Confidential Amounts

Commitment Schemes and Range Proofs

Kaspar Etter, November 2018

SHIFT CRYPTOSECURITY

Motivation

All amounts are visible to everyone in Bitcoin.

Increase privacy for individuals and businesses.

Solution: additively homomorphic commitments

Recap: Commitment Schemes

Commit yourself to a value so that others cannot learn your value before you reveal it to them later.

A commitment scheme has to be both:

- **Binding**: you can only reveal committed value
- **Hiding**: no one can determine committed value

Only one of these properties can be perfect, while the other is only computationally secure.

Example: Shared Coin Flip

Situation: Call with friend, cannot agree on a bar.

Flip a coin: Easy in person, difficult over phone.

Idea: Combine the outcome of separate coin flips.

Problem: Whoever tells first, risks being cheated.

Solution: Commit to your coin flip first, e.g. hash (flip, salt), and only reveal it after learning their flip.

Recap: Pedersen Commitment

Given group of order q with generators G and H.

Commit to secret s < q by choosing a random value r < q and computing **C = sG + rH**.

Scheme is **perfectly hiding** and computationally binding as long as no one know x so that H = xG.

Nothing-up-my-sleeve principle: H = Hash(G).

ElGamal commitment scheme: C = (sG + rH, rG); perfectly binding but only computationally hiding.

Linearity Property

$c(s_1) + c(s_2) = c(s_1 + s_2)$ with c(…) as commitment.

Pedersen commitments are linear:
$(s_1G + r_1H) + (s_2G + r_2H) = (s_1 + s_2)G + (r_1 + r_2)H$.

Instead of verifying that sum of all input amounts minus sum of all output amounts and transaction fee equals zero, all other nodes can verify that the same is true for the commitments of these values.

Make fee explicit and **prevent negative amounts**!

From Commitments to Signatures

Commitment **C = rG + aH** for amount a with role of G and H swapped, nobody knows the value c so that C = cG because nobody knows the value x so that H = xG.

If amount a is zero, commitment becomes **C = rG**, which is then a public key to the private key r.

If the amount is one, you can subtract 1H from the commitment to get the public key **C' = C – 1H**.

Range Proofs with Ring Signatures

Use ring signatures to prove that amount is 0 or 1 by signing with the private key of either C or C'.

If amount is neither 0 nor 1, H remains in both C and C', making it infeasible to produce signature.

Example: Prove that amount a is ≥ 0 and < 16.

Split commitment $C = C_1 + C_2 + C_3 + C_4$, sign as ($C_1$ or $C_1' = C_1 - 1H$) and (C_2 or $C_2' = C_2 - 2H$) and (C_3 or $C_3' = C_3 - 4H$) and (C_4 or $C_4' = C_4 - 8H$).
(C_{1-4} are commitments to zero or 1, 2, 4, 8 resp.)

Bullet Proofs

Bullet proofs are optimization of range proofs but they are outside the scope of this lecture, for now.