Bitcoin: Fungibility, Privacy & Identity

Adam Back
bitcoin / applied cryptographer

ecash & fungibility

- Paper cash bank notes are equal $1 = $1
- Legal precedent dating back to 1700s
- Scottish court case involving high value note
  - Court ruled confidence in cash would collapse if merchant was allowed to reclaim stolen note
- Idealized cryptographic ecash aims to enforce fungibility via indistiguishability rather than law
- trust in mathematics over law
  - bitcoin physical coin “vires in numeris”

central server blind ecash

- Chaum blind sig 1982 (blind RSA sig)
  - p = b^e*m mod n →
  - r = p^d = b*m^d mod n ←
  - c = r/b = m^d mod n (verify c^e =? m)
- Avoid existential signature forgery
  - c^e=m => c=m^d mod n (random looking m)
  - prevent with structure m = s||H(s)
- Double spend prevent: server stores serial no s
  - s randomly chosen by user (avoid reuse s or rejected)

privacy – blind server ecash

- cryptographically unlinkable, payer anonymous
- optionally payee anonymous (chaum)
  - if payee chooses s, b
- online (payee has to deposit asap)
- optionally linked to account or just cash exchanged
- chaum single denomination (per n keypair)
  - reduced anonymity set 1,2 4, 8.. denominations
- zero-trust: colluding merchant & payer cant link
- perfect fungibility/privacy; vulnerable to server shutdown

Digicash betabucks

- David Chaum's company (netherlands)
- Demo ecash server 1mil “betabuck” coins
- No banking interface, faucet coins on request
- People started selling things to try bootstrap
  - Relying on digicash promise to cap to 1mil coins
- Company went bankrupt
  - Double-spend database went offline
  - Spent vs unspent became unprovable

Brands credentials/ecash

- Stefan Brands (David Chaum's PhD student)
- Representation problem (extended Schnorr sig)
  - y=g^x*h^y*.. mod p for base g,h,..
- Blind schnorr signature extension
  - Blind (secret key) certificate
- Supports multiple denominations
- Flexible attribute certificates
- ZK provable formulae on attributes
  - Over 18 or Dutch citizen (but not which)

Brands high-level

- h=encode-attribs(attrs,x) (user pub h, pri x)
- h'=blind(h,b) → (b random blinding factr)
- p=blind-prove(h',attrs) → vrfy-attrs(h',attrs)
- ← s=sign(h')
- c=unblind(s,b) (blind cert c, encodes attrs)
- attr-prove(c,h,attrs)=? Valid (anyone verify)

hashcash

- Hashcash proof of work 1997 (Adam Back)
- Fully decentralized
- No coordinated inflation control
- anonymous/fungible as fresh coins only
- not respendable: one-use stamp for anti-DoS
- H(s,c)/2^(n-k) = 0 (brute force lsb == 0000...)
  - Where s is service string
  - c is counter (starting at random offset)

B-money / bit-gold

- B-money proposal 1998 (Wei Dai)
- Bit-gold proposal 1998 (Nick Szabo)
- Use hashcash for distributed mining
- Design outlines (not implemented)
  - broadcast transactions to group of servers
  - Inflation set by vote (b-money)
  - Inflation adjusted by collectible market (bit-gold)
- pseudonym based (like bitcoin)

Sander & Ta-Shma

- Blinding only works for central server
- auditable anonymous ecash paper (1999)
- ZKP of set-membership
- Using merkle tree and DLOG
- Can be decentralized as bank has no private key
- somewhat CPU expensive and largish proofs
- Later optimized by Zerocoin

Bitcoin (Nakamoto 2008)

- hashcash mining (like b-money/bit-gold)
- Dynamic difficulty / fixed supply curve
- Proof of work solution to byzantine generals
- Pseudonym based each coin is linkable
- Change making links
- Change combining links
- Overall quite linkable (Shamir & Dorit 2013)
  - Using network analysis of above links

Taint tracing

- due to online thefts & illicit use
- Ignoring fungibility some parties proposed to trace coins as a biz service (coin validation)
- Bad side effect could create a value run
  - If you hold a coin that is rejected by merchant
  - You try to sell it, maybe at a discount
  - Creates a run on bitcoin price?
  - Damages confidence as the 17th century case

Weak fungibility: Feature & bug

- Users: its somewhat private
- Crime investigation: its not very private
- Users want more privacy
- Investigation want same or less privacy
- Users/banks/biz want more fungibility
- Fungibility provides privacy as a side effect
- Sounds like a conflict

Identity

- Bitcoin privacy is fragile (Shamir & Dorit network analysis)
- Internet not very anonymous
  - Identify when paying (account, delivery addr), largely identified IP#
  - regulated biz require proof of identity
- Societal contract: reasonable suspicion required for tapping
- Criminal investigation
  - business entities required to keep records
  - Investigation via record subpoena
- National intelligence
  - post-Snowden: intelligence agencies extensively tapping & logging
  - extensive device, network compromise
  - so far seemingly fundamental limit – host security is hard

Zerocoin (Green, Miers 2013)

