Zerocash

Decentralized Anonymous Payments from Bitcoin

Oakland 2014

Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, Madars Virza

Presented by Gengmo Qi and Tianpeng Zhang

9th/Nov/2017

Agenda

- Background

- 3 initial attempts to construct a basic anonymous e-cash
- 3 attempts to extend its functionalities

- What’s happening now?

Motivation

Bitcoin's privacy problem

- Recall: How does Bitcoin prevent double-spending?
- Solution: broadcast every transaction into a public ledger (*blockchain*)

The cost: privacy.

- Purchase history (timing, amounts, merchant) seen by friends etc.
- Account balance revealed in every transaction.
- Merchant's cash flow exposed to competitors.

Motivation

Bitcoin's privacy problem

- Pseudonymous, but:

– Most users use a single or few addresses

– Transaction graph can be analyzed.

- Also: threat to the currency's fungibility.

"a dollar is a dollar, regardless of its history"

- Centralized: reveal to the bank.
- Decentralized: reveal to everyone???!

Previous attempts at Bitcoin anonymity

- Trusted mix (but: operator can trace/steal)
- Zerocoin: decentralized mix service for Bitcoin
  Limitations:
  - Performance: 45 kB/spend, ~0.5 s to verify. (for 128-bit security)
  - Single denomination (undivisible) ⇒ reveals amount
  - Reveal payment destinations; no direct transfer
  - Requires explicit “laundry” process.
- CoinJoin and others

- Goal: fully privacy-preserving
  - Anyone can post a transaction to anyone else, while provably hiding the payment
    (1) Sender (2)Receiver (3)Amount

Let’s try to design an anonymous coin from scratch

- Coin

  `sn` Serial number

  `A365e7006565f14342df9096b46cc7f1d2b9949367180fdd8de4090eee30bfdc`

- Minting
  - I hereby consume 1 BTC to create value-1 coin with serial number `sn`
- Spending
  - Consume the coin with serial number `sn`

Attempt #1: plain serial numbers

- Minting: I hereby spend 1 BTC to create value-1 coin `sn`
- Spending: I am using up a coin with unique `sn`

`sn`

Serial Number

`sn1`

`sn2`

`sn3`

`sn4`

`sn5`

……

Legend:

On public chain

Attempt #1: plain serial numbers

- Minting: I hereby spend 1 BTC to create value-1 coin sn
- Spending: ~~I am using up a coin with unique sn~~

Legend:

On public chain

sn

Serial Number

sn1

sn2

sn3

sn4

sn5

......

Prevents double spending? ✅

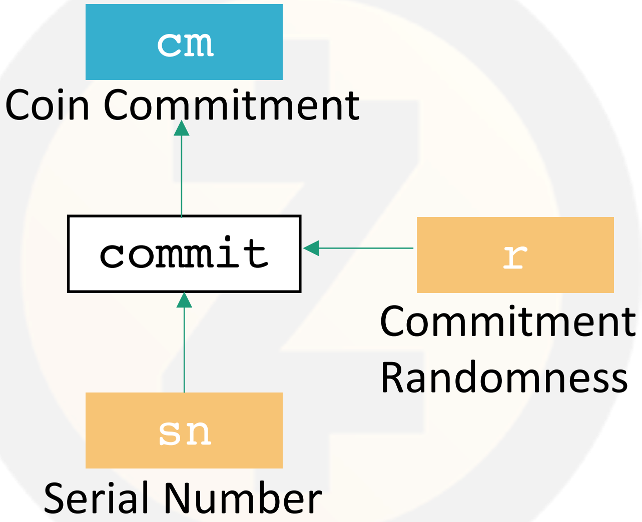
Prevents false spending? ❌

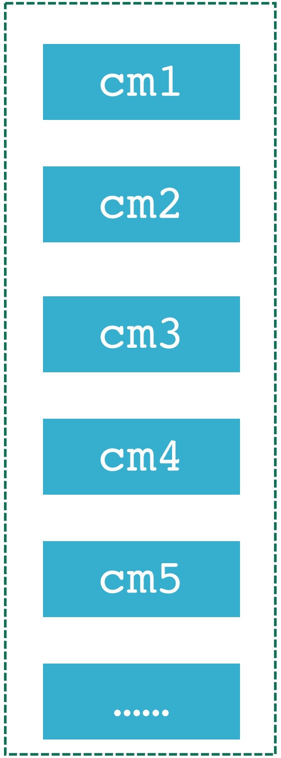
Mint & Spend unlinkable? ❌

Attempt #2: committed serial numbers

[Sander Ta-Shma 1999]

- Minting: I hereby spend 1 BTC to create value-1 coin *cm*





Legend:

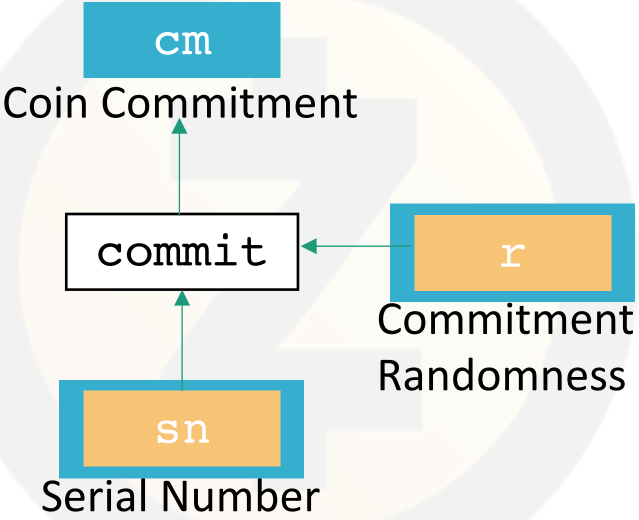
On public chain

In private wallet

Attempt #2: committed serial numbers

[Sander Ta-Shma 1999]

- Minting: I hereby spend 1 BTC to create value-1 coin *cm*
- Spending: I am using up a coin with *cm*, and here are its corresponding *sn* and *r*
  (sn,r)



Legend:

On public chain

In private wallet

cm1

cm2

cm3

cm4

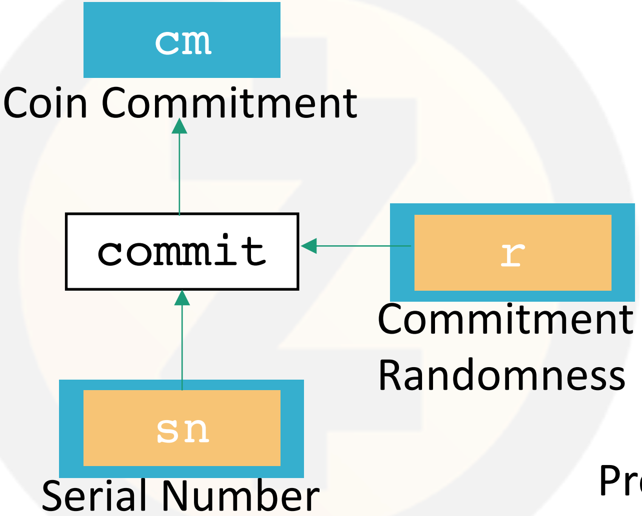
cm5

……

Attempt #2: committed serial numbers

[Sander Ta-Shma 1999]

- Minting: I hereby spend 1 BTC to create value-1 coin *cm*
- Spending: ~~I am using up a coin with *cm*, and here are its corresponding *sn* and *r*~~
  (sn,r)



Legend:

On public chain

In private wallet

cm1

cm2

cm3

cm4

cm5

……

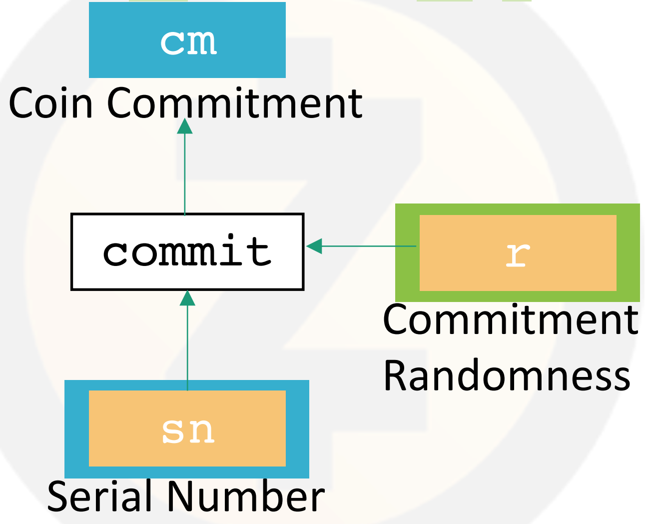
Prevents double spending? ✅

Prevents false spending? ✅

Mint & Spend unlinkable? ❌

Attempt #3: Zero Knowledge Proof of Commitment

- Minting: I hereby spend 1 BTC to create value-1 coin *cm*
- Spending: (sn,π) I am using up a coin with unique *sn*,
  I know *r* such that (1)a *cm* is in 'list of prior commitments'
  (2)*cm* = COMM(*sn*,*r*)



CMList

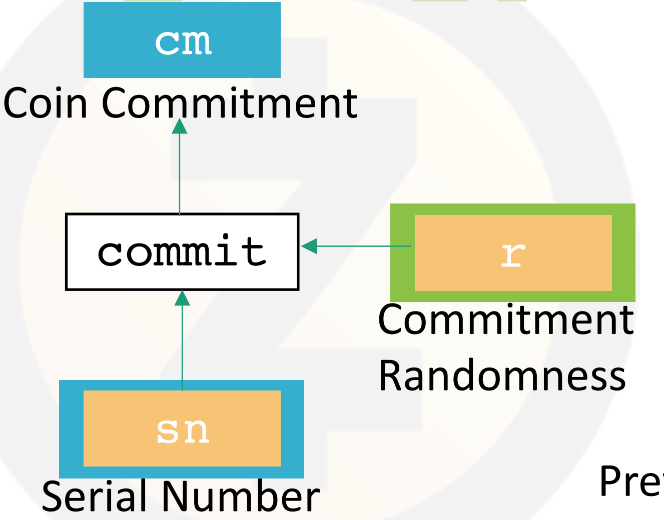
- cm1
- cm2
- cm3
- cm4
- cm5
- ……

Legend:

- On public chain
- In private wallet
- Prove to be known

Attempt #3: Zero Knowledge Proof of Commitment

- Minting: I hereby spend 1 BTC to create value-1 coin *cm*
- Spending: (sn, π) I am using up a coin with unique *sn*,
  I know *r* such that (1)a *cm* is in ‘list of prior commitments’ (2) *cm* = COMM(*sn*, *r*)



