Non-interactive Zero-Knowledge Proofs

Jens Groth
University College London
Zero-knowledge proof [GMR85]

Witness: Statement true because...

Prover

Verifier

OK, statement is true

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

Statement
Internet voting

- **Vote**: Encrypts vote to keep it private
- **Ciphertext**: Tally without decrypting individual votes

Voters → Ciphertext → 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

Cipherertext

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,  \( x \in L \)

Prover

Verifier

Proof  \( \pi \)
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 \( w \) 
\((x, w) \in R_L\)

Statement: \( x \in L \)

Prover

Verifier
NIZK proofs in the plain model only possible for trivial languages $L \in \text{BPP}$ [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]

Statement: $x \in L$

Common reference string
0100…11010

Proof: $\pi$

Prover
Verifier

$(x, w) \in R_L$
Common reference string (CRS)

- 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]

$$0110110101000101110100101$$
Zero-knowledge simulation

Common reference string
0100…11010

Statement: \( x \in L \)

Simulation trapdoor
\( \mathcal{A} \rightarrow \tau \)

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 $\sigma$
  – $P(\sigma,x,w)$: Generates a proof $\pi$
  – $V(\sigma,x,\pi)$: Outputs 1 (accept) or 0 (reject)
Public vs. private verification

- Publicly verifiable
  - K generates CRS $\sigma$
  - V checks proof given input $(\sigma, x, \pi)$

- Privately verifiable
  - K generates CRS $\sigma$ and private verification key $\omega$
  - V checks proof given input $(\omega, x, \pi)$
### 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 \( \pi \) 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

Perfect completeness: $\Pr[\text{Accept}] = 1$

- Common reference string $\sigma \leftarrow K(1^k)$
- Statement $x \in L$
- Witness $w$ so $(x,w) \in R$
- $P(\sigma, x, w) \rightarrow \pi$
- $V(\sigma, x, \pi) \rightarrow \text{Accept/reject}$
Soundness

Perfect soundness: ∀ Adv: Pr[Reject] = 1
Statistical soundness: ∀ Adv: Pr[Reject] \approx 1
Computational soundness: ∀ poly-time Adv: Pr[Reject] \approx 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 $\sigma$ $\leftarrow E(1^k) \rightarrow \xi$

Statement $x$

$\pi$

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

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

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

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

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

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

Computational ZK:
\[ \forall \text{ poly-time Adv}: \Pr[\text{Adv} \rightarrow 1|\text{Real}] \approx \Pr[\text{Adv} \rightarrow 1|\text{Simulation}] \]
Witness indistinguishability [FS90]

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

Statement \( x \in L \)

Witnesses \( w_0, w_1 \)

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

\( \Pr(\sigma, x, w_b) \rightarrow \pi \)

Guess \( \in \{0, 1\} \)

Perfect witness-indistinguishable: \( \forall \text{Adv}: \Pr[\text{Guess} = b] = \frac{1}{2} \)

Computational WI: \( \forall \text{poly-time Adv}: \Pr[\text{Guess} = b] \approx \frac{1}{2} \)
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

\[ \pi = (a, z) \]

\[ H(x, a) \]
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: \( x \in L \)

\[ \text{CRS} \]

\[ (x,w) \in R_L \]

\[ K(1^k) \rightarrow (pk,sk) \]

\[ E_{pk}(0;r_1) \]
\[ E_{pk}(1;r_2) \]
\[ E_{pk}(0;r_3) \]
\[ E_{pk}(1;r_4) \]
Statistical sampling

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

CRS

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Probably remaining pairs of encrypted bits are 00 and 11
Kilian-Petrak for instance consider 3SAT formulas

\((x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_4 \lor \neg x_5) \land (x_1 \ldots) \land \ldots \land (x_1 \ldots) \land (\neg x_1 \ldots)\)

