VSK Chains

Integrated Content and Currency Transaction Blockchains

C.E. Veni Madhavan, Ch. Srikanth, H.V. Kumar Swamy

09 Sept, 2017

Department of Computer Science and Automation
Indian Institute of Science, Bangalore

Outline

Contribution in this Paper

A conceptual design of a computational system for handling, in an integrated manner, distributed *transactional information* and the associated monetary *payment information*, arising in typical organizational applications.

1. a conceptual architectural design for chains of blocks of content transaction chains,
2. a conceptual architectural design for blocks of financial instrument transaction chains including a proposed form of denominational digital cash, and
3. a new, versatile cryptographic hash function to support all the chaining operations.

Blockchain Technology

Blockchain Technology

- A new way to store, manage, and distribute information
- Decentralized Public Database
- Allow anonymous transactions
  anyone can join to read, write and verify data
- Mining:
  - validate/process (valid) transactions
  - Enables the consensus among network nodes with out central authority
  - Irrevocable transaction history/database
  - based on difficulty of solving hard problem on cryptographic hash function
  - Usually miners are rewarded with "new coins"
- New block acceptance based on largest length blockchain

Types of Blockchains

- **Public/Permissionless Blockchain** (Eg. Bitcoin)
  - Completely decentralized ledger (database)
  - Anyone can be join the network
  - Anyone in the network can read, write and verify data
  - Transaction verification is time consuming
- **Private Blockchain** (Eg. Hyperledger, Corda, Gem Health)
  - Controlled by a *Trusted Party*
  - Limited/restricted access to data
  - Transactions are accepted by Trusted Party
- **Permissioned/Consortium Blockchain** (Eg. Quorum,)
  - Hybrid between Public and Private Blockchain
  - restricts the entities who can contribute to the consensus of the system state

Permissionless vs Permissioned Blockchains

- Permissionless Blockchains(Bitcoin): drawbacks
  - Governance,
  - Taxation,
  - Scalability/Computational Power,
  - ≈ 7 transactions per second (tps)

- Permissioned Blockchains: advantage
  - Scalability, more cost effective
  - Instant settlement,
  - Large number of transactions,
  - Financial/Medical sectors,
  - Small-value, Community-based transactions

Motivation

Motivation

- Community Based large number of small-value transactions with root of trust
- State Sponsored Cryptocurrencies
  Ecuador, China, BRICS cryptocurrencies, Estonia
- peer-to-peer transactions
  - Small-value, large-scale transactions,
  - Instant settlement,
  - Scalability
- Custom tokens with blockchain
  Ex. WhopperCoin - by Burger King
- Grant Management in a organization

VSK Chains - Design

VSK Chains - Design

VSK Chains consists of two parts:

1. the transactional information pertaining to a subject matter of an application,
   stored in *chains of blocks of plain content transaction chains*,
   denoted *BS*, and

2. the payment or monetary information pertaining to this transaction,
   stored in *chains of blocks of monetary transaction chains*,
   denoted *BM*.

Ex. Grants Management system of a university

VSK Chains - Design

Content Transaction Chains

Content Transaction Chains - Example

Grants Management System(GMS) of an university: involve

1. generation of grant for a project proposal, P by scientific investigators of an institution to a funding agency;
2. reviews of P by technical groups;
3. approval and funding of P;
4. receipt, accounting, auditing, and management until closure of P by the recipient institution;
5. project execution involving - technical work, financial expenditure, peer group interactions, purchases of equipment and services, etc.

Involve a chain of digital documentation - email, bills and receipts, reports, payment records.
Transparency and accountability at different levels led to permissioned/private blockchain formulation.

Content Transaction Chains - General Structure

Here plain content C will be viewed as a bit stream.
Generic format of transactions of the blocks of BS
$< T, X, Y, n, C, hc, hs, fl, bl >$ where,

T - a transaction ID.
- linked to the blockchain ID representing the transactional information chain and also to any applicable *coinchain* ID.
- link will be *null* in the case of pure content transactions

Content Transaction Chains - General Structure

Here plain content C will be viewed as a bit stream.
Generic format of transactions of the blocks of BS
$< T, X, Y, n, C, hc, hs, fl, bl >$ where,

X - the sender ID.
- unique identification tags issued by central authority
- The related payment information, generated by the sender, such as, remittance would be tagged with the corresponding coin transaction ID.