CMList

- cm1
- cm2
- cm3
- cm4
- cm5
- ……

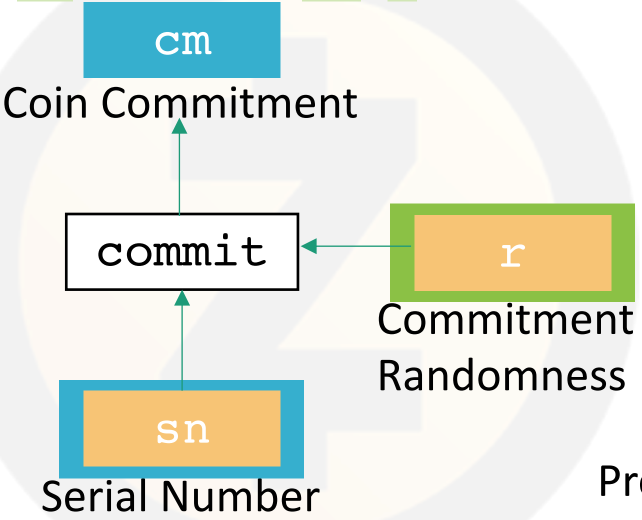
Legend:

- On public chain
- In private wallet
- Prove to be known

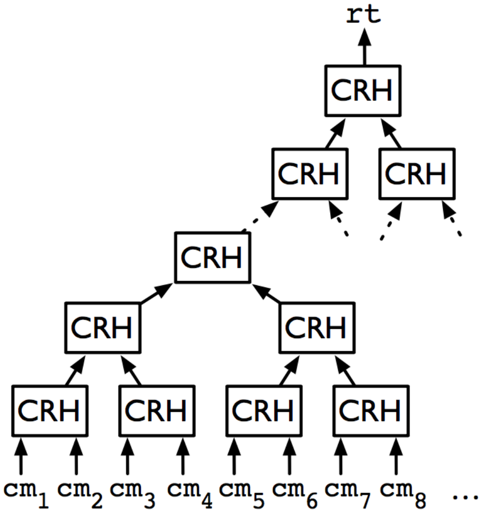
Prevents double spending? ✅
Prevents false spending? ✅
Mint & Spend unlinkable? ✅

Attempt #3: Zero Knowledge Proof of Commitment_v2

- Minting: I hereby spend 1 BTC to create value-1 coin *cm*
- Spending: I am using up a coin with unique *sn*,
  (sn,π) **I know *r* such that (1)a *cm* is in tree with *rt***
  (2)*cm* = COMM(*sn*,*r*)



(a) Merke tree over (cm_1, cm_2, ...)



Legend:

- On public chain
- In private wallet
- Prove to be known

Prevents double spending? ✅
Prevents false spending? ✅
Mint & Spend unlinkable? ✅

In proofs we trust

- Intuition: “virtual accountant/notary/witness” using cryptographic proofs.

- Desired proof properties:
  - *zero-knowledge*
  - *Succinct*
  - *Non-interactive*
  - *ARguments of Knowledge*

I am using up a coin with unique $\underline{\texttt{sn}}$,
I know $\underline{\texttt{r}}$ such that (1)a $\underline{\texttt{cm}}$ is in tree with $\underline{\texttt{rt}}$
(2)$\underline{\texttt{cm}}$ `=` `COMM(`$\underline{\texttt{sn}}$`,`$\underline{\texttt{r}}$`)`

zk-SNARKs(blackbox)

- *zero-knowledge, Succinct, Non-interactive ARguments of Knowledge*

"API":

$\texttt{Setup}(\mathit{stmt})$

$\pi \leftarrow \texttt{Prove}(\mathit{input})$

$\texttt{Verify}(\pi)$

→ libsnark

Quick recap before we proceed

In our best attempt so far:

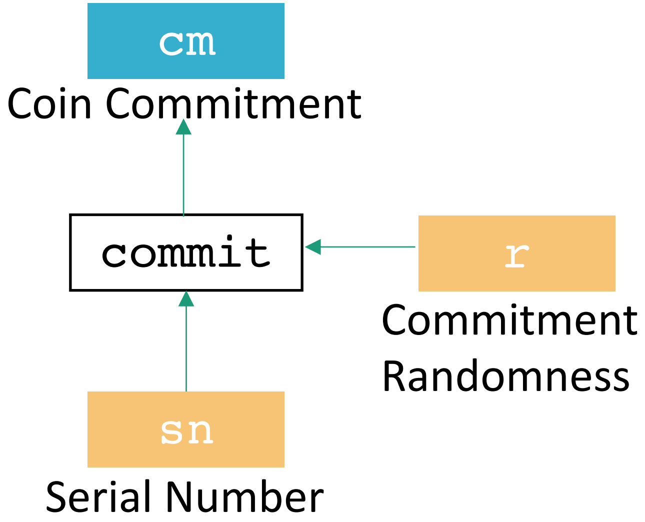
- (i)How did we create new coins? How to spend them?

(1)Mint

`Mint(cm)`

(2)Spend

`Spend(sn,`π`)`



Quick recap before we proceed

In our best attempt so far:

- (ii)Security:

How to prevent double spending? False spending?

(1)Double spending:

Serial Number **`sn`**

(2)False spending:

Knowing my **`cm`**, can't derive **`sn`** and **`r`**

Quick recap before we proceed

In our best attempt so far:

- (iii)How privacy is protected?
  - Unlink **`Spend`** and **`Mint`**.
  - Whenever I see a **`Spend`** transaction, I don't know which previous **`Mint`** transaction it corresponds to

Quick recap before we proceed

In our best attempt so far:

- (iv)What is zk-SNARK trying to convey? `Spend(sn,π)`

- **I know a secret randomness *r* such that** (won't tell you the value of ***r***)

- **(1) a *cm* is in tree with *rt***

it induces a commitment ***cm*** that belongs to the collection of all commitments that appear on the chain so far. (It is one of them, but I won't tell you which one is it)

- **(2) *cm* = COMM(*sn*,*r*)**

Moreover, the serial number ***sn*** I reveal, and the secret randomness ***r*** I am not telling you, give rise to that particular coin commitment ***cm***.

Attempt #4: variable denomination

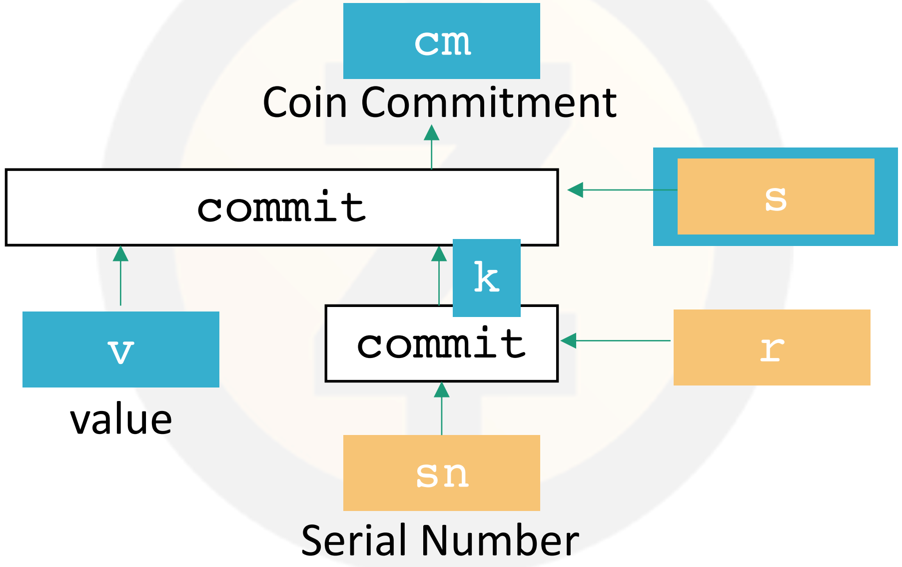
- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency

`Mint(cm,v,k,s)`

Legend:

On public chain

In private wallet



Attempt #4: variable denomination

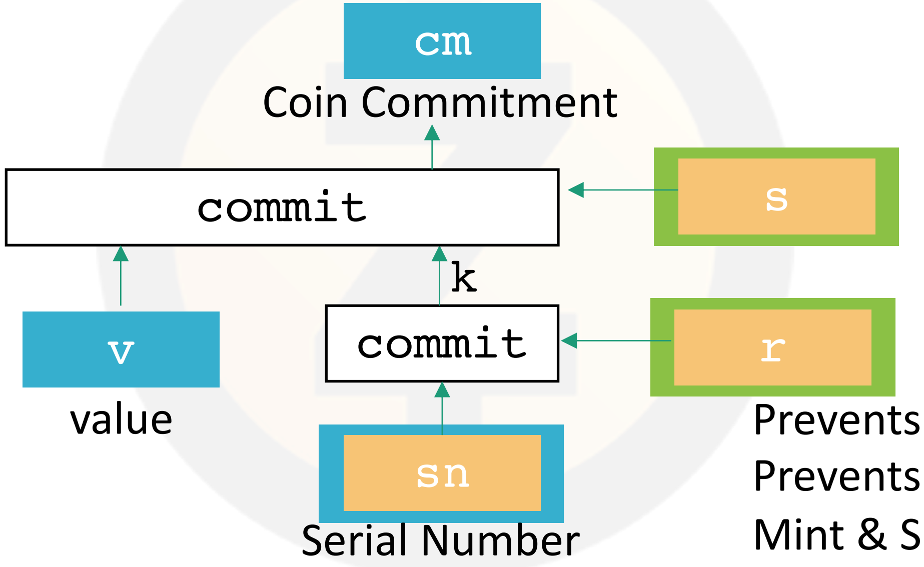
- Minting: I hereby spend *v* BTC to create value-1 coin *cm*,
  And here is *k*, *s* to prove consistency

`Mint(cm,v,k,s)`

Mint & Spend unlinkable?

