Formalizing and Securing Relationships on Public Networks

Nick Szabo

Originally published in 1997

Contents

Introduction

History has seen successive revolutions in the costs of doing global business. First transportation, then manufacturing, and recently communications costs have fallen dramatically. Yet there are still major barriers to doing business internationally. The cost of doing business globally is increasingly dominated by issues of jurisdiction, security, and trust: the costs of developing, maintaining, and securing our relationships.

Despite the recent rise of global computer networks, our institutions still take for granted that we live in a world of paper. We formalize our relationships with written contracts, written laws, and forms designed for paper. Our attitudes and laws regarding intellectual property and privacy have assumed a world of paper which is costly to copy. Increasingly, we can no longer take these deeply embedded, highly evolved paper institutions for granted. Nor, since these institutions involve complex human relationships, can we redesign them overnight. We are entering a period where civilization must once again adapt to a radical new media.

Over the long stretch civilization, paper represents only one of many technologies used to mediate commercial relationships. The Inca used quipu – accounts encoded on strings, a system with interesting tamper-resistance properties. Early Middle Eastern civilizations used clay tokens for thousands of years. These combined the function of, and were a precursor to, both cuneiform writing and coins. Coins started out as lumps of standardize metal and weight. Since these were too expensive to test during a normal business transaction, they came to be stamped by reputable or powerful authorities. Coins played a major role in commerce for thousands of years, but that era is now over.

Business is now dominated by paper and institutions of written literacy. Security measures have included chops, seals, and written signatures. Value has been transered via bills of exchange (which evolved into checks), bearer certificates, and accounts using the double-entry bookeeping system. Most importantly, we take for granted that contracts and law are written on this static medium, to be interpreted and enforced by human authorities.

We are now entering an era of online communications and software "literacy". The "physics of cyberspace", studied by computer scientists, are radically different from the properties of paper, to an even greater degree than paper was different from string, clay, and metal. Not only written but also aural, visual, and other sensory media can be combined. Most importantly, digital media are dynamic – they not only transmit information, but can also make some kinds of decisions. Digital media can perform calculations, directly operate machinery, and work through some kinds of reasoning much more efficiently than humans.

The movement from static to dynamic media promises to bring about a fourth cost revolution in the related areas of jurisdiction, trust, and security. Impacts on business will be felt in law, accounting, auditing, billing, collections, contracts, confidentiality, and so on: in short, the entire nature of our business relationships will be altered in ways only partially foreseeable.

The main traditional way to formalize a business relationship is the contract, a set of promises agreed to in a "meeting of the minds". We naturally think of contracts as written, but oral agreements are also considered contracts, and have been around since prehistory. The contract is the basic building block of a market economy. Over many centuries of cultural evolution has emerged both the concept of contract and principles related to it, encoded into common law. Such evolved structures are often prohibitively costly to rederive. If we started from scratch, using reason and experience, it could take many centuries to redevelop sophisticated ideas like contract law and property rights that make the modern market work. But the digital revolution

challenges us to develop new institutions in a much shorter period of time. By extracting from our current laws, procedures, and theories those principles which remain applicable in cyberspace, we can retain much of this deep tradition, and greatly shorten the time needed to develop useful digital institutions.

Computers make possible the running of algorithms heretofore prohibitively costly, and networks the quicker transmission of larger and more sophisticated messages. Furthermore, computer scientists and cryptographers have recently discovered many new and quite interesting algorithms. Combining these messages and algorithms makes possible a wide variety of new protocols. These protocols, running on public networks such as the Internet, both challenge and enable us to formalize and secure new kinds of relationships in this new environment, just as contract law, business forms, and accounting controls have long formalized and secured business relationships in the paper-based world.

In electronic commerce 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. These efforts are striving after common objectives, and converge on the concept of smart contracts[1].

Smart contracts reduce mental and computational transaction costs imposed by either principals, third parties, or their tools. The contractual phases of search, negotiation, commitment, performance, and adjudication constitute the realm of smart contracts. This article covers all phases, with an emphasis on performance. Smart contracts utilize protocols and user interfaces to facilitate all steps of the contracting process. This gives us new ways to formalize and secure digital relationships which are far more functional than their inanimate paper-based ancestors.

Contracts Embedded in the World

The basic idea behind smart contracts is that many kinds of contractual clauses (such as collateral, bonding, delineation of property rights, etc.) can be embedded in the hardware and software we deal with, in such a way as to make breach of contract expensive (if desired, sometimes prohibitively so) for the breacher. A canonical real-life example, which we might consider to be the primitive ancestor of smart contracts, is the humble vending machine. Within a limited amount of potential loss (the amount in the till should be less than the cost of breaching the mechanism), the machine takes in coins, and via a simple mechanism, which makes a freshman computer science problem in design with finite automata, dispense change and product according to the displayed price. The vending machine is a contract with bearer: anybody with coins can participate in an exchange with the vendor. The lockbox and other security mechanisms protect the stored coins and contents from attackers, sufficiently to allow profitable deployment of vending machines in a wide variety of areas.

Smart contracts go beyond the vending machine in proposing to embed contracts in all sorts of property that is valuable and controlled by digital means. Smart contracts reference that property in a dynamic, often proactively enforced form, and provide much better observation and verification where proactive measures must fall short.

As another example, consider a hypothetical digital security system for automobiles. The smart contract design strategy suggests that we successively refine security protocols to more fully embed in a property the contractual terms which deal with it. These protocols would give control of the cryptographic keys for operating the property to the person who rightfully owns that property, based on the terms of the contract. In the most straightforward implementation, the car can be rendered inoperable unless the proper challenge-response protocol is completed with its rightful owner, preventing theft. But if the car is being used to secure credit, strong security implemented in this traditional way would create a headache for the creditor – the repo man would no longer be able to confiscate a deadbeat's car. To redress this problem, we can create a smart lien protocol: if the owner fails to make payments, the smart contract invokes the lien protocol, which returns control of the car keys to the bank. This protocol might be much cheaper and more effective than a repo man. A further reification would provably remove the lien when the loan has been paid off, as well as account for hardship and operational exceptions. For example, it would be rude to revoke operation of the car while it's doing 75 down the freeway.

In this process of successive refinement we've gone from a crude security system to a reified contract:

```
(1) A lock to selectively let in the owner and
    exlude third parties;
(2) A back door to let in the creditor;
(3a) Creditor back door switched on only upon nonpayment
     for a certain period of time; and
(3b) The final electronic payment permanently switches
     off the back door.
```

Mature security systems will be undertaking different behavior for different contracts. To continue with our example, if the automobile contract were a lease, the final payment would switch off leasee access; for purchase on credit, it would switch off creditor access. A security system, by successive redesign, increasingly approaches the logic of the contract which governs the rights and obligations covering the object, information, or computation being secured. Qualitatively different contractual terms, as well as technological differences in the property, give rise to the need for different protocols.

Contemporary Practice

Accounting Controls

Outside of the financial cryptography community, and long predating it, there is a deep tradition of protocols used in the course of performing contracts. These protocols consist of a flow of forms ("data flow", canonically displayed in data flow diagrams), along with checks and procedures called "controls". Controls serve many of the same functions as cryptographic protocols: integrity, authorization, and so on. This article uses "control protocols" or simply "controls" to refer to this combination of data flow and controls.

Control protocols, and the professions of auditing and accounting[2] based on them, play a critical but ill-analyzed role in our economy. Economists lump them, along with other costs of negotiating and ensuring the performance of contracts, under their catch-all rubric of "transaction costs". But without controls, large corporations and the economies of scale they create would not be possible. Controls allow a quarrelsome species ill-suited to organizations larger than small tribes to work together on vast projects like manufacturing jumbo jets and running hospitals. These control protocols are the result of many centuries of business experience and have a long future ahead of them, but the digital revolution will soon cause these paper-era techniques to be dramatically augmented by, and eventually integrate into, smart contracts.