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: $|w| \leq |C|$

<table>
<thead>
<tr>
<th>CRS in bits</th>
<th>Proof in bits</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-Ostrovsky-Sahai 12</td>
<td>$O(k_G)$</td>
<td>$O(</td>
</tr>
<tr>
<td>Groth 10</td>
<td>$</td>
<td>C</td>
</tr>
<tr>
<td>Groth 10</td>
<td>$</td>
<td>C</td>
</tr>
<tr>
<td>Gentry 09</td>
<td>$\text{poly}(k)$</td>
<td>$</td>
</tr>
</tbody>
</table>
## Practice

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

<table>
<thead>
<tr>
<th></th>
<th>Circuit SAT</th>
<th>Practical statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inefficient</td>
<td>Damgård 92</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kilian-Petrank 98</td>
<td>1 GB</td>
</tr>
<tr>
<td>Efficient</td>
<td>Groth-Ostrovsky-Sahai 12</td>
<td>1 KB</td>
</tr>
<tr>
<td></td>
<td>Groth-Sahai 12</td>
<td></td>
</tr>
</tbody>
</table>
Non-interactive Zero-Knowledge Proofs from Pairings

Jens Groth
University College London
• 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

• Gen(1^k) generates (p, q, G, G_T, e, g)
• G, G_T finite cyclic groups of order n = pq
• Pairing e: G × G → G_T
  – e(g^a, g^b) = e(g, g)^{ab}
  – G = ⟨g⟩, G_T = ⟨e(g, g)⟩
• 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: \(c = g^ah^r\) \(r \leftarrow \mathbb{Z}_n\)

Decryption: \(c^q = (g^ah^r)^q = g^{qa}h^{qr} = (g^q)^a\)
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: \(c = g^a h^r\) \(r \leftarrow Z_n\)

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

\[ w_4 = \neg(w_1 \land w_2) \]
Prove

\[ 1 = \neg(w_3 \land w_4) \]
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: $e(c, cg^{-1}) = e(h, \pi)$

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

$\rightarrow w = 0 \mod p \text{ or } w = 1 \mod p$
Observation

<table>
<thead>
<tr>
<th>$b_0$</th>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$b_0 + b_1 + 2b_2 - 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

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

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

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

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

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

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 $g = h^\tau$ \quad \tau \leftarrow Z_n^*$

The simulation trapdoor is $\tau$

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

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

Prove

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

Using $w_2 = 0$, $w_3 = 0$

for the NAND proofs
Witness-indistinguishable 0/1-proof

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

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

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 \( \pi \) satisfying equation, no matter whether \( c \) contains 0 or 1
Zero-knowledge of full Circuit SAT proof

Sketch of proof:

\[ \Pr[\text{Adv} \rightarrow 1 | \text{Real proof}] \]

\[ \approx \Pr[\text{Adv} \rightarrow 1 | \text{Real proof on } h \text{ with order } n] \]

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

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

\[ = \Pr[\text{Adv} \rightarrow 1 | \text{Simulated proof}] \]
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
  $$c_i^q = (g^{w_i} h^{r_i})^q = (g^q)^{w_i}$$
<table>
<thead>
<tr>
<th>Known for all of NP?</th>
<th>Computational zero-knowledge</th>
<th>Perfect zero-knowledge (everlasting privacy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive proof</td>
<td>Yes</td>
<td>Yes [Brassard-Crepeau 1986]</td>
</tr>
<tr>
<td>Non-interactive proof</td>
<td>Yes</td>
<td>Yes [Groth-Ostrovsky-Sahai 2012]</td>
</tr>
<tr>
<td></td>
<td>[Goldreich-Micali-Wigderson 1986]</td>
<td></td>
</tr>
</tbody>
</table>
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 $\leftarrow K(1^k)$

$w_{\text{guilt}}$ witness for $C$ unsatisfiable

Comp. culpable soundness: $\forall$ poly-time Adv: $\Pr[\text{Reject}] \approx 1$
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 $w_{\text{guilt}}$ 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

- 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 \( \text{ord}(h) = q \)

\[ \sigma = (n, G, G_T, e, g, h) \]
where \( \text{ord}(h) = n \)
Non-interactive Zero-Knowledge Proofs from Pairings