- Spending: I am using up a coin with value *v* (unique) *sn*,
  I know some secret randomness *r*, *s* such that
  existence **(1)a *cm* is in the collection of all previous commitments**
  well-formed **(2) *cm* = COMM(*v*,*k*,*s*) && *k* = COMM(*sn*,*r*)**

`Spend(sn,` π`,v)`



Legend:

On public chain

In private wallet

Prove to be known

Prevents double spending? ✅

Prevents false spending? ✅

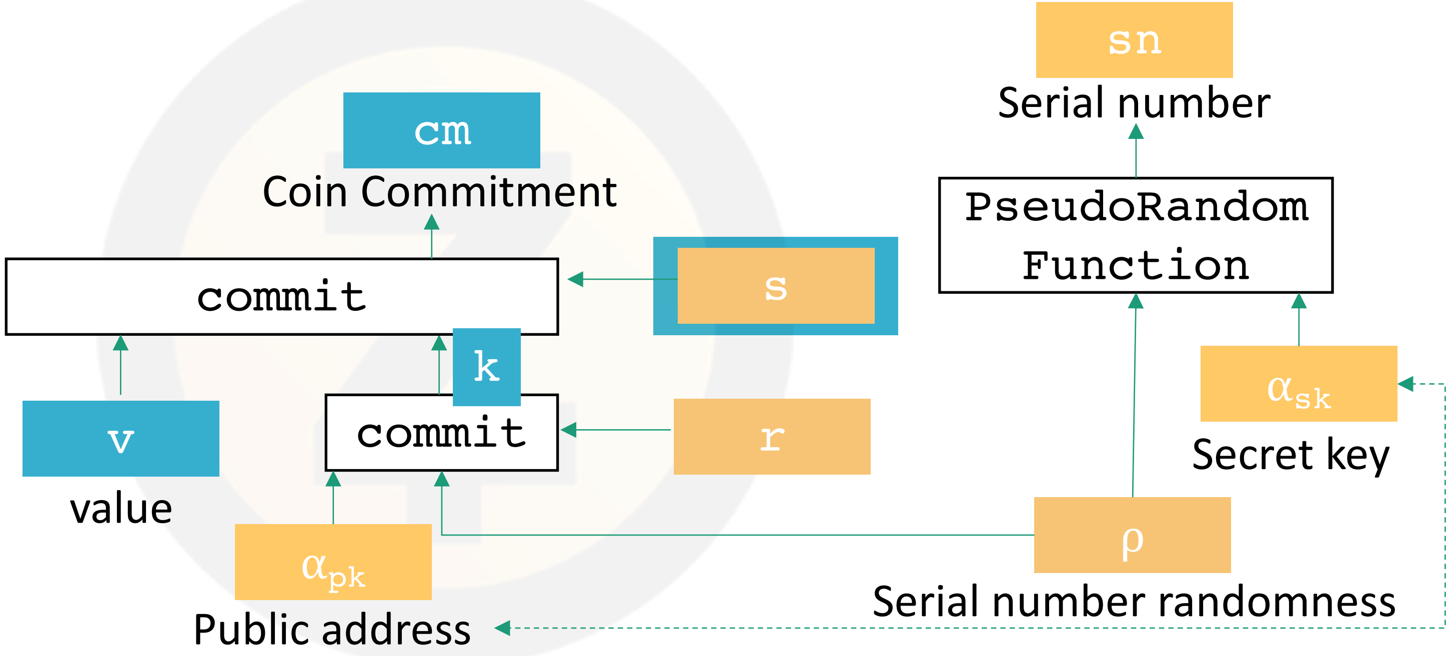
Mint & Spend unlinkable? ❓

Attempt #5: anonymous sender address

- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency

`Mint(cm,v,k,s)`

`Spend(sn,π,v)`



Legend:

On public chain

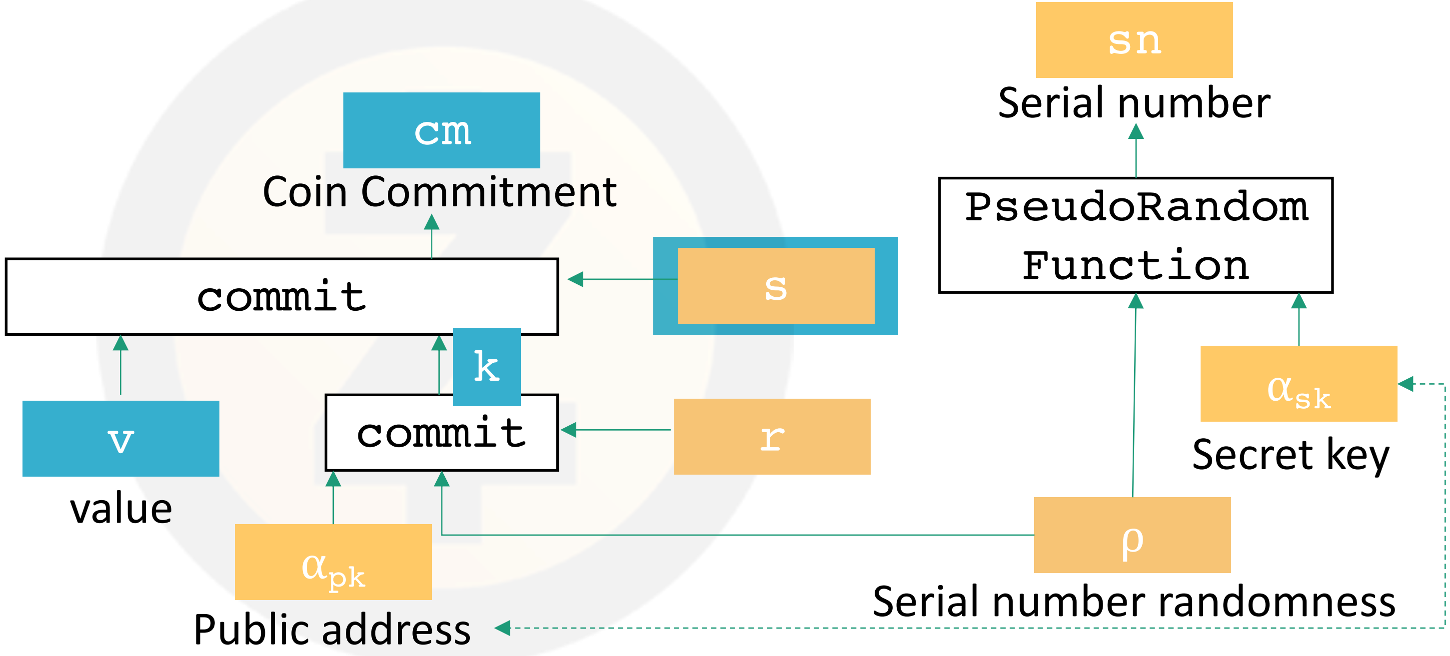
In private wallet

Attempt #5: anonymous sender address

- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency

`Mint(cm,v,k,s)`

`Spend(sn,π,v)`



Legend:

On public chain

In private wallet

Attempt #5: anonymous sender address

- Minting: I hereby spend *v* BTC to create value-1 coin *cm*, And here is *k*, *s* to prove consistency

`Mint(cm,v,k,s)`

- Spending: I am using up a coin with value *v* (unique) *sn*, **I know secret(k, r, s, ρ, α_{pk}, α_{sk}) that match secret *cm*:**

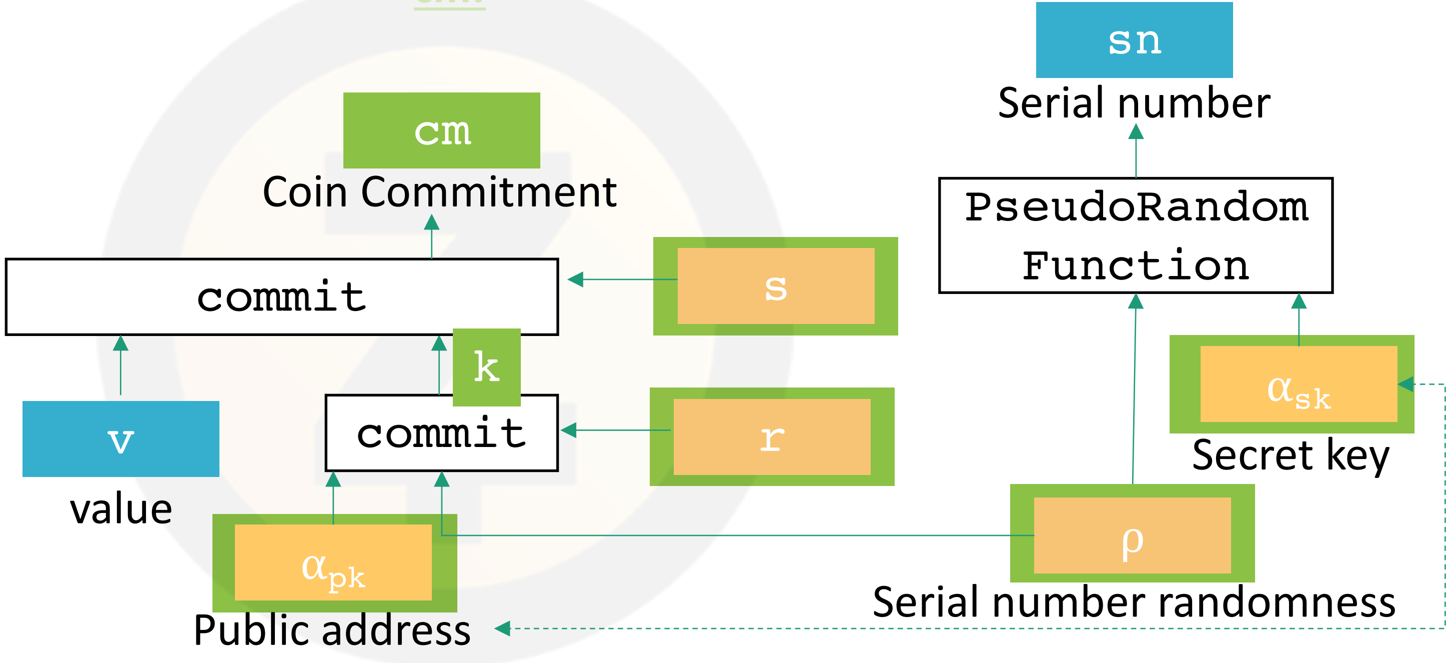
`Spend(sn,` π`,v)`

Legend:

On public chain

In private wallet

Prove to be known



Attempt #5: anonymous sender address

- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency

Mint(cm,v,k,s)

- Spending: I am using up a coin with value $\underline{v}$ (unique) $\underline{sn}$,
  I know secret($\underline{k}$, $\underline{r}$, $\underline{s}$, $\underline{\rho}$, $\underline{\alpha_{pk}}$, $\underline{\alpha_{sk}}$) that match secret $\underline{cm}$:

existence **(1) a $\underline{cm}$ is in the collection of all previous commitments**

