Non-interactive Zero-Knowledge Proofs

Jens Groth
University College London

3rd Bar-Ilan Winter School on Cryptography 2013

Zero-knowledge proof [

Zero-knowledge:
Verifier learns statement is true, but *nothing* else

Witness:
Statement true because…

Statement

OK, statement is true

Prover

Verifier

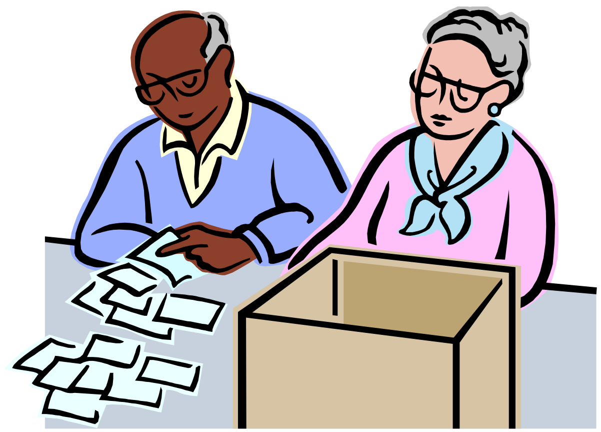
Internet voting

Vote

Encrypts vote to keep it private

Tally without decrypting individual votes

Ciphertext



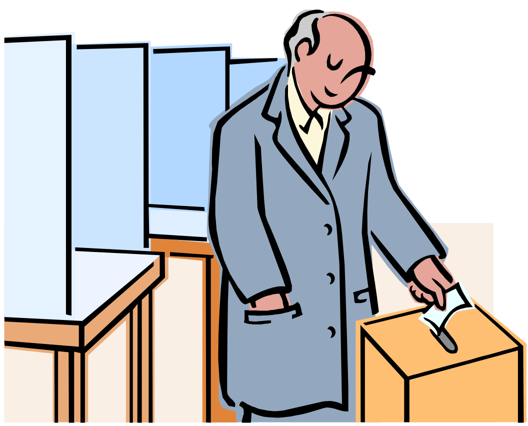
Voters

Election authorities

Election fraud

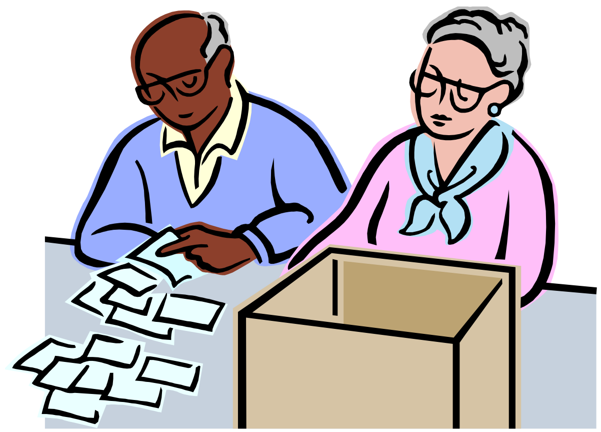
Not Bob

Encrypts -100 votes for Bob



Ciphertext →

Is the encrypted vote valid?



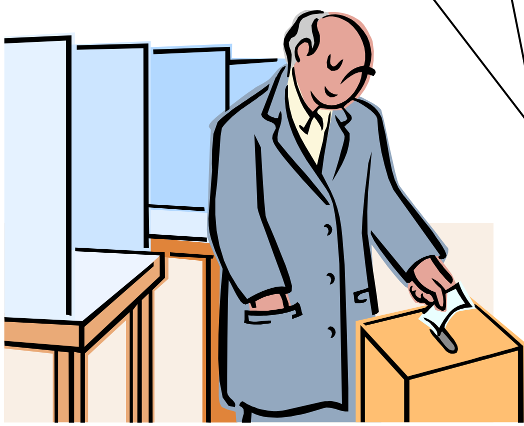
Voters

Election authorities

Zero-knowledge proof as solution

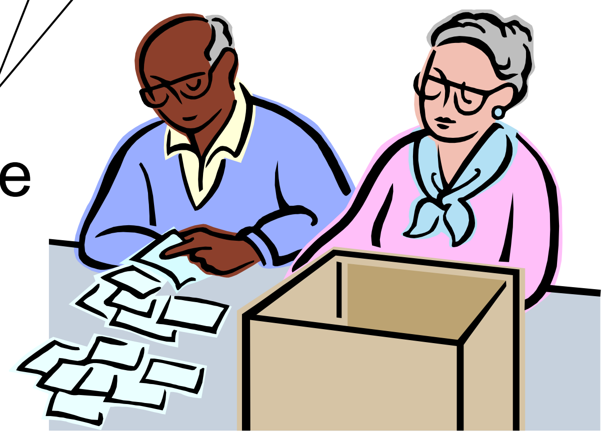
Zero-knowledge:
Vote is secret

Soundness:
Vote is valid



Ciphertext

Zero-knowledge proof for valid vote encrypted

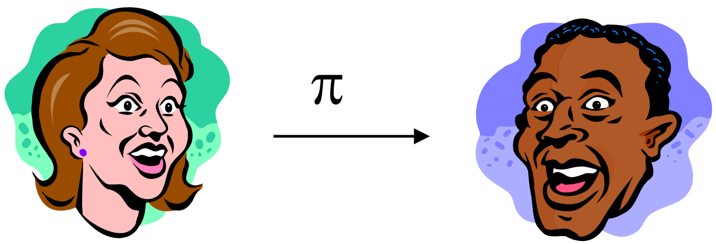


Voters

Election authorities

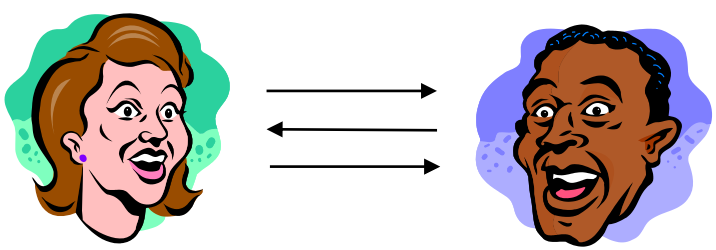
Round complexity

- Non-interactive zero-knowledge proof



Useful for non-interactive tasks
- Signatures
- Encryption
- …

- Interactive zero-knowledge proof



Non-interactive proofs

Witness w
$(x,w) \in R_L$

L language in NP defined by R_L

Statement: $x \in L$

OK, $x \in L$

Proof π $\longrightarrow$

Prover

Verifier

Non-interactive zero-knowledge (NIZK) proofs

- Completeness
  - Can prove a true statement
- Soundness
  - Cannot prove false statement
- Zero-knowledge
  - Proof reveals nothing (except truth of statement)

Zero-knowledge = Simulation

Witness w
$(x,w) \in R_L$

π →

Statement: $x \in L$

Prover

Verifier

NIZK proofs in the plain model only possible for trivial languages $L \in BPP$ [GO94]

Given probabilistic polynomial time algorithms P, V, S for prover, verifier and simulator

Decision algorithm for $x \in L$ or $x \notin L$

Run $S(x) \rightarrow \pi$

Return $V(\pi)$

If $x \notin L$: Soundness implies verifier algorithm rejects

If $x \in L$: Zero-knowledge; simulation looks like real proof

Completeness then means verifier accepts

Non-interactive zero-knowledge proof [BFM88]

Common reference string
0100…11010

$(x,w) \in R_L$

Statement: $x \in L$

Proof: π

$\longrightarrow$

Prover

Verifier

Common reference string (CRS)

01101101010001011110100101

- Can be uniform random or specific distribution
  - Key generation algorithm K for generating CRS
- Trusted generation
  - Trusted party
  - Secure multi-party computation
  - Multi-string model with majority of strings honest [GO07]

Zero-knowledge simulation

Simulation trapdoor

Common reference string
0100…11010 $\leftarrow S \rightarrow \tau$

$(x,w) \in R_L$

Statement: $x \in L$

$S(\tau,x) \rightarrow \pi$

Prover

Verifier

Publicly verifiable NIZK proofs

- NP language L
  - Statement $x \in L$ if there is witness w so that $(x,w) \in R_L$
- An NIZK proof system for R_L consists of three probabilistic polynomial time algorithms (K,P,V)
  - $K(1^k)$: Generates common reference string σ
  - $P(\sigma,x,w)$: Generates a proof π
  - $V(\sigma,x,\pi)$: Outputs 1 (accept) or 0 (reject)

Public vs. private verification

Anybody can check the proof

- Publicly verifiable
  - K generates CRS σ
  - V checks proof given input (σ,x,π)

Designated verifier with ω can check proof

- Privately verifiable
  - K generates CRS σ and pri[illegible]e verification key ω
  - V checks proof given input (ω,x,π)

Public vs. private verifiability

Public verifiability

- Sometimes required
  - Signatures
  - Universally verifiable voting
- Reusability
  - Proof can be copied and sent to somebody else
  - Prover only needs to run once to create proof π that convinces everybody
- Hard to construct

Private verifiability

- Sometimes suffices
  - CCA-secure public-key encryption, e.g., Cramer-Shoup encryption
- Cannot be transferred
  - For designated verifier only
- Easier to construct

