General Insurance Seminar

Connecting Today and Tomorrow

13 – 15 November 2016

Grand Hyatt
Melbourne

Actuaries Institute

This presentation has been prepared for the 2016 General Insurance Seminar.
The Institute Council wishes it to be understood that opinions put forward herein are not necessarily those of the Institute and the Council is not responsible for those opinions.

Blockchain, smart contracts and potential insurance application

D. Semenovich

- What exactly are blockchain and smart contracts?
- Some existing appliactions (decentralised lotteries, prediction markets).
- Potential insurance use cases.

What is the blockchain?

White papers talk about “automated audit”, “unprecedented financial services innovation”, “sales disruption” and the like, enabled by “the blockchain” — claims that are sure to leave a jaded practitioner skeptical.

Public blockchain technology actually consists of three relatively independent components:

- direct application of public key cryptography and related ideas to the facilitation of financial transaction,
- a significant innovation in distributed consensus algorithms,
- smart contracts.

Cryptographic hash functions

The basic primitives underlying blockchain system, *cryptographic hash functions*, accept a string of characters of any length and return a fixed length output, e.g. 16 bytes, while satisfying some additional properties:

Collision resistance means that it is difficult to find two such distinct values x and y that cause a collision, namely that $H(x) = H(y)$. While very many such pairs exist (by Dirichlet's principle), finding a collision for a given x should be computationally infeasible.

Hiding property means that given the output v of a hash function $v = H(r + x)$ it is impossible to discover what the input x was, where $+$ denotes concatenation and $r \in R$ is drawn uniformly at random from a sufficiently large set R.

Blockchain data structure

Cryptographic hash functions can be used to generate digests of arbitrary information, protecting the data against tampering — once the hash value of a file is computed, all one needs is this hash to verify the authenticity of the original file.

A series of files or data blocks where each block contains data as well as a hash of the previous block is called a *blockchain*.

This structure prevents tampering with any of the blocks as long as we know the hash of the latest block and new blocks can be added as required. Bitcoin uses blockchain to store its transaction history.

Digital signatures

This is another essential component of public blockchain technology.

The party that intends to digitally sign messages must generate a pair of values (s, p) jointly satisfying certain properties. Here s is the *secret key* that needs to be kept private and p is the *public key* made known.

- $v = \mathrm{sign}(s, m)$ — signing function takes a message m and a secret key s and returns signature v.

- $\mathrm{validate}(p, m, v)$ — validation function takes a public key p, a message m, and a signature v and evaluates to true if and only if the signature was indeed generated on the same message using the matching secret key.

These functions then satisfy the condition:

$$\mathrm{validate}\left(p, m, \mathrm{sign}(s, m)\right) = \mathrm{true}\,.$$

Final requirement is that it is not computationally feasible to fake signatures.

Public keys can be used to *identify* a person — in order for someone to speak for that identity they must have access to the corresponding secret key. Identity creation in this situation is completely decentralized, anyone can create (s, p) pairs at any time without notification.

In Bitcoin and other similar system, public keys are used as payment *addresses* or account numbers.

Cryptocurrency

Lets consider how to implement a simple payment system using the ideas so far.

We will need two types of transactions — one to *create* new coins by fiat and another to *pay* coins from an existing owner to a new owner. Transactions will be recorded in a blockchain data structure.

Only the scheme operator is authorized to *create* new coins. To do so, they sign a transaction containing coin id, its value and public key to which the coin is assigned.

Anyone who owns any coins is authorized to *pay* their coins to someone else. This transaction deletes some coins and creates new coins of the same total value assigned to a new set of public keys. The transaction is signed by the owner of the coins spent.

Scheme operator recognizes the payment transaction as valid and appends it to the blockchain if the following conditions are satisfied:

- the deleted coins are active — they were created in a previous transaction and not already deleted,
- the value of the coins deleted equals to the value of coins created,
- the transaction is signed by the secret key corresponding to the private key to which the coins were previously assigned (only the operator can create new coins by fiat).

The operator then publishes and signs the new block containing the transaction. The operator is unable to fake history of transactions as it would invalidate the blockchain or make payments on behalf of those public keys where they don't also control the secret key.

Distributed consensus

The major breakthrough of Bitcoin was the decentralised version of the procedure described in the previous section, eliminating the single point of failure in the operator.

A simplified version of the Bitcoin consensus protocol for payment transactions (ignore coin creation for the moment) is as follows:

- new payment transactions are broadcast to all participants,
- each participant collects new valid transactions (in the sense described earlier) into a block,
- at regular intervals a randomly chosen participant broadcasts its block,
- other participants accept the block only if all transactions are valid,

- participants demonstrate that they have accepted the block by including it in their version of the blockchain (i.e. its hash will appear in the next block that they collect).

This scheme eliminates the dependency on the central authority and has reasonable properties — invalid transaction blocks (e.g. someone trying to spend coins they dont own) get rejected by other (honest) participants, splits can be dealt with &c.