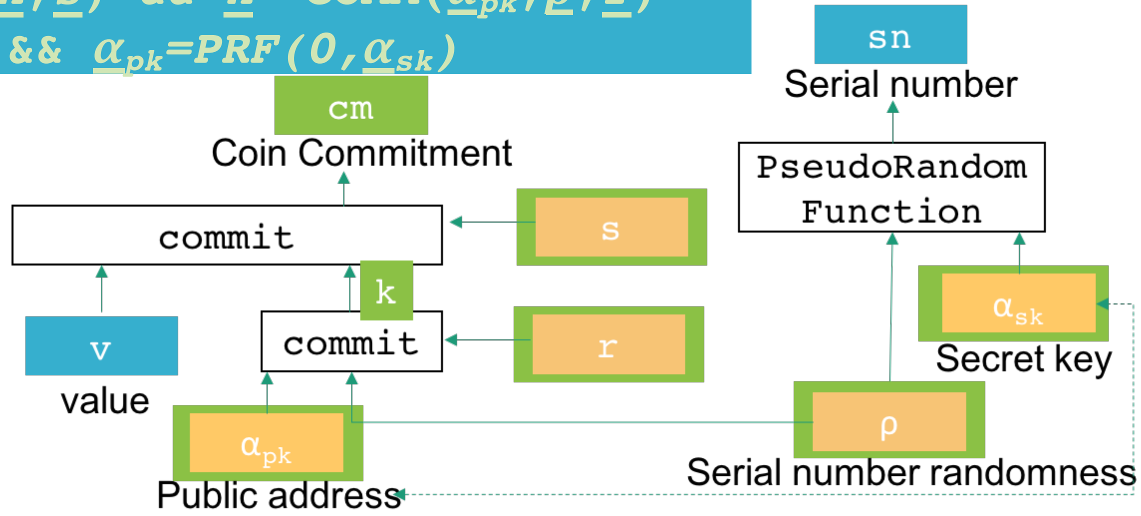
well-formed **(2) $\underline{cm}$ = COMM($\underline{v}$, $\underline{k}$, $\underline{s}$) && $\underline{k}$ = COMM($\underline{\alpha_{pk}}$, $\underline{\rho}$, $\underline{r}$)**

possession **(3) $\underline{sn}$ = PRF($\underline{\rho}$, $\underline{\alpha_{sk}}$) && $\underline{\alpha_{pk}}$ = PRF(0, $\underline{\alpha_{sk}}$)**

Spend(sn, π, v)

Legend:

- On public chain
- In private wallet
- Prove to be known



Attempt #5: anonymous sender address

- Minting:

I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$,
And here is $\underline{k}$, $\underline{s}$ to prove consistency

Mint(cm, v, k, s)

- Spending:

I am using up a coin with value $\underline{v}$ (unique) $\underline{sn}$,
I know secret($\underline{k}$, $\underline{r}$, $\underline{s}$, $\underline{\rho}$, $\underline{\alpha_{pk}}$, $\underline{\alpha_{sk}}$) that match secret $\underline{cm}$:

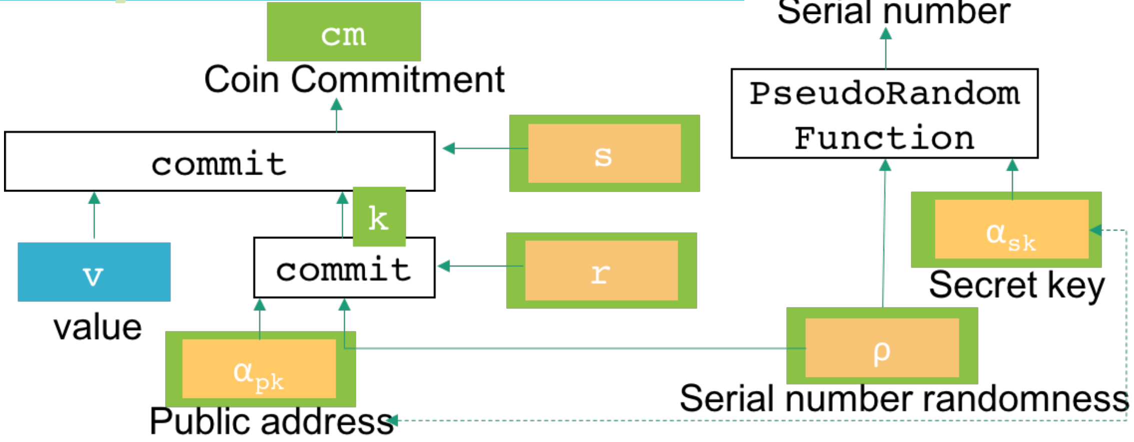
existence **(1)a $\underline{cm}$ is in the collection of all previous commitments**

well-formed **(2)$\underline{cm}$ = COMM($\underline{v}$, $\underline{k}$, $\underline{s}$) && $\underline{k}$ = COMM($\underline{\alpha_{pk}}$, $\underline{\rho}$, $\underline{r}$)**

possession **(3)$\underline{sn}$ = PRF($\underline{\rho}$, $\underline{\alpha_{sk}}$) && $\underline{\alpha_{pk}}$=PRF(0, $\underline{\alpha_{sk}}$)**

Spend(sn, π, v)

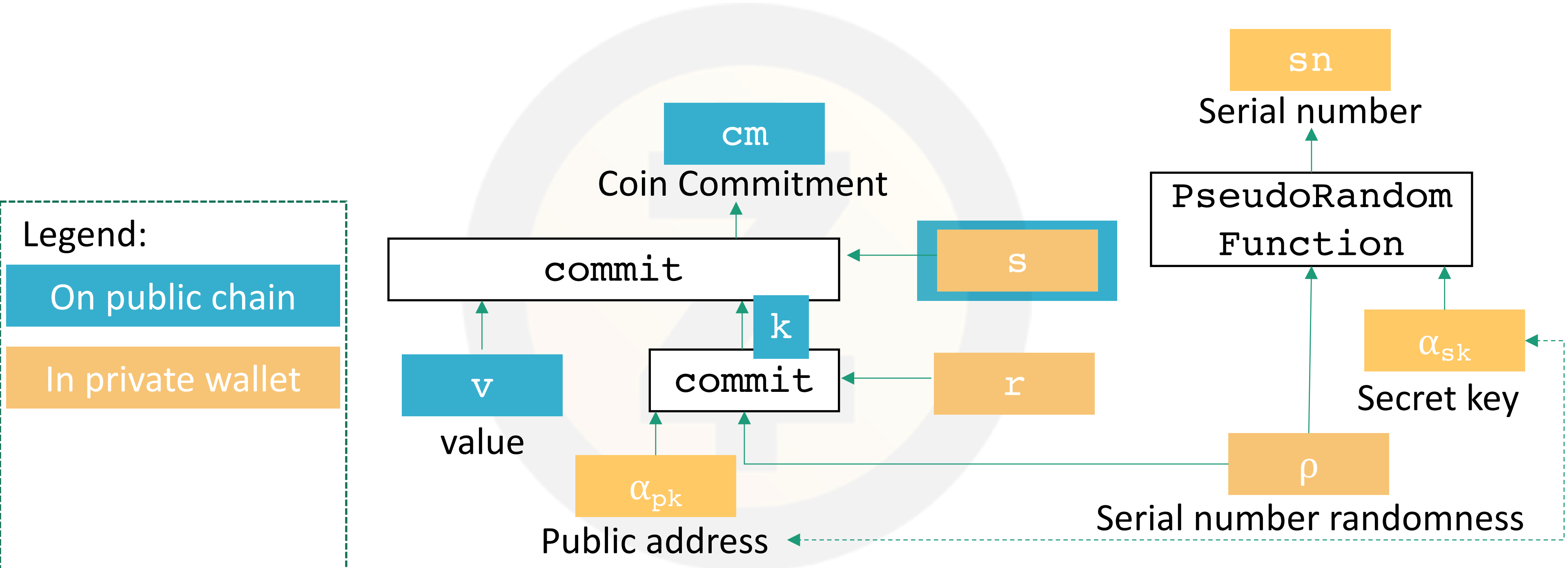
Prevents double spending? ✅
Prevents false spending? ✅
Variable denomination? ✅
Mint & Spend unlinkable? ❓
Other problems?



Attempt #6: sending direct payments

- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency

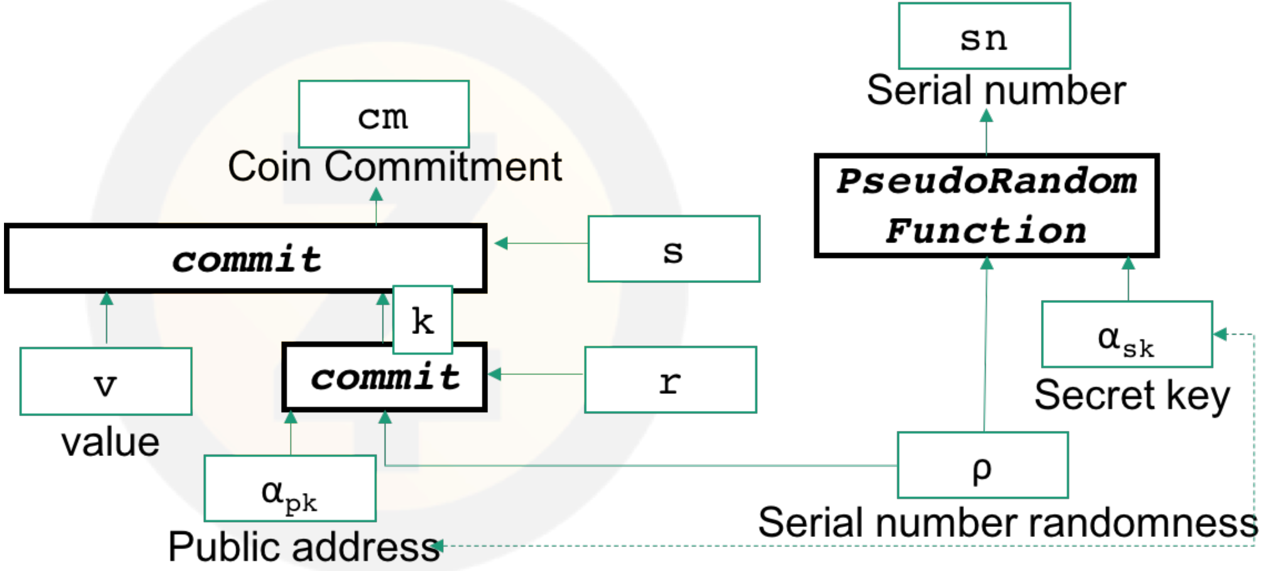
`Mint(cm,v,k,s)`



Attempt #6: sending direct payments

- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency
- Spending: Burn coin $\underline{sn^A}$ & create new coin with commitment cm^B