Controls enable auditing of contract performances, allowing more precise inference of the behavior of an agent. Auditing is costly, so it is undertaken by random sampling. Economists study the substitutability between the probability of verifying a breach and the magnitude of legal fines, where physical enforcement is used. Conceivably, one could substitute increasingly high penalties for increasingly rarer and less expensive auditing. However, this is not robust to real-world conditions of imperfect information.

Since controls primarily address the implicit contracts between employees and employer, there is little mapping from contract to control. A secondary function of controls to to monitor contracts with other organizations. Here there is some mapping, but it is confounded by the integration of the two functions in most controls. Rather than based on contractual terms, controls are typically based on managerial authorization.

Controls are typically based around amounts of money and quantities of goods. A canonical control is double entry bookkeeping, where two books are kept, and there must be arithmetic reconciliation between the books. To conceal an irregularity, necessary to omit from both sides, or to record entries offsetting the irregularity. Notice that there is a problem distinguishing error from fraud. This problem crops up in many areas in both auditing and smart contracts. To illustrate, here are two common control techniques:

Imprest: this is a family of controls involving the receipt or disbursement of bearer certificates (usually notes and coins). One example is the protocol used at most movie theaters. Entry is segregated from payment by introducing tickets and establishing two employee roles, the ticket seller in a booth, and the ticket stub

salesman at the entrance. Periodically, a bookkeeper reconciles the number of tickets with the total paid. Discrepancy again indicates fraud or error.

Customer audit: Techniques to get the customer to generate initial documentation of a transaction. For example, pricing goods at $.99 forces the employee to open the cash register to make change, generating a receipt.

A complete control protocol typically features the generation of initial documentation, segregation of duties, and arithmetic reconciliation of quantities of goods, standard service events, and money.

Of these, the segregation of duties deserves special comment.

In a large business, transactions are divided up so that no single person can commit fraud. Segregation of duties is an instance of the principle of required conspiracy. For example, the functions of warehouse/delivery, sales, and receipt of payments are each performed by different parties, with a policy that each party reports every transaction to a fourth function, accounting. Any singular reported activity (e.g., delivery without receipt of payment) indicates potential fraud (e.g., a delivery was made to a customer and the payment pocketed instead of being put into the corporate treasury). Segregation of duties is the auditor's favorite tool. Where it is absent the auditor cries "foul", just as a good engineer would react to a single point of failure. Many cryptographic systems have rightfully gone down to commercial failure because they ground down to trust in a single entity rather than segregating functions so as to require conspiracy.

There are least three significant differences between the scope and emphasis of smart contracts and controls. Controls are paper-era protocols designed around static forms, place little emphasis on confidentiality, and are based on management authorizations rather than one-to-one relationships.

Smart contracts can be based on a wide variety of interactive protocols and user interfaces, and can be involved in a wide variety of kinds of contractual performance. Control protocols, developed in the era of paper, are based on static forms passed as messages and processed in tables and spreadsheets. Controls focus on money and counts of standardized goods and service events, easily recorded by numbers and manipulated by arithmetic, while mostly ignoring other kinds or aspects of contractual performance. Checksums on numbers, the basis of reconciliation, are crude and forgeable compared to cryptographic hashes. Electronic Data Interchange (EDI) keeps these static forms and maintains reliance on controls. It uses cryptographic hashes for nothing more sophisticated than integrity checks on individual messages.

Controls place little emphasis on confidentiality, at least in the modern accounting literature. The emphasis on confidentiality in paper-era protocols is lacking because violation of often implicit confidences, via replication of data, was much more difficult with paper. Furthermore, technologies for protecting confidentiality while auditing were not feasible. Businesses traditionally trusted accounting firms with confidences, a trust that has eroded over the last century, and will erode still further as accounting firms start taking advantage of the vast amounts of inside and marketing information they are collecting from their customers' databases during audits. Using paper-based protocols in a digital world, there are few effective controls against the auditors themselves. Post-unforgeable transaction logs and multiparty secure computation, discussed below, indicate the possibility of cryptographic protocols to implement less relavatory but more effective auditing trails and controls; their use may be able to ameliorate the growing problems with data mining and breach of confidentiality.

Auditors place quite a bit of trust in management to authorize transactions in a secure and productive manner. Objecting to this dual trust in management and distrust of employees inherent in the accounting tradition, there has been a trend in the last two decades towards a loosening of controls as a part of hierarchy flattening and empowerment of professional employees. Unfortunately, loose controls have led to several recent scandals in the banking and investment trade. The most recent view is that there must be a learned tradeoff between controls and empowerment. The smart contract view is that we need smarter controls, originating at the ownership of the company, and entailing less asymmetry between management and other professional employees. This means converting many implicit employee contracts to more explicit smart contracts based on more direct relationships between owners (or at least their directors) and employees, and symmetric formalizations between employees.

Although most of these differences are biased against controls, these traditional protocols have a long future ahead of them, simply because they have a long past. They are highly evolved, hundreds of years old (double-entry bookkeeping, for example, predates the Renaissance). Smart contracts will incorporate many techniques and strategies from control protocols, such as generation of an initial record, segregation of duties, and reconciliation. It will not be long, however, before smart contracts start augmenting and transforming traditional business procedures, making a wide variety of new business structures possible and in the long run replacing traditional controls.

Electronic Data Interchange

Electronic Data Interchange (EDI) is the computer-to-computer communication of standardized business transactions between organizations, in a standard format that permits the receiver to perform the intended transaction. It renders traditional static business forms in cyberspace, and maintains the dependence on traditional controls. Beyond simple encryption and integrity checks, EDI does not take advantage of algorithms and protocols to add security and "smarts" to business relationships. It enables more rapid execution of traditional negotiation and performance monitoring procedures.

EDI loses some security features provided by physical paper (such as difficulty of copying) while not gaining advantages from the wide variety of protocols possible beyond simple message-passing of static forms. This article examples a much richer set of protocols.

EDI contracts tend to be merely reiterations of existing terms and conditions, with only some timing expectations changed for the electronic environment. By redesigning our business relationships to take advantage of a richer set of protocols, smart contracts can take us far beyond the paper-based paradigm of shipping around forms in a secure manner.

The following classification, derived from Sokol[3], illustrates the variety of business forms that have been rendered in electronic form:

```
Administrative
  Product code and price catalogs
  Catalog updates
  Forecasts and plans
  Deals and promotions
  Statements
Prepurchasing
  Requests for quote (&response)
  Inventory inquiry/advice
Purchasing
  Purchase order & acknowledgment
  Purchase order change & acknowledgment of change
  Material release
  Point of sale/inventory on hand
Shipping and Receiving
  Shipment status inquiry & response
  Advance shipment notification
  Bill of Lading
  Freight bill
Warehouse
  Inventory inquiry & status
  Shipping notice
  Receipt confirmation
  Shipment order
  Shipment confirmation
Customs
  Declaration
  Release
Billing and Paying
  Invoice
  Payment remittance
  Credit and debit memos
  Receipts
```

Automata as Authority

Focal (or Schelling) points are often designed and submitted into negotiations by one side or another, both to bias the negotiations and to reduce their cost. The fixed price at the supermarket (instead of haggling), the prewritten contract the appliance salesman presents you, etc. are examples of hard focal points. They are simply agreed to right away; they serve as the end as well as the beginning of negotiations, because haggling over whether the nearest neighbor focal point is better is too expensive for both parties.

There are many weak enforcement mechanisms which also serve a similar purpose, like the little arms in parking garages that prevent you from leaving without paying, the sawhorses and tape around construction sites, most fences, etc. Civilization is filled with contracts embedded in the world.

