Quo Vadis PKI?

IEEE International Conference on Public Key Infrastructure and its Applications: PKIA 2017
Bangalore 14-15, November 2017

C.E.Veni Madhavan

Informatics and Security Laboratory
Department of Computer Science and Automation
Indian Institute of Science, Bangalore

15 November 2017

Contents

1. Digital Evolutionary Threads

- DDBMS $\longrightarrow$ blockchains
- ecash $\longrightarrow$ cryptocurrencies
- PKI $\longrightarrow$ TSI (Trust Systems Infrastructure)

2. AAAARGH - PAIN, SKC, PKC, PKI

- accountability, auditability, authority, anonymity, governance, hype
- privacy, authenticity, integrity, non-repudiability
- symmetric key cryptography
- public key cryptography
- public key infrastructure

3. applications - documents, finance . . .

- digital tokens - dongles, wallets, . . .
- digital money - transactions, commerce, . . .
- digital finance - loans, deposits, inclusion, . . .
- digital lockers - certificates, contracts, . . .
- digital trust - PKI, KMC, TTP, blockchains, . . .

4. Blockchain Technology - Summary

- a trusted and immutable ledger
  - work factor for proof-generation - *worst-case* 2^{72} per block
  - each block header contains hash of previous block header
  - each block header contains root-hash of Merkle-hash-tree of the transactions included
  - verifiability - one time application of hash function
- a distributed, publicly verifiable ledger
  anyone can check validity of blocks & transactions:
  - using user public keys
  - tracing all previous blocks & transactions
- *very expensive integrity proofs*
- a public (anyone can join/verify data) blockchain is called *permissionless* blockchain

Consensus in Blockchain

other hard problems used in consensus rules:

- proof-of-stake: based on number of coins hold by the miner
- proof-of-space or *proof-of-capacity*: specified amount of disc space
- proof-of-activity: combination of proof-of-work & proof-of-stake
- proof-of-storage: specified disc space
- proof-of-burn: prove that certain amount of coins are burned (lost for ever)

Blockchains Vs Traditional data integrity

- blockchains provided trust, and data integrity based on hardness of solving mathematical problems (i.e., proof-of-work)
- traditional approach: Digital signature based on computational hardness of DLP/Integer Factoring/ECDSA
- a comparison:
  - blockchains - A single node (honest or dishonest), has to spend same amount of work (2^{72} hash computations)
  - traditional (RSA1024 bit) - honest node (with private key) can produce signature in few milliseconds
    dishonest node (for forgery) would try to factor RSA1024
    (will need only few hours of the proof-of-work network cited - computing 2^{72} hashes in 10 mins)

5. ecash, cryptocurrencies cryptonomics

- D.Chaum [1982], S.Brandt
- Micromint, Millicent, Netcard
- ecash - electronic analog of fiat cash
- withdrawal, spending, deposit protocols
- blind signatures, double-spending controls
- anonymity, privacy, fungibility, transferability
- mutual authentication, distributed operations
- hash chains with signatures on terminal coins
- large volume, small value, micro-finance operations
- our earlier paper - transferable ecash [Indocrypt 2000]
- our previous work - semi-distributed ecash [our CEFIPRA Project]
- our recent paper - VSKchains- semi-centralized [ADCOM Sep 2017]
- our recent paper - Cryptocurrencies: Science and Socio-Economics
- our current work - alternative proofs-of-work, algo, complexity, SUPW

The Bitcoin (Contd.)

- every transaction is recorded in a public ledger called *blockchain*
- blocks of chains are inked by a system of *hash addresses*
  i.e., each block contains hash of the previous block, hence immutable
- each block contains a collection of new transactions
- each block created by a process called *mining* or proof-generation
- mining, or finding a specific hash value, is computationally expensive
- miner is rewarded with a bitcoin in return for the computational effort

Transactions

- each transaction is like a double-entry bookkeeping ledger entry, containing several input and output entries
- each input refers to a (previous) transaction ID with unused bitcoin output entry
- each output refers to *payee* bitcoin address and bitcoin amount to be transferred
- bitcoin address is derived from user's public-private key pair (hence pseudo-anonymity)
- each transaction is digitally signed for *payer* authentication
- only *valid* transactions (i.e., previously unspent bitcoins checked by an address search) will be included in a new block (hence no double-spending)

Distributed Consensus

- consensus rules: pertaining to validity of blocks and transactions in it
- consensus throughout the network (without a central authority) achieved by *mining*
- mining involves solving a computationally hard problem that serves as a proof-of-work
- proof-of-work algorithm used in the Bitcoin is from a hash function SHA-256
- each node can independently verify the validity of a new block (including transactions in it)
- only a valid block will enter into blockchain, hence lead to precessing of new valid transactions