`Mint(cm,v,k,s)`
`Spend(sn^A,cm^B,π)`

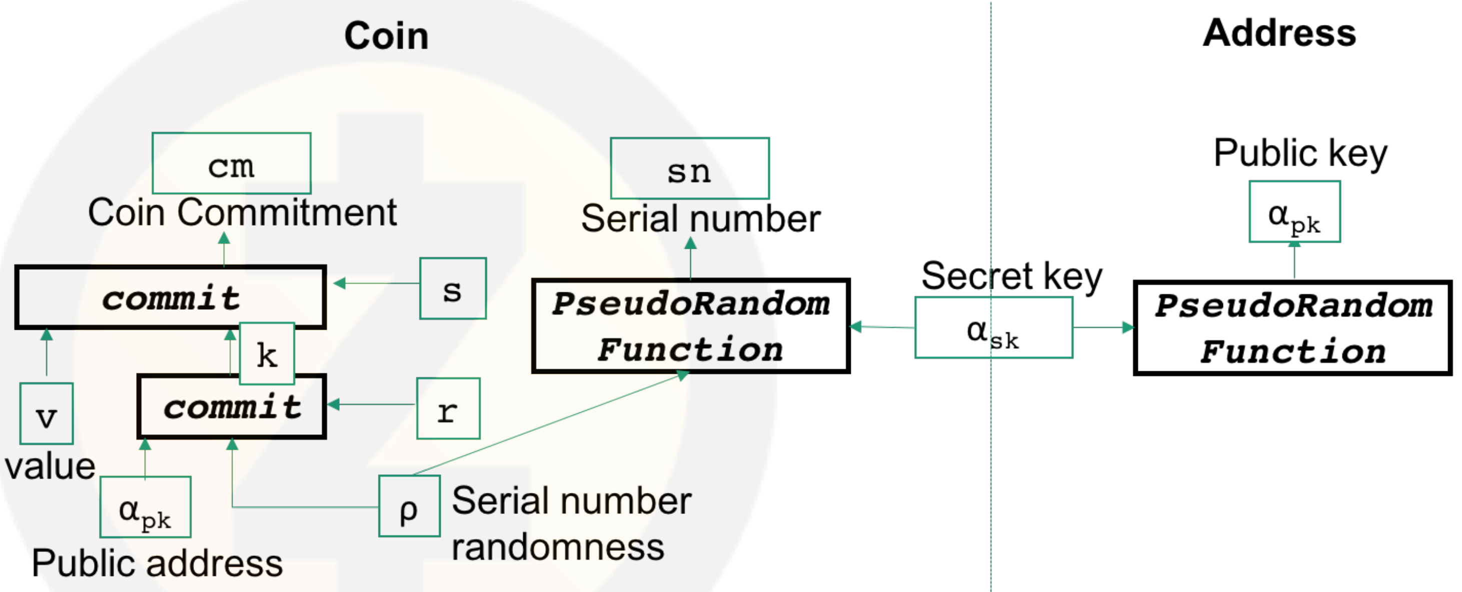


Attempt #6: sending direct payments

- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency

$$\mathtt{Mint(cm,v,k,s)}$$
$$\mathtt{Spend(sn^{A},cm^{B},}\pi\mathtt{)}$$

- Spending: Burn coin $\underline{sn^{A}}$ & create new coin with commitment cm^{B}



Attempt #6: sending direct payments

- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency

Mint(cm,v,k,s)
Spend(sn^A,cm^B,π)

- Spending: Burn coin $\underline{sn^A}$ & create new coin with commitment cm^B

I know secret(cm^A, v^A, k^A, r^A, s^A , ρ^A, $\alpha_{pk}{}^A$, $\alpha_{sk}{}^A$) (cm^B, v^B, k^B, r^B, s^B , ρ^B, $\alpha_{pk}{}^B$)

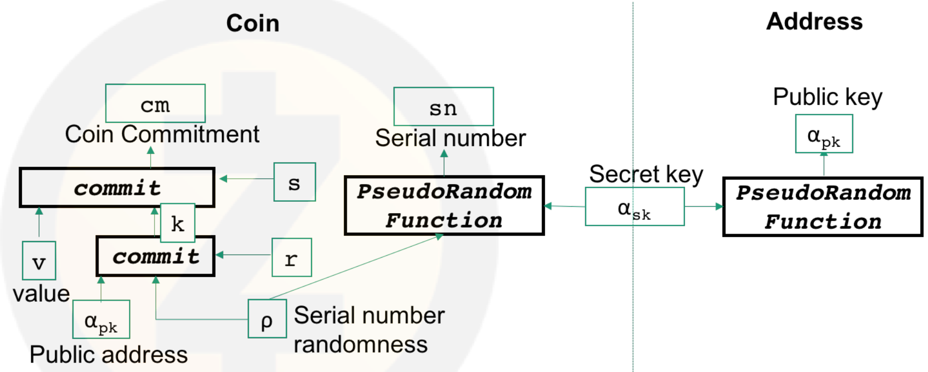
existence (1)a cm^A is in the collection of all previous commitments

well-formed (2)cm^A = COMM(v^A, k^A, s^A) && k^A = COMM($\alpha_{pk}{}^A$, ρ^A , r^A)

possession (3)sn^A = PRF(ρ^A , $\alpha_{sk}{}^A$) && $\alpha_{pk}{}^A$=PRF(0 , $\alpha_{sk}{}^A$)

well-formed (4)cm^B = COMM(v^B, k^B, s^B) && k^B = COMM($\alpha_{pk}{}^B$, ρ^B , r^B)

same value (5)v^A=v^B



Attempt #6: sending direct payments

- Minting: I hereby spend $\underline{v}$ BTC to create value-1 coin $\underline{cm}$, And here is $\underline{k}$, $\underline{s}$ to prove consistency

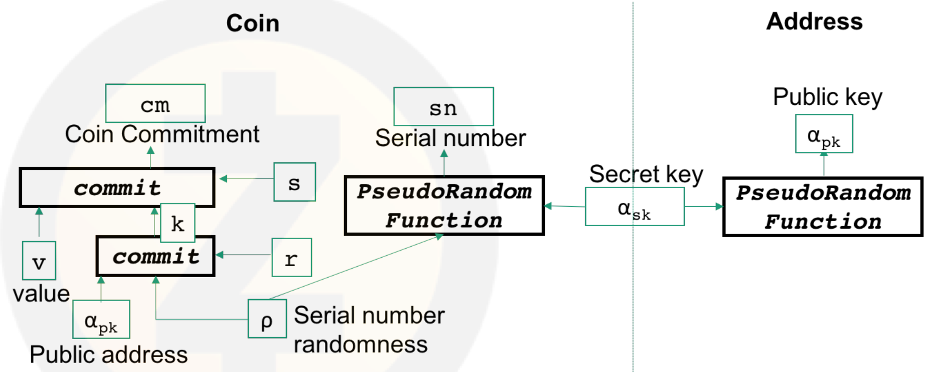
```
Mint(cm,v,k,s)
Spend(snᴬ,cmᴮ,π)
```

- Spending: Burn coin $\underline{sn^A}$ & create new coin with commitment cm^B

I know secret(cm^A, v^A, k^A, r^A, s^A, ρ^A, $\alpha_{pk}{}^A$, $\alpha_{sk}{}^A$) $\boxed{(cm^B, v^B, k^B, r^B, s^B, \rho^B, \alpha_{pk}{}^B)}$

existence	(1)a cm^A is in the collection of all previous commitments
well-formed	(2)cm^A = COMM(v^A, k^A, s^A) && k^A = COMM($\alpha_{pk}{}^A$, ρ^A, r^A)
possession	(3)sn^A = PRF(ρ^A, $\alpha_{sk}{}^A$) && $\alpha_{pk}{}^A$=PRF(0, $\alpha_{sk}{}^A$)
well-formed	(4)cm^B = COMM(v^B, k^B, s^B) && k^B = COMM($\alpha_{pk}{}^B$, ρ^B, r^B)
same value	(5)v^A=v^B

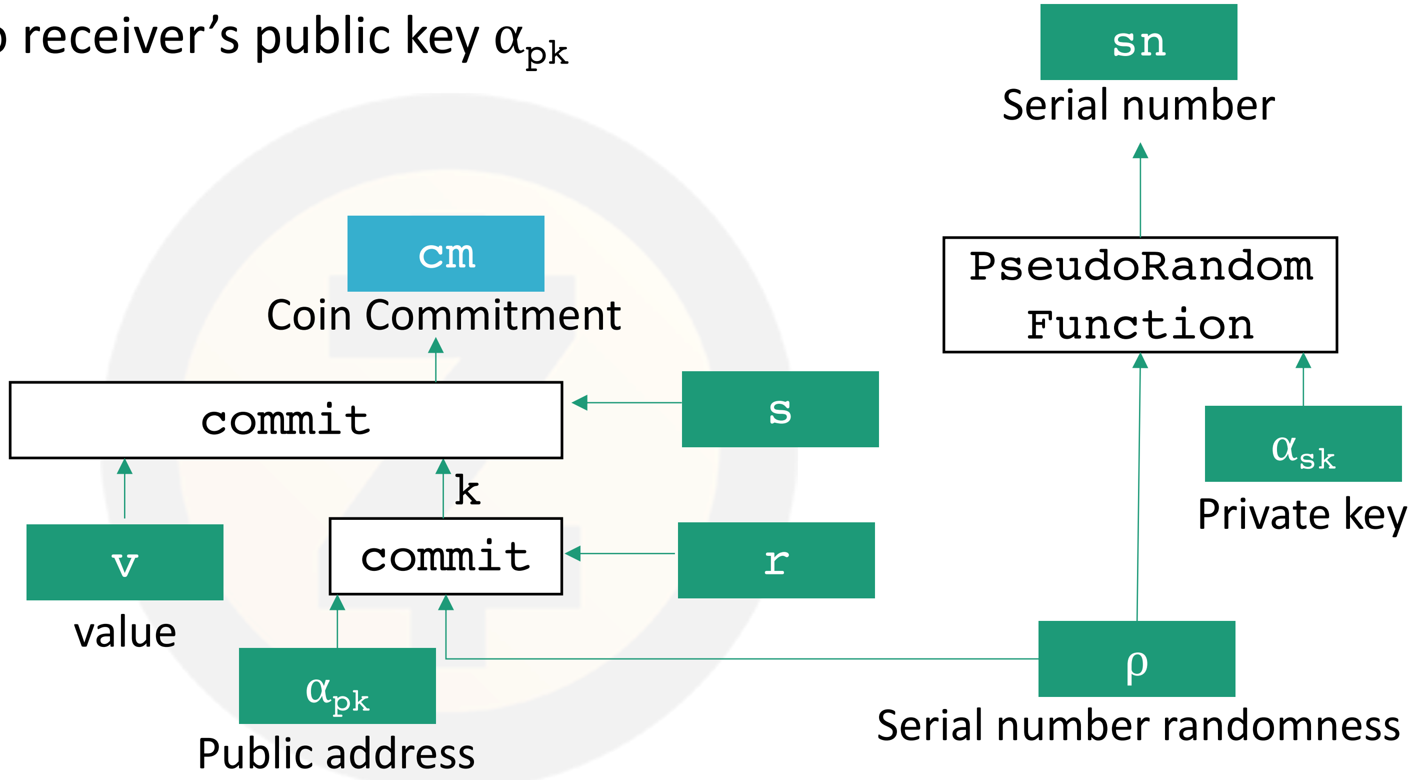
Prevents double spending? ✅
Prevents false spending? ✅
Variable denomination? ✅
Mint & Spend unlinkable? ✅