Jens Groth
University College London
# NIZK proof efficiency

<table>
<thead>
<tr>
<th>Circuit SAT</th>
<th>Practical statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inefficient</td>
<td>Hidden bits</td>
</tr>
<tr>
<td>Efficient</td>
<td>Groth-Ostrovsky-Sahai 12</td>
</tr>
</tbody>
</table>
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

\[ \mu_1, \ldots, \mu_m \in \mathbb{Z}_n, \Omega, g, u, v', v_1, \ldots, v_m \in G, A \in G_T \]

• Prover knows witness \( \theta_1, \theta_2, \theta_3, \theta_4 \in G \)

\[
e(\theta_1, \theta_2 \Omega) = A \quad e(\theta_2, u) = e(\theta_3, g)e(\theta_4, v') \prod_{i} v_i^{\mu_i}
\]

• 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

\[ a, c \in G \quad b, d \in \mathbb{Z}_n \]

\[ t = b + yd \mod n \]

\[ t_G = x^y a^y c^t \]

\[ t_T = e(t_G, ct_G^b) \]
Non-interactive cryptographic proofs for correctness of constructions

Yes, here is a proof.

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

$t = b + yd \mod n$

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

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

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 \(\tau \in \mathbb{Z}_n^*\)

- Commitment to group element \(x \in G\)
  \[c = x h^r\] \[r \leftarrow \mathbb{Z}_n\]

- 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 \(x^\lambda\)

- Simulation CRS: Perfect hiding commitments
  - When \(h\) has order \(n\) the commitment is a random group element
Homomorphic properties

- Commitments are homomorphic
  \[- (xh^r)(yh^s) = xyh^{r+s} \]
  \[- (g^xh^r)(g^yh^s) = g^{x+y}h^{r+s} \]
- Pairing commitments
  \[- e(xh^r, yh^s) = e(x, y)e(h, x^sy^rh^{rs}) \]
  \[- e(xh^r, g^yh^s) = e(g, x^y)e(h, x^sg^{yr}h^{rs}) \]
  \[- e(g^xh^r, g^yh^s) = e(g, g)^{xy}e(h, g^{xs+yr}h^{rs}) \]
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 \( \pi \) 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 \( \pi \) by checking
  \[ e(a, d)e(c, d) = t_T e(h, \pi) \]

• Soundness when \( \text{ord}(h) = q \)
  
  - 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 \( \pi \) by checking
  \[ e(a, d)e(c, d) = t_T e(h, \pi) \]
- Witness-indistinguishability when \( \text{ord}(h) = n \)
  - The commitments are perfectly hiding, so there are many different possible openings \( x, r, y, s \) of \( c, d \) satisfying the equation
  - However, since \( \text{ord}(h) = n \) there is a unique proof \( \pi \) 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 \( \pi \)
Full NIWI proof for a set of equations

- Suppose we have equations $eq_1, eq_2, ...$ of the form
  \[
  \prod_i e(a_i, x_i) \prod_{i,j} e(x_i, x_j)^{y_{ij}} = t_T
  \]
- We can give a NIWI proof that there are values $x_1, ..., x_m \in G$
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 $\mathbb{Z}_n$ 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
  \[
  \sum_j b_j y_j + \sum_{i,j} \gamma_{ij} y_i y_j = t \mod n
  \]
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?

- Commuting linear and bilinear map
- We will generalize this methodology
  - Groups can have prime or composite order
  - Pairing \( e: G_1 \times G_2 \rightarrow G_T \) 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.

\[
\begin{align*}
G \times G & \rightarrow G_T \\
\downarrow & \downarrow \\
G \times G & \rightarrow G_T \\
\downarrow & \downarrow \\
G_p \times G_p & \rightarrow G_{T,p} \\
\downarrow & \downarrow \\
(x, y) & \rightarrow t_T \\
\downarrow & \downarrow \\
(x h^r, y h^s) & \rightarrow t_T e(h, \pi) \\
\downarrow & \downarrow \\
(x^\lambda, y^\lambda) & \rightarrow t_T^\lambda
\end{align*}
\]
Modules

