The evolution of pairing-based zero-knowledge proofs

2nd ZKProof Workshop 2019

Jens Groth, DFINITY
Recurring motifs

- **Language**
  - Types of statements we can prove

- **Security**
  - Underpinning assumptions
  - Unconditional soundness vs unconditional zero-knowledge

- **Efficiency**
  - Prover computation, verifier computation, interaction, setup size, succinctness
Pre-pairing
Abiogenesis

- Goldwasser-Micali-Rackoff 85/89
  - Defined zero-knowledge proofs: complete, sound, zero knowledge
- Constructed interactive zero-knowledge proof systems
  - Quadratic residuosity as well as quadratic non-residuosity

\[ x \equiv w^2 \mod N \]
Cambrian explosion
Biodiversity

- Language
  - [GMW86/91] All of NP (Graph 3-colorability)
  - [FLS90/99] Hamiltonicity, [BCC86] Boolean circuit SAT, …

- Security
  - [GMW86/91] One-way functions suffice for computational zero-knowledge
  - [(BC86,C86)/BCC88] Zero-knowledge against unbounded adversaries

- Efficiency
  - [BFM88] Non-interactive zero-knowledge proofs in CRS model
  - [Kil92] Succinct interactive proofs

- Properties
  - [TW87,FFS88,FS90,BG92] Proof of knowledge
Devonian explosion

- [Sch90/91] Discrete log based signatures
- [CDS94,Cra96] Σ-protocols
- [CD98] Arithmetic circuit satisfiability

<table>
<thead>
<tr>
<th>Proof system</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>[CD98]</td>
<td>$O(N)$ elements</td>
</tr>
<tr>
<td>[Gro09]</td>
<td>$O(\sqrt{N})$ elements</td>
</tr>
<tr>
<td>[BCCGP16]</td>
<td>$O(\log N)$ elements</td>
</tr>
</tbody>
</table>

- [BBBPWM18] Bulletproofs
- [CS98] Non-interactive designated verifier proofs for linear relations
Pairing-based NIZK proofs
Adaptive radiation?

<table>
<thead>
<tr>
<th>ZK proof for NP-complete languages</th>
<th>Computational zero knowledge</th>
<th>Unconditional zero knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-interactive</td>
<td>Blum-Feldman-Micali 1988</td>
<td>?</td>
</tr>
</tbody>
</table>

- Are there non-interactive zero-knowledge proofs with everlasting privacy?
  - Fiat-Shamir suggested answer could be yes. But based on standard assumptions?
Exogenous genetic material

[BGN05] Pairing-based double-homomorphic encryption scheme

- Symmetric composite order groups of order $N = pq$ with $e: G \times G \rightarrow G_T$
- Public key $(g, h)$ where $\text{ord}(g) = N$ and $\text{ord}(h) = q$
  - Assumption: Hard to decide whether $h$ has full order $N$ or lives in subgroup of order $q$
- Encrypt small integer $m$ under public key $(g, h)$ as $g^mh^r$
- Additively homomorphically

\[
g^a h^r \cdot g^b h^s = g^{a+b} h^{r+s}
\]
- Multiplicatively homomorphically

\[
e(g^a h^r, g^b h^s) = e(g, g)^{ab} e(g^{(a+b)hr^s}, h)
\]
Transformation

- [BGN05] + NIZK \iff NIZK with perfect and everlasting zero knowledge

<table>
<thead>
<tr>
<th>ZK proof for NP-complete languages</th>
<th>Computational zero knowledge</th>
<th>Unconditional zero knowledge</th>
</tr>
</thead>
</table>

- Main idea to prove circuit satisfiability
  - Commit to wire values in circuit as \( g^a h^r, g^b h^s, g^c h^t, \ldots \)
  - Verifier can easily add committed elements \( g^a h^r \cdot g^b h^s = g^{a+b} h^{r+s} \)
  - Prover can demonstrate multiplicative relation \( e(g^a h^r, g^b h^s) = e(g^c h^t, g) e(\pi, h) \)
  - Perfectly sound if \( \text{ord}(h) = q \) and perfectly zero knowledge if \( \text{ord}(h) = N \)
Clade

- [GOSb] NIZK proofs based on prime-order groups with pairings
  - Also setup-free non-interactive witness-indistinguishable proofs
- [Gro06] NIZK proofs for practical language (pairing-product equations)
  - [BW06,BW07] NIZK proofs in pairing-based group signatures
- [GS08] Efficient proofs for practical languages
  - Practical language captures correctness of generic computation (here for symmetric pairing)
  - Statement: a public set of equations using generic operations and public values
    - Pairing-product equations: $e(A, X) \cdot e(X, Y)^y = 1$
    - Multi-exponentiation equations: $X^a \cdot B^y \cdot X^z = T$
    - Quadratic equations: $1 \cdot y + z \cdot z = t$
  - Witness: Secret field elements $y, z$ and group elements $X, Y$ satisfying all equations
Statement: Here is a ciphertext and a document. The ciphertext contains a digital signature on the document.