There are still two problems — we have no way to create new coins and in practice there is no reliable way to randomly choose a participant to nominate new block or to ascertain how many participants there actually are.

Proof of work

Bitcoin offered yet another ingenious solution to both of the earlier questions.

Instead of being chosen randomly to announce the next block, participants instead compete at solving cryptographic puzzles. This is called a *proof-of-work* scheme.

For a new block to be valid, it must contain a fixed length number, called *nonce*, such as that the hash of the nonce appended to the rest of the block falls below certain target value:

$$H(\mathrm{nonce} + \mathrm{new\_block}) < \mathrm{target}\,.$$

Since there is no better known strategy for finding nonces than simple enumeration, the participants win the competition essentially randomly, with the chance of winning proportional to the computational power they can bring to the problem. Competing to find nonces is also called *mining*. This approach makes it much more costly for malicious players to subvert the system.

Finally, to incentivize miners to solve hashing puzzles for new blocks, Bitcoin protocol allows them to include a transaction that *creates* coins of certain predefined value by fiat and assign it to the address of their choice.

Smart contracts

Bitcoin does not simply store addresses of coin recipients for payment transactions but instead allows for small custom programs to be executed that check validity of a supplied public key.

This is a very powerful idea as it makes transactions themselves programmable. Over time it was possible to use this functionality to implement ne Bitcoin applications without changing the underlying protocol. Examples include schemes that allow funds only to be claimed if k out of n potential beneficiaries supply their signatures, decentralised lotteries and betting.

More generally, these programmable transactions are known as *smart contracts*.

Applications of blockchain - loterries

A lottery is not dissimilar to an insurance pool — a large number of people deposit their money with a single counterparty who then disburses most of the money collected to a few randomly chosen individuals (after taking an often substantial fee).

Bitcoin and similar public blockchain systems offer a fascinating way to implement reliable distributed lotteries without a trusted central party (and potentially at much lower cost).

Lets say we have only 3 participants, called A, B and C. They send money to a specially crafted Bitcoin script which then pays out the total contribution back to one of them at random. The algorithm is as follows:

- Each participant picks a number — A chooses x, B chooses y and C chooses z. They then communicate $H(x)$, $H(y)$ and $H(z)$ together with their payment.

- A, B and C create a new transaction now incorporating x, y, and z. The payment is made to the $(x + y + z)\ \%\ 3$ participant.

This scheme has certain undesirable properties but rectifying them is quite involved. Much more realistic proposals already exist and it is likely we will see fully featured distributed cryptocurrency based lotteries before long.

Applications of blockchain - prediction markets

Another significant area of interest are so called *prediction* or *betting markets*. These offer a way for people to place bets on outcomes of a diverse range of events, from sports to elections and corporate financial results . Prediction markets are also favoured by economists as a mechanism to efficiently aggregate information from multiple sources.

Main components of a prediction market are as follows:

- A mechanism to accept funds into *escrow* and make payouts according to event outcome. This broadly corresponds to payment processing and treasury or capital management function of an insurance company.

- An *arbitration process* for determining the outcomes in question. Arbitration can be both decentralized (by consensus of market participants, another group such miners or even another market) and centralized, provided by a trusted third party (any signed data feed can ultimately be used). Further questions arise when the outcomes are ambiguous or not a matter of public record. In insurance context this would correspond to claims assessment.

- An *order book* or similar mechanism (such as a market maker relying on a *scoring rule*) for participants to find counterparties to trade with. An order book contains *bids* and *asks*. A bid is a buy order and ask is a sell order. Typically the ask price is higher than the bid, otherwise two participants are matched up and a trade occurs, eliminating one of the orders. This has no close analogue in insurance outside of risk securitasion.

Following is an example of how a very simple bet can be implemented using the Bitcoin protocol:

- An arbitrator creates two pairs of keys (s_0, p_0) and (s_1, p_1) for No and Yes outcomes of a certain event and publishes the public keys. Once the outcome is known, they also publish the corresponding secret key.

- A wishes to bet on a Yes outcome and B on a No outcome. They deposit money to a bitcoin script from which payments can be withdrawn either using signatures from A and Yes or B and No.

Potential insurance use cases

Digital certificates of insurance — using digital signatures on certificates of insurance to allow decentralised verification.

Insurance pools — these are very similar to decentralised lotteries just with a different set of rules to determine the beneficiaries, arbitration can be implemented either through vote of participants or a trusted third party.

Insurance markets — essentially the same as prediction markets (may need "insurable interest" constraints), insurance linked securities with parametric triggers already come very close.

Core systems for smart contracts — internal systems implementing a centralised version of blockchain can allow rapid innovation in product design.

Conclusions

- Many insurance use cases do not necessarily require full public blockchain.

- Early successes will likely be those that can leverage work in lotteries and prediction markets. Low friction automated outcome arbitration appears to be the key.

- Ultimate possibilities of public blockchain technology are well beyond the imagination of this author.