• An abelian group \((A, +, 0)\) is a \(\mathbb{Z}_p\)-module if \(\mathbb{Z}_p\) acts on \(A\) such that for all \(r, s \in \mathbb{Z}_p, a, b \in A\)
  - \(1a = a\)
  - \((r + s)a = ra + sa\)
  - \(r(a + b) = ra + rb\)
  - \(r(sa) = (rs)a\)

• If \(p\) is a prime then \(A\) is a vector space

• Examples
  - \(\mathbb{Z}_p, G_1, G_2, G_T, G_1^2, G_2^2, G_T^4\) are \(\mathbb{Z}_p\)-modules
Modules with bilinear map

• We will be interested in finite $\mathbb{Z}_p$-modules $A_1, A_2, A_T$ with a bilinear map $\cdot_A : A_1 \times A_2 \to A_T$

• Examples:
  - pair: $G_1 \times G_2 \to G_T$ \( (x, y) \mapsto e(x, y) \)
  - exp: $G_1 \times \mathbb{Z}_p \to G_1$ \( (x, y) \mapsto x^y \)
  - exp: $\mathbb{Z}_p \times G_2 \to G_2$ \( (x, y) \mapsto y^x \)
  - mult: $\mathbb{Z}_p \times \mathbb{Z}_p \to \mathbb{Z}_p$ \( (x, y) \mapsto xy \text{ mod } p \)
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

• Linear maps and modules
  \[ A \xrightarrow{i} B \xrightarrow{p} C \]

• Elements \( u_1, u_2, \ldots, u_m \in B \)

• Commit to an element \( x \in A \)

\[ c = i(x) + \sum_{i} r_i u_i \]

• Perfectly hiding \( x \) if \( i(A) \subseteq \langle u_1, \ldots, u_m \rangle \)

• Perfectly binding to \( p(c) \)
  – For soundness, we want \( p(u_i) = 0 \)
Example

- Linear maps and modules
  \[
  i \quad p \\
  G \to G^2 \to G
  \]
  \[
  i: x \to (1, x) \\
  p: (a, b) \to ba^{-\alpha}
  \]

- 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 \( p(i(x)) = x \)
Commuting linear and bilinear maps

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

\[
\begin{align*}
A_1 \times A_2 & \xrightarrow{i_A} A_T \\
i_1 \downarrow & \downarrow i_2 \downarrow \quad \downarrow i_T \\
B_1 \times B_2 & \xrightarrow{i_B} B_T \\
p_1 \downarrow & \downarrow p_2 \downarrow \quad \downarrow p_T \\
C_1 \times C_2 & \xrightarrow{i_C} C_T
\end{align*}
\]

- 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

\[ G_1 \times G_2 \xrightarrow{e} G_T \]

\[ i_1 \downarrow \quad i_2 \downarrow \quad \downarrow i_T \]

\[ G_1^2 \times G_2^2 \xrightarrow{\otimes} G_T^4 \]

\[ p_1 \downarrow \quad p_2 \downarrow \quad \downarrow p_T \]

\[ G_1 \times G_2 \xrightarrow{e} G_T \]

\[ (x, y) \to e(x, y) \]

\[ (1, x), (1, y) \to (1, 1, 1, e(x, y)) \]

- \[ p_1(a, b) = ba^{-\alpha}, p_2(c, d) = dc^{-\beta} \]

- \[ (a, b) \otimes (c, d) = (e(a, c), e(a, d), e(b, c), e(b, d)) \]

- \[ p_T(a, b, c, d) = dc^{-\beta} (ba^{-\beta})^{-\alpha} \]

ElGamal decryption with keys \( \alpha, \beta \), respectively
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 $u_1, ..., u_m \in B_1, v_1, ..., v_n \in B_2$

• Two indistinguishable types of CRS
  – WI CRS has $i_1(A_1) \subseteq \langle u_1, ..., u_m \rangle, i_2(A_2) \subseteq \langle v_1, ..., v_n \rangle$
  – Soundness CRS has $p_1(u_i) = 0$ and $p_2(v_j) = 0$
Statement

