The following paper was originally published in the
Proceedings of the Second USENIX Workshop on Electronic Commerce
Oakland, California, November 1996

Token-Mediated Certification and Electronic Commerce

Daniel E. Geer, Donald T. Davis
Open Market, Inc.

Token-Mediated Certification and Electronic Commerce

Daniel E. Geer, Jr., Sc.D., Donald T. Davis
Open Market, Inc.
Cambridge, Mass. 02142
`{geer,ddavis}@openmarket.com`

Abstract

Public key technology presumes the availability of certificates and certifying authorities (CAs) living within a shallow hierarchy rooted at a few ($n \ll 100$) public CAs. We propose an alternative that lessens the day-to-day dependence on centralized CAs while deepening the certificate tree. We do this by suggesting that smartcards provide CA functions, thus re-framing some payment problems as simpler authorization problems.

1. Rationale by way of introduction

Smartcards will, beyond any doubt, dominate all commerce and will be how electronic commerce and conventional commerce merge into a single commerce. Irresistible global forces are behind this, and no one company has either the mass or the momentum to change it. Getting it right will have historical significance. We think we have — we encapsulate transactional commerce in public key certificates issued by token devices. These ubiquitous, personal certification authorities would not supplant centralized CAs, but would perform temporary local name-to-key bindings, primarily for access-control. We suggest that by easing some of access-control's scaling problems in this way, we gain the opportunity to re-frame some payment problems as simpler authorization problems.

The peculiar advantage of smartcard-issued certificates is that they are temporary and cheap because they are not public documents. A smartcard-issued certificate is the public key equivalent of a symmetric session key; we've brought together the benefits of asymmetric key-management and of low-cost, low-value temporary keys. Indeed, smartcard-issued certificates do not require the same kind of trust that conventional public key certificates do. Instead, a personal CA's certificates are in a sense reflexive: when Bob's smartcard issues a certificate for Alice, Alice will use the cert not to communicate with others, but only for her future communications with Bob. A personal CA could operate in this way without a smartcard, but installing this function in a smartcard protects the bearer from theft of the CA's private-key.[1]

2. One, two, many CAs

We suggest a novel idea: suppose a smartcard had the ability to issue public key certificates? How might such CA-smartcards be advantageous in electronic commerce? But before we list what smartcard-CAs can offer for merchants and consumers, we'll describe a few transaction scenarios. The reader will note that in each example the smartcard-issued cert is a single-use delegated authorization granting access to the issuer's resource.

- A consumer uses his smartcard to buy a video-download from Sony's Web page while he's at work. The smartcard issues an authorization certificate to Sony's download server. Sony's server calls the consumer's home-PC's modem, and presents the authorization certificate in a "May I download?" request. The consumer's home-PC uses the consumer's *public-key* to validate the cert, and permits the download. By the time the customer gets home, the video is ready to watch.
- A consumer uses his smartcard to subscribe to an online magazine, but the magazine doesn't appear on a Web site (because the Web is slow), and it doesn't appear in his mailbox (because the magazine is too bulky for e-mail). Rather, the

[1] For the purposes of this paper, we use the term "smartcard" to mean a handheld cryptographic token that communicates via a reader with the desktop CPU, and whose functions are not limited to memory insertion and retrieval. We include neither one-time password tokens nor memory only "chip cards" in this class.

magazine spontaneously appears on the consumer's local disk, or in his home-directory. The consumer does receive a brief e-mail notice of each magazine's arrival. The consumer's computer accepts the magazine's asynchronous download, because the publisher presents an authorization certificate that the consumer's smartcard issued when the consumer subscribed.