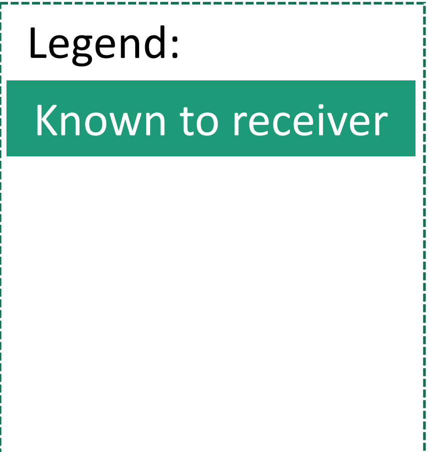


Attempt #6: sending direct payments

Sender send coin secrets(v,ρ,r,s) to receiver:

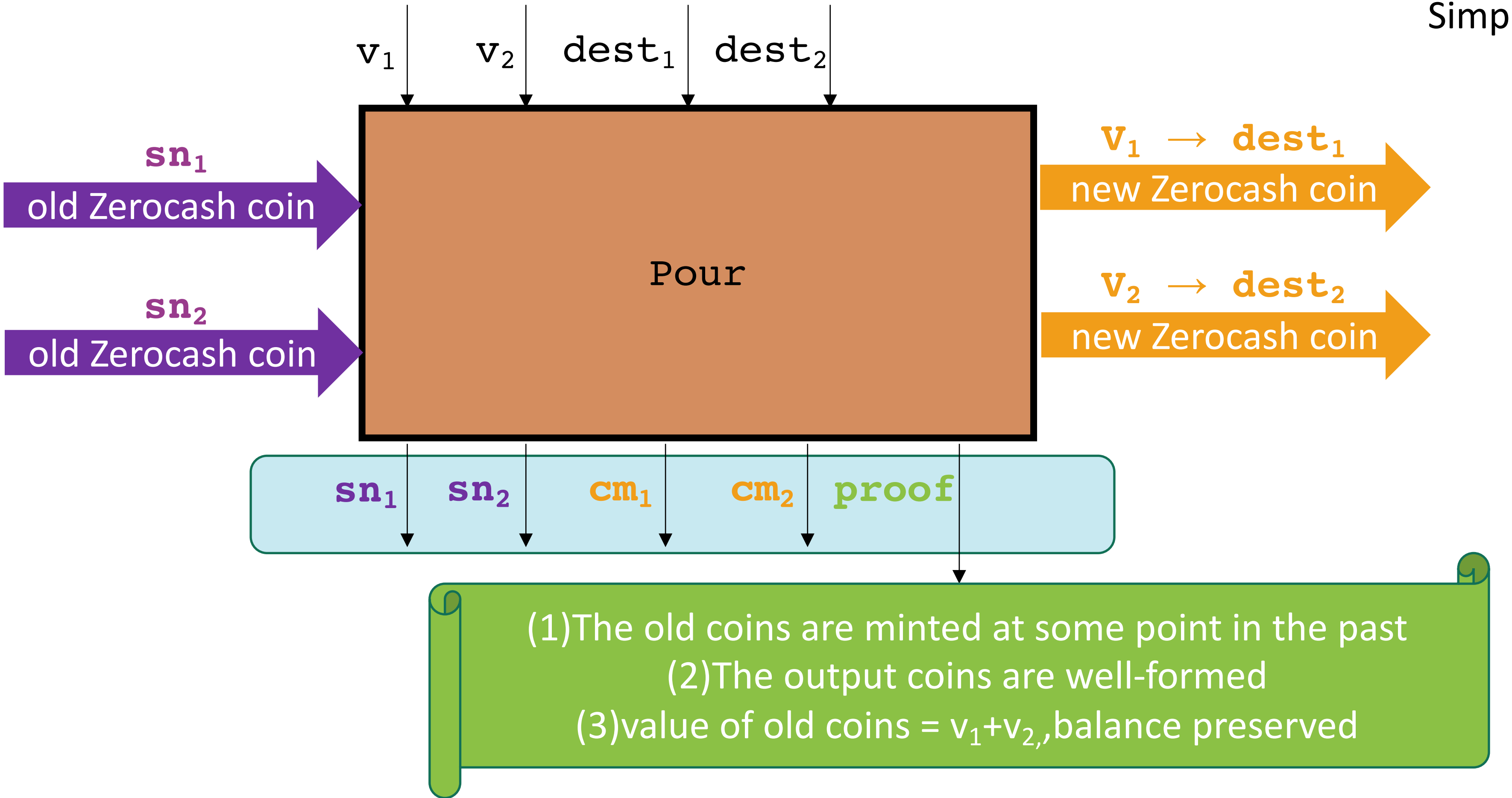
- (1) out of band
- (2) encrypted to receiver's public key α_{pk}





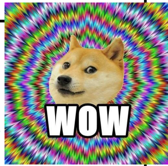
Pouring Zerocash coins

Simplified



What does a transaction look like?

root	1c4ac4a110e863deeca050dc5e5153f2b7010af9
sn_1	a365e7006565f14342df9096b46cc7f1d2b9949367180fdd8de4090eee30bfdc
sn_2	6937031dce13facdebe79e8e2712ffad2e980c911e4cec8ca9b25fc88df73b52
cm_1	a4d015440f9cfae0c3ca3a38cf04058262d74b60cb14ecd6063e047694580103
cm_2	2ca1f833b63ac827ba6ae69b53edc855e66e2c2d0a24f8ed5b04fa50d42dc772
pubkeyHash info	8f9a43f0fe28bef052ec209724bb0e502ffb5427
SigPK	2dd489d97daa8ceb006cb6049e1699b16a6d108d43
Sig	f1d2d2f924e986ac86fdf7b36c94bcdf32beec15a38359c82f32dbb3342cb4bedcb78ce116bac69e
MAC_1	b8a5917eca1587a970bc9e3ec5e395240ceb1ef700276ec0fa92d1835cb7f629
MAC_2	ade6218b3a17d609936ec6894b7b2bb446f12698d4bcafa85fcbf39fb546603a
ciphertext_1	048070fe125bdaf93ae6a7c08b65adbb2a438468d7243c74e80abc5b74dfe3524a987a2e3ed075d54ae7a53866973eaa5070c4e0895 4ff5d80caae214ce572f42dc6676f0e59d5b1ed68ad33b0c73cf9eac671d8f0126d86b667b319d255d7002d0a02d82efc47fd8fd648 057fa823a25dd3f52e86ed65ce229db56816e646967baf4d2303af7fe09d24b8e30277336cb7d8c81d3c786f1547fe0d00c029b63bd 9272aad87b3f1a2b667fa575e
ciphertext_2	049311081431 9b0b5cabb9a9225062354987c8b8f604d96985ca52c71a77055b4979a50099cefc5a359bdf0411983388fa5de840a0d 64816f1d9f38641d217986af98176f420caf19a2dc18c79abcf14b9d78624e80ac272063e6b6f78bc42c6ee01edfbcddbeb60eba586 eaecd6cb017069c8be2ebe8ae8a2fa5e0f6780a4e2466d72bc3243e873820b2d2e4b954e9216b566c140de79351abf47254d122a35f 17f840156bd7b1feb942729dc
zkSNARKproof	a4c3cad6e02eec51dc8a37ebc51885cf86c5da04bb1c1c0bf3ed97b778277fb8adceb240c40a0cc3f2854ce3df1eafdcefccc532bc5afaefefe9d3975726f2ca829228 6ca8dd4f8da21b3f98c61fac2a13f0b82544855b1c4ce7a0c9e57592ee1d233d43a2e76b9bdeb5a365947896f117002b095f7058bdf611e20b6c2087618c58208e3 658cfcc00846413f8f355139d0180ac11182095cdee6d9432287699e76ed7832a5fc5dc30874ff0982d9658b8e7c51523e0fa1a5b649e3df2c9ff58dc05dac7563741 298025f806dfbe9cfe5c8c40d1bf4e87dacb11467b9e6154fb9623d3fba9e7c8ad17f08b17992715dfd431c9451e0b59d7dc506dad84aef98475d4be530eb501925 dfd22981a2970a3799523b99a98e50d00eaab5306c10be5



Timeline of Zcash

2013 IEEE Symposium on Security and Privacy

Zerocoin: Anonymous Distributed E-Cash from Bitcoin

Ian Miers, Christina Garman, Matthew Green, Aviel D. Rubin
The Johns Hopkins University Department of Computer Science, Baltimore, USA
{imiers, cgarman, mgreen, rubin}@cs.jhu.edu

2014 IEEE Symposium on Security and Privacy

Zerocash: Decentralized Anonymous Payments from Bitcoin

Eli Ben-Sasson[*], Alessandro Chiesa[†], Christina Garman[‡], Matthew Green[‡], Ian Miers[‡], Eran Tromer[§], Madars Virza[†]
[*]Technion, `eli@cs.technion.ac.il`
[†]MIT, `{alexch, madars}@mit.edu`
[‡]Johns Hopkins University, `{cgarman, imiers, mgreen}@cs.jhu.edu`
[§]Tel Aviv University, `tromer@cs.tau.ac.il`

Zcash begins

Zooko Wilcox | Oct 28, 2016

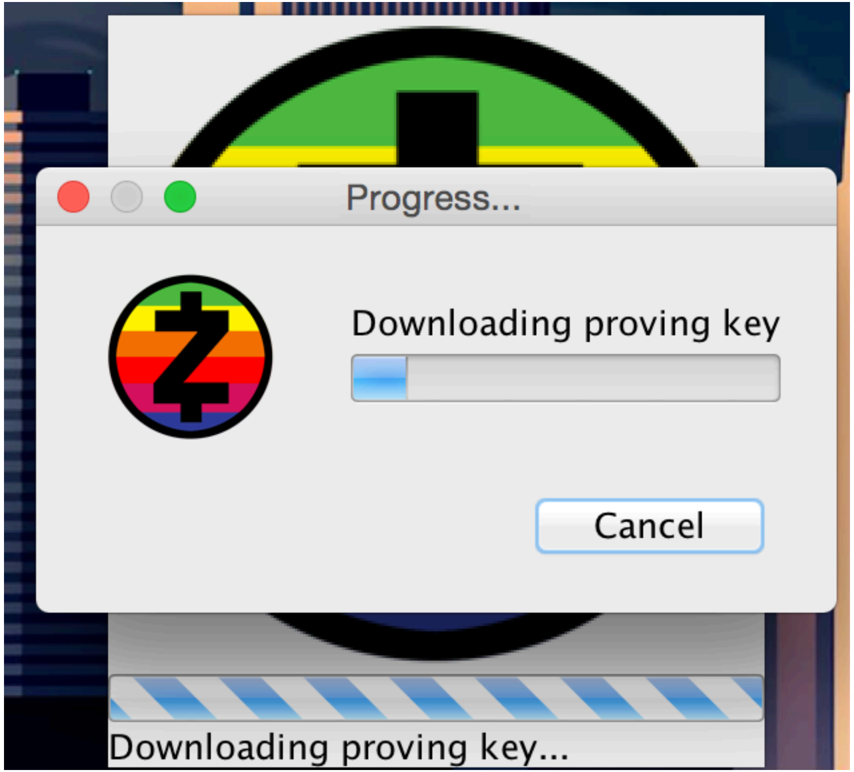
The Zcash blockchain is live! We released the genesis block this morning, and people all around our planet have begun mining and transacting on it.