More subtle examples include taxi meters, cash register readouts, computer displays, and so on. As with hard focal points, the cost of haggling can often be reduced by invoking technology as authority. "I'm sorry, but that's what the computer says", argue clerks around the world. "I know I estimated $50 to get to Manhattan, but the meter reads $75", says the taxi driver.

Dimensions of Contract Design

Economists stress two properties important to good contract design: *observability* by principals and *verifiability* by third parties such as auditors and adjudicators. From the traditions behind contract law and the objectives of data security, we derive a third objective, *privity*. We flesh out the dimensions of contract design by disentangling mental from computational transaction costs, classifying the kinds of enforceability, characterizing the temporal phases of contracting, and discussing the nature of tradeoffs between the three design objectives.

Mental and Computational Transaction Costs

The costs that smart contracts address are lumped by economists under the catch-all rubric of "transaction costs". We can divide these into mental and computational transaction costs.

One major category of costs include the cost of anticipating, agreeing to, and clearly writing down the various eventualities. These are largely mental transaction costs, although online research tools, for example, may bring more information about eventualities.

Most contractual dispute involves an unforeseen or unspecified eventuality. We lack a good model for this. Such a model would account for the computational costs of foreseeing these eventualities, some of which may be uncomputable (and therefore of infinite cost). Where eventualities remain unspecified, contracts remain incomplete.

Where counterparties lack focal points, they lack a meeting of minds. Negotiation addresses this gap; the farther apart the focal points (in terms of value), the more expensive the negotiations. There are a variety of institutions of negotiation, which economists study under the rubric of "mechanisms". These range from simple haggling to sophisticated auctions and exchanges.

Contracting Phases

For the temporal phases of contracting we use the following schema, classified according to the two-phase model used in economics:

```
Ex-Ante
    Search
    Negotiation
    Commitment
Ex-Post
    Performance
    Adjudication
```

Smart contracts often involve trusted third parties, exemplified by an intermediary, who is involved in the performance, and an adjudicator, who is invoked to resolve disputes arising out of performance (or lack thereof). Intermediaries can operate during search, negotiation, commitment, and/or performance. Hidden knowledge, or adverse selection, occurs ex-ante; hidden actions (moral hazards) occur ex-post.

Here are some examples of contemporary electronic commerce activities and the phases of contracting they deal with:

```
EDI                   commitment, performance
Contract drafting     negotiations
Web surfing           search
Payment                       performance
Online exchange       search, negotiation, commitment
"I agree" button              commitment
```

This article covers all phases, with a particular emphasis on performance.

Observability, Hidden Knowledge, and Hidden Actions

The first objective of smart contract design is observability, the ability of the principals to observe each others' performance of the contract, or to prove their performance to other principals. The field of accounting is, roughly speaking, primarily concerned with making contracts an organization is involved in more observable.

Economists discuss "hidden knowledge", also known as "adverse selection", which can occur due to lack of ability to observe potential counterparties during the search and negotiation phases. Another major problem is "hidden actions", also known as "moral hazard", which can occur due to the lack of observability and ability to drop out of contract during the performance phase of a contract.

One important task of smart contracts, that has been largely overlooked by traditional EDI, is critical to "the meeting of the minds" that is at the heart of a contract: communicating the semantics of the protocols 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.

Here's a small example of smart fine print:

```
if (x == true) {
    printf("x is false");
}
```

Without user interfaces smart contracts are largely invisible, like the electronics in newer car engines. This is both a blessing – counterparties don't have to feel like they're dealing with user-hostile computers – and a curse – the "smart fine print" problem of hidden actions.

To properly communicate transaction semantics, 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. For example, encryption can be shown by putting the document in an envelope, and a digital signature by affixing a seal onto the document or envelope.

Online Enforceability

Amid all the hype about "information warfare", lost in the noise is the fact that it is impossible to commit an act of physical violence over the Net. That includes not only all physical crimes of coercion, but also arrest, incarceration, and other traditional methods of law enforcement. Because of this fact, and the jurisdictional swamp that is the multinational Internet, this article concentrates on means of protecting against breach and third parties that do not rely on law enforcement.

We can categorize the security measures against breach, eavesdropping, and interference in the following manner:

```
Proactive
    - breaching actions rendered impossible
    - either side can drop out with minimal loss upon counterparty
       breach
Reactive
    Deterrence
    - reputation
    - physical enforcement
          third parties: tort law
    Damage Recovery
    - secured transaction
    - reputation
    - physical enforcement
          principals: contract law
          third parties: tort law
```

Currently, the most prevalent forms of security software are not proactive cryptography, but reactive and panoptic methods like virus scanning software, filtering firewalls, traceroutes of attackers, etc. Once modern cryptographic protocols are more widely deployed, the balance will likely shift towards preventative security.

Verifiability by Adjudicators

Reactive measures rely upon two areas: verifiability and penalties. As discussed in the section on accounting controls, under ideal economic conditions, the statistical distribution of verification failures is known, so that verification costs and penalties are can be traded off neatly. But with imperfect information, the jurisdictional swamp, and lack of collateral or other security, collection of damage awards is even more severely limited than in contracts confined to traditional geographic jurisdictions. Reputation costs may be the only practical source of penalties in many cases. For reactive measures to work, high verifiability is critical.

So our second objective is *verifiability*, the ability of a principal to prove to an adjudicator that a contract has been performed or breached, or the ability of the adjudicator to find this out by other means. The disciplines of auditing and investigation roughly correspond with verification of contract performance.

Privity: Protection from Third Parties

Our third objective of smart contract design is privity, the principle that knowledge and control over the contents and performance of a contract should be distributed among parties only as much as is necessary for the performance of that contract. This is a generalization of the common law principle of contract privity, which states that third parties, other than the designated adjudicators and intermediaries, should have no say in the enforcement of a contract. To maintain knowledge and control, performance must be encapsulated: protected from outside influences, especially sophisticated attacks. This is the idea behind both the legal doctrine of privity, which restricts redress to the parties to a contract, and the idea of property rights.

Attacks against privity are epitomized by third parties Eve the eavesdropper, a passive observer of contents or performance, and malicious Mallet, who actively interferes with performance or steals service. Under this model privacy and confidentiality, or protecting the value of information about a contract, its parties, and its performance from Eve, is subsumed under privity, as are property rights. The most common definitions of "security" in the online world roughly correspond to the goal of privity.

Our generalized privity thus encompasses property rights as stable objects linked to particular contracts (and thereby the parties in privity to such contracts, the "owners"). Privity creates a clear boundary within which operate a coherent set of rights, responsibilities, and the knowledge with which to carry out those responsibilities and protect those rights. Clarified boundaries also allow accountability. Protection from extraneous interference allows us to focus responsibility for the consequences of contract-related activity onto the parties to the contract.

Trading Off Contract Design Objectives

Privity says that we want to minimize vulnerability to third parties. Verifiability and observability often require that we invoke them. An intermediary must be trusted with some of the contents and/or performance

of the contract. An adjudicator must be trusted with some of the contents, and some of the history of performance, and to resolve disputes and invoke penalties fairly. In smart contract design we want to get the most out of intermediaries and adjudicator, while minimizing exposure to them. One common outcome is that confidentiality is violated only in case of dispute.

Many kinds of specific performance are often entrusted to intermediaries. We must be able to trust the intermediary (credit agency, anti-virus software vendor, certificate intermediary, digital cash mint, etc.) with their particular claims (about creditworthiness, dangerous byte patterns, identity, conservation of the money supply, etc.) As Ronald Reagan remarked in a slightly different context, "trust but verify". To deserve our trust, intermediaries must convince us that their claims are true. We need to be able to "ping" their veracity, verifying that certain claimed transactions in fact occurred. An entire profession exists in market economies to perform this function: auditing.

Ideally, observability and verifiability can also include the ability to differentiate between intentional violations of the contract and good faith errors, but this is difficult in practice, since the difference is often largely one of subjective, unrevealed intent.