Proof-of-Work

- the hash function $\mathcal{H}$ = SHA256 is used in the Bitcoin
- hard Problem: find a string (or part of a string), called *nonce*, which gives specified hash (say, certain leading bits are zero)
- given $x, y = H(x)$, find nonce n, such that $H(x + n) = z$, where $x =< 0, \ldots, 0, b_{72}, \ldots, b_{255}$ is a specified form of bit-string output
- verification is easy: Apply SHA256 once to check whether the leading bits of hash of *solution string* are zero
- How HARD is hard?
  prob $(b_i = 0) = \frac{1}{2}$, (i.e.,i^{th}-bit zero)
  Probability that first ℓ bits are zeros is $\frac{1}{2^\ell}$
  Need to try 2^ℓ nonces

Proof-of-Work

- difficulty level : No. of first ℓ bits
- difficulty level maintained so that new valid block can be found in around 10 minutes
- based on avg. computation time for last 2100 blocks
- current status (Oct. 2017): (source: blockchain.info)
  total hash rate available $\sim 8 * 10^{18} \simeq 2^{62.795}$ per second
- total hashes per block *for 10 minute tasking* $\sim 8 * 10^{18} * 10 * 60 \simeq 2^{72}$
  equivalent : first 72 bits are zeros
- hard problem is equivalent to:
  given a hash function and a partial output find a pre-image

Zerocash

- Zk-SNARK - efficient variant of zero-knowledge proof of knowledge (GMR89) to show some proof that one owns k coins, without showing the coins, by giving only 1 bit that one knows secret keys controlling the k coins
- succinct non-interactive arguments of knowledge
- (Ben-Sasson, Chiesa, Garman, Green, 2014)
  zerocash : decentralized anonymous payment from bitcoin
- Bitcoin not completely anonymous, although multiple identities are used
- de-anonymization is possible with blockchain ledger transaction graphs
- use of mixes (or laundries) of coin pools has limitations
- privacy : fiat cash >> ecash >> Bitcoin
- Zerocoin (MGGR13) uses, like ecash, zero knowledge proofs to thwart transaction graph analyses
- redeeming zero-coins requires double discrete log proofs of knowledge

Pinocchio

- Pinocchio Coin: building Zerocoin from a Pairing-based Proof System (DanezisFournetKohlweissParno13)
- The original Zerocoin protocol relies on the Strong RSA assumption and double-discrete logarithm proofs - performance constraints
- a variant of the Zerocoin protocol using instead elliptic curves and bilinear pairings. The proof system makes use of modern techniques based on quadratic arithmetic programs resulting in smaller proofs and quicker verification.

Our Reckoning: TKG, BC-CC PoW as a Service (PoWaaS); quantum indulgence, quantum readiness

- Trust-as-a-Service (TaaS) : authorized TKG (with digital certificate)
- Proof-of-Work as a Service (PaaS) $\leftrightarrow$ algorithm cycles for sale!
- independent TTP / TKG as service for *identity based signatures*
- high-throughput, custom hash functions $\leftrightarrow$ research services
- petahash/sec arbiter networks $\leftrightarrow$ grid resource services
- cryptocurrency exchange networks $\leftrightarrow$ cryptonomics services
- quantum cryptanalysis of BC, CC:
  quantum search (Grover), quantum DLP, IFP (Shor), realistic quantum attacks (ABTLST, ArXiv: [quant-ph] 28 Oct 2017)
- some quantum-safety-measures: extreme parametrics; ASIC-agnostics; (Memory-Hard (MH) computations) (egalitarian mining!)
- MH candidates: finding predicated hash-collisions; finding special subgraphs; Equihash - generalized birthday problem
  (MY thoughts: problems from number theory and cryptography)

6. alternatives to present form of PKI

- pomcor.com: [K.Lewison and F.Corella, Oct 2016]
- traditional: knowledge-based verification (KBV) for remote identity proofing
- new : rich credential enabling a 3-factor authentication - (“has”, “knows”, “is”)
- asserting credentials on the blockchain with on-chain storage backing them with a PKI implemented on the blockchain (without CRL, OCSP (online certificate status protocol) queries
- utilize NFC payment or h/w identity tokens used for in-person transactions

Alternatives: PKI, Algorithms, Architectures

- PKI: complex to install, maintain; costly to issue and distribute; costly to recover and validate; certificate revocation
- QuoVadis Netherlands : PKI Root Signing
- enterprise CAs to chain themselves under QuoVadis trusted root embedded in main-stream browsers
- identity based encryption (IBE) - uses Private Key generator (PKG/TKG)
- certificate-less PKC
- CEBOT: certificate enrollment for billions of things (Sep 2015, Sweden) (super-lightweight protocol)
- Authenticated key-exchange without PKI [Hao, Ryan, 2006] (uses double DLP Diffie-Hellman with ZKIP)