Zcash as of this morning

All ▾ | Coins ▾ | Tokens ▾ | USD ▾ | ← Back to Top 100

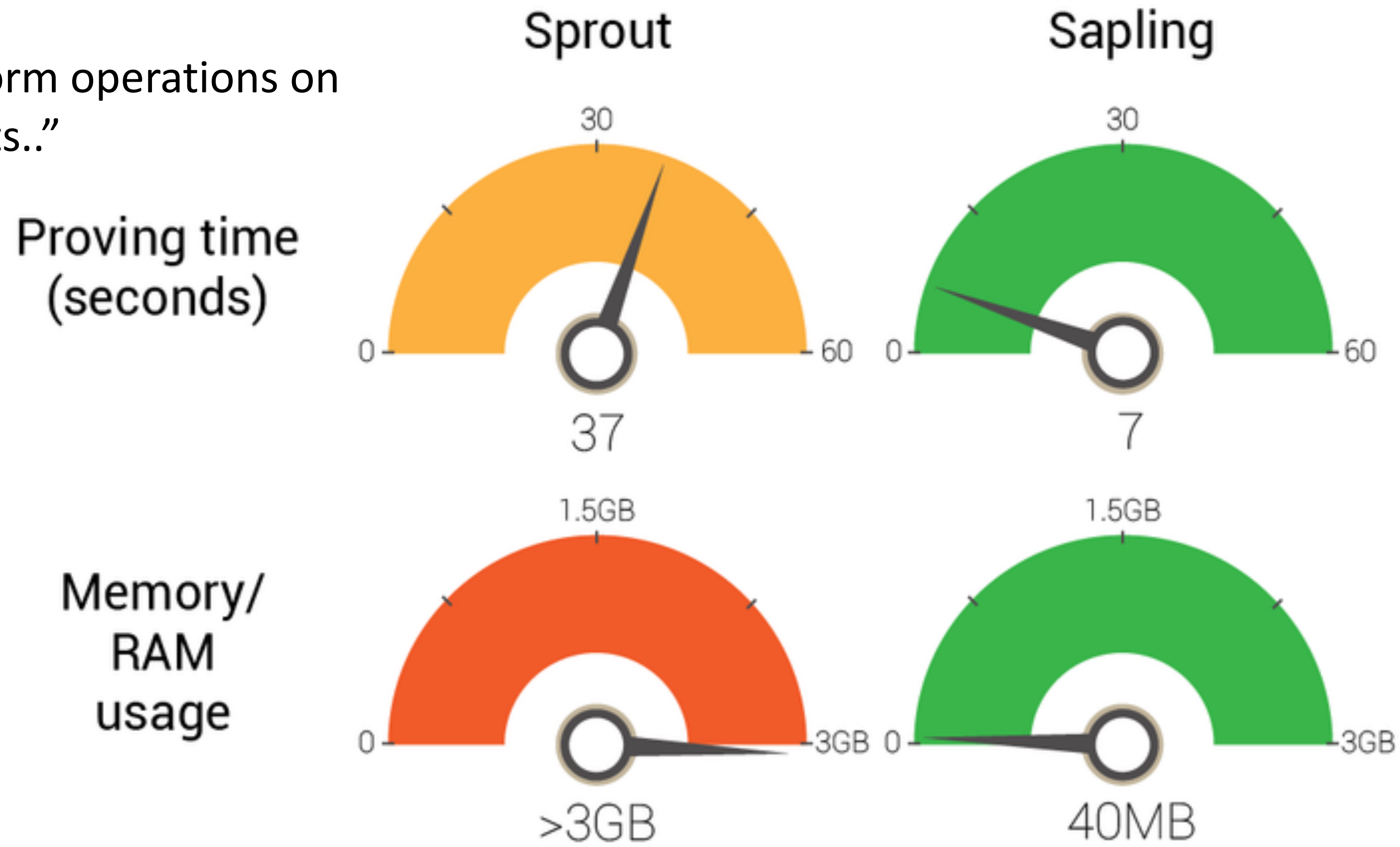
#	Name	Symbol	Market Cap	Price	Circulating Supply	Volume (24h)	% 1h	% 24h	% 7d
1	Bitcoin	BTC	$122,574,167,168	$7352.50	16,671,087	$4,614,130,000	-1.47%	2.25%	8.51%
2	Ethereum	ETH	$29,463,011,035	$308.16	95,608,529	$973,376,000	-0.55%	4.55%	5.43%
3	Bitcoin Cash	BCH	$10,676,720,231	$636.56	16,772,475	$890,417,000	2.28%	2.57%	21.76%
4	Ripple	XRP	$8,360,226,531	$0.216971	38,531,538,922 *	$160,370,000	-0.02%	3.63%	11.86%
5	Litecoin	LTC	$3,335,373,595	$62.07	53,738,882	$337,594,000	-0.65%	0.87%	16.13%
6	Dash	DASH	$2,401,876,167	$312.87	7,677,037	$103,209,000	-0.99%	7.14%	15.89%
7	NEO	NEO	$2,013,882,000	$30.98	65,000,000 *	$122,593,000	0.74%	18.28%	18.80%
8	Monero	XMR	$1,720,036,265	$112.22	15,327,496	$81,072,200	-1.62%	12.08%	32.99%
9	NEM	XEM	$1,705,464,000	$0.189496	8,999,999,999 *	$6,447,280	-2.27%	5.25%	11.26%
10	Ethereum Classic	ETC	$1,356,419,613	$13.95	97,257,388	$127,993,000	-0.48%	-1.38%	35.04%
11	IOTA	MIOTA	$1,308,091,424	$0.470616	2,779,530,283 *	$44,598,200	-0.37%	21.91%	31.19%
12	Qtum	QTUM	$881,557,511	$11.97	73,647,244 *	$196,600,000	0.64%	7.76%	19.32%
13	OmiseGO	OMG	$794,299,223	$7.78	102,042,552 *	$85,668,400	1.63%	22.51%	26.47%
14	Lisk	LSK	$682,120,729	$5.95	114,622,875 *	$33,198,300	-1.12%	16.16%	32.75%
15	Cardano	ADA	$648,324,548	$0.025006	25,927,070,538 *	$6,346,360	-0.40%	13.47%	3.13%
16	Zcash	ZEC	$639,994,999	$248.21	2,578,431	$59,987,700	-1.24%	3.11%	11.98%

What is happening?



Criticism: memory usage

*“We have designed an elliptic curve called
Jubjub
which is efficient to perform operations on
inside of zk-SNARK circuits..”
---Sep 13,2017



*Cultivating Sapling: Faster zk-SNARKs. (n.d.). Retrieved November 07, 2017, from https://z.cash/blog/cultivating-sapling-faster-zksnarks.html

Criticism: 'Ceremony'

- Recall zk-proof "API":

  `Setup(`$stmt$`)`

  $\pi \leftarrow$ `Prove(`$input$`)`

  `Verify(`π`)`

- Who will do the parameter setup for ZCash?
- How can we trust these people?

FYI: Details in trusted setup

- `Setup` generate fixed keys used by all provers and verifiers.
- If `Setup` is compromised at the dawn of the currency, attacker could later forge coins.
- Run once. Once done and intermediate results erased, no further trust
- Anonymity is unaffected by corrupted setup
- Can be done by an MPC protocol, secure if even one of the participants is honest.

[Ben-Sasson Chiesa Green Tromer Virza 2015]

Criticism: ‘Ceremony’

Simple Parameter Generation

- Generate (Public key/ Private key) pair
- Keep Public key for future use
- Delete Private key

Toxic Waste

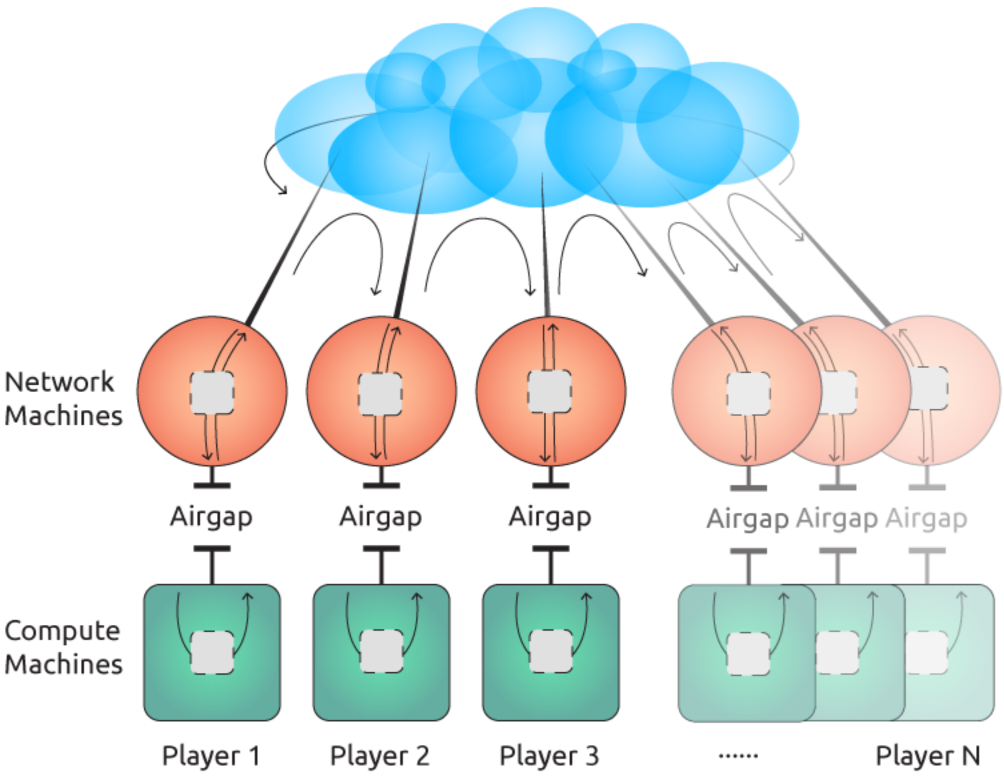
Criticism: 'Ceremony'

Multiparty Computation Protocol

- Generate shards of public key/private key
  - (Pub_1,Pri_1), (Pub_2,Pri_2),…
  - Combine Pub_1, Pub_2 … to be Public Key
  - Delete **ANYONE** in Pri_1, Pri_2,…

Toxic Waste

Criticism: ‘Ceremony’



As soon as any one of the Witnesses deleted their private key shard, then the toxic waste could never be created.