Building Blocks of Smart Contract Protocols

Protocols

A *protocol*[5] in computer science is a sequence of messages between at least two computers. At a higher level of abstraction, a protocol consists of algorithms communicating via messages. These programs act as proxies, or agents, for human users, who communicate their preferences via users interfaces. We distinguish protocol endpoints by names such as "Alice" and "Bob", but it should be kept in mind that the end points are really computer processing units, which may or may not be under the control of, or taking actions contrary to the intent of, the human user. Human users typically do not have full knowledge of the protocol in question, but rather a metaphorical understanding obtained via user interface, manuals, and so on. Unlike most real-world contracts, protocols must be unambiguous and complete.

Protocols come in three basic types. I have modified the terminology of Schneier[6] to match more closely to the corresponding business terminology:

```
self-enforcing:  Alice <--> Bob,
mediated:   Alice <--> intermediary <--> Bob
adjudicated:  (Alice <--> Bob) --> [evidence] --> adjudicator
```

The corresponding smart contracts elaborate on "Alice" to distinguish between the software (in two components, the endpoint of protocol and the user interface), and Alice herself. Cryptographic and other computer security mechanisms give us a kit of tools and parts from which we can build protocols, which form the basis of smart contracts.

The "Physics of Cyberspace"

The security properties of physical media are based on physical properties we often take for granted, for example the unforgeability of an atom of gold. The structural constraints ("physics") of cyberspace relevant to security are described by the mathematical theories studied by computer scientists, especially in the specialty called cryptography. Here are the important "fundamental particles" of the cryptographic universe:

```
-- pseudorandom function families -->
        secret key encryption, hash, MAC, ...

-- trapdoor one-way functions -->
        public key encryption

-- pseudorandom bit generators -->
        generate keys, padding, cookies
```

```
-- information-theoretic/unconditional -->
        one-time pads
```

These "particles" are potent building blocks for engineering secure protocols. Imagine a material so tough, it is completely impervious to a supernova, and so cheap you could use it to make walls, locks, safes, and envelopes to protect everyday items. This is not just a metaphor: cracking a 4,096 bit RSA key with best known algorithm really would require more electrical power for the computers than the power produced by a supernova. These cryptographic primitives promise to be a main driving force of the fourth cost revolution for global business.

Cryptographic Protocols

A family of protocols, called cryptographic protocols because their first application was computerized "secret writing", provide many of the basic building blocks that implement the improved tradeoffs between observability, verifiability, privity, and enforceability in smart contracts. Contrary to the common wisdom, obscurity is often critical to security. Cryptographic protocols are built around foci of obscurity called keys. A key's immense unknown randomness allows the rest of the system to be simple and public. The obscurity of a large random number, so vast that a lucky guess is astronomically unlikely, is the foundation upon which cryptographic protocols, and smart contracts based on them, are built.

Two significant cautions are in order when thinking about how cryptographic protocols can be used in online relationships. The first is that protocols usually provide security "up to" some assumption. This assumption is a remaining weak point which a complete working system must address in some reasonable manner. One common endpoint is is assumptions about trusted third parties. Often the degree or function of the trust is not well specified, and it is up to the real-world systems analyst to characterize and ameliorate these exposures. The best mediated protocols only trust the intermediary or counterparty with a well limited function.

Even without trusted third parties, cryptographic protocols often ground out in trust of the counterparty. For example, encryption of a message provides confidentiality up to the actions of parties with decrypting keys. Encryption does not stop key holders from posting plain text to Usenet. We cannot just say that encryption provides "confidentiality" and leave our concern for confidentiality at that.

The second caution is that much of the terminology used in the cryptographic literature to name protocols ("signatures", "cash", etc.) is misleading. Sometimes the terminology falls short on substantial matters: a "digital signature", for example, is not biometric and is based on a key that can easily be copied if not protected by another mechanism. Often cryptographic protocols can be generalized to much wider purposes than implied by the label. For example, "digital cash" is a very general protocol which can implement a wide variety of bearer certificates and conservation wrappers for distributed objects.

Attacks against Smart Contracts

Protocols for smart contracts should be structured in such a way as to make their contracts

a. robust against naive vandalism, and
b. robust against sophisticated, incentive compatible (rational) attack

A vandal can be a strategy or sub-strategy of a game whose utility is at least partially a function of one's own negative utility; or it can be a mistake by a contracting party to the same effect. "Naive" simply refers to both lack of forethought as to the consequences of an attack , as well as the relatively low amount of resources expended to enable that attack. Naive vandalism is common enough that it must be taken into consideration. A third category, (c) sophisticated vandalism (where the vandals can and are willing to sacrifice substantial resources), for example a military attack by third parties, is of a special and difficult kind that doesn't often arise in typical contracting, so that we can place it in a separate category and ignore it here. The distinction between naive and sophisticated strategies has been computationally formalized in algorithmic information theory.

The expected loss due to third party attack is called the exposure. The cost of third parties to defeat the security mechanism is the disruption cost. If the disruption cost is greater than the expected benefit, we can

expect an incentive compatible attacker to disrupt the security.

Public and Secret Key Cryptography

One of the drivers of the trust cost revolution will likely be the wide variety of new cryptographic protocols that have emerged in recent years. The most traditional kind of cryptography is *secret key* cryptography, in which Alice and Bob (our exemplar parties to a smart contract) use a single shared, prearranged key to encrypt messages between them. A fundamental problem we will see throughout these protocols is the need to keep keys secret, and *public key* cryptography helps solve this. In this technique, Alice generates two keys, called the private and public keys. She keeps the private key secret and well protected, and publishes the public key. When Bob wishes to send a message to Alice, he encrypts a message with her public key, sends the encrypted message, and she decrypts the message with her private key. The private key provides a "trapdoor" that allows Alice to compute an easy inverse of the encryption function that used the public key. The public key provides no clue as to what the private key is, even though they are mathematically related. The RSA algorithm is the most widely used method of public key cryptography.

Public Authentication

Public key cryptography also makes possible a wide variety of *digital signatures*. These proves that a piece of data (hereafter referred to as just an "object") was in active contact with the private key corresponding to the signature: the object was actively "signed" with that key. There are two steps to an authentication protocol: signing and verification. These may occur synchronously, or, in many public protocols, a signature may be verified at some distant time in the future.

The digital signature probably should have been called a "digital stamp" or "digital seal" since its function resembles more those methods than an autograph. In particular, it is not biometric like an autograph, although incorporation of a typed-in password as part of the private key used to sign can sometimes substitute for an autograph. In many Asian countries, a hand-carved wooden block, called a "chop", is often used instead of autographs. Every chop is unique, and because of the unique carving and wood grain cannot be copied. A digital signature is similar to the chop, since every newly generated key is unique, but it is trivial to copy the key if obtained from the holder. A digital signature relies on the assumption that the holder will keep the private key secret.

A *blind signature* publically authenticates privy information (but can we use non-privy signatures blindly as well?). This is a digital signature and secret-key encryption protocol that together have the mathematical property of commutativity, so that they can be stripped in reverse of the order they were applied. It's like stamping an unknown document through carbon paper (without having to worry about smudging). The effect is that Bob "signs" an object, for which he can verify its general form, but cannot see its specific content. Typically the key of the signature defines the meaning of the signed object, rather than the contents of the object signed, so that Bob doesn't sign a blank check. Blind signatures used in digital bearer certificates, where Bob is the clearing agent, and in Chaumian credentials, where Bob is the credential issuer.

Privy Authentication

The blind signature is one example of the many "magic ink signatures" cryptographers have invented. Another class of these protocols are used to limit the parties allowed to either verify the signature or to learn the identity of the signer. The most privy are the zero-knowledge proofs, where only the counterparty can authenticate the prover. Designated confirmer signatures allow the signer to designate particular counterparties as verifiers. For example, a business could give particular auditors, investigators, or adjudicators the authority to verify signed objects, while other third parties, such as competitors, can learn nothing from the signature. Group signatures allow members to sign as an authentic member of a group, without revealing which member made the signature.