Completeness

Common reference string $\sigma \leftarrow K(1^k)$

Statement $x \in L$

Witness w
so $(x,w) \in R$

$P(\sigma,x,w) \rightarrow \pi$

$V(\sigma,x,\pi) \rightarrow$
Accept/reject

Perfect completeness: Pr[Accept] = 1

Soundness

Common reference string σ $\longleftarrow$ $K(1^k)$

$\longrightarrow$ Statement $x \notin L$

Adaptive soundness:
The adversary first sees
CRS and then cheats

π

$V(\sigma,x,\pi) \rightarrow$
Accept/reject

Perfect soundness: ∀ Adv: Pr[Reject] = 1
Statistical soundness: ∀ Adv: Pr[Reject] ≈ 1
Computational soundness: ∀ poly-time Adv: Pr[Reject] ≈ 1

Proofs vs. arguments

- Proof
  - Perfect or statistical soundness
  - No unbounded adversary can prove a false statement
- Argument
  - Computational soundness
  - No probabilistic polynomial time adversary can prove a false statement

Proof of knowledge [DP92]

Common reference string σ $\longleftarrow$ $E(1^k) \rightarrow \xi$

$\longrightarrow$ Statement x

π

$V(\sigma,x,\pi) \rightarrow$ Accept/reject

Extractor E:
$E(\xi,x,\pi) \rightarrow w$

Perfect proof of knowledge: $\forall$ Adv: $\Pr[(x,w)\in R_L \mid \text{accept}] = 1$

Statistical PoK: $\forall$ Adv: $\Pr[(x,w)\in R_L \mid \text{accept}] \approx 1$

Comp. PoK: $\forall$ poly-time Adv: $\Pr[(x,w)\in R_L \mid \text{accept}] \approx 1$

Zero-knowledge

$K(1^k) \rightarrow \sigma$

$(x,w) \in R_L$

$P(\sigma,x,w) \rightarrow \pi$

$\rightarrow$ 0/1

Multi-theorem ZK [FLS99]
The adversary can get many real/simulated proofs

$\tau \leftarrow S_1(1^k) \rightarrow$ σ

$(x,w) \in R_L$

$S_2(\sigma,\tau,x) \rightarrow$ π

$\rightarrow$ 0/1

Perfect ZK: $\Pr[\text{Adv} \rightarrow 1 | \text{Real}] = \Pr[\text{Adv} \rightarrow 1 | \text{Simulation}]$

Computational ZK:

$\forall$ poly-time Adv: $\Pr[\text{Adv} \rightarrow 1 | \text{Real}] \approx \Pr[\text{Adv} \rightarrow 1 | \text{Simulation}]$

Witness indistinguishability [FS90]

Common reference string $\sigma \longleftarrow K(1^k)$

Statement $x \in L$

Witnesses w_0, w_1

$(x,w_0),(x,w_1) \in R_L$

$\longleftarrow$

$b \leftarrow \{0,1\}$

$P(\sigma,x,w_b) \rightarrow \pi$

$\longrightarrow$

$\longrightarrow$ Guess $\in \{0,1\}$

Perfect witness-indistinguish.: $\forall$ Adv: Pr[Guess = b] = ½

Computational WI: $\forall$ poly-time Adv: Pr[Guess = b] $\approx$ ½

Witness-indistinguishability vs. zero-knowledge

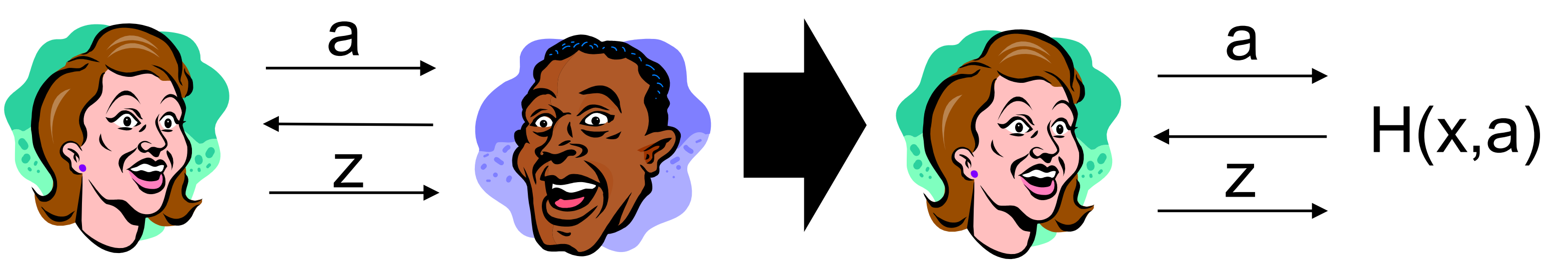
- Zero-knowledge implies witness-indistinguishability
  - Reveals nothing, in particular not which witness used
- Witness-indistinguishability weaker than ZK
  - Suppose all witnesses for the same statement in L have the same prefix, then a WI proof may reveal that prefix
    - w_0 = 100100101 11011
    - w_1 = 100100101 00100

WI proof may reveal 100100101

  - If each statement has only one witness, then the WI proof may reveal the entire witness
    - Statement: (u,v) ElGamal encryption of 1, i.e., $(u,v) = (g^r,h^r)$
    - Witness-indistinguishable proof: r

Fiat-Shamir heuristic [FS86]

- Take an interactive ZK argument where verifier's messages are random bits (public coin argument)
- Let the CRS describe a hash-function H
- Replace the verifier's messages with hash-values from the current transcript



- NIZK argument $\pi = (a,z)$

Fiat-Shamir heuristic

- Efficient NIZK arguments that work well in practice
- Hopefully they are secure
  - Can argue heuristically that they are computationally sound in the random oracle model [BR93], where we pretend H is a truly random function
  - But in real life H is a deterministic function and there are instantiations of the Fiat-Shamir heuristic [GK03] that yields insecure real-life schemes

Encrypted random bits [BFM88]

Statement: $x \in L$

$(x,w) \in R_L$

$K(1^k) \rightarrow (pk,sk)$

CRS

$E_{pk}(0;r_1)$

$E_{pk}(1;r_2)$

$E_{pk}(0;r_3)$

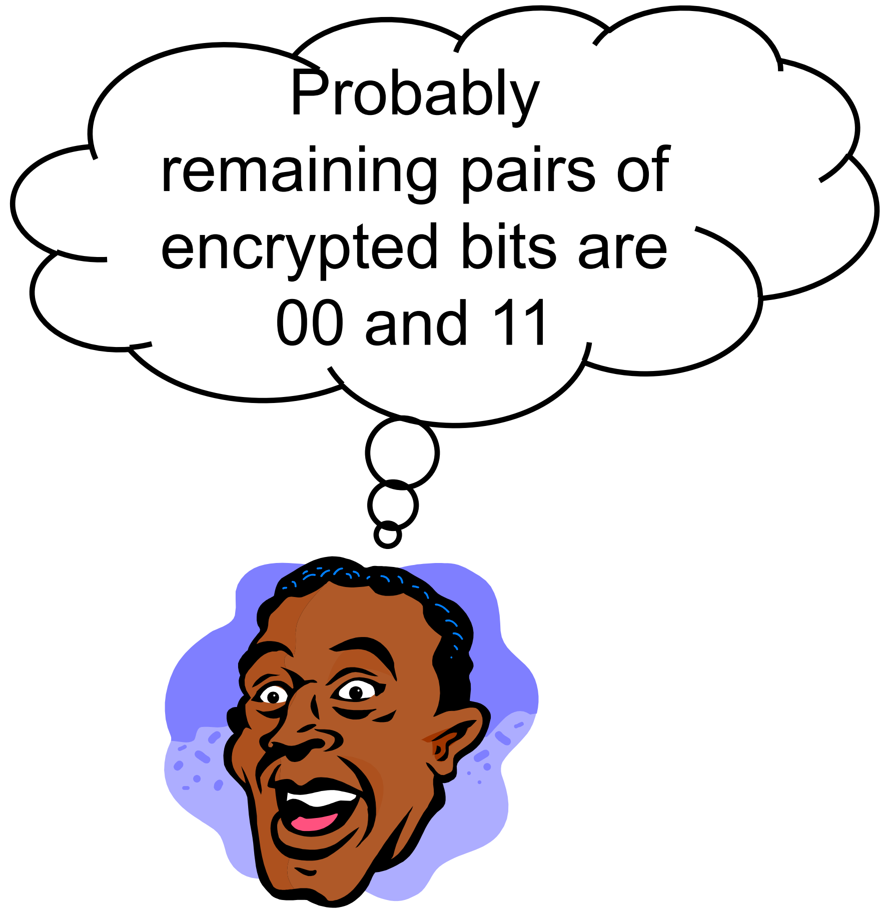
$E_{pk}(1;r_4)$

Statistical sampling

- Random bits not useful
- Use statistical sampling to get hidden bits with structure

CRS

1	1
1	0
0	0
0	1



Reveal certain bits from structures

Reveal: ?0 ⊕ 1? ⊕ ?1 = 0

10 ∨ 11 ∨ 11

- Kilian-Petrank for in formulas

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4 \vee \neg x_5) \wedge (x_1 \ldots) \wedge \ldots$$
$$\ldots \wedge (x_1 \ldots) \wedge (\neg x_1 \ldots)$$

- They give method to assign hidden pairs of bits to each literal in a consistent manner such that
  - If literal is true the pair is 01 or 10, if literal is false the pair is 00 or 11
  - Pairs for literals corresponding to different appearances of same variable are consistent with each other
- With satisfying assignment possible to XOR all clauses to 0
- With an unsatisfied clause 50% chance bits do not XOR to 0

NIZK proofs for Circuit SAT

- Security level: 2^{-k}
- Trapdoor perm size: $k_T = \text{poly}(k)$
- Group element size: $k_G \approx k^3$
- Circuit size: $|C| = \text{poly}(k)$
- Witness size: $|w| \leq |C|$

	CRS in bits	Proof in bits	Assumption
G-Ostrovsky-Sahai 12	$O(k_G)$	$O(\|C\| \cdot k_G)$	Pairing-based
Groth 10	$\|C\| \cdot k_T \cdot \text{polylog}(k)$	$\|C\| \cdot k_T \cdot \text{polylog}(k)$	Trapdoor perms
Groth 10	$\|C\| \cdot \text{polylog}(k)$	$\|C\| \cdot \text{polylog}(k)$	Naccache-Stern
Gentry 09	$\text{poly}(k)$	$\|w\| \cdot \text{poly}(k)$	FHE + NIZK

Practice

Statement: Here is a ciphertext and a document. The ciphertext contains a digital signature on the document.

