Scalable Distributed Key Generation

Robert Chen
Mentored by Alin Tomescu

2019 PRIMES Conference
5/19/19
Background: Digital Signatures

Signer

M = “Hello.”
σ = Sign(M, SK_{Signer})

Verifier

Verify(σ, M, PK_{Signer}) = true
Background: Threshold Signatures

\[
\sigma_1 = \text{Sign}(M, SK_1) \\
\sigma_2 = \text{Sign}(M, SK_2) \\
\vdots \\
\sigma_n = \text{Sign}(M, SK_n)
\]

Signature Shares

\[\sigma = \text{Sign}(M, SK_{\text{group}})\]

Verify(\(\sigma\), M, PK_{\text{group}}) = true
Distributed Key Generation (DKG)

\[ S_1 \rightarrow S_2 \rightarrow \cdots \rightarrow S_n \rightarrow SK_{\text{group}} \]

\[ PK_{\text{group}} \]

\[ SK_{\text{group}} \]
Distributed Key Generation (DKG)

# malicious < t
Distributed Key Generation (DKG): Applications

- Generating secret keys for threshold signature schemes
- Generating random nonces for Schnorr threshold signatures
- Random beacons
- Proactive Secret Sharing
## Contributions

<table>
<thead>
<tr>
<th>DKG scheme</th>
<th>Per-player bandwidth</th>
<th>Per-player computation time (deal + verify)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feldman DKG</td>
<td>$O(nt)$</td>
<td>$O(nt)$</td>
</tr>
<tr>
<td>Kate DKG</td>
<td>$O(n)$</td>
<td>$O(nt)$</td>
</tr>
<tr>
<td>AMT DKG</td>
<td>$O(n \log(n))$</td>
<td>$O(n \log(n))$</td>
</tr>
</tbody>
</table>
DKG Outline

- Each player $i$ acts as a **dealer** and “shares” a secret $s_i$ with all other players via Verifiable Secret Sharing (VSS)

- **Our contribution:** We show how to do VSS in $O(n \log n)$ time rather than $O(nt)$ time, which helps scale DKG

$$SK = \sum s_i$$
Secret Sharing (SS)

- Dealer picks a secret $s$ and “shares” it with all other players such that $t$ out of $n$ can reconstruct it

$$p(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{t-1} x^{t-1}$$

$$s = c_0$$
Secret Sharing (SS)

Dealer

Pick a p(x)

p(1)

p(2)

⋯

p(n)
Verifiable Secret Sharing (VSS)

Dealer

Pick a \( p(x) \)

\[ y_1 = p(1) \]
\[ y_2 = p(2) \]
\[ \vdots \]
\[ y_n = p(n) \]

Verify \((y_1, \pi_1)\)

Verify \((y_2, \pi_2)\)

Verify \((y_n, \pi_n)\)
Polynomial Commitments

- Polynomial commitment to $p(x)$ is $g^{p(\alpha)}$
- How do we provide evaluation proofs $\pi_i$ that a value $p(i) = y$ and verify proof against commitment $g^{p(\alpha)}$?
- Polynomial remainder theorem:
  $p(x) - y = q(x)(x-i)$ if and only if $p(i) = y$
- **Proof**: Commitment to quotient $g^{q(\alpha)}$
- Verify? Check using magic! (bilinear pairings)
- Dealer: $O(nt)$ time to compute evaluation proofs
Solution: Multipoint Evaluation

- Need to build evaluation proofs that \( p(i) = y \)
- Key idea: multipoint evaluation is just a tree of polynomials. We commit to some of them and obtain proofs too.

\[ p(x) \]
\[ p(1) \quad p(2) \quad p(3) \quad \ldots \ldots \ldots \quad p(n) \]

\( O(n \log^2 n) \) rather than \( O(nt) \)
Multipoint Evaluation

\[ p / (x-1)(x-2)(x-3)(x-4) \]

\[ r_1,4 / (x-1)(x-2) \]

\[ q_{1,2}, r_{1,2} \]

\[ r_{1,4} / (x-1) \]

\[ q_1, r_1 \]

\[ r_{1,2} / (x-2) \]

\[ q_2, r_2 \]

\[ r_{3,4} / (x-3) \]

\[ q_3, r_3 \]

\[ r_{3,4} / (x-4) \]

\[ q_4, r_4 \]

\[ r_1 = p \% (x-1) = p(1) \]

\[ r_2 = p \% (x-2) = p(2) \]

\[ r_3 = p \% (x-3) = p(3) \]

\[ r_4 = p \% (x-4) = p(4) \]

\[ O(n \log^2 n) \]
Authenticated Multipoint Evaluation Trees (AMT)

\[ p / (x-1)(x-2)(x-3)(x-4) \]

\[ r_{1,4} / (x-1)(x-2) \]

\[ q_{1,2}, r_{1,2} \]

\[ r_{1,4} / (x-3)(x-4) \]

\[ q_{3,4}, r_{3,4} \]

\[ r_{1,4} / (x-1) = p(1) \]

\[ r_{2} = p \% (x-2) = p(2) \]

\[ r_{3} = p \% (x-3) = p(3) \]

\[ r_{4} = p \% (x-4) = p(4) \]

\[ \pi_3 = (g^{q_{1,4}(\alpha)}, g^{q_{3,4}(\alpha)}, g^{q_3(\alpha)}) \]

\[ p(x) = q_{1,4}(x-1)(x-2)(x-3)(x-4) + q_{3,4}(x-3)(x-4) + q_3(x-3) + p(3) \]

\[ O(n \log^2 n) \]

\[ \rightarrow O(n \log n) \]
Recap

- DKG - generate shared SK and PK, requires each player to perform a VSS
- VSS - pick polynomial $p$ and send $p(i)$ to each player $i$, needs to compute proofs that $p(i)$ is valid using polynomial commitments
- Polynomial commitments - existing schemes like Kate take $O(nt)$ to compute all proofs, AMT provides all proofs in $O(n \log^2 n)$ time
- Result: Faster DKG that scales to tens of thousands of players.
Results

DKG per-player deal+verify times (or VSS total time)

- Feldman
- Kate et al
- AMT

Time (in seconds)

Total # of players

10.78 hours
18.52 minutes
Acknowledgements

I would like to thank:
● My mentor, Alin Tomescu, for his support and guidance
● Srini Devadas, for coordinating CS-PRIMES
● My parents and family
● MIT-PRIMES program
Thank you!

Questions?