<table>
<thead>
<tr>
<th>Inefficient</th>
<th>Kilian-Petrrank 1994</th>
<th>Groth 2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficient</td>
<td>Groth-Ostrovsky-Sahai 2006</td>
<td>Groth-Sahai 2008</td>
</tr>
</tbody>
</table>
Pairing-based SNARKs
Pterygota

[Kil92] Probabilistically checkable proofs → succinct zero-knowledge arguments

[FS86] Hash + public-coin proofs → non-interactive arguments

[Mic00(94)] Succinct non-interactive arguments
Succinctness

- [GW11] SNARG = Succinct non-interactive argument
  - Need non-falsifiable assumptions to build SNARGs
- [BCCT12] SNARK = Succinct non-interactive argument of knowledge
  - ...and zk-SNARK = zero-knowledge SNARK
  - Added ease of verification requirement - verifier time polynomial in statement size

Succinct = almost zero-knowledge

Succinct = easy verification
Pairing-based SNARKs

- [Gro10] Short pairing-based non-interactive zero-knowledge arguments
  - Preprocessing zk-SNARK, but the term had not been coined yet 😊

<table>
<thead>
<tr>
<th>Common reference string</th>
<th>Proof size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(N^2)$ elements</td>
<td>$O(1)$ elements</td>
</tr>
<tr>
<td>$O(N^{2\over 3})$ elements</td>
<td>$O(N^{2\over 3})$ elements</td>
</tr>
</tbody>
</table>

And $O(N^2)$ computation for prover and verifier, where $N$ is size of Boolean circuit or arithmetic circuit

- Power knowledge of exponent assumption
  - CRS includes group elements $g, g^x, ..., g^{xq}, g^\alpha, g^{\alpha x}, ..., g^{\alpha xq}$
  - Can only compute $c, \hat{c} = c^\alpha$ by using known $a_0, ..., a_q$ in $c = \prod (g^{xi})^{a_i}$ and $\hat{c} = \prod (g^{\alpha xi})^{a_i}$
  - Symmetric pairing, so can verify $\hat{c} = c^\alpha$ by checking $e(c, g^\alpha) = e(\hat{c}, g)$
Balancing polynomials in the exponent

● Core idea

○ For multiplication gates, suppose we have commitments $A = \prod_{i=0}^{n-1} (g^{x^i})^{a_i}, B = \prod_{j=0}^{n-1} (g^{xjn})^{b_j}$

$$e(A, B) = e\left(g, \prod_{i,j} (g^{x^{i+jn}})^{a_ib_j}\right) = e(g, g)^{\sum_{i=0}^{n-1} a_ib_ix^{i(1+n)}+\sum_{i \neq j} a_ib_jx^{i+jn}}$$

○ So given commitment $C = \prod_{i=0}^{n-1} (g^{x^i})^{c_i}$ and constant commitment $D = \prod_{i=0}^{n-1} (g^{xjn})^{-1}$

$$e(A, B)e(C, D) = e(g, g)^{\sum_{i=0}^{n-1} (a_ib_i-c_i)x^{i(1+n)}} \cdot e(g, g)^{\sum_{i \neq j} \text{something} \cdot x^{i+jn}} = e(g, \pi)$$

where the left-hand side disappears if and only if $c_i = a_ib_i$

○ The main thrust of the proof system is to design the CRS so the proof $\pi$ cannot have $i = j$ coefficients, but can eliminate any $i \neq j$ coefficients of the in-the-exponent polynomials
Intelligent design

<table>
<thead>
<tr>
<th>Idea</th>
<th>CRS</th>
<th>Proof size</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Lip13] Sum-free sets</td>
<td>$N^{1+o(1)}$ elements</td>
<td>$O(1)$ elements</td>
</tr>
<tr>
<td>[GGPR13] QSP and QAP</td>
<td>$O(N)$ elements</td>
<td>$O(1)$ elements</td>
</tr>
</tbody>
</table>

- Quadratic span programs and quadratic arithmetic programs
  - Instead of coefficients of polynomials, look at distinct evaluation points $r_1, \ldots, r_n$
  - Lagrange polynomials $\ell_1(x), \ldots, \ell_n(x)$ such that $l_i(r_k) = 1$ if $i = k$ and otherwise $\ell_i(r_k) = 0$
  - Given commitments $A = g^{\sum a_i \ell_i(x)}, B = g^{\sum b_j \ell_j(x)}, C = g^{\sum c_i \ell_i(x)}$ and constant $D = g^{-\sum \ell_i(x)}$
    
    $$e(A, B)e(C, D) = e(g, g)^{\sum (a_i b_i - c_i) \ell_i^2(x)} \cdot e(g, g)^{\sum_{i \neq j} \text{something} \cdot \ell_i(x) \ell_j(x)} = e(\pi, g^{\prod (x-r_i)})$$
    
    if and only if $a_i b_i = c_i$, since $\ell_i^2(r_k) = 1$ if and only if $i = k$, and $\ell_i(r_k)\ell_j(r_k) = 0$ in all $r_k$
Arithmetic circuit satisfiability

- Universal CRS, works for any circuit [Gro10, Lip13]
  - Statement consists of circuit description, public inputs, and witness of private inputs
- Specialized CRS, tailored to specific circuit [GGPR13]
  - Circuit is fixed, statement consists of public inputs, and witness of private inputs
  - With logarithmic overhead use universal circuit, adaptive with CRS size $O(N \log N)$
Past, present and future
Phylogenetic web

Pairings

PZK → NIZK

Pairing NIZK → NIZK for pairings

Knowledge of exponent

Pairing SNARK

Succinct Σ

Programming

Applications
Sexual reproduction

Theory 💖 Practice

- [PHGR13] Pinocchio implements pairing-based zk-SNARKs for simple C
- [BFR+13, WSR+15,BCG+13] Pantry, Buffet, Libsnark
- Language

R1CS

Compilers, general computation
Adaptation

- **Language**
  - Pairing-friendly languages, R1CS, general computation
  - Programming languages, interoperability

- **Security**
  - Setup: multi-crs, mpc-generated, updatable
  - Formal verification, usable security, convincing people

- **Efficiency**
  - Asymmetric pairings, nesting of proofs, commit-and-prove
  - Research & development
Invasive Species