7. many signatures: technical departure

- (*) bilinear pairings
- (*) challenge-response
- ( ) identity based encryption
- ( ) blind, undeniable, multi-party
- ( ) algorithms, protocols for blockchains
- (*) zero-Knowledge interactive proof (ZKIP)
- ( ) attacks on algorithms, protocols, computations

Pairing based Signatures

- *Bilinear Pair* : $e : G_1 X G_2 \rightarrow H$, where e is poly-time computable, $\forall m, n \in Z, P_1 \in G_1, P_2 \in G_2$ $e(n \cdot P_1, m \cdot P_2) = n.m \cdot (P_1, P2)$
- a pair of groups is $G_1 = Z_p^*$ and $G_2 = E(F_q)$
- a pair of groups is $G_1 = E(F_{q_1})$ and $G_2 = E(F_{q_2})$
- a pairing based short signature scheme in the above case (using the Tate-Lichtenbaum pairing) is as follows:
  - *signature setup* :
  - Let $G = G_1 = G_2$ be the additive group of points on elliptic curve $G = E(F_q)$ and let B be the *base* point of large order in G.
  - Let α_P and $E_P = \alpha_P \cdot B$ be the private and public keys of P.
  - *signature generation* :
    $P \rightarrow V$: compute $M = h_1(m), M \in G, S = \alpha_P \cdot M$, send $< S, m >$.
  - *signature verification* :
    $V : M = h_1(m), u = e(E_P, M), v = e(B, S)$ and check $u = v$
  - *proof* :
    $u = e(E_P, M) = \alpha_P.1 \cdot e(B, M) = 1.\alpha_P \cdot e(B, M) = e(B, S)$

CR Schemes

- Challenge-Response (CR Schemes) (based on digital signatures)
- Let e_P, d_P be the public and private keys of P
- V $\rightarrow P : r_V \in_R Z$
- P $\rightarrow V : r_P \in_R Z$, sends $C =< r_P, s(= E(r_P||r_V, d_P)) >$
- V : recovers from C the components corresponding to r_P and r_V (from s by computing $E(s, e_P)$) and checks for match

ZKIP Schemes - Idea

- a system setup - (TTP, TKG, TSI)
- P holds a public-private key pair
- a three pass interaction between P and V
- P chooses a random *commitment* and sends a *witness* to V
- V sends a random *challenge* to P
- P sends a *response* using, the private key, the commitment and the challenge
- V computes the expected response using the commitment, the challenge, public key, and establishes probabilistically, the possession of knowledge by P
- *All the ZKIP schemes, described as authentication or identification schemes, also serve as signature schemes by treating the challenge as the message/digest to be signed*

ZKIP Schemes - Realization

Feige-Fiat-Shamir Scheme

- *system setup*: $p, q \equiv 3 \pmod 4; n = p.q$
- *P setup*: $x_1, \ldots, x_t \in_R Z_n^*; u_1, \ldots, u_t \in_R \{0,1\}$
  compute $y_i = (-1)^{u_i}.(x_i^2)^{-1} \pmod n, i = 1, \ldots, t$
  make $< y_1, \ldots, y_t >$ public and keep $< x_1, \ldots, x_t >$ private
- *protocol*:
  $P \rightarrow V : c \in_R Z_n^*, z \in_R \{0,1\}, w \equiv (-1)^z.c^2 \pmod n$
  (commitment)
  $V \rightarrow P :< h_1, \ldots, h_t >, h_i \in_R \{0,1\}$ (challenge)
  $P \rightarrow V : r \equiv c.\prod_{i=1}^{t} x_i^{h_i} \pmod n$ (response)
  V computes $v \equiv r^2 \prod_{i=1}^{t}(y_i)^{h_i} \pmod n$, verifies $v \equiv \pm w \pmod n$

FFS scheme is based on *hardness* of the *SQRT modulo composite* problem
Other, protocols are :
GQ scheme based on hardness of the *RSA* problem and
Schnorr scheme based on hardness of the *DLP* problem

Quo Vadis PKI?

Cryptography is a tale of many intrigues from state-craft to PKI,
with a rich blend of math, computing, trade, and commerce variety.
Algebra, number theory, statistics, algorithmics based schemata
are the bases for securing, citizen, business and government data.
Hence PKI evolution is of great concern to society and crypto community.

cevm
15 November, 2017