- A purchasing officer needs to buy something that requires countersignature. He generates a new public key pair and sends a combination of his certificate and the (sealed) new public-key to a higher-ranking officer. The higher-ranking officer mints a new certificate, one that has both the increased authority and a limitation thereon for the particular purpose at hand. Possibly, the cert is returned encrypted in the public-key of the card so that it is only usable after a session in which the card is involved.
  Alternately, the senior officer can craft the public key pair on behalf of the junior officer. The relevant private-key is returned within the cert either singly sealed in the (junior officer's) role's public-key or doubly sealed in the role's public-key *and* the public-key of the junior officer's smartcard's public-key (thereby denying the opportunity for further delegation). Such a scenario, the senior officer generates the public key pair, is inherently consistent with escrow.
  In either case, the junior officer can use that certificate as an otherwise ordinary client certificate in a transaction to which he is not otherwise entitled.
- An executive wishes to delegate to a subordinate the right to process the executive's e-mail while the executive is on a 10 day vacation. The executive issues a cert for the role "process executive's mail" with limited time-validity. Because the cert is a delegation cert, the cryptographically signed mail will be signed with the delagatee key, *i.e.*, "on behalf of."
- An investor issues a cert to a broker combining a "limit order" and a "trade at close" directive, *i.e.*, an order to trade at a range of prices if those prices obtain near a market's closing moment. Such a cert satisfies all the ordinary requirements of a secure communication, *viz.*, it is verifiably authentic, authoritative, non-repudiable, and can be issued such that it must be confirmed at the moment of use (permitting revocation thereby). Members of the exchange could issue certs that enable investors to book trade orders directly through to the floor of the exchange with such certs containing whatever volume or credit limits as represent the assumed risk of trade delegation to the investor.
- A consumer writes a cert instead of a paper check.[2] The cert authorizes the consumer's bank to debit the consumer's checking account upon presentation by a named or un-named identity. A "stop payment" is a kind of certificate revocation, and remains a single message between the consumer and his bank. It, too, has a transactional semantic.
- The consumer uses a "personal-proxy webserver" such as Open Market's OM-Express to download a catalog, does off-line shopping, and places the order whenever it is convenient to do so by uploading a collection of "purchase" certs.

In each of these scenarios, the CA-smartcard (CA-sc) performs a handshake that authenticates both parties and converts a new public-key into a new certificate. We will rely on X.509v3 extensibility by including, as an attribute-value pair, the ancestor certificate that contributes to the current one. In this way, it will not be possible to disguise (or repudiate) the delegation of authority that the cert chain represents. The flow would be:

CA-sc <--> merchant	exchange public key certs
CA-sc <--> merchant	mutual authentication, session-key
merchant ---> CA-sc	new public-key, MAC
CA-sc ---> merchant	new cert: "Merchant may download"
time may pass	
merchant ---> consumer	new cert, req-to-download
consumer ---> merchant	consumer-cert, mutual auth, "OK"
merchant ---> consumer	purchased data

As usual, the CA (smartcard CA or CA-sc here) never sees the other party's private-key. The new certificate is essentially a proxy authorization, but with a reflexive flavor; it is a grant of authority by me to myself and on my own behalf, rather than a letter of introduction to others yet un-named. This reflexivity avoids questions of trust management [BlFeLa96] because the user is simply trusting his own key to represent his past wishes to him himself. The consumer authorizes the merchant to manipulate the consumer's resources or data. The merchant exercises this delegated right, on the consumer's behalf.

[2] Or, a consumer writes a cert instead of an electronic check such as in the NetChecque system [NeMe95] which quite cleverly preserves the flow and semantics of paper checks in an asynchronous electronic medium such as e-mail.

CA-smartcards make several things easier in electronic transactions:

#1 Fulfillment can be greatly delayed after payment, without access-control administrivia on either end. In particular, no ACL is necessary (a huge scalability bonus).

#2 The merchant can deliver the service or product asynchronously. Current Web-commerce assumes that the customer initiates the fulfillment connection.

#3 The consumer can receive services and products without interactivity, without conscious effort, and without using his private-key.

#4 Delegation of roles becomes easy and assumes a transactional semantic that can include counting functions.

And why are these important?

#1 Makes electronic commerce easier to set up and manage, for everyone.

#2 Will make certain services easier to commercialize. For example, a long-haul bulk-transport service might accept a source-URL, a target URL, and an authorization cert from the customer (at payment-time). Later, when the service's proprietary lines become available for a gigabyte transfer, the service could use the cert to authenticate itself to the source and target, before starting the transfer. The customer need not be online during the transfer itself. In this area, new services are enabled rather simply automating traditional commerce.

#3 Makes electronic shopping easier and more transparent for consumers. It also becomes safer, because the home computer can receive delivery even in the consumer's absence, without exposing the smartcard or the CA's private-key to theft.

#4 In the end, we believe that delegation of roles and commercial transactions merge. (A businessman may not care about this merger. Yet.)

