Backdooring Cryptocurrencies

...

The Underhanded Crypto Contest Winners

What is the Underhanded Crypto Contest?

- It's like the Underhanded C Contest.
- Add hard-to-spot backdoors to crypto.
  - Protocols.
  - Implementations.
- It's important because:
  - Makes us better at spotting backdoors.
    - And, by extension, unintentional bugs.
  - Predicts future classes vulnerabilities.
- This is the third year.

2014

- Winner: Tiny AES implementation for IoT devices.
  - It magically becomes vulnerable when you implement IPSec with it.
  - Re-uses the same buffer for the key and CTR-mode nonce.
  - Uses `typedef unsigned char boolean` for the “re-run the key schedule parameter.”
  - Encrypt something that’s not a multiple of 16 bytes => Key schedule gets re-run.
  - But... the buffer contains the nonce, so everything from then on is decipherable.
- Runner up: Stern’s Zero-Knowledge Identification Protocol
  - Hamming weight function has an integer bug.
  - Makes tricking the prover a lot easier.

2015

- Mini contest held during DEF CON 23.
- Category 1: GnuPG
  - GnuPG re-randomizes the top 32 bits of DSA nonces.
  - The patch makes it look like it's clearing the nonce when it's no longer needed.
  - But... now the nonce is all zero except those top 32 bits.
- Category 2: Password Hashing
  - User-enumeration side-channel defense.
  - Check the password against a hash of a random string.
  - But... the random string comes from `rand()`.

2016

- Cryptocurrency theme.
- Only one entry, but it's good!
- Thank you to our sponsors for the prize money:

Gold: 

Silver: 

- Thank you to our judges:
  - JP Aumasson
  - Gustavo Banegas
  - Eric Wustrow

Winner: Jonas Nick
https://nickler.ninja

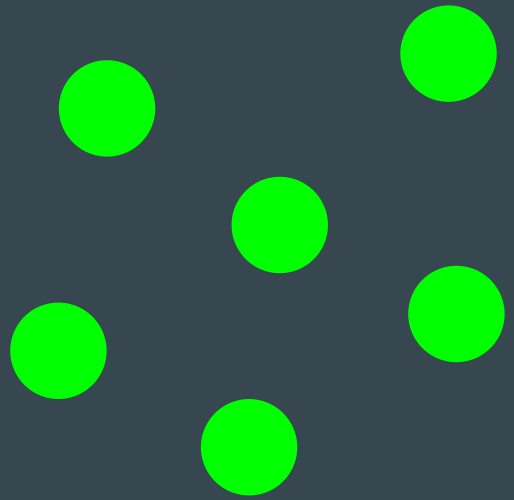
The Winner

- Here it is:

3LEETmEZWJX9ULbsFQVgL2QgGCJHPZJVaJ

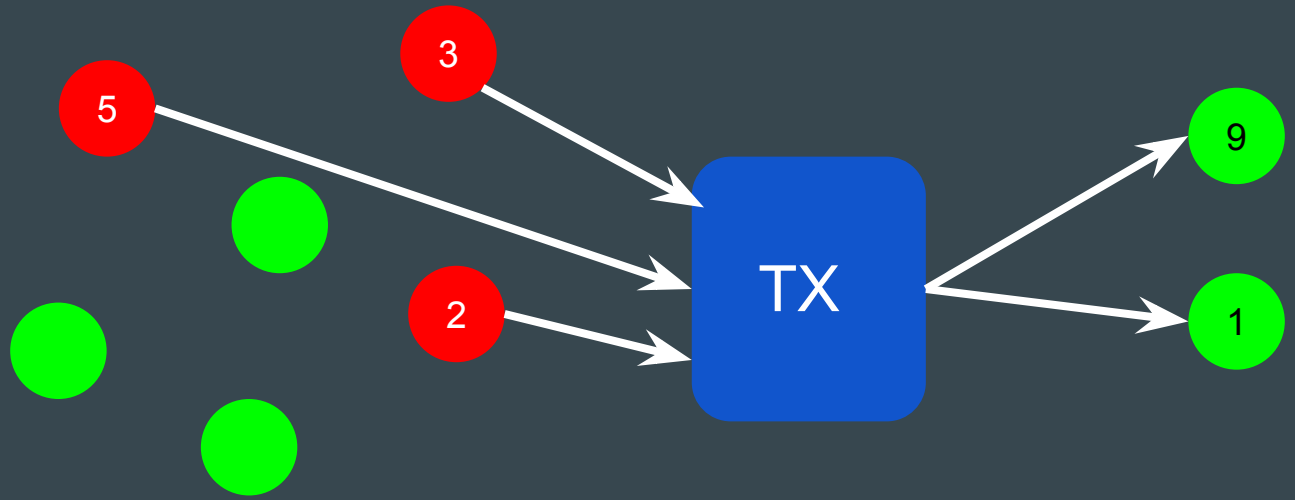
- It’s a Bitcoin address.
- Money sent there is “destroyed,” … *but not really.*
- Burn addresses are useful for...
  - Counterparty
  - Proof-of-Burn