	Circuit SAT	Practical statements
	1 GB	
Inefficient	Damgård 92 Kilian-Petrank 98	**1 KB**
Efficient	Groth-Ostrovsky-Sahai 12	Groth-Sahai 12

Non-interactive Zero-Knowledge Proofs from Pairings

Jens Groth

University College London

3rd Bar-Ilan Winter School on Cryptography 2013

Groth-Ostrovsky-Sahai 2012 (2006)

- NIZK proof for Circuit SAT
- Perfect completeness, perfect soundness, computational zero-knowledge
- Common reference string: O(1) group elements
- Proofs: O(|C|) group elements

Composite order bilinear group

- $\mathrm{Gen}(1^k)$ generates (p, q, G, G_T, e, g)
- G, G_T finite cyclic groups of order $n = pq$
- Pairing $e: G \times G \to G_T$
  - $e(g^a, g^b) = e(g, g)^{ab}$
  - $G = \langle g \rangle, G_T = \langle e(g, g) \rangle$
- Deciding group membership, group operations, and bilinear pairing efficiently computable
- Subgroup decision assumption
  - Given (n, G, G_T, e, g, h) hard to distinguish whether h has order q or h has order n

BGN encryption [Boneh-Goh-Nissim 05]

Public key:	(n, G, G_T, e, g, h)	h has order q
Secret key:	p, q	$n = pq$
Encryption:	$c = g^a h^r$	$r \leftarrow \mathbf{Z}_n$
Decryption:	$c^q = (g^a h^r)^q = g^{qa} h^{qr} = (g^q)^a$ Compute discrete logarithm if a small	

BGN encryption is IND-CPA secure if the subgroup decision assumption holds

Sketch of proof

By subgroup decision assumption public key looks the same as if h had order n. But if h had order n, ciphertext would have no information about the plaintext a.

Commitment

Public key: (n, G, G_T, e, g, h) h has order q

Commitment: $c = g^a h^r$ $r \leftarrow \boldsymbol{Z}_n$

Perfectly binding: Unique $a \; mod \; p$

Computationally hiding: Indistinguishable from h order n

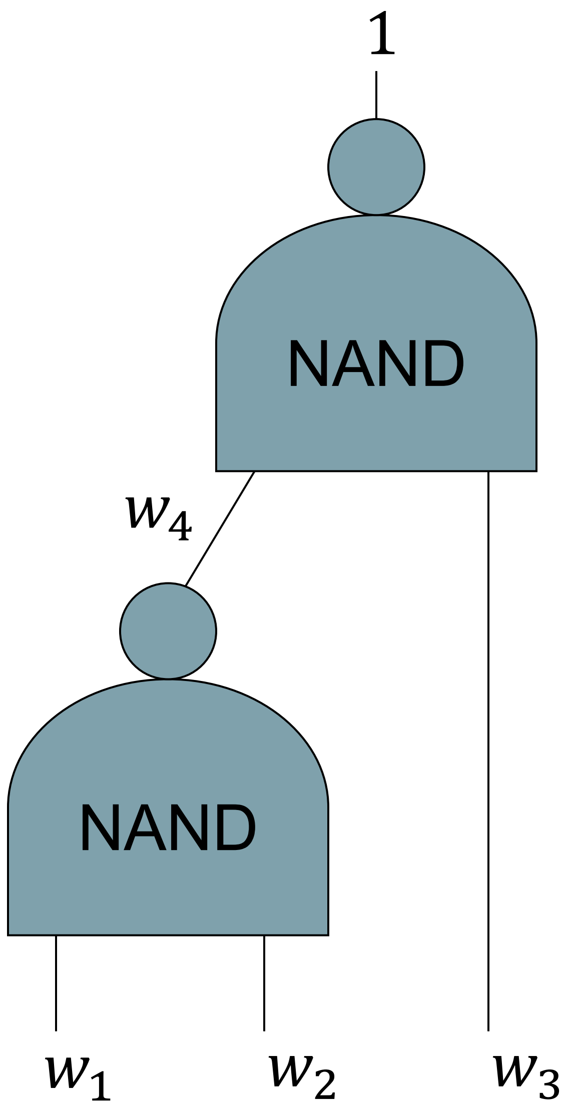
Addition: $(g^a h^r)(g^b h^s) = g^{a+b} h^{r+s}$

Multiplication: $e(g^a h^r, g^b h^s)$

$$= e(g^a, g^b) e(h^r, g^b) e(g^a, h^s) e(h^r, h^s)$$

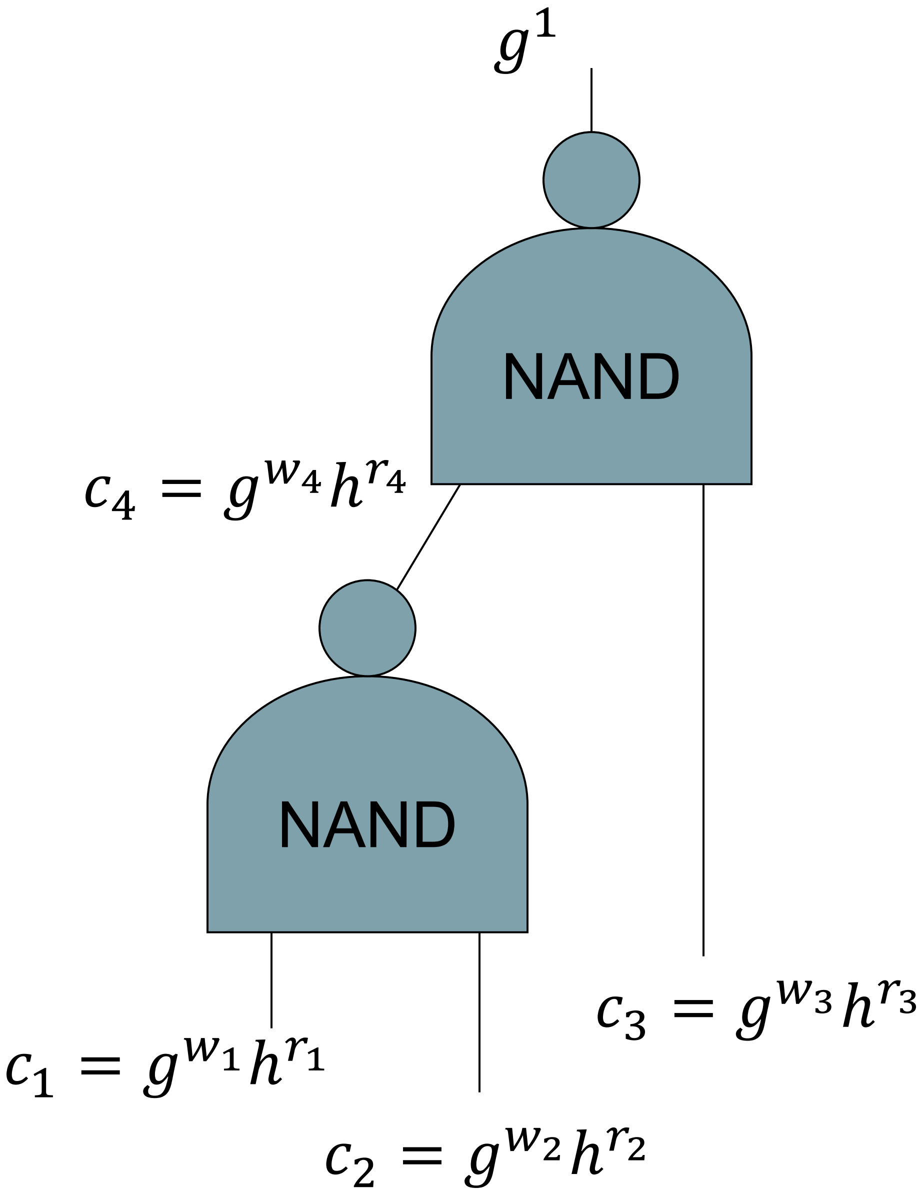
$$= e(g, g)^{ab} e(h, g^{as+rb} h^{rs})$$

