Radoslaw Krzeski
Summary of Topic XI – Smart Contracts

Formalizing and Securing Relationships on Public Networks
The article begins by using an interesting analogy, where a contract is a set of promises through "the meeting of minds", and the formalization of this relationship. Contracts and principles have been encoded into law of mankind history. It is argued that this formalization does take time, and such poses new challenges in the digital sphere, where laws are new or not formalized. Deriving from exciting contracts we can shorten this creation of new contracts and enforcing institutions significantly. Different protocols allow for this functionality to be implemented in the digital world.

Previous attempts have failed to cross the worlds of economics and cryptography, but meet in the concept of smart contracts. They utilize protocols and user interfaces to facilitate all steps needed in contract, and to formalize digital relationships.

An example is used where the lease of a Smart Car can be controlled through smart contracts. The usage of the word controls is imminent, as such to describe the data flow to ensure integrity, authorization and critical forms. Previous control has typically been around amounts of money and quantity of good.
Paper contracts are designed to be static, place little emphasis on confidentiality and are often based on authorization from upper management. Checksums and bases reconciliations are primitive compared to cryptographic hashes ,
(and when viewed from within a business also entails less asymmetry and more direct one-to-one relationship between owners and other employees of a contract).

Types of controls:
Imprest: Family of control involving receipt or disbursement of bearer certificate (movie theatre/tickets as an example)
Customer Audit: where the customer needs to generate initial documentations.
Segregation of duties: where multiple parties discourage of avoid the possibility of fraud.

As controls(paper-based) have loosened due to the flattening of hierarchies, they have led to scandals in investment and banking industry. A learned trade-off must be found, where there must be more explicit contracts between owners and employees and symmetric formalizations between employees.

Keyword: EDI- Electronic Data Interchange; Client-to-Client communication of standardized business transactions between organizations. Enables more rapid execution of negotiations and performance monitoring, but lack the full abilities of a smart contract. A wide variety of contract terms and agreements have already been formalized using EDI`s.

Good contract design stress two important contract designs: observability and verifiability by third parties. The article derives a third design however; privity (recognized by law, and where "contents and performance of a contract should be distributed among parties only as much as is necessary of the performance of that contract" (used as means for protection against the third party, provide a clear boundary)).

These properties also have mental and computational costs:
Mental has the cost of anticipating, agreeing to, and writing down the various eventualities. These can be derived from a function, with the ability to calculate cost from a given set of eventualities. Compared to the traditional EDI, semantics have a more natural place in SC`s.

A smart contract should have contracting phases, and as such marked by the Ex-ante and Ex-post phases. They also often use a trusted third-party, and "adjudicator". Hidden knowledge and hidden actions are also mention as abilities of a smart contract, but deemed viewable if observability is of concern. It is a both a blessing and a curse, where too much information hinders action, to little becomes a privacy concern.

In essence, the ability that cryptographic protocols provide us with basic messaging channels, an obscure hash pattern can then provide the initialization of smart contracts. This is often used by the means of private and public keys, along with digital signatures.

<u>Other keywords in the article</u>: Post-unforgeable Transactions logs, Mutually Confidential Computation, Bearer Certificates, Unlikable Transfers, Conserved Objects, Digital Cash, Credentials, Watermarks, Pseudonymous Credit Ratings, Secured Credit, Ripped Instruments, Interval Instruments,

Blockchains and Smart Contracts for the Internet of Things
This paper aims to examine if blockchain technology is a good fit for the IoT-sphere. It reviews the peer-to peer network the usually use, and how trust work in between them without an intermediary. Further on the paper looks at how this

combination between IoT and blockchain facilities the sharing of services and resources to create marketplaces. These are to be cryptographically verified in many different workflows. Privacy and transactional values are also discussed.

The main spark in interest among institutions seems to be the ability to avoid an intermediary to perform decentralized transactions. The lack of authority is promising and hopefully resource-saving (trustless networks). The addition of smart contracts that reside on the chain is also interesting. Apart from the general introductions to blockchain abilities, the paper discusses how Nick Szabo introduced the concept of Smart Contracts back in 1994. IT assumes a network where every node runs a virtual machine able to run the code in SC`s, and that the blockchain network acts a distributed VM.