Bitcoin 101



Unspent Transaction Output (UTXO) Set

Bitcoin 101

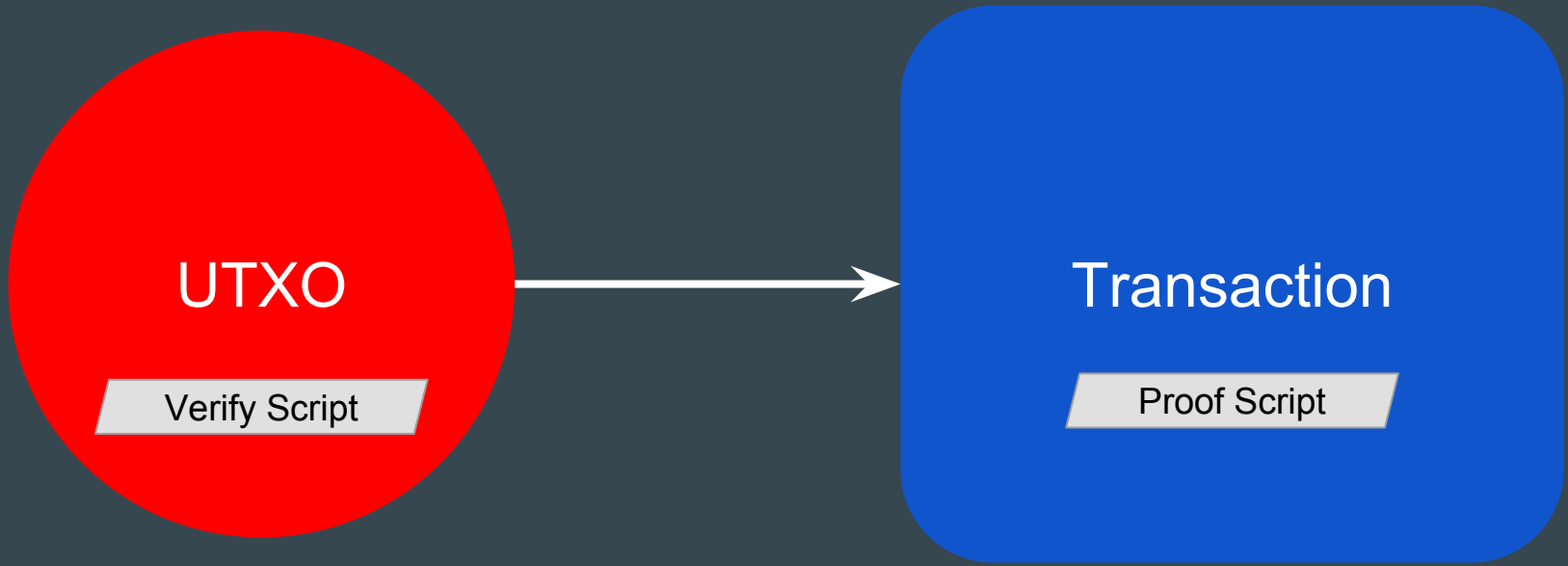


Unspent Transaction Output (UTXO) Set

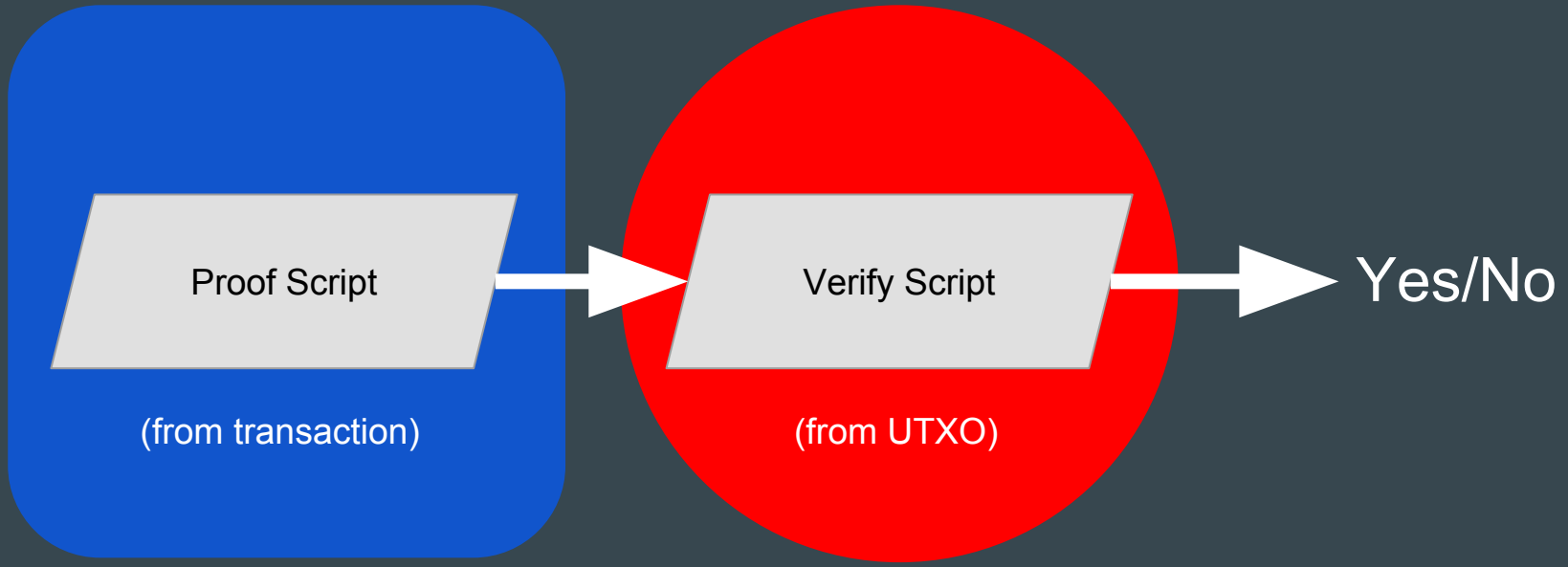
Bitcoin 101

Unspent Transaction Output (UTXO) Set

Bitcoin 101



Bitcoin 101


Proof Script
(from transaction)
Verify Script
(from UTXO)
Yes/No

Winner: Jonas Nick
https://nickler.ninja

Why it's a Burn Address

To spend coins sent to 3LEETmEZWJX9ULbsFQVgL2QgGCJHPZJVaJ you must satisfy this Verify Script:

```
304402203c5288058306b3bf5cd8202413b867...   (signature)
2
023b439207c8a0a082a5c5a968632be9a363f5...   (public key 1)
0316b0dbf710b8739eec21a806e7142db1755a...   (public key 2)
2
OP_CHECKMULTISIG
```

Winner: Jonas Nick
https://nickler.ninja

Why it's a Burn Address

To spend coins sent to 3LEETmEZWJX9ULbsFQVgL2QgGCJHPZJVaJ the transaction must satisfy this Verify Script:

1. The transaction's signature (by private key 1) must be 304402203c528...
2. You must provide a signature of the transaction (by private key 2).

Winner: Jonas Nick
https://nickler.ninja

How to Spend Burnt Coins

- Problem: Burn address creator has to predict the spending transaction's hash.
- Not possible, but there's a convenient bug in bitcoin:

```
uint256 SignatureHash(const CScript& scriptCode, const CTransaction& txTo, unsigned int nIn, int nHashType)
{
    static const uint256 one(uint256S("0000000000000000000000000000000000000000000000000000000000000001"));