• The statement consist of quadratic equations $eq_1, \ldots, eq_N$ 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 values

$$
\tilde{x} = (x_1, \ldots, x_m) \quad \tilde{y} = (y_1, \ldots, y_n)
$$

that satisfy all equations $eq_1, \ldots, eq_N$

• Simplified notation

$$
\tilde{a} \cdot \tilde{y} + \tilde{x} \cdot \tilde{b} + \tilde{x} \cdot \Gamma \tilde{y} = t
$$
Commitment to witness

- Prover commits in $B_1, B_2$ to all secret elements

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

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

\[
\tilde{c} = i_1(\tilde{x}) + R\tilde{u} \quad \tilde{d} = i_2(\tilde{y}) + S\tilde{v}
\]
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
\[
\begin{align*}
\vec{a}' & = p_1(i_1(\vec{a})) & \vec{b}' & = p_2(i_2(\vec{b})) & t' & = p_T(i_T(t)) \\
\vec{x}' & = p_1(\vec{c}) & \vec{y}' & = p_2(\vec{d})
\end{align*}
\]

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

\[ G_1 \times G_2 \rightarrow G_T \]
\[ (x, y) \rightarrow e(x, y) \]

\[ i_1 \downarrow \quad i_2 \downarrow \quad \downarrow i_T \]
\[ G_1^2 \times G_2^2 \rightarrow G_T^4 \]
\[ ((1, x), (1, y)) \rightarrow (1, 1, 1, e(x, y)) \]

\[ p_1 \downarrow \quad p_2 \downarrow \quad \downarrow p_T \]
\[ (x, y) \rightarrow e(x, y) \]

- \[ p_1(i_1(\tilde{a})) = \tilde{a} \quad p_2(i_2(\tilde{b})) = \tilde{b} \quad p_T(i_T(t)) = t \]
- Projection therefore gives us the original
equation is satisfied by \( \tilde{x} = p_1(\tilde{c}) \) and \( \tilde{y} = p_2(\tilde{d}) \)
\[ \tilde{a} \cdot \tilde{y} + \tilde{x} \cdot \tilde{b} + \tilde{x} \cdot \Gamma \tilde{y} = t \]
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
  \( i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2(\vec{b}) + \vec{c} \cdot \Gamma\vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \vec{\phi} \cdot \vec{\nu} \)

• The prover can create a proof \( \vec{\pi} \in B_2^n, \vec{\phi} \in B_1^m \)
  \( \vec{\pi} = R^T(i_2(\vec{b}) + \Gamma\vec{d}) \quad \vec{\phi} = S^T(i_1(\vec{a}) + \Gamma^T i_1(\vec{x})) \)
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?
  \[ i_1(\vec{a}) \cdot \vec{d} + \vec{c} \cdot i_2(\vec{b}) + \vec{c} \cdot \Gamma \vec{d} = i_T(t) + \vec{u} \cdot \vec{\pi} + \phi \cdot \vec{\nu} \]

- For non-unique proofs, we will randomize them such that any witness yields a uniform random distribution over proofs satisfying the equation
  - Observe
    \[ \vec{u} \cdot \vec{\pi} + \phi \cdot \vec{\nu} = \vec{u} \cdot (\vec{\pi} + T\vec{\nu}) + (\phi - T^T\vec{u}) \cdot \vec{\nu} \]
  - On a WI CRS \( \vec{\pi} \in \langle \vec{\nu} \rangle \) so \( \vec{\pi}' = \vec{\pi} + T\vec{\nu} \) is random in \( \langle \vec{\nu} \rangle \)
  - Randomise \( \vec{\phi}' = \vec{\phi} - T^T\vec{u} + \vec{w} \) with random \( \vec{w} \cdot \vec{\nu} = 0 \)
    - May require CRS to contain information to make it possible to pick random \( \vec{w} \in \langle \vec{u} \rangle \) such that \( \vec{w} \cdot \vec{\nu} = 0 \) (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

\[
\hat{a} \cdot \hat{y} + \hat{x} \cdot \hat{b} + \hat{x} \cdot \Gamma \hat{y} = t
\]

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

\[
\hat{a}' \cdot \hat{y}' + \hat{x}' \cdot \hat{b}' + \hat{x}' \cdot \Gamma \hat{y}' = t'
\]