We only need ONE honest witness

Criticism: ‘Ceremony’

The Crazy Security Behind the Birth of Zcash, the Inside Story

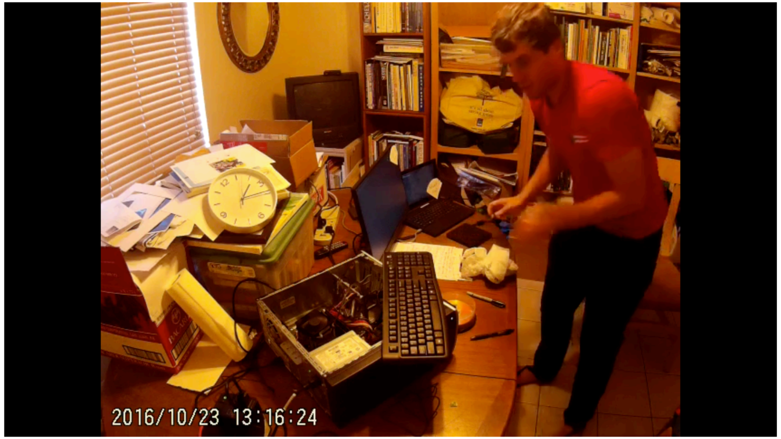
By Morgen E. Peck
Posted 2 Dec 2016 | 18:50 GMT

Photo: Morgen Peck

Paranoia, the destroyer: Za Wilcox, brother of Zcash CEO Zooko Wilcox, sets about destroying a computer used to generate the cryptographic parameters needed to start Zcash

Zcash's public parameters were generated using this protocol in a ceremony that took place on October 21-23. The ceremony involved six participants, each in their own location, each with their own hardware:

1. Andrew Miller
2. Peter Van Valkenburgh
3. John Dobbertin (pseudonym)
4. Zooko Wilcox
5. Derek Hinch
6. Peter Todd



ZCash: Orlando Station Report

Criticism: 'Ceremony'

OCT
31

Improved zk-SNARK Multi-party Computation Protocol

Sean Bowe, Ariel Gabizon and Ian Miers | Oct 31, 2017

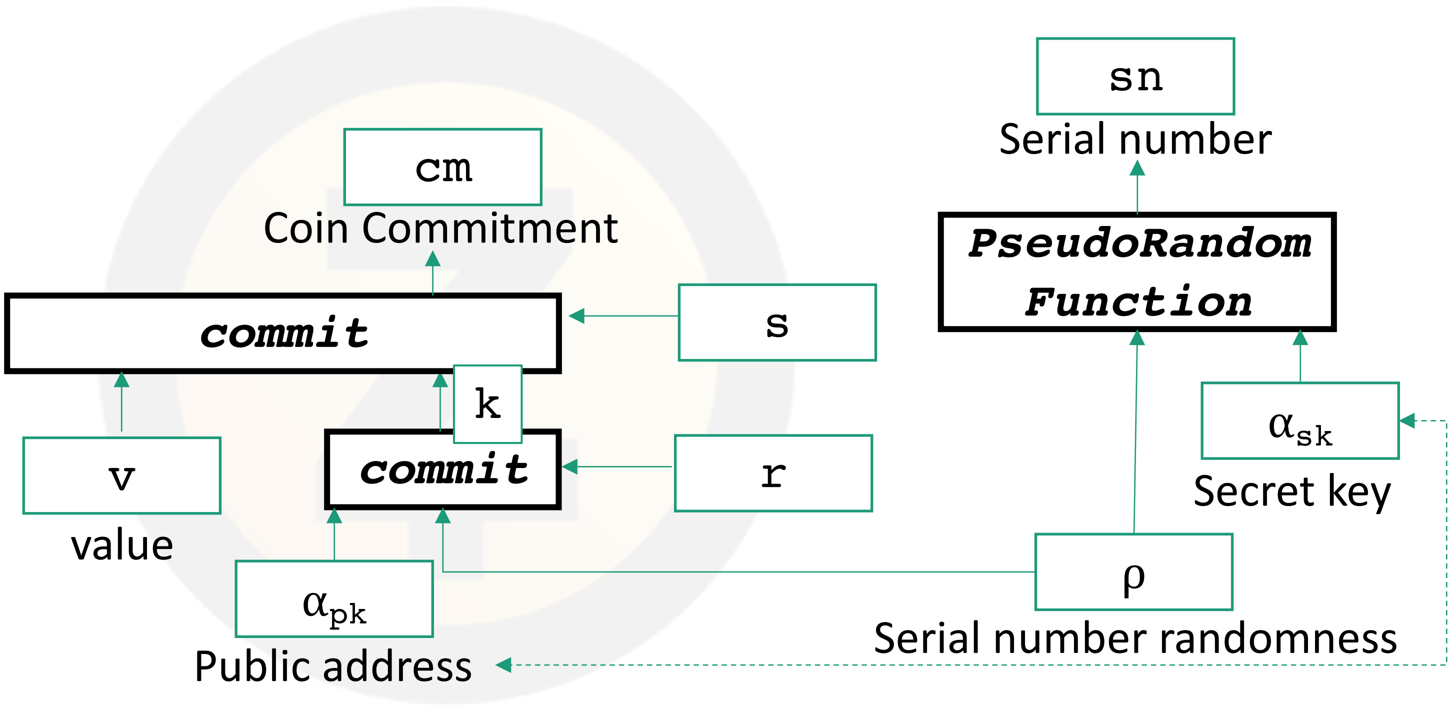
zk-SNARKs – the zero-knowledge proofs at the core of Zcash – require a parameter generation ceremony to take place for every statement that you wish to create proofs about. Although privacy is protected by zk-SNARKs unconditionally, if this ceremony is compromised it becomes possible to counterfeit Zcash. It is thus important for us to ensure these parameters are created securely.

Last year, Zcash performed such a ceremony using a multi-party computation (MPC) protocol. These protocols have the property that only *one* party needs to be uncompromised for the resulting parameters to be secure. In other words, in order to compromise the ceremony, *every* participant needed to be compromised.

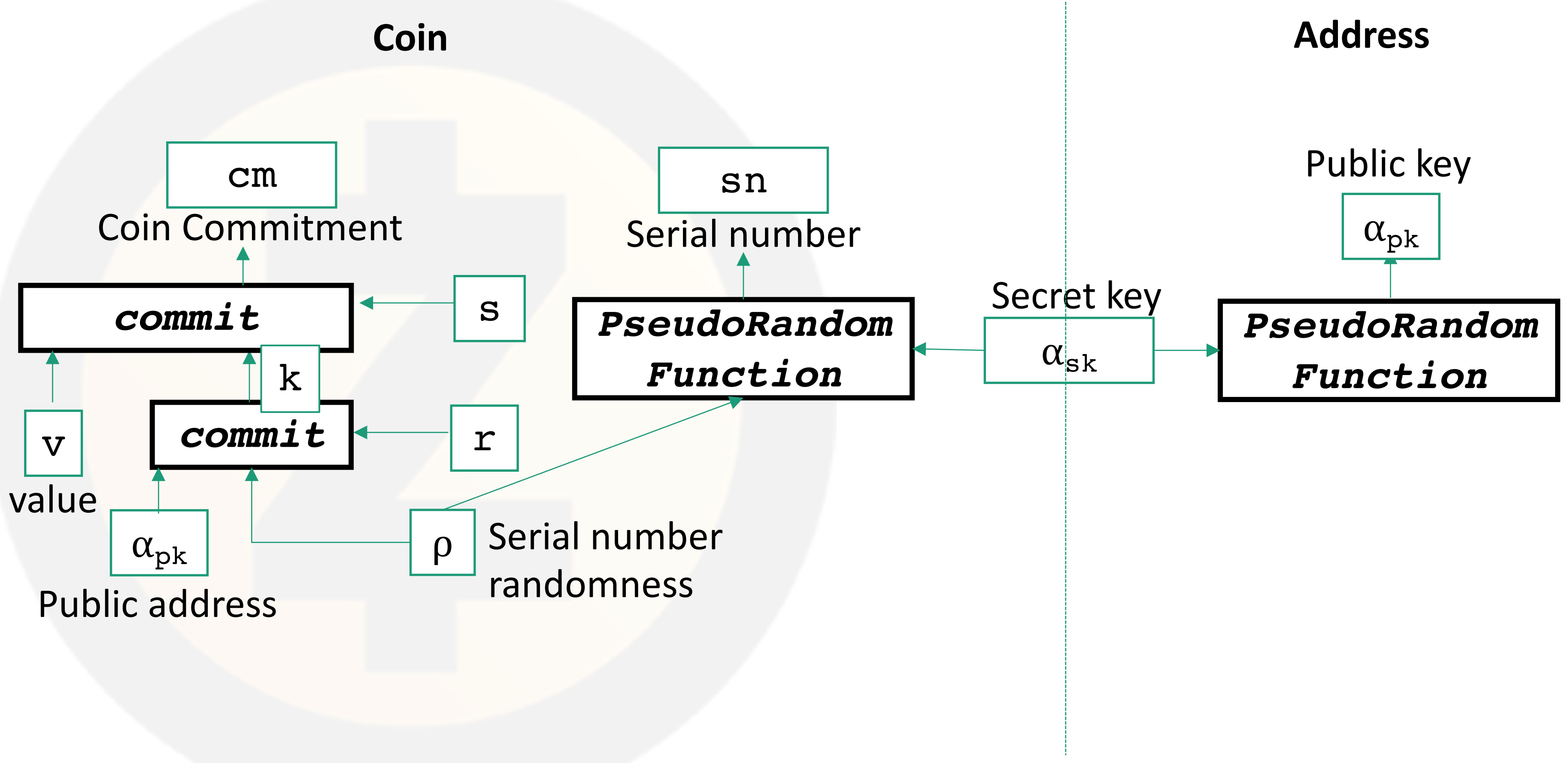
Takeaways

- 1. Maximum anonymity you can get
- 2. At a high-level, zk-SNARKs look intuitive
- 3. Under utilized
- 4. Further discussions

Appendix 1:A concise view



Appendix 2: Another view



Acknowledgements

- This paper presentation is part of COMP6111C:Blockchain and Cryptocurrency Technologies in 2017 Fall at HKUST, taught by Prof. Dimitris Papadopoulos
- The slides are prepared based on the sources listed below. The presenters would like to thank the authors for making the information publicly available online.

- E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In *IEEE Symposium on Security and Privacy*, 2014
- E. Tromer. Information Security – Theory vs. Reality, 0368-4474-01, Winter 2015-2016, Lecture 12:Verified computation and its applications
- "Zerocash: improving Bitcoin using SNARKs", *YouTube*, 2014. [Online]. Available: https://www.youtube.com/watch?v=S6qOj9ap6RM. [Accessed: 16- Nov- 2017].

- The slides have not been updated since then and some information may be outdated, if you have any questions, please feel free to reach out to the original paper authors, or the presenters gq35@cornell.edu, tzhang@g.harvard.edu