NIZK proof for Circuit SAT



Circuit SAT is NP complete

NIZK proof for Circuit SAT



Prove $w_1 \in \{0,1\}$

Prove $w_2 \in \{0,1\}$

Prove $w_3 \in \{0,1\}$

Prove $w_4 \in \{0,1\}$

Prove
$w_4 = \neg(w_1 \wedge w_2)$

Prove
$1 = \neg(w_3 \wedge w_4)$

Proof for c containing 0 or 1

Write $c = g^w h^r$ (unique $w \; mod \; p$ since h has order q)

Recall $e(c, cg^{-1}) = e(g,g)^{w(w-1)} e(h, g^{(2w-1)r} h^{r^2})$

Proof $\pi = g^{(2w-1)r} h^{r^2}$

Verifier checks: $e(c, cg^{-1}) = e(h, \pi)$

$$\rightarrow \; e(g,g)^{w(w-1)} e\left(h, g^{(2w-1)r} h^{r^2}\right) = e(h, \pi)$$

$\rightarrow \; w = 0 \; mod \; p$ or $w = 1 \; mod \; p$

Observation

b_0	b_1	b_2	$b_0+b_1+2b_2-2$
0	0	0	-2
0	0	1	0
0	1	0	-1
0	1	1	1
1	0	0	-1
1	0	1	1
1	1	0	0
1	1	1	2

$b_2 = \neg(b_0 \wedge b_1)$

if and only if

$b_0 + b_1 + 2b_2 - 2 \in \{0,1\}$

Proof for NAND-gate

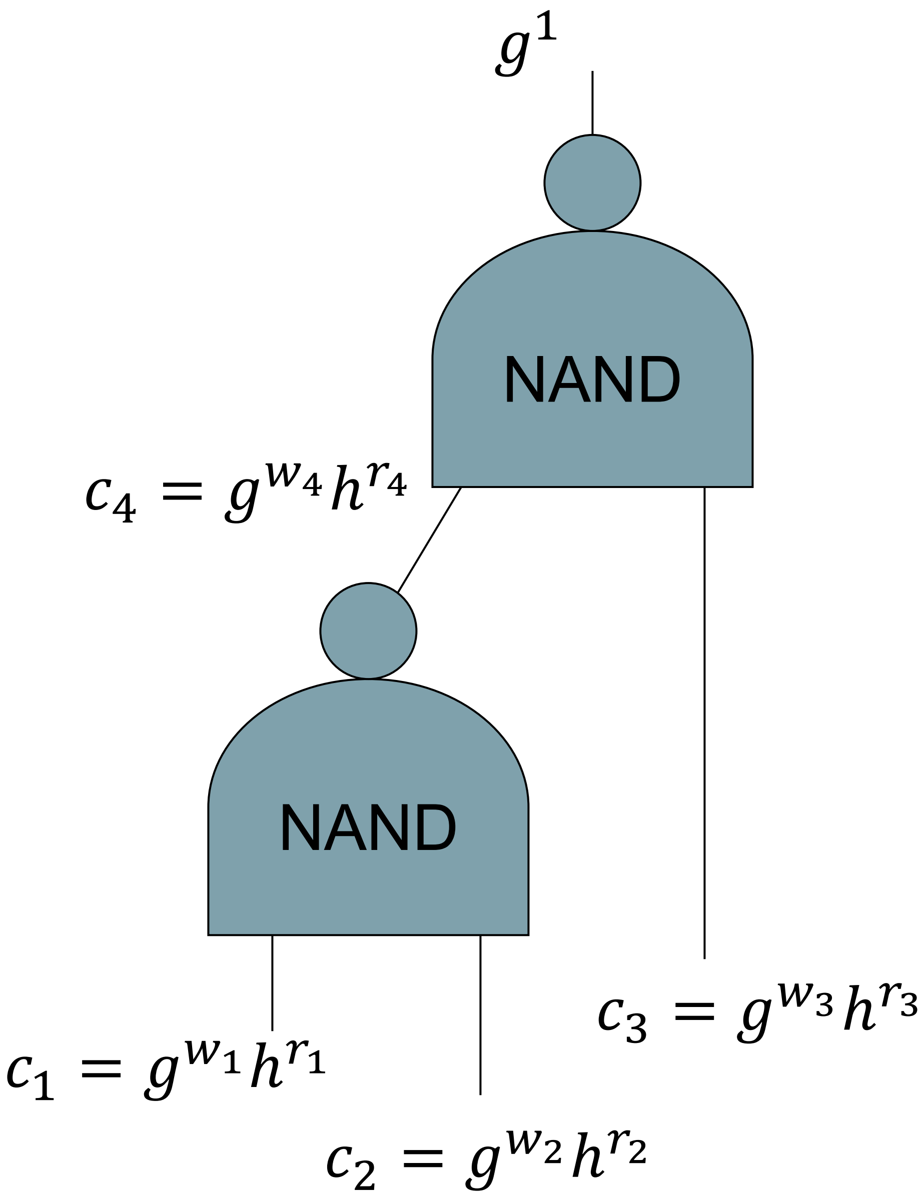
Given c_0, c_1, c_2 containing bits b_0, b_1, b_2
wish to prove $b_2 = \neg(b_0 \wedge b_1)$

$$b_2 = \neg(b_0 \wedge b_1) \quad \text{if} \quad b_0 + b_1 + 2b_2 - 2 \in \{0,1\}$$

$$c_0 c_1 c_2^2 g^{-2} = g^{b_0+b_1+2b_2-2} h^{r_0+r_1+2r_2}$$

Prove $c_0 c_1 c_2^2 g^{-2}$ contains 0 or 1

NIZK proof for Circuit SAT



Prove $w_1 \in \{0,1\}$

Prove $w_2 \in \{0,1\}$

Prove $w_3 \in \{0,1\}$

Prove $w_4 \in \{0,1\}$

Prove
$$w_4 = \neg(w_1 \wedge w_2)$$

Prove
$$1 = \neg(w_3 \wedge w_4)$$

CRS (n, G, G_T, e, g, h)
CRS size $3k_G$
Proof size $(2|w| + |C|)k_G$

Zero-Knowledge

Subgroup decision assumption

Hard to distinguish whether h has order q or n

Simulated common reference string

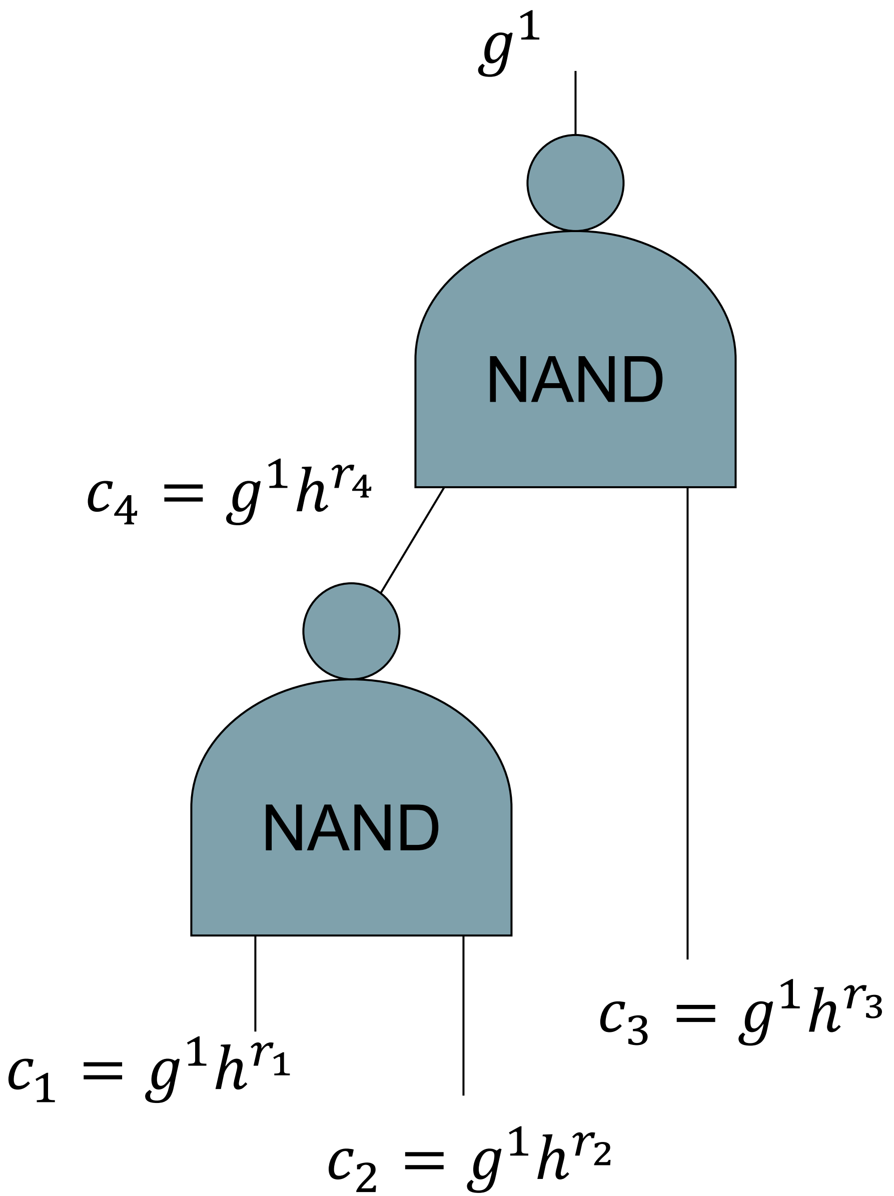
h order n by choosing $g = h^{\tau}$ $\tau \leftarrow \boldsymbol{Z}_n^*$