- 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

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

- On a WI CRS the commitments and proofs \( \hat{c}, \hat{d}, \hat{\pi}, \hat{\phi} \) 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 $A_1 = Z_p$
  – 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 $eq_1, ..., eq_N$ of the form
  \[ x_0 \cdot (-t) + \tilde{a} \cdot \tilde{y} + \tilde{x} \cdot \tilde{b} + \tilde{x} \cdot \Gamma \tilde{y} = 0 \]
- Choosing $x_0 = 0, \tilde{x} = \tilde{0}, \tilde{y} = \tilde{0}$ gives the simulator a witness satisfying all equations simultaneously
- And because $c_0 = i_1(1) = i_1(0) + \tilde{r}^T \tilde{u}$ 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 $x_i \in G_1, y_j \in G_2, \hat{x}_i, \hat{y}_j \in \mathbb{Z}_p$
  - Pairing product equations
    $$\prod_j e(a_j, y_j) \prod_i e(x_i, b_j) \prod_{i,j} e(x_i, y_j)^{\gamma_{ij}} = e(g, g)^0$$
  - 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 $G_2$

<table>
<thead>
<tr>
<th>Cost of each variable/equation</th>
<th>$G_1$</th>
<th>$G_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables $x \in G_1, \hat{x} \in \mathbb{Z}_p$</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Variables $y \in G_2, \hat{y} \in \mathbb{Z}_p$</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Pairing product equations (zero-knowledge if all $t_T = 1$)</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Multi-exponentiations in $G_1$</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Multi-exponentiations in $G_2$</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Quadratic equations in $\mathbb{Z}_p$</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Non-interactive Zero-Knowledge Proofs from Pairings – extra remarks

Jens Groth
University College London
<table>
<thead>
<tr>
<th>CRS-free proofs for all of NP?</th>
<th>Zero-knowledge</th>
<th>Witness-indistinguishability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive proofs</td>
<td>4 rounds</td>
<td>2 rounds</td>
</tr>
<tr>
<td>Non-interactive proofs</td>
<td>Impossible</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Naïve idea for NIWI proofs in the plain model

Statement: $x \in L$

Prover

CRS $\sigma$

Proof $\pi$

Verifier

No, maybe you used a simulation CRS
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

Statement: $x \in L$

Prover

CRS $\sigma_0, \sigma_1$
Proof $\pi_0, \pi_1$

Verifier

At least one CRS is sound. So either $\pi_0$ or $\pi_1$ shows that $x \in L$
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

\[(\sigma_0, \sigma_1) \leftarrow K_{\text{related}}(1^k, b) \quad (\sigma_b \text{ is WI CRS})\]
\[\pi_0 \leftarrow P(\sigma_0, C, w)\]
\[\pi_1 \leftarrow P(\sigma_1, C, w)\]

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

• Verification

Check \((\sigma_0, \sigma_1)\) related so at least one is sound
Check \((\sigma_0, C, \pi_0)\) is valid proof
Check \((\sigma_1, C, \pi_1)\) is valid proof
Witness-indistinguishability

Given circuit C and two witnesses \( w_0, w_1 \)

- Generate \( \sigma_0 \) as WI CRS and \( \sigma_1 \) as perfect sound CRS
  
  - Proof using \( w_0 \) on \( \sigma_0 \)
  
  - Proof using \( w_0 \) on \( \sigma_1 \)
  
  - Proof using \( w_1 \) on \( \sigma_0 \)
  
  - Proof using \( w_0 \) on \( \sigma_1 \)

- Switch to \( \sigma_0 \) perfect sound CRS and \( \sigma_1 \) WI CRS
  
  - Proof using \( w_1 \) on \( \sigma_0 \)
  
  - Proof using \( w_0 \) on \( \sigma_1 \)
  
  - Proof using \( w_1 \) on \( \sigma_0 \)
  
  - Proof using \( w_1 \) on \( \sigma_1 \)

- Switch back to \( \sigma_0 \) being WI CRS and \( \sigma_1 \) perfect sound CRS