3. Smartcard CAs: Core of the idea

When Bob's smartcard issues a cert to Alice, no-one else has any reason to trust Bob's claims about Alice's key. Alice's cert is really only a message from Bob to himself, saying in effect, "Earlier, Alice authenticated herself once with this key-pair and once with a trusted one, so I know that this key is hers." Alice's cert validates her temporary public-key to Bob; this gives her a quick way to re-authenticate with Bob, or for him to send private messages to her, without Bob's having to remember a shared symmetric-key for long periods of time. Note that Alice and Bob could achieve the same effect with other mechanisms, but certification seems to be the simplest and most natural.

Further, with X.509 V3's extended certificate formats, Bob can include in Alice's cert her access-permissions for Bob's resources. Issuing such an authorization cert relieves Bob of the need to store or remember Alice's rights, and this approach, pursued adroitly, turns out to have scaling benefits. Finally, and crucially, Bob can receive, validate, and act upon this authorization cert, without using his smartcard or his private-key. This means that even in Bob's absence, Bob's computer can securely act on his behalf, without using his private-key and thus without exposing it to theft. This no-risk activity is unusual in public key cryptography.

4. Needs of electronic commerce

As the Internet prepares to deploy public key cryptography on a large scale for commercial applications, many in the industry eagerly welcome its unusual security features:

• Extreme cryptographic strength,
• Trustless administration,
• Anonymous cash protocols, and
• Non-repudiable signatures.

However, we argue that these features are not particularly necessary or useful for electronic commerce. We believe that the main commercial value of public key crypto lies in deeper, less-appreciated features:

• Asynchronous authentication,
• Geographic reach, and
• Access-control scaling.

Smartcard-issued certificates help public key to fulfill these deeper needs of electronic commerce. In the rest of this section, we will justify this reassessment of public key cryptography. First, we will discuss these true merits, and then we will explain our deprecation of public key's supposed merits.

4.1. Why Public Key Lends Itself to Commerce

Fast start

Much of public key's advantage over symmetric key security is transitory, because public key authentication is fairly costly, except when no security infrastructure is available. Once a security infrastructure is in place for all of the Internet, symmetric key authentication will offer lower per-transaction costs than public key. Ordinarily, public key cryptography is deemed essential for electronic commerce because (where its infrastructure is in place) public key cryptography absolutely minimizes the marginal cost of set-up that must precede a secure transaction between two commerce-ready parties. Stating this again for emphasis: Public key technology minimizes the cost of set-up that must precede a secure communication; it does not necessarily minimize overall security costs nor does its cost structure adapt to the risk involved in a particular transaction, nor do the costs and benefits distribute evenly. Nevertheless, we defer further discussion of cost minimization to other fora and merely declare that public key technology will dominate commerce.

Geographic Reach

Public key cryptography allows self-initiated secure communication between people who have no security administration in common. While this geographic reach will contribute greatly to the growth and adoption of electronic commerce, it probably won't be that important once commercial networking technology matures. Public key technology is particularly advantageous when key-bearing users are too thinly spread across the network to be united efficiently in a single security administration. That is precisely the current state of electronic commerce: relatively few consumers possess cryptographic credentials of any kind, and few of the institutions on the network are preparing to issue cryptographic credentials. Thus, if a buyer and seller on the Internet have cryptographic keys at all, their keys are unlikely to come from the same place.

Further, when on-line security servers are rare, it's unlikely that a widely-separated buyer and seller will both have access to the same security server. In this situation, compatibility of disparate credentials is easier with public keys than it would be with symmetric key cryptography, and users are better off without having to rely on an online intermediary. However, as commerce becomes pervasive, buyers and sellers will be more likely to get their credentials from the same issuing authority, wide-area symmetric key-distribution-centers (KDCs) will be more pervasive, too, and public key will lose its geographic advantage. For example, if one or a few ISP's gain national hegemony, they may prefer to offer symmetric key credentials to clients, and to use public key credentials for inter-server applications. Thus, in essence, public key cryptography will provide a long-distance scaffold, on which we will build a national infrastructure for commerce.