The simulation trapdoor is τ

Commitments are now perfectly hiding trapdoor commitments

$$g^1 h^r = g^0 h^{r+\tau}$$

Simulation



Prove $w_1 \in \{0,1\}$

Prove $w_2 \in \{0,1\}$

Prove $w_3 \in \{0,1\}$

Prove $w_4 \in \{0,1\}$

Prove

$$w_4 = \neg(w_1 \wedge w_2)$$

Prove

$$1 = \neg(w_3 \wedge w_4)$$

Using $w_2 = 0, w_3 = 0$ for the NAND proofs

Witness-indistinguishable 0/1-proof

Write $c = g^1 h^r$ or $c = g^0 h^{r+\tau}$

$$e(c, cg^{-1}) = e(h, g^r h^{r^2}) \text{ or } e(c, cg^{-1}) = e(h, g^{-(r+\tau)} h^{(r+\tau)^2})$$

Proof $\pi = g^r h^{r^2}$ or $\pi = g^{-(r+\tau)} h^{(r+\tau)^2}$

Verifier checks $e(c, cg^{-1}) = e(h, \pi)$

Perfect witness-indistinguishable when h has order n since there is unique π satisfying equation, no matter whether c contains 0 or 1

Zero-knowledge of full Circuit SAT proof

Sketch of proof:

Pr[Adv→1|Real proof]

≈ Pr[Adv→1|Real proof on h with order n]

= Pr[Adv→1|Hybrid proof where h has order n and commitments to 1. The simulator uses trapdoor to open them to real witness and gives real proofs]

= Pr[Adv→1|Hybrid proof where h has order n and commitments to 1. The simulator uses trapdoor to open some commitments to 0 in NAND proofs]

= Pr[Adv→1|Simulated proof]

Composable zero-knowledge

- Real common reference string computationally indistinguishable from simulated common reference string
- Real proof on simulated common reference string perfectly indistinguishable from simulated proof on simulated common reference string

NIZK proof for Circuit SAT

- Commit to all wires w_i as $c_i = g^{w_i} h^{r_i}$
- For each i prove c_i contains 0 or 1
- For each NAND prove $c_0 c_1 c_2^2 g^{-2}$ contains 0 or 1
- Total size: $2|w| + |C|$ group elements

- Perfect completeness, perfect soundness, composable zero-knowledge
- Also, perfect proof of knowledge

$$c_i^q = (g^{w_i} h^{r_i})^q = (g^q)^{w_i}$$

Known for all of NP?	Computational zero-knowledge	Perfect zero-knowledge (everlasting privacy)
Interactive proof	Yes [Goldreich-Micali-Wigderson 1986]	Yes [Brassard-Crepeau 1986]
Non-interactive proof	Yes [Blum-Feldman-Micali 1988]	Yes [Groth-Ostrovsky-Sahai 2012]

Perfect zero-knowledge

- Instead of h with order q, use h with order n

- Easy to verify that we have perfect completeness
- As argued earlier we have perfect zero-knowledge
- What about soundness?

”Natural” computational soundness fails

- Start with h of order n and Adversary that produces a false statement and a valid proof
- Switch to h of order q, which Adversary cannot distinguish from order n. Therefore Adversary still produces a statement and a valid proof
- We now have non-adaptive soundness, when statement is independent of CRS. Otherwise a false statement has been proven with h of order q
- But there is a problem with adaptive soundness
  - Consider the statement "h has order q"

Adaptive culpable soundness

Common reference string $\leftarrow K(1^k)$

C, w_{guilt}

Proof π

w_{guilt} witness for C unsatisfiable

Comp. culpable soundness: $\forall$ poly-time Adv: Pr[Reject] ≈ 1

Computational culpable soundness

Sketch of proof:

- Imagine poly-time Adversary could break culpable soundness; after seeing CRS where h has order n, Adversary makes valid $(C, w_{\text{guilt}}, \pi)$.
- By subgroup decision assumption approximately same success probability for Adversary producing valid $(C, w_{\text{guilt}}, \pi)$ when h has order q.
- But w_{guilt} guarantees C is unsatisfiable and when h has order q the perfect soundness guarantees C is satisfiable.

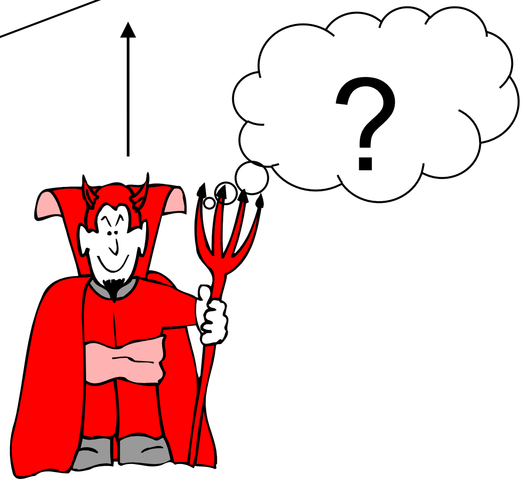
Culpable soundness the ”right” definition

- Abe-Fehr 07 show that impossible to achieve perfect zero-knowledge and the ”natural” adaptive soundness definition with standard direct black-box methods
- Often a non-satisfiability witness exists
  - Consider for instance verifiable encryption; here the secret key is a witness for the plaintext not being m
- Computational culpable soundness sufficient for constructing universally composable NIZK proofs

Groth-Ostrovsky-Sahai 12

$\sigma = (n, G, G_T, e, g, h)$
where $ord(h) = q$

- NIZK proof for Circuit SAT
- Perfect binding key
  – Perfect completeness
  – Perfect soundness
  – Computational zero-knowledge
- Perfect hiding key
  – Perfect completeness
  – Culpable soundness
  – Perfect zero-knowledge



$\sigma = (n, G, G_T, e, g, h)$
where $ord(h) = n$

Non-interactive Zero-Knowledge Proofs from Pairings

Jens Groth

University College London

3rd Bar-Ilan Winter School on Cryptography 2013

NIZK proof efficiency

	Circuit SAT	Practical statements
Inefficient	Hidden bits	Groth 06
Efficient	Groth-Ostrovsky-Sahai 12	Groth-Sahai 12 Coming next

Our goal

- We want high efficiency. Practical non-interactive proofs!
- We want non-interactive proofs for statements arising in practice such as ”the ciphertext c contains a signature on m”. No NP-reduction!

Example: Boyen-Waters 07 group signatures

- Statement

  $\mu_1, \ldots, \mu_m \in \boldsymbol{Z}_n, \Omega, g, u, v', v_1, \ldots, v_m \in G \,, A \in G_T$
- Prover knows witness $\theta_1, \theta_2, \theta_3, \theta_4 \in G$

$$e(\theta_1, \theta_2\Omega) = A \qquad e(\theta_2, u) = e(\theta_3, g)e(\theta_4, v' \prod_i v_i^{\mu_i})$$

- The group signature on $M = (\mu_1, \ldots, \mu_m)$ is a six element proof of knowledge $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2)$

* Boyen-Waters 07 NIZK proof independent of our work

Constructions in bilinear groups

$$a, c \in G \quad b, d \in \boldsymbol{Z}_n$$

$$t = b + yd \bmod n$$

$$t_G = x^y a^y c^t$$

$$t_T = e(t_G, c t_G^b)$$

Non-interactive cryptographic proofs for correctness of constructions

Yes, here is a proof.

Are the constructions correct? I do not know your secret x, y.

$$t = b + yd \bmod n$$

$$t_G = x^y a^y c^t$$

$$t_T = e(t_G, ct_G^b)$$

Proof

Commitment to group elements

- Common reference string (n, G, G_T, e, g, h)
  - Real CRS: h has order q
  - Simulation CRS: $g = h^\tau$ with $\tau \in \boldsymbol{Z}_n^*$
- Commitment to group element $x \in G$

$$c = xh^r \qquad\qquad r \leftarrow \boldsymbol{Z}_n$$

- Real CRS: Perfect binding in order p subgroups
  - Let $\lambda = 1 \; mod \; p, \lambda = 0 \; mod \; q$ then $c^\lambda = x^\lambda h^{\lambda r} = x^\lambda$ determines x^λ
- Simulation CRS: Perfect hiding commitments
  - When h has order n the commitment is a random group element

Homomorphic properties

- Commitments are homomorphic
  - $(xh^r)(yh^s) = xyh^{r+s}$
  - $(g^x h^r)(g^y h^s) = g^{x+y} h^{r+s}$
- Pairing commitments
  - $e(xh^r, yh^s) = e(x, y)e(h, x^s y^r h^{rs})$
  - $e(xh^r, g^y h^s) = e(g, x^y)e(h, x^s g^{yr} h^{rs})$
  - $e(g^x h^r, g^y h^s) = e(g, g)^{xy} e(h, g^{xs+yr} h^{rs})$