Content Transaction Chains - General Structure

Here plain content C will be viewed as a bit stream.
Generic format of transactions of the blocks of BS
$< T, X, Y, n, C, hc, hs, fl, bl >$ where,

Y - the receiver ID.
- unique identification tags issued by central authority
- The related payment information, generated by the receiver, such as, receipts, delivery order release, would be tagged with the corresponding coin transaction ID.

Content Transaction Chains - General Structure

Here plain content C will be viewed as a bit stream.
Generic format of transactions of the blocks of BS
$< T, X, Y, n, C, hc, hs, fl, bl >$ where,

n - denotes the content count.
- used to denote the chain lengths or block sizes.

Content Transaction Chains - General Structure

Here plain content C will be viewed as a bit stream.

Generic format of transactions of the blocks of BS

$< T, X, Y, n, C, hc, hs, fl, bl >$ where,

C - a content transferred from X to Y

- multiple views of the same content are permissible
- organized in the form of a chain of blocks

Each block B_i represents a chain of transactions.

Transactions in a block B_i by $B_{i,j}$.

The chain structure of transactions in a block can be written as $B_{i,j+1} =< hc||B_{i,j}||hs >$, where $hc = \rho(B_{i,j-1})$ called *input hash*, and $hs = \rho(hc||B_{i,j})$ called *output hash*.

Content Transaction Chains - General Structure

Here plain content C will be viewed as a bit stream.
Generic format of transactions of the blocks of BS
$< T, X, Y, n, C, hc, hs, fl, bl >$ where,

hc, hs hc denotes the input hash, and
hs denotes the output hash value

A pair of hash values for the i-th block, Bh_i, Bf_i, denoting the *input hash* and *output hash* (or the block header and block footer of B_i, with $Bh_i = Bf_{i-1}$, $Bf_i = \rho(Bh_i||B_i)$.

Content Transaction Chains - General Structure

Here plain content C will be viewed as a bit stream.
Generic format of transactions of the blocks of BS
$< T, X, Y, n, C, hc, hs, fl, bl >$ where,

fl, bl - forward and reverse links pointing to the content database using chosen key fields such as date or transaction ID.
- search dictionaries and Merkle trees based on header and footer hash values can be set up to facilitate efficient blockchain search.

Content Transaction Chains - Shared Ledgers

- Subsets of users in a network can have a secured, auditable way of handling distributed transactional and financial information.
- The basic design features have been built to correspond closely to real-world, organizational business processes.
- Such processes have been designed and used to audit whether good, regulatory practices in accounting, management and transparency of operations have been followed.
- In our simulation we study this by generating, a collection of standardized, formatted, transactions in the form of relational tuples and then handling the tuples in a variety of ways of multiple blockchains.

Chains of Blocks of Monetary Transactions

Money and e-Money

- Book *Ascent of Money*, by Niall Ferguson, begins with a,
  "bread, cash, dosh, dough, loot, lucre, moolah, readies, the wherewithal: call it what you like, money matters."
  and ends with "... financial markets are like the mirror of mankind, revealing every hour of every working day the way we value ourselves and the resources of the world around us"
- Physical cash seems attractive for medium and large denominations [ref. 6].
- Paying by cash or cheque than by card (debit or credit) increases one's emotional attachment to the object being bought [ref 6].
- e-Money/ecash: introduced by David Chaum in 1982, led to enormous research on handling various issues of privacy, anonymity, security and double-spending using cryptography principles.
- Earlier an algorithm [ref. 1] for online, transferable ecash system that employed multiple verification authorities, and digital signatures.