Protection of Keys

So far, we've assumed parties like Alice and Bob are monolithic. But in the world of smart contracts, they will use computer-based software agents and smart cards to do their electronic bidding. Keys are not

necessarily tied to identities, and the task of doing such binding turns out to be more difficult than at first glance. Once keys are bound, they need to be well protected, but wide area network connections are notoriously vulnerable to hacking.

If we assume that the attacker has the ability to intercept and redirect any messages in the network protocol, as is the case on wide area networks such as the Internet, then we must also assume, for practical all commercial operating systems, that they would also be able to invade client if not merchant computers and find any keys lying on the disk.

There's no completely satisfactory solution to end point operations security from network-based attacks, but here's a strategy for practically defanging this problem for public-key based systems:

All public key operation can be performed inside an unreadable hardware board or smart card on a machine with a very narrow serial-line connection (ie, it carries only a simple single-use protocol with well-verified security) to a dedicated firewall. This is economical for high traffic servers, but may be less practical for individual users. Besides better security, it has the added advantage that hardware speeds up the public key computations.

If Mallet's capability is to physically seize the machine, a weaker form of key protection will suffice. The trick is to hold the keys in volatile memory. This makes the PC proof from physical attacks – all that needed to destroy the keys is to turn off the PC. If the key backups can be hidden in a different, secure physical location, this allows the user of this PC to encrypt large amounts of data both on the PC itself and on public computer networks, without fear that physical attack against the PC will compromise that data. The data is still vulnerable to a "rubber hose attack" where the owner is coerced into revealing the hidden keys.

Capabilities

Object-oriented, or capability, security is a deep and promising area, but beyond the scope of this article. Capabilities can potentially simplify the design of many distributed security protocols. Instead of developing a new or modified cryptographic protocol for each contracting problem, capabilities may allow us to design a rich variety of distributed security protocols over a common cryptographic framework.

For more information see Introduction to Capability Based Security.

Quora

Quorum distribution of performance or control over resources can be based on the secret sharing of keys needed to perform or control a resource. These are also known as threshold techniques. These are methods of splitting a key (and thus control over any object encrypted with that key) into N parts, of which only M are needed to recreate the key, but less than M of the parts provide no information about the key. Secret sharing is a potent tool for distributing control over objects between principals.

Markus Jacobsson has designed a quorum of mints for signing digital coins, for example. Quorum establishes a "required conspiracy" of M out of N to perform a function, providing an option for stronger protection than the typical 2 out of N used in segregation of duties, and greater confidence in the security underlying the segregation.

Post-Unforgeable Transaction Logs

Traditionally, auditors have contacted counterparties in order to verify that a transaction actually took place (The "principle of required conspiracy" at work again). With post-unforgeable logs, via a hierarchical system of one-way hash functions, a party can publically commit to transactions as they are completed by publishing signed cumulative hashes of the transaction stream. The confidentiality of the transaction is fully maintained until an auditor "pings" the transaction to determine its actual nature. The counterparty identity can remain confidential, because it is not required to establish the other facts of the transaction. The only attack is to forge transactions in real time, as the transaction itself takes place, which in most practical cases will be

unfeasible. Most accounting fraud involves analyzing sets of completed transactions and then forging them to make them compute to a desired counterfactual result.

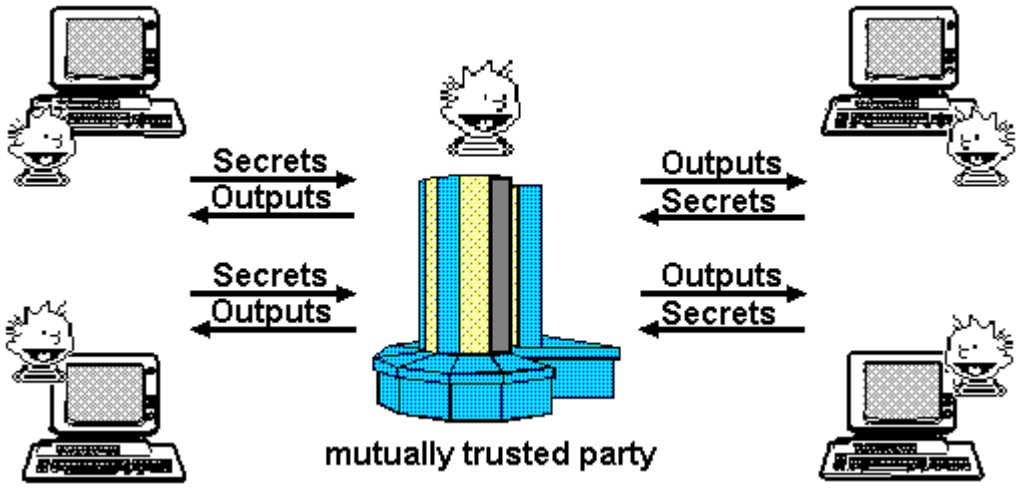
Mutually Confidential Computation

Cryptographers have developed protocols which create virtual machines between two or more parties. Multiparty secure computation allows any number of parties to share a computation, each learning only what can be inferred from their own inputs and the output of the computation. These virtual machines have the exciting property that each party's input is strongly confidential from the other parties. The program and the output are shared by the parties. So, for example, we could run a spreadsheet across the Internet on this virtual computer. We would agree on a set of formulas, set up the virtual computer with these formulas, and each input our own private data. We could only learn only as much about the other participants' inputs as we could infer from our own inputs and the output.

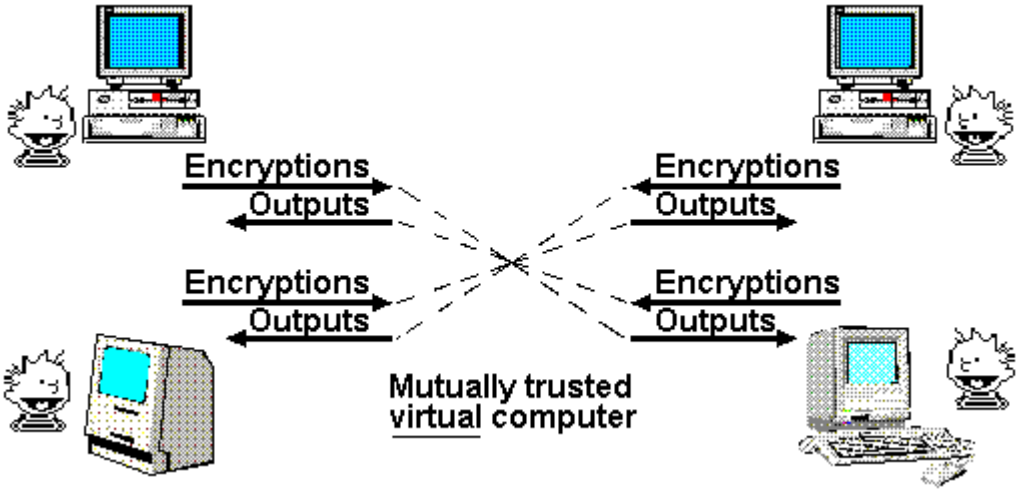
There are two major complications. The first is that this virtual computer is very slow: one machine instruction per network message. The second is that some parties learn the results before others. Several papers have disussed the fraction of parties one must trust in order to be assured of learning the correct output. The mechanism must be constructed so that a sufficient number of parties have an incentive to pass on the correct result, or reputation, side contracts, etc. used to the same effect.

With these caveats, any algorithmic intermediary can, in principle, be replaced by a trustworthy virtual computer. In practice, because of the two complications, we usually construct more limited protocols out of more efficient elements.

