27. Implement Ring Signatures with Schnorr for Anonymous Group Spending in JavaScript

Why: Schnorr's linear properties enable ring signatures, allowing spends from a group without revealing the signer. This hack provides deniability on BSTY, cool for anonymous collective funds in 2026, hiding individual contributions in veteran pools.

How to Implement: Mix real key with decoys; compute ring sig. Use noble-secp256k1.

```
javascript
const secp = require('@noble/secp256k1');


// Hack: Ring signature
function ringSign(message, realPrivKey, decoyPubKeys) {
    const realPubKey = secp.getPublicKey(realPrivKey);
    const ring = [realPubKey, ...decoyPubKeys];


    const e = secp.utils.sha256(message);  // Challenge
    const r = secp.utils.randomPrivateKey();  // Random
    const commitments = ring.map(pub => secp.getSharedSecret(r, pub));


    // Simplified ring (full impl needs Borromean or CLSAG for efficiency)
    const sig = secp.schnorr.sign(message, realPrivKey);
    return { sig, ring };  // Verify with aggregate check
```

```
}

const sig = ringSign('tx_hash', 'real_priv_hex', ['decoy1', 'decoy2']);
console.log(sig);
```

Analysis: Rings of size 10+ obscure signer with 1/n probability; Taproot integrates as single sig. BSTY's low fees make large rings viable, defeating ownership clustering in 2026 tools, enhancing group privacy.