Chains of Blocks of Monetary Transactions

VMCoins

VMCoins

- generated and issued by the *Coin Issuing Authority* also aliased as *Coin Supplying Authority* (CSA)
- The CSA server will maintain a coin database, denoted the vault $\mathcal{V}$, which will capture the life of every coin issued.
- The users will be individual customers or commercial merchants, owning digital databases called *wallets*, denoted $\mathcal{W}$, and *tills*,$\mathcal{T}$, respectively.
- the tills will reside in desktop/laptop/tablet/computers or smart-phones and the wallets will reside in the mobiquitous smart-phones or even feature phones

VMCoins - structure

N - number of coins.

The coins are issued initially, in stacks of m coins each as done by the mint of the State.

$$C_i :< H_i, T_i, E_i >,\ i = 1, \ldots, N$$

1. Two main entities - *head*, *tail*, denoted H_i, T_i and an optional entity , *edge* denoted E_i.
2. A coin is identified uniquely, by the stack ID and its sequence number in the stack.
3. The coin format includes requisite information for authentication, accounting and auditing purposes.

VMCoins - structure

N - number of coins.

The coins are issued initially, in stacks of m coins each as done by the mint of the State.

$$C_i :< H_i, T_i, E_i >,\ i = 1, \ldots, N$$

head : $H =< j, d, s >$.

1. Here j, represents the sequence number of the coin in the stack and corresponds to the hash value (stored separately) in a hash chain with $C_i = \rho(C_{i-1}), i = 1, \ldots, m$,
2. C_0 is a pre-specified, initial constant.
3. The last coin in the chain, C_m will be digitally signed by the *Coin Issuing Authority*
4. d, represents the denomination (or token value), and
5. s represents the stack id.

VMCoins - structure

N - number of coins.

The coins are issued initially, in stacks of m coins each as done by the mint of the State.

$$C_i :< H_i, T_i, E_i >,\ i = 1, \ldots, N$$

tail : $T =< s, t, n >$

1. s represents the *payer*,
2. t the *payee* and n the spending count.
3. every transaction is logged by the movement of the circulating coin
4. the payer ID and the payee ID are embedded

VMCoins - structure

N - number of coins.

The coins are issued initially, in stacks of m coins each as done by the mint of the State.

$$C_i :< H_i, T_i, E_i >,\ i = 1, \ldots, N$$

edge : $E =< \lambda, \gamma, \tau >$

1. λ, γ, τ represent, the spatial coordinates of latitude and longitude and the time of transaction, respectively.
2. Above feature is *optional*, and has been included to show that The system can be useful in certain type of tracking of individual coins on a selective or default basis.

VMCoins - Vault, Tills, Wallets

The vault $\mathcal{V}$ is a central dynamic database, containing information on all coin transactions.
The generic format is $< T, X, Y, n, C, hc, hs, fl, bl >$.

T - transaction ID
- linked with coins of the same transaction, through the *fl*, *bl* links
- linked to the blockchain ID representing the transactional information chain that engenders this payment

VMCoins - Vault, Tills, Wallets

The vault $\mathcal{V}$ is a central dynamic database, containing information on all coin transactions.
The generic format is $< T, X, Y, n, C, hc, hs, fl, bl >$.

X - the payer ID from whose wallet the coin for the payment is transferred.
- unique identification tags issued to an individual by the Organization
- would be maintained (chained) in the appropriate blockchain $\mathcal{B}$ and tagged with the current transaction ID T.

VMCoins - Vault, Tills, Wallets

The vault $\mathcal{V}$ is a central dynamic database, containing information on all coin transactions.
The generic format is $< T, X, Y, n, C, hc, hs, fl, bl >$.

Y - the payee ID to whose wallet $\mathcal{W}$ or till $\mathcal{T}$
the payment is transferred.
- all the *four* pair-wise payment transactions T between the two entities, customer and merchant, may be represented uniformly by the denotation payer and payee.