Trusted Third Party:



Mathematically Trustworthy Protocol:



Multiparty secure computer theory, by making possible privy virtual intermediation, has major implications for all phases of contracting. This can be seen most clearly in the area of negotiations. A "mechanism" in economics is an abstract model of an institution which communicates with its participants via messages, and whose rules can be specified algorithmically. These institutions can be auctions, exchanges, voting, and so on. They typically implement some kind of negotiation or decision making process.

Economists assume a trusted intermediary operates the mechanism. Here's a simple example of using this virtual computer for a mechanism. Alice can submit a bid price, and Bob an ask price, to their shared virtual computer which has one instruction, "A greater than B?". The computer then returns "true" if Alice's bid is greater than Bob's offer. A slightly more sophisticated computer may then decide the settlement price according to a number of different algorithms (Alice's bid, Bob's ask, split the difference, etc.) This implements the mechanism "blind bargaining" with no trusted intermediary.

In principle, since any computable problem can be solved on this virtual computer (they are "Turing complete"), any computable economic mechanism can be implemented without a trusted intermediary. In practice, these secure virtual computers run very slowly (one virtual machine instruction per network message), and the order in which participants learn results often matters. But the existence proof, that any economic mechanism can be run without a trusted intermediary, up to temporal issues, is very exciting. This means that, in principle, any contract which can be negotiated through a trusted third party (such as an auction or exchange) can be negotiated directly. So, in some abstract sense, the only remaining "hard" problems in smart contract negotiations are (a) problems considered hard even with a trusted intermediary (for the standard economic reasons), (b) nonsimultaneity problems in learning the decision, and (c) the task of algorithmically specifying the negotiating rules and output contract terms (This includes cases where an intermediary adds knowledge unavailable to the participants, such as a lawyer giving advice on how to draft a contract).

Applying this kind of analysis to the performance phase of contracts is less straightforward. For starters, economic theories of the performance phase are not as well developed or simple as the mechanism theory of negotiations. Indeed, most economic theory simply assumes that all contracts can be perfectly and costlessly enforced. Some of the "transaction cost" literature has started to move beyond this assumption, but there are few compelling results or consensus theories in the area of techniques and costs of contract enforcement.

Performance phase analysis with multiparty secure computer theory would seem to apply only to those contracts which can be performed inside the virtual computer. But the use of post-unforgeable auditing logs, combined with running auditing protocols inside the shared virtual computer, allows a wide variety of performances outside the virtual computer to at least be observed and verified by selected arbitrators, albeit not proactively self-enforced.

The participants in this mutually confidential auditing protocol can verify that the books match the details of transactions stored in a previously committed transaction log, and that the numbers add up correctly. The participants can compute summary statistics on their confidentially shared transaction logs, including cross-checking of the logs against counterparties to a transaction, without revealing those logs. They only learn what can be inferred from the statistics, can't see the details of the transactions. Another intriguing possibility is that the virtual computer can keep state over long periods of time, allowing sophisticated forms of privy and self-enforcing secured credit.

With mutually confidential auditing we will be able to gain high confidence in the factuality of counterparties' claims and reports without revealing identifying and other detailed information from the transactions underlying those reports. These provide the basis for solid reputation systems, and other trusted third party systems, that maintain integrity across time, communications, and summarization, and preserve confidentiality for transaction participants. Knowing that mutually confidential auditing can be accomplished in principle may lead us to practical solutions.

Contracts with Bearer

Bearer Certificates

Bearer certificates implement transferable rights on standardized contracts. Each kind of contract (for example, each denomination of "coin" in digital cash) corresponds to a digital signature, just as each issue of Federal Reserve Notes or stock certificates corresponds to a particular plate.

In the most straightforward bearer certificate protocol, the issuer and transfer agent (the same entity, for our purposes, though they can easily be unbundled) create a serial number (really a large unguessable random number, rather than a sequence), and add it to a list of issued certificates. The transfer agent clears a transfer by checking the signature to identify and nature of the bearer contract and verify that it was made, then looking on that contract's issued list to make sure the serial number is there, then removing the serial number. Alternatively, the issuer can let the issuee make up the serial number, then, when cleared, check the signature and put the number on the list of cleared certificates. The signature provides the assurance that the certificate is indeed the the particular kind of contract with bearer, while the serial number assures that the same instance of that contract is not cleared or redeemed more than once. In these simple versions, the transfer agent can link the transferee to the transferor for all transfers. To implement the privacy characteristics of coins and physical bearer certificates, we need to add unlinkability features.

Unlinkable Transfers

Unlinkability can be provided by combining the second variation above, a list of cleared certificates, with blind signatures and a mixing effect. Enough instances of a standardized contract are issued over a period of time to create a mix. Between the issuing and clearing of a certificate, many other certificates with the same signature will be cleared, making it highly improbable that a particular clearing can be linked to a particular issue via the signature. There is a tradeoff between the mixing effect and the exposure to the theft of a "plate" for a particular issue: the smaller the issue, the smaller the exposure but the greater the linkability; a larger issue has both greater exposure and greater confidentiality.

Blind signatures can be used to make certificate transfers unlinkable via serial number. Privacy from the transfer agent can take the form of transferee-unlinkability, transferor-unlinkability, or "double blinded" where both transferor and transferee are unlinkable by the transfer agent or a collusion of a transfer agent and counterparty.

Bearer certificates come in an "online" variety, cleared during every transfer, and thus both verifiable and observable, and an "offline" variety, which can be transferred without being cleared, but is only verifiable when finally cleared, by revealing any the clearing name of any intermediate holder who transferred the object multiple times (a breach of contract).

This unlinkability is often called "anonymity", but the issue of whether accounts are issued to real names or pseudonyms, and whether transferor and transferee identify themselves to each other, is orthogonal to unlinkability by the transfer agent in the online model. In the off-line model, account identification (or at least a highly reputable and/or secured pseudonym) is required: passing an offline certificate a second time reveals this identity. Furthermore, communications channels can allow Eve to link transferor and transferee, unless they take the precaution of using an anonymous remailer. Online clearing does make lack of identification a reasonable option for many kinds of transactions, although common credit and warrantee situations often benefit from or even require identification.

When confronting an attempted clearing of a cleared serial number, we face an error-or-fraud dilemma similar to the one we encountered above in double entry bookkeeping. The ecash™ protocol from DigiCash actually takes advantage of this ambiguity, second-transferring certificates on purpose to recover from a network failure. When certificates are lost over the net it is not clear to the transferor whether they have been received and cleared by the transferee or not. Second-transferring directly with the transfer agent resolves the ambiguity. This only works with the online protocol. The issue of distinguishing error from fraud is urgent in the offline protocol, but there is as yet no highly satisfactory solution. This problem is often intractable due to the subjectivity of intent.

Conserved Objects

Issuance and cleared transfer of references to a distributed object conserves the usage of that object. This object becomes "scarce" in economic terms, just as use of physical objects is finite. Conserved objects

provide the basis for a software economics that more closely resembles economics of scarce physical objects. Conserved objects can be used to selectively exclude not only scarce physical resources (such as CPU time, network bandwidth and response time, etc.), but also fruits of intellectual labor – as long as one is willing to pay the price to interact with that information over the network rather than locally (cf. content rights management). Conservation immunizes objects and the resources they encapsulate to denial of service attacks. Bearer certificate protocols can be used to transfer references to a particular instance or set of instances of an object, just as they can be used to transfer other kinds of standardized rights.

Digital Cash

Digital cash is the premier example of a digital bearer certificate. The issue and transfer agent is called a "mint". Bearer certificate protocols enable online payment while honoring the characteristics desired of bearer notes, especially unforgeability (via the clearing mechanism) and transfer confidentiality (via mixing and blinding).

To implement a full transaction of payment for services, we often need need more than just the digital cash protocol; we need a protocol that guarantees that service will be rendered 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. Potential smart contract protocols in this area are discussed under Credit.