They excel when tasked to manage data-drive interactions between entities on the network, or as exemplified in the article, an account based account & withdrawal system. It also argues that a proper blockchain should not have any non-deterministic behavior, in the way that such behavior is not possible or rejected by the contract as invalid.

DAO`s are also mentioned – decentralized autonomous organizations, constructs that may change behavior if it is coded. Further on, a few terms and definitions are made in attempt to split the different types of blockchain networks:

- Public & permissionless networks vs private and permissioned networks
- Ability to transact or mine
- UTXCO transactions or smart contracts (account-based, arbitrary logic)

An interesting statement is made where the operationalized cost of keeping IoT devices updated is rising, along with users being prone to dislike tracking and gathering of data. Blockchains seem to be a perfect fit for this, and as such a use case is imagined, where a company deploys a "smart contract that allows them to store the hash of the latest firmware update on the network".
Devices know about this, either by having it baked into them, or probing for a discovery service. After enough devices have the hash and firmware, the nodes sustain the SC`s long after the manufacturer shut down the original node.
Add a billing layer and you have a marketplace (device may or may not charge fee to cover infrastructure costs).
Slock, with its locked doors are covered as an real-life example. TransActive Grid is another, Filament a third.

Issues that may arise in this world, is the risk of a lower transaction processing throughput, due to the nature of blockchains. Scalability is an ongoing issue. Sharding is an option that is still lacking.
Maintaining privacy is another issue, as all transactions are open and inspectable by the ledger, patterns can be made. Miners can also be unfair, and censor block proposment. Penalizations system should be in place. Which bring to light that legal enforceability might be limited. Few organs are equipped to handle faulty contracts and fraud in the digital chain. A dual integration with real-world rules might be the solution.

Other problems mentioned were; the expected value of tokens, whether the full disclosure of contracts is a good thing, or self-destruct issues where contracts to not perform as intended. Other support services (DNS and message channel) might be needed.

Step by Step Towards Creating a Safe Smart Contract
The article begins by outlining that since cryptocurrencies move values, they should also have a consistent focus on the security aspect. The draw their experience from teaching the topic to undergraduate students. They also made a smart contract course open for all, which might be interesting to check out.

The class created smart contracts with "real currency" in a "play-for-keeps". If inserted into buggy code you risk losing the currency – requires economic thinking. Contracts must be written to ensure fairness, and watch out for cheats. Extra scrutiny required when the students were using the Ethereum Serpent language.

Apart from the introduction to the underlaying chain that smart contract appends to, and how they run/execute on all nodes, the article describes the basic attributes of a contract: program code, a storage file, and an account balance. Contracts can be created by posting a transaction to the blockchain. Becomes fixed when "uploaded". Program code is executed by all miners if activated by a transaction/message, from another user or contract. Trusted as such as a third party for correctness and availability, but not for privacy. Since the lab uses Ethereum, Gas also becomes a variable to watch for.

The lab had two phases: Creation phases, where contracts were created of free choice. Issues were discussed. Then Amendment phase came along, where students were given option to correct their code. Three classes of typical mistakes were seen:

- Logical errors (leaking money, third player inserted into Rock, Paper, Scissor game, plaintext messages, neglected state check)
- Not using cryptography (cleartext once again, epoch need to be used, biding and hiding)
- Misaligned Incentives (deviating player escapes a loss, can be controlled by deadlines) + call-stack bug (1024KB)

The article concludes that Serpent is still under development, and that smart contracts will benefit from language improvement in the future. It also argues that smart contract might be a smart way into the subject of cybersecurity due to the close nature to cryptography.

References:

Formalizing and Securing Relationships on Public Networks
Nick Szabo
http://ojphi.org/ojs/index.php/fm/article/view/548/469

Section II.D in Blockchains and Smart Contracts for the Internet of Things
Konstantinos Christidis and Michael Devetsikiotis
http://ieeexplore.ieee.org/document/7467408/

Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab
Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi
https://eprint.iacr.org/2015/460.pdf