VMCoins - Vault, Tills, Wallets

The vault $\mathcal{V}$ is a central dynamic database, containing information on all coin transactions.
The generic format is $< T, X, Y, n, C, hc, hs, fl, bl >$.

n - denotes the spending count.
- This is initially 0, for a coin C, in the first, issue, transaction i.e., between the CSA and a customer or merchant.
incremented by 1 every spending/transfer
- (optional) can have expiry

Complete framework of VSKChains

- registration : fcid $\rightarrow$ F; userid $\rightarrow U_i$; vendorid $\rightarrow V_j$;
- in generic transactions involve entities: F, U, V and icash $P_{..}$
- K_{FU} = gensymkey (F, U); exchange K_{FU}; K_{FV} = gensymkey (F, V); exchange K_{FV}; K_{UV} = gensymkey (U, V); exchange K_{UV};
- U $\rightarrow$ F : rqst $eca(val) = ecashadvance(val)$;
- at F : val $\rightarrow$*coinchain* / *couponchain* / *tokenchain* (with **VMcoin**);
- $T0_{FU} =< tid, eca(val), U >$;
- F $\rightarrow$ U : $T1_{FU} =< s_F(H(T0_{FU}), E_{K_{FU}}(val)) >$; ecash P_{FU}
- **iblockchain** : $\uparrow ibc(T1_{FU}, P_{FU})$; (with **SVKCS**);
- U $\rightarrow$ V : transactions T_{UV}, ecash payments P_{UV};
- **iblockchain** : $\uparrow ibc(T_{UV}, P_{UV})$;
- V $\rightarrow$ F : transactions, ecash payments $\rightarrow$ electronic payment advice / cheque / IMPS / NEFT / RTGS settlements P_{FV};
- $T_{FV} = E_{K_{FV}}(transaction)$; **iblockchain** : $\uparrow ibc(T_{FV}, P_{FV})$;
- U $\rightarrow$ F : eca(val) settlements; **iblockchain** : $\uparrow ibc(T_{UF}, P_{UF})$;

Digital Storage Space

- Currency in circulation in India as on 8 November 2016, was 93 Bn (billion) coins (of value Rs.211 Bn) and 87 Bn coins (of value Rs. 16 Tn (trillion)).
- VMCoin requires 12 bytes for the meta data and 64 bytes for the hash values and another 52 bytes for other auxiliary pointers.
- the 180 Bn coins, would require 32 Tn bytes.
- Can be spread over about 1024 nodes (corresponding to the district, state and central servers) in about a manageable system of 32 Gigabyte storage servers.

Note: VMCoin can represent any kind of tokens

VSK hash function

Hash Functions

- Two main cryptography algorithms used
  - Digital Signatures
  - Hash functions
- We introduce new hash function called SVK hash functions

VSK hash function

Certain Sequences of Arithmetic Progressions(AP)

Certain Sequences of Arithmetic Progressions(AP)

- **Notation**: $A(a, d)$ denotes the AP with leading term a, and common difference d,
  i.e., $A(a, d) : a, a + d, a + 2d, \ldots$.
- Given $a_0, d_0 \in \mathbb{Z}$, s.t. $\gcd(a_0, d_0) = 1$, construct a sequence of arithmetic progressions
  $\mathfrak{S}(a_0, d_0) = \{A(a_0, d_0), A(a_1, d_1), A(a_2, d_2), \ldots\}$ with the terms of the progressions satisfying an *invertibility property* $\mathcal{P}$.
  - i.e., for any $i, j \geq 0$
    $(a_i + jd_i)(a_{i+1} + jd_{i+1}) \equiv 1 \pmod{a_i + (j+1)d_i}$.
  - In other words, for any two consecutive progressions A, A' in the collection, the j^{th} terms of A, A' are multiplicative inverses of each other modulo the $(j+1)^{th}$ term of A.

Examples

Given co-prime pair $(11, 25)$,

$A(11, 25)$: 11, 36, 61, ...