Adversary knows C, \( w_0, w_1 \) and sees \((\sigma_0, \sigma_1, \pi_0, \pi_1)\)
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
    $$c = xh^r \quad \pi = a^r$$
    which is verified by checking $e(a, c) = e(h, \pi)$
  - Given commitment and proof $c, \pi$ we can rerandomize
    $$c' = ch^s \quad \pi' = \pi a^s$$
  - Or we can modify the commitment and proof
    $$c'' = cb^{-1} \quad \pi'' = \pi^t$$
  - Which shows $x''$ satisfies $e(a^t, x''b) = 1$
## Short pairing-based NIZK arguments

<table>
<thead>
<tr>
<th></th>
<th>CRS</th>
<th>Size</th>
<th>Prover comp.</th>
<th>Verifier comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abe-Fehr 07</strong></td>
<td>O(1) group</td>
<td>O(n) group</td>
<td>O(n) expo</td>
<td>O(n) pairing</td>
</tr>
<tr>
<td></td>
<td>Dlog &amp; knowledge of expo.</td>
<td>Comp. sound</td>
<td></td>
<td>Perfect ZK</td>
</tr>
<tr>
<td><strong>Groth 10</strong></td>
<td>O(n^2) group</td>
<td>O(1) group</td>
<td>O(n^2) mult</td>
<td>O(n) mult</td>
</tr>
<tr>
<td><strong>Groth 10</strong></td>
<td>O(n^{2/3}) group</td>
<td>O(n^{2/3}) group</td>
<td>O(n^{4/3}) mult</td>
<td>O(n) mult</td>
</tr>
<tr>
<td></td>
<td>q-CPDH and q-PKE</td>
<td></td>
<td>Comp. sound</td>
<td>Perfect ZK</td>
</tr>
<tr>
<td><strong>Lipmaa 12</strong></td>
<td>n^{1+o(1)} group</td>
<td>O(1) group</td>
<td>O(n^2) add</td>
<td>O(n) mult</td>
</tr>
<tr>
<td><strong>Lipmaa 12</strong></td>
<td>n^{1/2+o(1)} group</td>
<td>n^{1/2+o(1)} group</td>
<td>O(n^{3/2}) add</td>
<td>O(n) mult</td>
</tr>
<tr>
<td></td>
<td>Λ-PSDL and Λ-PKE</td>
<td></td>
<td>Comp. sound</td>
<td>Perfect ZK</td>
</tr>
<tr>
<td><strong>Gennaro-Gentry-</strong></td>
<td>O(n) group</td>
<td>7 group</td>
<td>O(n log n) mult</td>
<td>O(n) mult</td>
</tr>
<tr>
<td><strong>Parno-Raykova</strong></td>
<td></td>
<td></td>
<td>Comp. sound</td>
<td>Perfect ZK</td>
</tr>
</tbody>
</table>
Knowledge commitment [G10]

- **Commitment key**
  \[ ck = \left( g, g_1, g_2, \ldots \right) = \left( g, g^x, g^{x^2}, \ldots \right) \]
  \[ \left( \hat{g}, \hat{g}_1, \hat{g}_2, \ldots \right) = \left( g^\alpha, g^{\alpha x}, g^{\alpha x^2}, \ldots \right) \]

- **Commit to** \( a_1, a_2, \ldots, a_q \in \mathbb{Z}_p \) as
  \[
  \begin{align*}
  \left( c \right) &= \left( g^r \prod_{i \in [q]} g_i^{a_i} \right) \\
  \left( \hat{c} \right) &= \left( \hat{g}^r \prod_{i \in [q]} \hat{g}_i^{a_i} \right)
  \end{align*}
  \]

- 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 \( a_1, \ldots, a_q \)
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 \((g, g_1, \ldots, g_q) = (g, g^x, \ldots, g^{x^q})\)

• 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

- random oracle
- knowledge extract.
- pairing-based
- factoring-based
- trapdoor perm.
- one-way functions

Risk

Size

AF07, GW11

Mic
GGPR
Gro

CDS
GOS
Gro
Dam
BDMP
BFM
Gro
KP
FLS