Content Rights Management

Content protection contracts are valuable in that they incentive publishers to allow users to view content directly, rather than indirectly and partially via queries to remote servers. Content protection of software distributed online would allow it to be run locally rather than remotely, while enforcing the contract rights and copyrights of the publisher against the user. This local usage billing of software often goes under the rubric of "superdistribution"[11].

Watermarks

Watermark schemes work by altering less significant bits of content – usually a picture; sound works less well and text is difficult. These altered bits typically contain the identities of the publisher and viewer, and perhaps other information related to the contract. The idea is that, when investigators scan released content, the watermark will finger the breacher of the contract (or violator of copyright law).

Watermark investigation can be assisted by a quite inexpensive technique, Web spiders. These spiders look for redistributed watermarked material on the Web. The customer originating the copy can then be fingered.

One attack against watermarks is to overwrite likely watermark bits with other patterns legitimate to viewing software. The entanglement of watermark bits with bits important to the picture can be made rather obscure, but not strongly so by the standards of cryptography. Another attack is to steal content from a customer and distribute it as is. The watermark will finger the victim, rather than the thief.

All watermark schemes can be defeated with sufficient effort. These schemes can then be distributed as software worldwide. Once the initial effort is put into breaking a scheme, the marginal cost of breaking it is minimal. Furthermore, once the watermark is removed, the content can be distributed and even published[12] with secure anonymity.

In sum, watermark schemes can add significant risk to the copying of of low value or ephemeral information. This will be sufficient for many kinds of content, such as news or product updates. It won't stop, for long, the redistribution of high-value content. Since watermarks require traceable identification, they reduce customer privacy and require the inconvenience of registration and authentication, adding to the transaction costs of content purchase.

Controlled CPUs

Contrary to the hype, there is no strong content protection software. Watermarks are as close as we've come, and they fall far short of the standards of computer security. Large sums have gone into attempts to develop such technology, resulting in hundreds of patents but no substantial results.

As a result, some publishers have begun putting their research dollars into a radical alternative, innocuously dubbed the "secure CPU" (SPU)[13]. This is a CPU that is "secure" against the owner of the computer! To enforce copyright or content contracts, the SPU monitors all content-related activity. Some marketing literature even lists, alongside the traditional copyright, a new "right" of publishers to monitor the usage of their content. Remarkably enough, these panoptic non-personal computers are the focus of major R&D efforts.

The radical SPU projects demonstrate both the high value of content contracts to publishers and the high price we have to pay to maintain the paper-era intellectual property model online. Strong content protection would be valuable in going beyond indirect and partial viewing of content on servers, to viewing content directly and locally. The price is the loss of control over our own computers, and loss of privacy over our activities on those computers.

The online content market is squeezed from above and below. From above, by the ease of redistributing high-value content. From below, by the mental transaction costs of charging for low value content – costs to which the requirements of registration and traceable identification add substantially. The size of the market in between is an open question. "Information wants to be free", but authors and publishers want to be paid for it. The current content market for more difficult to copy media, such as books, films, CD-ROM, and so forth is large, in the hundreds of billions of dollars per year. But on the Internet, free content dominates. Distributing ephemeral content in the form of service subscriptions is in most cases a more viable way to go. It remains to be seen how large the Internet content market will become, and to what extent customers will tolerate impositions on privacy and control of their computers in order to obtain legal content.

Reputation Systems

Reputation can be viewed as the amount of trust an agent has created for himself[14]. Reputation systems ultimately need to be based on fact rather than mere opinion or faith to be effective. For example, if we are to have a good credit rating system, we need to be confident that the credit record assembled by the agency is sufficiently accurate. Reputation information is typical gathered and distributed by intermediaries trusted to perform this task. Reputation can take the form of a public database (such as credit rating services) or credentials issued by the tracking agency and carried by the user. A bearer doesn't want to show his negative credentials, so credentials are often only positive. But we want to protect ourselves against negative behavior sources well as search out positive sources.

Tags that bundle the results of a wide variety of transactions - global names, or universal IDs, or "True Names" – may provide the most incentive for parties to carry their negative credentials. Most people have accumulated enough positive reputation is some areas that it is well-nigh impossible for them to start over their entire lives as newcomers.

Robin Hanson[15] has observed that in a world of global names, the use of a local name may signal the hiding of negative credentials, so that the use of global names is in equilibrium. A further problem with local names is that our relationships are often not neatly compartmentalizable into standard service types, and even where they are we might like to expand them into new areas. On the other hand, local names are essential for privacy. I suggest that we will want to reveal progressively more local names to our counterparties as our relationships with them become closer and more co-exposed.

While the global name equilibrium may hold for many of our relationships, there may be plenty of areas where the privity benefits of localizing names outweigh the costs of being less or unable to differentiate newcomers from hostiles. For example, the preference-tracking service at www.firefly.com increases participation via the use of pseudonyms, thereby protecting customers from exposure to strangers who might abuse that information. On the other hand, credit transactions typically demand identifying information, because the contractual exposure typically outweighs benefits of privity.

Global name public keys, which have many drawbacks in terms of privity, may be the best way to track negative reputation, but they are no panacea. There is an important conundrum in an ID-based key system: the conflict between the ability to get a new key when the old one is or could be abused by another (key revocation), and the ability of another to be sure they are dealing with the same person again. This may also provide an opportunity for parties to selectively reveal positive credentials and hide negative ones. For example, a person with a bad credit rating could revoke the key under which that rating is distributed and create a new one, while selectively updating their positive credentials to the new key (e.g., have their alma mater create a new diploma).

The current universal (non-cryptographic) key in the U.S., the social security number (SSN), is very difficult to revoke; it's much easier to change your name. This policy is probably no accident, since the biggest economic win of global name identification is the tracking of negative reputations, which revocation can defeat. As long as the SSN is a shared database key, not used for the purpose of securely identifying a faceless transaction, there is little need for revocation beyond the undesired erasure of negative history. Combining a secret authentication key, which must be revocable, with a public universal ID is quite problematic.

Credit

One of the basic outstanding problems in smart contracts is the ensurement of credit. This comes up not only in loans, but in any other contract which involves a temporal lag between performance and reciprocal performance of the contractual terms.

In current practice, there are several partially effective processes for ensuring future performance:

- Reputation (especially credit reports): often effective, but only to a point, as it is often hard for the debtor to accurately judge the future reputational effects of an action (e.g., failure to pay a bill, taking out too large a loan, etc.) that has clear, local, beneficial effects today. There is more imbalance in knowledge between current and distant consequences among individual consumers, but even among large organizations with high credit ratings it is not an irrelevant factor.
- Secured transactions: liens, escrow, etc.
- Garnishment of future income
- Law enforcement, especially to enforce transfer of control over liened assets, garnishment, etc.

These processes have a fundamental property in common – they violate the privity of credit transactions – in other words, they bring in third parties to track reputations or enforce repayment. Do credit transactions entail a fundamental imbalance in incentives that can only be redressed by bringing in third parties, or can the security protocols be discovered which allow credit with minimal or no third party involvement?

Local Name Credit Ratings

Three important variables have been proposed for reputation economics:

```
operating value: expected future profits, given the reputation
throw-away value: profit from cheating, which ruins reputation
replacement cost: cost of recreating reputation
```

In turn, Peter Swire[16] describes two problems facing inadequately secured or unsecured loans to "credit names":

Adverse selection: Prior deadbeats can start fresh by signing up for the new service. Going in, it will be biased in favor of deadbeats. This problem may be addressed by using Chaumian credentials. These allow the established positive reputations of previous names to be carried over to the credit name, without allowing anyone to link the two names. Entrants without positive reputations can be rejected.

