Modern ZK Crypto

Session 4: Circom 2

Vivek Bhupatiraju
Personae / 0xPARC

01/17/23

vivboop

- Graduated from MIT in Spring ‘22, CS major + math minor
- Cryptography researcher at Personae / 0xPARC
  - Client-side ZK proving
  - Investigating how ZK can change how we express ourselves online (more ownership of personal data, enabling more pseudonymous characters, avoiding misinformation)
- Started through CS-PRIMES at MIT in 2016! Applied cryptography research in the Devadas Lab
- I really like music & decor !

Today's session

- Simple ZK signature scheme
- Simple group signatures scheme
- Merkle trees to enable larger groups
- **snarkjs** compilation pipeline

Signatures & group signatures

- Normal signature
  - KeyGen → (sk, pk): selects a random **secret key** sk and corresponding **public key** pk
  - Sign(m, sk) → s: given a message m and secret key, outputs a **signature** s
  - Verify(m, s, pk) → 1/0: given a message m, a signature s, and a public key pk, verifies if signature is valid
- Should be basically impossible be able to generate valid signature without knowledge of secret key

Signatures & group signatures

- Group signature for group **G**
  - KeyGen → (sk_i, pk_i): selects a random set of **secret keys** sk_i and corresponding **public keys** pk_i for each member of group
  - GroupSign(m, sk_i, G) → : given a message m and secret key, outputs a **signature** s
  - GroupVerify(m, s, G) → 1/0: given a message m, a signature s, and the group **G**, verifies if the signature came from the group
- Bespoke cryptographic solutions are usually fairly complicated