NIWI proof example

- Consider an equation

$$e(a,y)e(x,y) = t_T$$

- Commitments to variables

$$c = xh^r, d = yh^s$$

- Proof that committed values satisfy the equation

$$\pi = a^s x^s y^r h^{rs}$$

- Verify proof π by checking

$$e(a,d)e(c,d) = t_T e(h,\pi)$$

- Completeness

$$e(a, yh^s)e(xh^r, yh^s)$$
$$= e(a,y)e(x,y)\, e(h, a^s x^s y^r h^{rs})$$

NIWI proof example

- Consider an equation

$$e(a, y)e(x, y) = t_T$$

- Verify proof π by checking

$$e(a, d)e(c, d) = t_T e(h, \pi)$$

- Soundness when $ord(h) = q$
  - Let $\lambda = 1 \; mod \; p, \lambda = 0 \; mod \; q$ and raise to $\lambda = \lambda^2 \; mod \; n$ on both sides of verification equation
  $$e(a^\lambda, d^\lambda)e(c^\lambda, d^\lambda) = t_T^{\lambda^2} e(h^\lambda, \pi^\lambda) = t_T{}^\lambda$$
  - We see $x = c^\lambda, y = d^\lambda$ satisfy the equation in the order p subgroups of G, G_T

NIWI proof example

- Consider an equation

$$e(a, y)e(x, y) \ = t_T$$

- Verify proof π by checking

$$e(a, d)e(c, d) \ = t_T e(h, \pi)$$

- Witness-indistinguishability when $ord(h) = n$
  - The commitments are perfectly hiding, so there are many different possible openings x, r, y, s of c, d satisfying the equation
  - However, since $ord(h) = n$ there is a unique proof π satisfying the verification equation
  - Two openings x_0, r_0, y_0, s_0 and x_1, r_1, y_1, s_1 of c, d that satisfy the original equation therefore give the same π

Full NIWI proof for a set of equations

- Suppose we have equations $eq_1, eq_2, \ldots$ of the form

$$\prod_i e(a_i, x_i) \prod_{i,j} e(x_i, x_j)^{\gamma_{ij}} = t_T$$

- We can give a NIWI proof that there are values

$$x_1, \ldots, x_m \in G$$

  satisfying all the equations simultaneously
  – Commit to each variable x_i
  – Make a NIWI proof for each equation eq_k
- Commitments and proofs cost 1 group element each

Together with commitments to exponents in $\boldsymbol{Z}_n$ we get NIWI proof for simultaneous satisfiability a set of equations $eq_1, eq_2, \ldots$ that can be a mix of

– Pairing product equations

$$\prod_i e(a_i, x_i) \prod_{i,j} e(x_i, x_j)^{\gamma_{ij}} = t_T$$

– Multi-exponentiation equations

$$\prod_j a_j{}^{y_j} \prod_i x_i^{b_i} \prod_{i,j} x_i{}^{\gamma_{ij} y_j} = t_G$$

– Quadratic equations

$$\sum_j b_j y_j + \sum_{i,j} \gamma_{ij} y_i y_j = t \bmod n$$

Properties of the NIWI proof

- Two types of common reference string
  - Real CRS: h has order q
  - WI CRS: h has order n
  - Real and WI reference strings computationally indistinguishable
- Perfect completeness on both types of strings
- Real CRS: Perfect soundness in order p subgroups
  - Commitments perfectly binding and equation proofs perfectly sound
- WI CRS: Perfect witness-indistinguishability
  - Commitments perfectly hiding so can contain any valid witness
  - The equation proofs are perfectly witness-indistinguishable, so do not reveal anything about the witness inside the commitments

What makes the NIWI proof work?

$$\begin{tikzcd}
G \arrow[d] & \times & G \arrow[d] \arrow[r] & G_T \arrow[d] \\
G \arrow[d] & \times & G \arrow[d] \arrow[r] & G_T \arrow[d] \\
G_p & \times & G_p \arrow[r] & G_{T,p}
\end{tikzcd}
\qquad
\begin{tikzcd}
(x \arrow[d] & , & y) \arrow[d] \arrow[r] & t_T \arrow[d] \\
(xh^r \arrow[d] & , & yh^s) \arrow[d] \arrow[r] & t_T e(h,\pi) \arrow[d] \\
(x^\lambda & , & y^\lambda) \arrow[r] & t_T^\lambda
\end{tikzcd}$$

- Commuting linear and bilinear map
- We will generalize this methodology
  - Groups can have prime or composite order
  - Pairing $e: G_1 \times G_2 \to G_T$ with $G_1 \neq G_2$ or $G_1 = G_2$
  - Many different assumptions: Subgroup decision, SXDH (i.e., DDH in both groups), decision linear, etc.

Modules

- An abelian group $(A, +, 0)$ is a $\boldsymbol{Z}_p$-module if $\boldsymbol{Z}_p$ acts on A such that for all $r, s \in \boldsymbol{Z}_p, a, b \in A$
  - $1a = a$
  - $(r + s)a = ra + sa$
  - $r(a + b) = ra + rb$
  - $r(sa) = (rs)a$
- If p is a prime then A is a vector space
- Examples
  - $\boldsymbol{Z}_p, G_1, G_2, G_T, G_1^2, G_2^2, G_T^4$ are $\boldsymbol{Z}_p$-modules

Modules with bilinear map

- We will be interested in finite $\boldsymbol{Z}_p$-modules A_1, A_2, A_T with a bilinear map $\cdot_A: A_1 \times A_2 \to A_T$
- Examples:
  - $pair: G_1 \times G_2 \to G_T \qquad (x, y) \mapsto e(x, y)$
  - $exp: G_1 \times \boldsymbol{Z}_p \to G_1 \qquad (x, y) \mapsto x^y$
  - $exp: \boldsymbol{Z}_p \times G_2 \to G_2 \qquad (x, y) \mapsto y^x$
  - $mult: \boldsymbol{Z}_p \times \boldsymbol{Z}_p \to \boldsymbol{Z}_p \qquad (x, y) \mapsto xy \bmod p$

Statements we want to prove

- Statements consisting of quadratic equations $eq_1, \ldots, eq_N$ in A_1, A_2, A_T of the form
$$\sum_j a_j \cdot y_j + \sum_i x_i \cdot b_i + \sum_{ij} x_i \cdot \gamma_{ij} y_j = t$$
- The prover knows secret witness
$$\vec{x} = (x_1, \ldots, x_m) \quad \vec{y} = (y_1, \ldots, y_n)$$
that satisfies all equations $eq_1, \ldots, eq_N$
- Simplify notation using vectors and matrices
$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$$

Commitments in modules

Easy to compute

- Linear maps and modules

Hard to compute

$$A \xrightarrow{i} B \xrightarrow{p} C$$

- Elements $u_1, u_2, \dots, u_m \in B$
- Commit to an element $x \in A$

$$c = i(x) + \sum_i r_i u_i$$

- Perfectly hiding x if $i(A) \subseteq \langle u_1, \dots, u_m \rangle$
- Perfectly binding to $p(c)$
  - For soundness, we want $p(u_i) = 0$

Example

- Linear maps and modules

$$G \xrightarrow{i} G^2 \xrightarrow{p} G \qquad \begin{array}{l} i: x \to (1, x) \\ p: (a, b) \to ba^{-\alpha} \end{array}$$

- Elements $u_1 = (g, g^\alpha), u_2 = (h, h^{\alpha+\tau})$
  - If the DDH problem is hard in G cannot distinguish whether $\tau = 0$ or $\tau \neq 0$
- Commitment to $x \in G$

$$c = (g^{r_1} h^{r_2}, x(g^{r_1} h^{r_2})^\alpha h^{\tau r_2})$$

  - If $\tau \neq 0$ this is a perfectly hiding commitment
  - If $\tau = 0$ the commitment is an ElGamal encryption of x and p is the ElGamal decryption algorithm
    - Note $p(u_1) = p(u_2) = 1$ and $p\big(i(x)\big) = x$

Commuting linear and bilinear maps

- CRS defines $\boldsymbol{Z}_p$-modules $A_1, A_2, A_T, B_1, B_2, B_T$, C_1, C_2, C_T and (bi)linear maps $i_1, i_2, i_T, p_1, p_2, p_T, \cdot_A, \cdot_B, \cdot_C$

$$\begin{array}{ccccc}
A_1 & \times & A_2 & \xrightarrow{\cdot_A} & A_T \\
i_1 \downarrow & & i_2 \downarrow & & \downarrow i_T \\
B_1 & \times & B_2 & \xrightarrow{\cdot_B} & B_T \\
p_1 \downarrow & & p_2 \downarrow & & \downarrow p_T \\
C_1 & \times & C_2 & \xrightarrow{\cdot_C} & C_T
\end{array}$$

- Prover's witness is in A_1, A_2
- Will commit and make proofs in B_1, B_2
- Soundness will hold in C_1, C_2, C_T

Example

$$\begin{tikzcd}
G_1 \arrow[d, swap, "i_1"] & \times & G_2 \arrow[d, swap, "i_2"] \arrow[r, "e"] & G_T \arrow[d, "i_T"] \\
G_1^2 \arrow[d, swap, "p_1"] & \times & G_2^2 \arrow[d, swap, "p_2"] \arrow[r, "\otimes"] & G_T^4 \arrow[d, "p_T"] \\
G_1 & \times & G_2 \arrow[r, "e"] & G_T
\end{tikzcd}
\qquad
\begin{tikzcd}
(x, y) \arrow[d, swap, "i_1\ i_2"] \arrow[r] & e(x, y) \arrow[d, "i_T"] \\
((1, x), (1, y)) \arrow[d, swap, "p_1\ p_2"] \arrow[r] & (1,1,1, e(x, y)) \arrow[d, "p_T"] \\
(x, y) \arrow[r] & e(x, y)
\end{tikzcd}$$

