More recent papers and essays on smart contracts, commercial controls and security.

Smart Contracts

Glossary

A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitration and enforcement costs, and other transaction costs[1].

Some technologies that exist today can be considered as crude smart contracts, for example POS terminals and cards, EDI, and agoric allocation of public network bandwidth.

Digital cash protocols[2,3] are fine examples of smart contracts. They enable online payment while honoring the characteristics desired of paper cash: unforgeability, confidentiality, and divisibility. When we take a second glance at digital cash protocols, considering them in the wider context of smart contract design, we see that these protocols can be used to implement a wide variety of electronic bearer securities, not just cash. We also see that to implement a full customer-vendor transaction, we need more than just the digital cash protocol; we need a protocol that guarantees that product will be delivered if payment is made, and vice versa. Current commercial systems use a wide variety of techniques to accomplish this, such as certified mail, face to face exchange, reliance on credit history and collection agencies to extend credit, etc. Smart contracts have the potential to greatly reduce the fraud and enforcement costs of many commercial transactions. Digital cash protocols use several of the rich new building blocks coming out of the fields of cryptography and computer science. Most of these components have not yet been widely exploited to facilitate contractual arrangements, but the potential is vast. These subprotocols include Byzantine agreement, symmetric and asymmetric encryption, digital signatures, blind signatures, cut & choose, bit commitment, multiparty secure computations, secret sharing, oblivious transfer, and multiparty secure computation. All of these except the first are described in [2,3].

The consequences of smart contract design on contract law and economics, and on strategic contract drafting, (and vice versa), have been little explored. As well, I suspect the possibilities for greatly reducing the transaction costs of executing some kinds of contracts, and the opportunities for creating new kinds of businesses and social institutions based on smart contracts, are vast but little explored. The "cypherpunks"[4] have explored the political impact of some of the new protocol building blocks. The field of Electronic Data Interchange (EDI), in which elements of traditional business transactions (invoices, receipts, etc.) are exchanged electronically, sometimes including encryption and digital signature capabilities, can be viewed as a primitive forerunner to smart

contracts. Indeed those business forms can provide good starting points and channel markers for smart contract designers.

One important task of smart contracts, that has been largely overlooked by traditional EDI, is communicating the semantics of the transaction to the parties involved. There is ample opportunity in smart contracts for "smart fine print": actions taken by the software hidden from a party to the transaction. For example, grocery store POS machines don't tell customers whether or not their names are being linked to their purchases in a database. The clerks don't even know, and they've processed thousands of such transactions under their noses. Thus, via hidden action of the software, the customer is giving away information they might consider valuable or confidential, but the contract has been drafted, and transaction has been designed, in such a way as to hide those important parts of that transaction from the customer.

To communicate transaction semantics well, we need good visual metaphors for the elements of the contract. These would hide the details of the protocol without surrendering control over the knowledge and execution of contract terms. A primitive but good example is provided by the SecureMosiac software from CommerceNet. Encryption is shown by putting the document in an envelope, and a digital signature by affixing a seal onto the document or envelope. On the other hand, Mosaic servers log connections, and sometimes even transactions, without warning users -- classic hidden actions.

Another area that might be considered in smart contract terms is synthetic assets[5]. These new securities are formed by combining securities (such as bonds) and derivatives (options and futures) in a wide variety of ways. Very complex term structures for payments (ie, what payments get made when, the rate of interest, etc.) can now be built into standardized contracts and traded with low transaction costs, due to computerized analysis of these complex term structures. Synthetic assets allow us to arbitrage the different term structures desired by different customers, and they allow us to construct contracts that mimic other contracts, minus certain liabilities. As an example of the latter, synthetic assets have been constructed that mimic the returns of stocks in German companies, without requiring payment of the tax foreigners must pay to the German government for capital gains in German stocks. It's important to note that these synthetics do _not_ confer voting rights as do the originals. It might be possible to add smart contract protocols to transfer voting rights to the synthetic. Of course, these protocols might have to be quite secure to withstand attacks from the third party jurisdiction, whose transaction cost (the tax) is being arbitraged away by the synthetic asset.

Finally, we can extend the concept of smart contracts to property. Smart property might be created by embedding smart contracts in physical objects. These embedded protocols would automatically give control of the keys for operating the property to the agent who rightfully owns that property, based on the terms of the contract. For example, a car might be rendered inoperable unless the proper challenge-response protocol is completed with its rightful owner, preventing theft. If a loan was taken out to buy that car, and the owner failed to make payments, the smart contract could automatically invoke a lien, which returns control of the car keys to the bank. This smart lien might be much cheaper and more effective than a repo man. Also needed is a protocol to provably remove

the lien when the loan has been paid off, as well as hardship and operational exceptions. For example, it would be rude to revoke operation of the car while it's doing 75 down the freeway.

Smart property may be a ways off, but digital cash and synthetic assets are here today, and more smart contract mechanisms are being designed. So far the design criteria important for automating contract execution have come from disparate fields like economics and cryptography, with little cross-communication: little awareness of the technology on the one hand, and little awareness of its best business uses other. The idea of smart contracts is to recognize that these efforts are striving after common objectives, which converge on the concept of smart contracts.

References:
[1] _The New Palgrave: Allocation, Information, and Markets_
[2] Bruce Schneier, _Applied Cryptography_ (digital cash
objectives are on pg. 123)
[3] _Crypto_ and _Eurocrypt_ conference proceedings, 1982-1994.
[4] "Crypto Rebels", Wired #2, also cypherpunks mailing list
(mail to majordomo@toad.com with body "subscribe cypherpunks")
[5] Perry H. Beaumont, _Fixed Income Synthetic Assets_

More recent papers and essays on smart contracts and the history and future of commercial controls and security.