A sequence of APs $\mathfrak{S}(11, 25)$ with invertibility property $\mathcal{P}$ is:

$$
\begin{array}{lcccccc}
A(11,25) & : & 11, & 36, & 61, & 86, & \ldots \\
A(23,16) & : & 23, & 39, & 55, & 71, & \ldots \\
A(17,7) & : & 17, & 24, & 31, & 38, & \ldots \\
A(12,5) & : & 12, & 17, & 22, & 27, & \ldots \\
A(10,3) & : & 10, & 13, & 16, & 19, & \ldots \\
A(4,1) & : & 4, & 5, & 6, & 7, & \ldots \\
A(4,1) & : & 4, & 5, & 6, & 7, & \ldots \\
 & & & \vdots & & &
\end{array}
$$

Interesting Facts

- One can construct many sequences of APs for given co-prime pair (a_0, d_0).
- If common differences are in non-increasing order then there exists a *unique* sequence of APs.
  In previous example: $\{25, 16, 7, 5, 3, 1, 1, \ldots\}$.
- Always sequence of common difference ends with 1
- Hence only *finitely many distinct* arithmetic progressions in a collection.

Observed Facts

- Given co-prime pair (a_0, d_0), let $\mathfrak{S}(a_0, d_0)$ denote the finite (and unique) sequence of APs satisfying:
  1. the terms of the progressions satisfy invertibility property $\mathcal{P}$
  2. the common differences of the progressions are in non-increasing order.
- Construction of $\mathfrak{S}(a_0, d_0)$, imitates the *Euclidean GCD algorithm*
- Number of terms in $\mathfrak{S}(a_0, d_0)$ will be $O(\log(d_0))$
- Total time complexity is $O(\log^3 d_0)$
- Computation of leading terms of each progression in $\mathfrak{S}(a_0, d_0)$ is easy

Grouping in $\mathfrak{S}(a_0, d_0)$

- A sub-collection $\mathcal{G}$ of consecutive progressions of $\mathfrak{S}(a_0, d_0)$ is called a *grouping* if it satisfies the following two properties:
  1. the difference between the common differences of any two consecutive progressions in $\mathcal{G}$ is same,
  2. $\mathcal{G}$ is maximal.
- The difference between the consecutive common differences called the *second common difference* corresponding to $\mathcal{G}$
- The number of progressions in a grouping $\mathcal{G}$ as the size of $\mathcal{G}$, and denoted by $|\mathcal{G}|$.
- **Note:** any two consecutive groupings share an arithmetic progression

Example: Groupings in $\mathfrak{S}(11, 25)$

$\mathfrak{S}(11, 25)$ has two groupings:

$\mathcal{G}_1 = \langle A(11, 25), A(23, 16), A(17, 7) \rangle$,

$\mathcal{G}_2 = \langle A(17, 7), A(12, 5), A(10, 3), A(4, 1) \rangle$.

The second common difference corresponding to $\mathcal{G}_1$ is 9.

The second common difference corresponding to $\mathcal{G}_2$ is 2.

The Groupings share the progression $A(17, 7)$.

The sizes of $\mathcal{G}_1$ and $\mathcal{G}_2$ are 3 and 4, respectively.

Grouping in $\mathfrak{S}(a_0, d_0)$

Given (a_0, d_0),

- easy to compute number of grouping in $\mathfrak{S}(a_0, d_0)$
- easy to compute size of any grouping in $\mathfrak{S}(a_0, d_0)$
- easy to compute any member of any grouping in $\mathfrak{S}(a_0, d_0)$
- Total time complexity is $O(\log^3 d_0)$

Inverse: What about computing (a_0, d_0), given *certain* terms of $\mathfrak{S}(a_0, d_0)$?

VSK hash function

Inverse Problem

A Problem on Leading terms

- *For a given m integers $f_1, f_2, \ldots, f_m$, does there exist a pair of integers x and y such that f_j, $1 \leq j \leq m$, are the leading terms of some progressions in $\mathcal{A}(x, y)$?*
- Sub cases of the problem
  - (**Case 1.**) Given leading terms of any *three* consecutive progressions in $\mathfrak{S}(x, y)$, we can compute pair (x, y), uniquely, with time complexity polynomial in size of input.
  - (**Case 2.**) Given f_1, f_2, the number of pairs x, y s. t. f_1, f_2 are the leading terms of some consecutive progressions in $\mathfrak{S}(x, y)$ is bounded by the number of divisors of $f_1 f_2 - 1$. Computing pairs is equivalent to factoring integers.

For more details refer 2,3 and 4.