- $p_1(a, b) = ba^{-\alpha}, p_2(c, d) = dc^{-\beta}$ ElGamal decryption with keys α, β, respectively
- $(a, b) \otimes (c, d) = \big(e(a, c), e(a, d), e(b, c), e(b, d)\big)$
- $p_T(a, b, c, d) = dc^{-\beta}\left(ba^{-\beta}\right)^{-\alpha}$

Common reference string

- CRS has modules $A_1, A_2, A_T, B_1, B_2, B_T, C_1, C_2, C_T$ and (bi)linear maps $i_1, i_2, i_T, p_1, p_2, p_T, \cdot_A, \cdot_B, \cdot_C$ and elements $u_1, \dots, u_m \in B_1, v_1, \dots, v_n \in B_2$
- Two indistinguishable types of CRS
  - WI CRS has $i_1(A_1) \subseteq \langle u_1, \dots, u_m \rangle, i_2(A_2) \subseteq \langle v_1, \dots, v_n \rangle$
  - Soundness CRS has $p_1(u_i) = 0$ and $p_2(v_j) = 0$

Statement

- The statement consist of quadratic equations $eq_1, \dots, eq_N$ in A_1, A_2, A_T of the form
$$\sum_j a_j \cdot y_j + \sum_i x_i \cdot b_i + \sum_{ij} x_i \cdot \gamma_{ij} y_j = t$$
- The prover knows values
$$\vec{x} = (x_1, \dots, x_m) \quad \vec{y} = (y_1, \dots, y_n)$$
that satisfy all equations $eq_1, \dots, eq_N$
- Simplified notation
$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$$

Commitment to witness

- Prover commits in B_1, B_2 to all secret elements

$$c_i = i_1(x_i) + \sum_k r_{ik} u_k \qquad d_j = i_2(y_j) + \sum_k s_{jk} v_k$$

- Let $\vec{c} = (c_1, \ldots, c_m)$ and $\vec{d} = (d_1, \ldots, d_n)$ then

$$\vec{c} = i_1(\vec{x}) + R\vec{u} \qquad \vec{d} = i_2(\vec{y}) + S\vec{v}$$

NIWI proofs

- For each equation
$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$$
the prover creates a NIWI proof $\vec{\pi} \in B_2^n, \vec{\phi} \in B_1^m$

- For each equation the verifier checks
$$i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2(\vec{b}) + \vec{c} \cdot \Gamma \vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{v}$$

Soundness

- For each equation the verifier checks
  $i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2(\vec{b}) + \vec{c} \cdot \Gamma\vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{v}$
- On a soundness string $p_1(\vec{u}) = \vec{0}, p_2(\vec{v}) = \vec{0}$
- We define

$$\vec{a}' = p_1(i_1(\vec{a})) \quad \vec{b}' = p_2\big(i_2(\vec{b})\big) \quad t' = p_T(i_T(t))$$

$$\vec{x}' = p_1(\vec{c}) \qquad \vec{y}' = p_2(\vec{d})$$

  Projecting the verification equation to C_1, C_2, C_T

$$\vec{a}' \cdot \vec{y}' + \vec{x}' \cdot \vec{b}' + \vec{x}' \cdot \Gamma\vec{y}' = t' + 0 + 0 = t'$$

Example

$$
\begin{array}{ccccc}
G_1 \times G_2 & \xrightarrow{e} & G_T & & (x,y) \to e(x,y) \\
i_1\downarrow \quad i_2\downarrow & \bigotimes & \downarrow i_T & & i_1\downarrow \quad i_2\downarrow \qquad \downarrow i_T \\
G_1^2 \times G_2^2 & \rightarrow & G_T^4 & & ((1,x),(1,y)) \to (1,1,1,e(x,y)) \\
p_1\downarrow \quad p_2\downarrow & & \downarrow p_T & & p_1\downarrow \quad p_2\downarrow \qquad \downarrow p_T \\
G_1 \times G_2 & \xrightarrow{e} & G_T & & (x,y) \to e(x,y)
\end{array}
$$

- $p_1(i_1(\vec{a})) = \vec{a} \quad p_2\left(i_2\left(\vec{b}\right)\right) = \vec{b} \qquad p_T(i_T(t)) = t$
- Projection therefore gives us the original equation is satisfied by $\vec{x} = p_1(\vec{c})$ and $\vec{y} = p_2(\vec{d})$

$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{y} = t$$

Completeness

- The prover has commitments
$$\vec{c} = i_1(\vec{x}) + R\vec{u} \quad \vec{d} = i_2(\vec{y}) + S\vec{v}$$
- For each equation the committed witness satisfies
$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{y} = t$$
- For each equation the verifier checks
$$i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2\left(\vec{b}\right) + \vec{c} \cdot \Gamma\vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{v}$$
- The prover can create a proof $\vec{\pi} \in B_2^n, \vec{\phi} \in B_1^m$
$$\vec{\pi} = R^T\left(i_2\left(\vec{b}\right) + \Gamma\vec{d}\right) \qquad \vec{\phi} = S^T(i_1(\vec{a}) + \Gamma^T i_1(\vec{x}))$$

Witness-indistinguishability

- WI CRS $i_1(A_1) \subseteq \langle \vec{u} \rangle, i_2(A_2) \subseteq \langle \vec{v} \rangle$
- The commitments $\vec{c}, \vec{d}$ are perfectly hiding
- What about the proofs?
  $i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2(\vec{b}) + \vec{c} \cdot \Gamma \vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{v}$
- If $\vec{\pi}, \vec{\phi}$ are unique then we have perfect WI
- For non-unique proofs, we will randomize them such that any witness yields a uniform random distribution over proofs satisfying the equation

Witness-indistinguishability

- What about the proofs?
  $$i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2\left(\vec{b}\right) + \vec{c} \cdot \Gamma\vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{v}$$
- For non-unique proofs, we will randomize them such that any witness yields a uniform random distribution over proofs satisfying the equation
  - Observe
    $$\vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{v} = \vec{u} \cdot (\vec{\pi} + T\vec{v}) + \left(\vec{\phi} - T^T\vec{u}\right) \cdot \vec{v}$$
  - On a WI CRS $\vec{\pi} \in \langle\vec{v}\rangle$ so $\vec{\pi}' = \vec{\pi} + T\vec{v}$ is random in $\langle\vec{v}\rangle$
  - Randomise $\vec{\phi}' = \vec{\phi} - T^T\vec{u} + \vec{w}$ with random $\vec{w} \cdot \vec{v} = 0$
    - May require CRS to contain information to make it possible to pick random $\vec{w} \in \langle\vec{u}\rangle$ such that $\vec{w} \cdot \vec{v} = 0$ (but often not needed)

Overview

- CRS defines Z_p-modules $A_1, A_2, A_T, B_1, B_2, B_T, C_1, C_2, C_T$ and (bi)linear maps $i_1, i_2, i_T, p_1, p_2, p_T, \cdot_A, \cdot_B, \cdot_C$ and $\vec{u}, \vec{v}$ and $\vec{w}$-info

$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{y} = t$$

$$i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2(\vec{b}) + \vec{c} \cdot \Gamma\vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{v}$$

$$\vec{a}' \cdot \vec{y}' + \vec{x}' \cdot \vec{b}' + \vec{x}' \cdot \Gamma\overrightarrow{y'} = t'$$

- Prover’s witness is in A_1, A_2
- Commitments and proofs are in B_1, B_2
- Soundness holds in C_1, C_2, C_T

Zero-knowledge

$$i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2(\vec{b}) + \vec{c} \cdot \Gamma\vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{v}$$

- On a WI CRS the commitments and proofs $\vec{c}, \vec{d}, \vec{\pi}, \vec{\phi}$ are perfectly witness-indistinguishable
- Are the commitments and proofs also ZK?
- Problem
  - Cannot simulate proofs without knowing a witness!