Asynchronous Authentication

Perhaps the most important feature of public key cryptography is that its basic mode of operation is asynchronous; it does not require two-way network connections. In fact, the sender and recipient of an authenticated message need never be on the network at the same time.[3] Asynchronous messaging is well-suited to the complex multi-party protocols that will likely be common in electronic commerce. For example, a typical electronic transaction might involve six or more communicating parties, all widely separated on the network:

- a customer,
- a merchant or content provider,
- their 2 financial institutions,
- an authentication service such as a CA,
- an access-control manager.

Asynchronous messaging can allow the transaction to conclude with a fairly simple set of protocols, without maintaining several simultaneous connections amongst the participants, and without imposing extra process-state on any of the central servers. Thus, correct use of public key cryptography can make the design of secure and failure-tolerant protocols easier than with symmetric key cryptography, albeit at some cost in speed and key-management. [Da96]

---

[3] Asynchronous messaging is possible with symmetric key systems too, but the public key mechanism is simpler. See [DaSw92] and [LaAbBuWo91].

4.2. Bad Reasons to Use Public Key for Commerce

Perfect Strength:

Though public key offers the possibility of unbreakable security, this ideal requires such long keys as to obstruct real-time applications. In practice, commerce applications often choose short key-sizes so as to keep response-times and computational burdens reasonable. For example, VISA's STT payments protocol [VI90] specified 512-bit key-pairs for client certificates even though the insecurity of such short keys had widely been accepted since Lenstra's success in factoring RSA-120, fully two years earlier. [DeDoLeMa93]

Most RSA users and application developers are more conservative and opt for 768-bit or 1024-bit keys, on the grounds that these lengths give good security for an acceptable performance cost. But, the expert opinion is that these key-lengths too will fall within 10 and 20 years, respectively. [Od96] Thus, RSA's potential for extremely high security is limited by the practical need for fast protocols.

This area is changing quickly. The rise of elliptic curve algorithms may well make the RSA patent worthless before it expires, but elliptic curve crypto does not bring perfect security within reach. Elliptic curve does run much faster than the public key algorithms it replaces (Diffie-Hellman and DSA), but EC is not so fast as to outrun this performance/security tradeoff.

Trustlessness

Public key protocols do not expose clients' keys to administrators, so even if a PK administrator is corrupt, he can't learn clients' secrets, and he can't impersonate clients. However, this property isn't important in commercial operations, which invest trust in many other people besides the security administrator, anyway. Trust relationships are a normal part of commercial and corporate existence. For example, corporations routinely entrust sensitive documents and plans to couriers, parcel-delivery services, advertising agencies, financial institutions, and corporate partners. Businesses also necessarily place a great deal of trust in individuals, including company staff, consultants, and lawyers. Similarly, citizens and consumers place comparable trust in corporations, in civil authorities, and in other people, during the course of our everyday economic activities.

Anonymity

Another form of commercial trust holds when people and corporations trust one another to keep some transactions private. Several proposed electronic cash mechanisms have been designed to allow consumers to make anonymous electronic purchases. However, in conventional commerce, explicitly anonymous transactions are rare, and guarantees of anonymity are rarer still. Cash is technically anonymous, but cash expenditures are really only quasi-anonymous because we usually use cash in face-to-face purchases. There are several reasons for anonymity's rarity in conventional commerce:

- Anonymous transactions are hard to arrange and protect,
- Civil authorities have found that anonymity can undermine tax laws, and
- Anonymity is most necessary in illicit commerce.

Electronic commerce can make anonymous transactions easier to perform, but can do little about the other constraints. Thus, we see little need in electronic commerce for anonymous network mechanisms.

Non-Repudiation

Perhaps the most dramatic feature of public key cryptography is the non-repudiable digital signature, which does nothing to hide an electronic message's content but instead assures every reader of the message's authenticity. Digital signatures are almost perfectly analogous to handwritten document signatures, so it seems that electronic contracts could become commonplace in commerce, once the legislatures and courts agree that digital signatures should be legally binding.

It seems to us, though, that digital contracts are a solution in search of a problem. Crucially, how will digital contracts reduce transactional labor costs or improve a business' economies of scale? Without addressing cost-reduction needs such as these, digital contracts cannot attract markets and customers to the Internet.