- Optimized set membership ZKP
  - More efficient Sander & Ta Shma design
  - Using Benaloh & de Mare RSA accumulator
- Good fungibility/privacy
- Still inefficient:
  - 1 minute to create coin, 20-40kB per coin
  - 1 denomination
  - Or 1,2,4.. denomination & reduced anonymity set
  - RSA accumulator has trap door (forge coins, still private)

Zerocash (2014 Green, et al)

- Using SCIP/SNARKS (2013 Ben-Sassoon)
- ZKP of set-membership SNARK
  - Program implements SHA256 Merkle tree
- Better
  - multiple denominations, compact proofs (< 300bytes)
  - big creation params (> 1GByte)
  - moderately expensive creation
  - Practical but still has a trap door
  - New crypto (cryptanalysis risk?)

anonymous ecash

- Zerocash alt-coin (Green, Miers plan)
  - Setup trapdoor assurance ceremony
  - Better bitcoin peg rather than alt?
- Or trap-door free ZC3, efficient etc
- Cryptographic fungibility good
- But is Society ready for full anonymous ecash?
  - transfer $1b or 1c
  - Completely anonymous: how much, who, or when
  - No one can undo/block/freeze it

Crypto fungibility as building block

- Cryptographic fungibility at transaction layer
- Identity at payment level
- Analogy: identity to buy gun, but pay in cash
- Optional certified ID for regulated business
- Most users are not bad actors
- Business keep records
- Subpoena good actors for info on bad
- Replicates status quo, avoid pre-emptive surveillance

More privacy

- Encrypt certified identity (regulated biz scenario)
  - subpoena for record + court order CA to decrypt
- Multiple self-asserted identity (pseudonym) Non regulated scenario
- Network logging by recipient
- Intelligence community: logging, back-doors, human int.
- Societal contract:
  - Full cryptographic fungibility
  - Privacy possible & practical
  - investigation possible
- Full identity does not prevent:
  - identity theft; banks now (HSBC laundered ~$1b)

Short term

- CoinJoin (Maxwell 2013)
  - Trustless multiple input multiple output tx
- Merge avoidance (Hearn 2013)
  - Pay to multiple addresses in parts
- One-use address (Nakamoto 2008)
  - Avoid direct linking
- CoinSwap (Maxwell 2013)
  - trustless paired A->B and B->C
- Coin control (intelligent change management)

Address limitations

- Donation address (static, more linkable)
- Smart phone wallets reusing addresses
  - No HD address support yet
- Users dont understand one-use address
- Reusable address (full node only – trial decrypt using DH)
- Prefix for SPV, but reduces anonymity set worse flow
- Bloom filter for SPV (some ambiguity)
  - Not much ambiguity or more query bandwidth
- IBE address

IBE address

- ID based encrypt (Weil Pairing Boneh Franklin 2001)
- User acts as own IBE server
- Sender computes per block/epoch pub key
- Encrypt for pub key
- SPV user delegates decrypt capability to node
  - Calculates private key for epoch key
- Node cant correlate payments to IBE addr diff epoch
- Compact query. More CPU for node. Query fee?

Homomorphic Encrypted value

- Another aspect of privacy is amounts
  - Salary, business model, wealth, safety
- FHE is slow impractically inefficient (~10^7x)
- Single HE is efficient
  - E(a)+E(b)=E(a+b) eg el gamal, paillier
- But wraps a+b=c (mod n)=> a+b=c+kn
  - Add ZKP range proof to prevent wrap

Bitcoin HE value

- Use ZKP range proof (Schoenmakers 2000?)
- Optimize a bit
  - 8 byte unencrypted value
  - 1kB encrypted value
- Can add up change
  - Pederson commitment C=xG+vH (c=g^x*h^v)
  - x1G+v1H=?x2G+v2H+x3G+cH
  - Add up unencrypted fee also
- Normal bitcoin linkability, but value privacy

Applications HE value

- Preserving commercial confidentiality
- Auditable business risk
  - Insurance coverage
- Smart contract:
  - Insurance company cant issue policies
  - If next policy is > reinsurance coverage
- Bitcoin audited company
  - Income, expenditure, dividends, salary
  - all public auditable - no off balance sheet risk, systemic risk audit
  - Auditable leverage ratios

RingCoin

- Curiosity: can use ZKP homomorphic value
- plus ZKP generic “OR” construct
- Involve coin as transaction input IF:
  - know private key (you own the coin)
  - OR you are taking 0 value from the encrypted val
- Name from Ring signature
  - Like multiparty sig where you prove 1 of n
  - Without cooperation from other n-1
- More private because you chose randomly
- Or choose plausible other spenders

committed-transaction

- Send encrypted transaction
- Miners validate not double-spent
- Wait 6-blocks, then reveal key to network
- Miner can not tell sender, recip, amount
- To undo miner has to orphan own work
- User can reveal more tx
- Makes miner policy uneconomic

Committed-tx protocol vote

- Protocol defines only committed tx
- miner rejects committed tx
- Then miner forms alt with no users
- Users define protocol
- Hashrate falls
  - Rest of miners continue
  - Tolerated hostile miner limit

Respendable commited-tx

- Can respend tx in comitted form
- Send key to recipient
- Full node only, quite private
- But over time coins circulate and eveyone
- in coin path can see history

end