Zero-knowledge

- Strategy
  - Set up WI CRS so that the simulator can find a witness
- Consider the case where $A_1 = \boldsymbol{Z}_p$
  - On the WI CRS we have $i_1(A_1) \subseteq \langle \vec{u} \rangle$ so
$$i_1(1) = i_1(0) + \vec{r}^T \vec{u}$$
  - The simulator will use $\vec{r}$ as the simulation trapdoor
- Rewrite all the equations $eq_1, \ldots, eq_N$ to the form
$$1 \cdot (-t) + \vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = 0$$

Zero-knowledge simulation

- Consider 1 to be an extra variable x_0 where we use commitment $c_0 = i_1(1)$
- We now have equations $eq_1, \ldots, eq_N$ of the form
$$x_0 \cdot (-t) + \vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{y} = 0$$
- Choosing $x_0 = 0, \vec{x} = \vec{0}, \vec{y} = \vec{0}$ gives the simulator a witness satisfying all equations simultaneously
- And because $c_0 = i_1(1) = i_1(0) + \vec{r}^T\vec{u}$ on a WI CRS the simulator has an opening of c_0 to 0 that it can use in all the NIWI proofs
- Each commitment is perfectly hiding and each proof perfectly WI, so this is a perfect simulation

Example

- Consider equations over $x_i \in G_1, y_j \in G_2, \hat{x}_i, \hat{y}_j \in \boldsymbol{Z}_p$
  - Pairing product equations

$$\prod_j e(a_j, y_j) \prod_i e(x_i, b_j) \prod_{i,j} e\left(x_i, y_j\right)^{\gamma_{ij}} = e(g,g)^0$$

  - Multi-exponentiation equations in G_1 (similar for G_2)

$$\prod_j {a_j}^{\hat{y}_j} \prod_i x_i^{\beta_i} \prod_{i,j} {x_i}^{\gamma_{ij}\hat{y}_j} = t_{G_1}$$

  - Quadratic equations

$$\sum_j \alpha_j \hat{y}_j \sum_i \hat{x}_i \beta_i + \sum_{i,j} \gamma_{ij} \hat{x}_i \hat{y}_j = t \bmod p$$

- Using $x_i = 1, y_j = 1$, $\hat{x}_i = 0, \hat{y}_j = 0$ we can simulate

Efficiency in the example

- Proofs for $e: G_1 \times G_2 \rightarrow G_T$ setting where DDH problem hard in both G_1 and G_2

Cost of each variable/equation	G_1	G_2
Variables $x \in G_1, \hat{x} \in \boldsymbol{Z}_p$	2	0
Variables $y \in G_2, \hat{y} \in \boldsymbol{Z}_p$	0	2
Pairing product equations (zero-knowledge if all $t_T = 1$)	4	4
Multi-exponentiations in G_1	2	4
Multi-exponentiations in G_2	4	2
Quadratic equations in $\boldsymbol{Z}_p$	2	2

Non-interactive Zero-Knowledge Proofs from Pairings – extra remarks

Jens Groth

University College London

CRS-free proofs for all of NP?	Zero-knowledge	Witness-indistinguishability
Interactive proofs	**4 rounds**	**2 rounds**
Non-interactive proofs	**Impossible**	**Yes**

Naïve idea for NIWI proofs in the plain model

No, maybe you used a simulation CRS

Statement: $x \in L$

CRS σ

Proof π

Prover

Verifier

NIWI proofs in the plain model [GOS12]

- Naïve idea: Provers picks both CRS and proof
  - Not convincing
- Better idea: Prover picks two CRSs and proofs
  - The two CRSs related such that at least one is guaranteed to be sound
  - But the verifier cannot tell which one is the sound string

NIWI proofs in the plain model

At least one CRS is sound. So either π_0 or π_1 shows that $x \in L$

Statement: $x \in L$

CRS σ_0, σ_1
Proof π_0, π_1

Prover

Verifier

NIWI proof in the plain model

- Better idea: Prover picks two CRSs and proofs
  - The two CRSs related such that at least one is guaranteed to be sound
  - But the verifier cannot tell which one is the sound string
- Requirements
  - Prover can pick two related CRSs such that either CRS can give witness-indistinguishability
  - The verifier can check that at least one CRS is sound, but not distinguish the sound CRS from the WI CRS

Suitable groups

- BGN group of composite order $n = pq$ not good because hard to tell whether h has order q
- Prime order groups better
  - For instance $e: G \times G \rightarrow G_T$ with prime order p
  - A CRS specifies $(f, 1, h), (1, g, h), (u, v, w)$
  - Write $(u, v, w) = (f^r, g^s, h^{r+s+t})$
  - If $t = 0$ perfect WI and if $t \neq 0$ perfect soundness
  - Decision linear assumption says hard to distinguish
- Related CRSs
  - $\sigma_0 = (p, G, G_T, e, f, g, h, u_0, v_0, w_0)$
  - $\sigma_1 = (p, G, G_T, e, f, g, h, u_0, v_0, w_0 h)$

NIWI proof in plain model

- Statement: C
- Proof

  $(\sigma_0,\sigma_1) \leftarrow K_{related}(1^k,b)$ (σ_b is WI CRS)

  $\pi_0 \leftarrow P(\sigma_0,C,w)$

  $\pi_1 \leftarrow P(\sigma_1,C,w)$

  The proof is $\pi = (\sigma_0,\sigma_1,\pi_0,\pi_1)$
- Verification

  Check (σ_0,σ_1) related so at least one is sound

  Check (σ_0,C,π_0) is valid proof

  Check (σ_1,C,π_1) is valid proof

Witness-indistinguishabilit

Adversary knows C,w_0,w_1 and sees (σ_0,σ_1,π_0,π_1)

Given circuit C and two witnesses w_0, w_1

- Generate σ_0 as WI CRS and σ_1 as perfect sound CRS

  Proof using w_0 on σ_0 Proof using w_0 on σ_1

  Proof using w_1 on σ_0 Proof using w_0 on σ_1

- Switch to σ_0 perfect sound CRS and σ_1 WI CRS

  Proof using w_1 on σ_0 Proof using w_0 on σ_1

  Proof using w_1 on σ_0 Proof using w_1 on σ_1

- Switch back to σ_0 being WI CRS and σ_1 perfect sound CRS

Special properties of pairing-based proofs

- Proofs consist of group elements and they are verified by pairing product equations
  - We can give an NIWI proof that there exists an NIWI proofs that a statement is true
- Proofs may be modified or randomized
  - Noted by [BCCKLS09] and used in delegatable credentials
  - Controlled malleable proofs formalized in [CKLM12]

Randomization of proofs

- Pairing-based NIZK proofs may be randomized
- Example
  - Consider statement $e(a,x)=1$ in BGN group
  - An NIZK proof would consist of a commitment and proof
    $$c = xh^r \qquad \pi = a^r$$
    which is verified by checking $e(a,c)=e(h,\pi)$
  - Given commitment and proof c,π we can rerandomize
    $$c' = ch^s \qquad \pi' = \pi a^s$$
  - Or we can modify the commitment and proof
    $$c'' = cb^{-1} \qquad \pi'' = \pi^t$$
  - Which shows x'' satisfies $e(a^t, x''b\ ) = 1$

Short pairing-based NIZK arguments

	CRS	Size	Prover comp.	Verifier comp.
Abe-Fehr 07	O(1) group	O(n) group	O(n) expo	O(n) pairing
	Dlog & knowledge of expo.		Comp. sound	Perfect ZK
Groth 10	$O(n^2)$ group	O(1) group	$O(n^2)$ mult	O(n) mult
Groth 10	$O(n^{2/3})$ group	$O(n^{2/3})$ group	$O(n^{4/3})$ mult	O(n) mult
	q-CPDH and q-PKE		Comp. sound	Perfect ZK
Lipmaa 12	$n^{1+o(1)}$ group	O(1) group	$O(n^2)$ add	O(n) mult
Lipmaa 12	$n^{1/2+o(1)}$ group	$n^{1/2+o(1)}$ group	$O(n^{3/2})$ add	O(n) mult
	Λ-PSDL and Λ-PKE		Comp. sound	Perfect ZK
Gennaro-Gentry-	O(n) group	7 group	O(n log n) mult	O(n) mult
Parno-Raykova			Comp. sound	Perfect ZK

Knowledge commitment [G10]

- Commitment key

$$ck = \begin{pmatrix} g, g_1, g_2, \dots \\ \hat{g}, \hat{g}_1, \hat{g}_2, \dots \end{pmatrix} = \begin{pmatrix} g \ , & g^{x} \ , & g^{x^2}, \dots \\ g^{\alpha}, & g^{\alpha x}, & g^{\alpha x^2}, \dots \end{pmatrix}$$

- Commit to $a_1, a_2, \dots, a_q \in \boldsymbol{Z}_p$ as

$$\begin{pmatrix} c \\ \hat{c} \end{pmatrix} = \begin{pmatrix} g^r \prod_{i \in [q]} g_i^{a_i} \\ \hat{g}^r \prod_{i \in [q]} \hat{g}_i^{a_i} \end{pmatrix}$$

- Can verify commitment correct $e(c, \hat{g}) = e(\hat{c}, g)$
- Power Knowledge of Exponent assumption
  - Impossible to make correct commitment without knowing r and $a_1, \dots, a_q$

Homomorphic property

- We now have a perfectly hiding commitment scheme using just two group elements to commit to a set of q *known* values $a_1, \ldots, a_q$
- The commitment scheme is homomorphic

$$(g^r \prod_{i\in[q]} g_i{}^{a_i})(g^s \prod_{i\in[q]} g_i^{b_i}) = g^{r+s} \prod_{i\in[q]} g_i^{a_i+b_i}$$

- We can add multiple committed values in a verifiable way using only a few group elements

Polynomial balancing

- Recall $(g, g_1, \dots, g_q) = (g, g^x, \dots, g^{x^q})$
- Commitment is $c = g^r \prod_{i\in[q]} g_i^{a_i} = g^{r+\sum_{i\in[q]} a_i x^i}$
- Pairing two commitments correspond to computing a committed product of polynomials

$$(r + \sum_i a_i x^i)(s + \sum_j b_j x^j)$$

- Carefully create large polynomial equations that are satisfied if and only if the statement is true
- Use proofs to cancel out extra polynomial terms

Size vs. assumption