Electronic commerce, growing as it is on an untrustworthy medium, tends toward a more paranoid trust model than a secure Internet would inspire. For example, signatures and anonymity have loomed larger than necessary in electronic commerce discussions because the Internet's insecurity arouses an unfocused and all-encompassing distrust in all of us. Eventually, commerce networks will provide "good-enough" end-to-end security transparently, electronic commerce's social cynicism will subside, and electronic commerce will fall back on traditional commercial trust-models, by-and-large. People will remember that after all, non-repudiable contracts aren't necessary for most transactions.

So, we disagree with the common claim that electronic commerce needs public key technology so as to gain absolute privacy, trustless administration, perfect anonymity, and electronic contracts. While we disparage these claims, we still accept the value of public key cryptography in commercial networking.

We believe that electronic commerce's primary benefit is scale, and that security issues are secondary. With networked marketing and networked payments, merchants gain larger markets and lower labor costs for transaction processing. Banks and credit-card companies expect to gain a high volume of per-transaction fees. Accordingly, to meet these expectations, public key's contribution to electronic commerce should address scaling issues. It is this need, at bottom, that smartcard-CAs address.

5. Raw Material

If we want to build a commerce system that includes smartcard-CAs, what do we have to work with? We suggest that it is smartcards, client software, certification, and delegation.

5.1. Smartcards

Treating this in biologic style, a "card" is a family of devices that "fit in a wallet." It has two genera, a "dumbcard" which is a passive device and a "smartcard" which is an active device. Within dumbcards, there are magnetic and non-magnetic species, *i.e.* requiring a reader or not requiring a reader. Within smartcards, there are wallet cards and PC/MCIA cards, distinguished mostly by the additional capacity that the much increased silicon real estate that PC/MCIA affords. All smartcards are gaining cost-effective capacity quickly, and some predict that they will shortly include biometrics such as thumbprint readers on wallet cards.

The availability of readers will sort itself out quickly. German Point-of-Sale conversion is happening now, Fischer International has a reader in a 3.5" floppy case, the Defense Logistics Agency began requiring PC/MCIA-capable machines beginning earlier this year, PC/MCIA-based smartcard readers are now circa $100 and cell phones and network computers alike now have the implicit premise of a smartcard as their chief protection. Other such examples seem to be appearing weekly.

We are interested in both the credit-card and the PC/MCIA cards which have a cryptographic co-processor and the ability to store at least two public key certificates, one for the card itself and one for the holder of the card. (With respect to some commercial systems, notably VISA's SET and Nortel's Entrust system, separate key pairs for signing and for sealing are required, so the number "2" may be higher.) We suggest the minimum of two certificates and a cryptographic co-processor to permit secure key management on the card, to admit the role of a security officer, and to separate "memory-only" functions from smartcard functions.

For important transactions to absolutely rely upon the smartcard, the private-key-half (of the public key pair) must never leave the card (the "trusted computing base"), but corporate convenience may well lead to key pair generation and escrow offboard the card. Regardless, we assume that the browser will be able to retrieve the user's "client certificate" whether via a PlugIn, a JAVA applet, or via some native browser capacity.

5.2. Client software

Client software to use such cards will likewise sort itself out quickly. Netscape is actively looking to add both PlugIn and direct browser support for public key smartcards, and is rumored to have engaged an offshore company to incorporate smartcard support (and full strength crypto) directly into the Netscape Navigator Web browser. Microsoft is similarly inclined; Microsoft's Crypto API performs client-side authentication based on certificates, secure certificate storage, and basic encryption/signature services. This Crypto API supports smartcards, in that smartcards can plug into the Crypto API layer from below to export their services to application software.

5.3. Certification and certs

In this section, we review the basics of public key infrastructure: certificates and certification authorities.

In commerce as in other social matters, people use names to identify other people, singly and in groups. However, the Internet community has learned through hard experience that simple exchange of names doesn't work well in packet communications. Computers can identify one another reliably only by their use of encryption keys. As the purpose of a certificate is to publicly bind a name to a key, we transpose our social identities into this electronic medium by means of public key certificates. The certification authority's job is to perform this transposition faithfully.

An RSA key-pair is in itself an electronic identity. When we see two widely-separated messages, both signed with the same private-key, we know the messages come from the same author, even if we don't know the author's name. In other words, without a certificate to bind a name ("identity") to that key we nonetheless have "reproducibility;" for many electronic applications, this is all we need.