The endgame problem: A credit name can establish a good credit rating over time. When the limit is high enough, the borrower can quickly spend it all. A malicious borrower, with a good rating established under a previous name, can systematically profit at the expense of the lender, if the throw-away value is greater than

the replacement cost. To address this problem, creditors will have to charge higher rates to new credit names and raise credit limits more slowly than for traceable names. Honest borrowers will subsidize the dishonest, to an even greater extent than they do in the current credit card system.

Secured Credit

Secured credit need not violate privity if the physical control over the securing property can be shared. So that, for example, automobile credit can be secured as long as repossession is possible, as described in the example above.

A standard mechanism of secured credit applicable online is the escrow. An escrow is an intermediary trusted to hold messages until messages from both sides are received, and, optionally, their contents verified - to extent the content is verifiable, and at the expense of some privity. The escrow then sends the messages off to their recipients, along with receipts. Messages can contain any sort of data: content, a bearer certificate, etc.

Ripped Instruments

Alice wants a New York City cab ride for which she's willing to pay $100, but she doesn't trust Bob the taxi driver to get her there on time if she pays up front. Bob in turn doesn't trust Alice to pay at the end of the trip. Commerce can be consummated by Alice tearing a $100 bill and giving half to Bob. After the trip she gives the other half to Bob, which he can then reassemble into a negotiable $100 bill. Alice loses her incentive to not pay. Bob gains incentive to get her there on time as promised. Both have made what economists call a "credible commitment" to perform their respective parts of the contract. Markus Jacobsson has digitized this idea, coming up with a protocol for ripped digital cash. As with many other aspects of digital cash, the idea can be further generalized to rip some other kinds of bearer instruments – specifically, those whose value can be divided roughly in half. If the transfer is double-blinded the transfer agent has no knowledge of the participants and therefore no bias to favor one over the other. The transfer agent must, however, be able to assess proof of performance, and the protocol is only workable where such proof (in the form of proof of receipt of a message, for example) is available.

The ripped bill is similar to using the transfer agent as an escrow agent. An advantage over using an escrow agent is that the need for extra anonymous channels between the parties and the escrow is avoided. A disadvantage is that the transfer agent now has taken on the major additional job of acting as an adjudicator, assessing proofs of performance (or at the very least, must be responsible for subcontracting out this job and implementing the adjudicator's judgement).

Credit Cards

Credit cards provide relatively little protection from third parties, especially in the area of privacy, but they do have an interesting contractual feature worth noting, the chargeback. With chargebacks customers can get refunds on allegedly unwanted merchandise. The issuer tracks the number of chargebacks both for customers and merchants; too many chargebacks can get you booted out of the system. This provides an efficient mechanism for refunds without resorting to expensive tort proceedings. Many customers who read the fine print or otherwise learn about chargeback limits often do chargebacks despite receiving and enjoying the merchandise; there is no practical way for the issuer to detect such fraud, and so it can only be pruned by limiting the number of chargebacks per customer. Some merchants complain vociferously about such customer "theft", and it seems to make possible coordinated attacks to put merchants out of business, but nevertheless merchants sign up for credit cards, because that's what their customers have signed up for. The chargeback feature makes customers more comfortable purchasing goods of unknown quality, especially mail-order and over the Internet. Chargeback provides a crude but effective partial solution to the information asymmetry problem between retailers and consumers.

Interval Instruments

"Time release" money that becomes good only after a certain date, and "interval money", that would expire after a certain date have been proposed. These can be implemented by a digital mint expiring or activating special issues of digital cash, or by a third party issuing escrowed keys at specific times. Since these keys are

encrypted against the escrow agent, and that agent doesn't know what they will be used for, the escrow agent has no incentive to cheat. A generalization of this is that transfer and redeemability are each associated with interval sets, or validity periods when each can and cannot be performed. This is analogous to clipping coupons on bonds.

Known Borrowers of Unknown Amounts

Hal Finney[17] has described a loan mix, to unlink borrowers from amount borrowed. The identity of the potential borrowers is still public, as well as the system for enforcing payment, but the actual amount loaned or borrowed remains unknown. The system starts with participants putting unknown amounts into a pot and getting receipts (bearer bonds) for these amount. All participants then borrow a standard amount. Whether a participant is a net borrower or a net creditor, and of what amount, remains private. When the loan is due all participants repay the standard amount, and the creditors reclaim the amounts on their bearer bonds. The amount actually borrowed (or, if negative, loaned) is the public amount borrowed minus the amount put into the pot. One consequence is that while negative reputations can still be accumulated when participants fail to pay back the standard amount, positive reputations are minimal, since participants who borrow and loan are indistinguishable. If future creditors put stock in positive participation, one could gain a credit rating by perpetually participating as a net borrower of zero, by loaning and borrowing the same amounts.

Conclusion

Smart contracts combine protocols, users interfaces, and promises expressed via those interfaces, to formalize and secure relationships over public networks. This gives us new ways to formalize the digital relationships which are far more functional than their inanimate paper-based ancestors. Smart contracts reduce mental and computational transaction costs, imposed by either principals, third parties, or their tools.

Mark Miller[18] foresees that the law of the Internet, and the devices attached to it, will be provided by a grand merger of law and computer security. If so, smart contracts will be a major force behind this merger.

Notes

A previous version of this paper appeared in the peer-reviewed journal First Monday, at http://www.firstmonday.dk/issues/issue2_9/szabo/index.html

Many of the links above and references here rely on URLs that can be found in the online edition of this paper, at http://szabo.best.vwh.net/caymanpaper.html.

1. The author has been refining these ideas since the early 1990's. A variety of earlier articles on this topic can be found at http://szabo.best.vwh.net ↩

2. George H. Bodnar and William S. Hopwood, 1987.*Accounting Information Systems*. 3rd ed. Boston: Allyn and Bacon. ↩

3. Phyllis K. Sokol, 1995. *From EDI to Electronic Commerce: a business initiative*. New York: McGraw-Hill. ↩

4. Oliver Hart, 1989. "Incomplete Contracts," In: John Eatwell, Murray Milgate, and Peter Newman (eds.), *The New Palgrave: Allocation, Information, and Markets*. New York: Norton.

5. Bruce Schneier,1996. *Applied Cryptography*. 2nd ed. New York: Wiley. ↩

6. John Bouvier, 1856. *A Law Dictionary: Adapted to the Constitution and Laws of the United States of American and of the Several States of the American Union*. Rev. 6th ed. ↩

7. Michael Polanyi, *Personal knowledge: Towards a post-critical philosophy*. Chicago: University of Chicago Press.

8. The economics of distributed knowledge is studied by the Austrian school; in particular see Friedrich Hayek, "On the Use of Knowledge in Society."

9. Vernon V. Palmer, 1992. *The Paths to Privity: The History of Third-Party Beneficiary Contracts at English Law*. San Francisco: Austin and Winfield.

10. Lance Cotrell, 1995. "Mixmaster & Remailer Attacks."

11. Brad Cox, 1995. *Superdistribution:: Objects as Property on the Electronic Frontier*. Reading, Mass.: Addison-Wesley. ↩

12. Ian Goldberg and David Wagner, 1997. "Enabling Anonymous Publishing on the World Wide Web." ↩

13. Olin Silbert, David Bernstein, and David Van Wie, 1996. "Securing the Content, Not the Wire for Information Commerce." ↩

14. Joseph M. Reagle Jr., 1996. "Trust in Electronic Markets," First Monday, Volume 2, number 2 (August). ↩

15. Robin Hanson, personal communication. ↩

16. Peter Swire, 1997. "The Uses and Limits of Financial Cryptography: A Law Professor's Perspective." ↩

17. Hal Finney, 1997. "Anonymous Credit" posts. ↩

18. Mark Miller, 1997. "The Future of Law," paper delivered at the *Extro 3* Conference (August 9). ↩

Please send your comments to nszabo (at) law (dot) gwu (dot) edu
Back | Index

- About
- Contact
- Donate BTC
- Atom feed
- GitHub