VSK hash function

Hash function based on $\mathfrak{S}(a_0, d_0)$

Hash function based on $\mathfrak{S}(a_0, d_0)$

- Fix a master parameter d_0.
- A hash function G is defined as follows.
  For an input a_0, the hash value $y = G(a_0, d_0)$ is given by a term of some random progression of the collection $\mathfrak{S}(a_0, d_0)$.
- The index of the chosen random progression is predetermined.

Efficiency:

- In case of G, the cost is quadratic in $\log_2 d_0$.
- Experiments study: for 1024-bit d_0 and $a_0 < d_0$, the cost of producing a term of $\mathfrak{S}(a_0, d_0)$ is 173 micro seconds on 2.7GHz, 64-bit (GMP library).

Advantage of master parameter d_0:

- Different category transactions uses different d_0

Generation of VM coins

- Standard method: apply a cryptographic hash function successively k times to a seed s_0
- output of each application of the hash function is considered as a *coin*
- Each coin in the stack is associated with the stack ID number.

Using SVK hashchains:

- generation of coins makes use of the object $\mathfrak{S}(a_0, d_0)$.
- different denomination uses different master parameter d_0.
- random $a_0 (< d_0)$ is chosen as the seed to produce a stack of coins c_1, c_2, $\ldots$, c_k where
  - c_i is the leading term of a progression of r_i^{th} grouping of $\mathfrak{S}(a_0, d_0)$.
- The chosen progression should belong to distinct groupings, i.e., $r_1 < r_2 < \ldots < r_k$.

Note: The integrity of coins is dependent on the difficulty of finding another number a_0' so that the sequence $\mathfrak{S}(a_0', d_0)$ produces the same coins.

Computing Hash of a Transaction

- Transaction $X = \langle x_1, x_2, \ldots, x_t \rangle$
- Each x_i is a l-bit binary data chunk.
- a naive method of computing the hash of the transaction X:
  Set $HO = 0$,
  For $1 \leq i \leq t$ do the following
  - $x_i' \leftarrow M(HO, x_i)$
  - $H_i \leftarrow Bits(x_i', L_i, D_i)$
  - $HO \leftarrow HO \oplus H_i$
- For example, M is defined as $x_i' = 32\text{-bit-LSB}(HO) \oplus x_i$.
- The value of H_i is a 256-bit integer extracted using x_i'-th term of the progression $A(L_i, D_i)$ of $\mathfrak{S}(a_0, d_0)$.

Note: If the number of chunks of the transaction X exceeds the number of groupings of $\mathfrak{S}(a_0, d_0)$, then the pair (a_0, d_0) is updated to a new pair (a_0', d_0').

Conclusion

- presented a design of an integrated system of chains of blocks of transactions for handling both content and currency
- new ideas of (i) digital denominational coins and (ii) hash function based on arithmetic progressions
- It is centralized, but can be converted to public chains
- Some prototype have been tested.

References

References

1. R. Sai Anand, C. E. Veni Madhavan, "An Online, Transferable E-Cash Payment System", Progress in Cryptology - INDOCRYPT 2000.
2. Ch. Srikanth, H.V. Kumar Swamy and C.E. Veni Madhavan, "New Cryptographic Systems based on Certain Sequences of Arithmetic Progressions", National Workshop on Cryptography, 2016
3. Ch. Srikanth, "Computational and Number Theoretic Aspects of Certain Collections of Arithmetic Progressions", Indian Institute of Science, 2017.
4. Ch. Srikanth, C.E. Veni Madhavan and H.V. Kumar Swamy, "Family of PRGs based on Collections of Arithmetic Progressions", Cryptology ePrint Archive, Report 2017/324, 2017.
5. A. M. Shah, N. Eisenkraft, J. R. Bettman and T. L. Chartrand, "Paper or Plastic?": How we pay influences post-transaction connection, Journal of Consumer Research, 2016
6. G. Amromin and S. Chakravorti, Debit card and cash usage: a cross-country analysis", Federal Reserve Bank of Chicago, Report Wp 2007-04, 2007

Thank You