z, g^\gamma) = e(g_v^{V_{mid}} g_w^{W_{mid}} g_y^{Y_{mid}}, g^{\beta\gamma})$$



# References

- Pairing-Based Cryptographic Protocols : A Survey  
<https://eprint.iacr.org/2004/064.pdf>
- DDH and CDH Problems <https://www.ee.iitb.ac.in/~sarva/courses/EE720/2019/notes/lecture-21.pdf>
- Jens Groth's lecture in the 9th BIU Winter School on Cryptography
  - <https://cyber.biu.ac.il/event/the-9th-biu-winter-school-on-cryptography/>
  - NIZKs from Pairings <https://cyber.biu.ac.il/wp-content/uploads/2019/02/BarIlan2019.pdf>
  - NIZKs from Pairings  
[https://www.youtube.com/watch?v=\\_mAKh7LFPOU](https://www.youtube.com/watch?v=_mAKh7LFPOU)
- Pinocchio: Nearly Practical Verifiable Computation,  
<https://eprint.iacr.org/2013/279.pdf>
- Why and How zk-SNARK Works by Maksym Petkus  
<https://arxiv.org/abs/1906.07221>
- Sections 7, 8 of *Quadratic Span Programs and Succinct NIZKs without PCPs*, GGPR13 <https://eprint.iacr.org/2012/215>