    if (nIn >= txTo.vin.size()) {
        //  nIn out of range
        return one;
    }

    // Check for invalid use of SIGHASH_SINGLE
    if ((nHashType & 0x1f) == SIGHASH_SINGLE) {
        if (nIn >= txTo.vout.size()) {
            //  nOut out of range
            return one;
        }
    }
```

- Yes, that 304402203c5288058306b3bf... is just a signature of 1.

Winner: Jonas Nick
https://nickler.ninja

How to Spend Burnt Coins

1. Create a transaction to spending the burnt UTXO.
2. Make it trigger the SIGHASH_SINGLE bug, so its hash is 1.
3. Sign it with private key 2.

It's NOBUS!

Winner: Jonas Nick
https://nickler.ninja

- "Nobody But Us" can exploit the backdoor.

1. The transaction's signature (by private key 1) must be 304402203c528...
2. You must provide a signature of the transaction (by private key 2).

Only the burn address creator knows that private key!

Drawing Conclusions

- Specific: In-band signalling is bad!
- General: Baked-in bugs increase mental burden.
  - Maybe fork more often to fix consensus bugs.
  - We were convinced by reasoning at the protocol level, but an easy-to-forget implementation detail brought our logic crashing down.

The Future of Underhanded Crypto

- There will be a 2017 contest.
- Broader scope, more like the 2014 one.
- Feedback: contact@underhandedcrypto.com

Thank you!

nccgroup

freedom from doubt