For example, a file server might reasonably grant administrative rights to any client that can sign messages with an admin key-pair P_a. The file-server has no interest in the administrator's social identity, and the access-control mechanism can perfectly consistently "name" administrators by their keys, instead of by their conventional UIDs. This idea of using public keys as names is a cornerstone of the Simple Public Key Infrastructure (SPKI) proposal. [El96]

For commerce, though, we need more than electronic consistency. A certificate is what tells us, "Only Alice owns key-pair K." From this, we can hold Alice's social identity responsible for her electronic activities, as when:

• A merchant must re-possess Alice's purchase for non-payment,
• Her Web store's sales-tax payments are past due, or
• Her "GET RICH NOW" pyramid scheme collapses.

More generally, there are many commercial transactions in which prepayment is impossible and, in all such cases, plaintiffs need recourse to street addresses, summonses, and hard cash. At bottom, this is why "named key" certificates are necessary: to bind our electronic identities to our social identities.

In conventional applications of public key cryptography, public key certificates are issued only by trusted services called "Certification Authorities." A CA signs messages of the form, "Alice's public-key is NNNNNNN." The CA's job is to check Alice's identity before signing the cert, and perhaps also to ensure that she actually holds the corresponding private-key. Usually, the CA verifies Alice's identity by looking at her driver's license or some other photo ID. Other users can use NNNNNNN to prove Alice's identity, but only after they verify the CA's signature on Alice's cert. The user community trusts the CA not to bind one person's name to another person's key.

Practical resistance to attack on public key based security requires that public keys be packaged as certificates, and, given these constraints, certifying authorities are essential to commerce. In fact, the service of a public certifying authority is the only part of commerce that neither merchants, buyers nor financial processors provide and which, in addition, does not lend itself to being a commodity.

Certifying authorities are difficult to operate, the business model is hardly yet worked out, revocation and roll-over remain unfunded liabilities, *etc.* Indeed, the risk-benefit equation of operating a commercial certifying authority has been such that even superbly well prepared, widely experienced players (such as BBN) have demurred. We should also note that commercial CAs (such as VeriSign) have tried without success to sell certs as a commodity.

In calling CA services a scarce commodity with substantial barriers to entry, we describe a problem we wish here to solve. We propose an alternative that lessens the day-to-day dependence on centralized CAs while deepening the certificate tree.

5.4. Commerce as Delegation

The term "commerce" is slippery, but we here define it to mean simply a transactional exchange between two parties. In particular, the participants are not required to live within the same organizational structure. In this sense commerce can be a small dollar-value retail purchase but it can also be counter-signing a personnel report. Practically, we focus on business settings though we do not believe this to be crucial.

Going one step further, if the reader can accept a claim that money is simply generic authority, then a purchase and a delegation are virtually identical in the sense of a transfer of authority. It is a subtle but small step; colloquial use has heretofore associated "transaction" with the mutual exchange of value and "delegation" as but half that, a one-way transfer. Overall, this commercial use of SC-coined delegations has the advantage that it does commerce securely yet with fewer special-purpose financial protocols. We believe that as a primitive, SC-certs can simplify financial protocols generally. Our reasons for this claim are:

• Commercial transactions usually involve more than two or three parties,
• Adding secure connections to a protocol adds a lot of complexity,
• Some parties' involvement can be reduced to asynchronous authorization messages,
• Creating and interpreting authorization certs is straightforward.

Even though designing public key authentication protocols is harder than commonly appreciated [AnNe95], we believe that authorization security is easier than multi-party authentication.

For example, If Alice wants to buy a movie from Sony, she issues two certs to Sony: a "Sony may download" cert, and a "Sony may withdraw $3 from my account" cert. Insofar as it's true that sc-certs can replace e-comm protocols, this may be because we're breaking down the synchronous connections of the current transaction models. Sony gets two certs from Alice and uses them later at disparate times. This breaks a protracted & secure application into three almost unrelated messages, so the result doesn't look much like a single protocol.

6. Authorization

There are three great styles of authorization, to wit, authentication of principals coupled to distributed ACLs, role-authentication coupled to policy directives either in software or policy databases, and capabilities with varying degrees of coupling to (an owner's) identity. We suggest the following bet, *viz.*, the authorization solution that matters in an electronic world is that of roles. The "role" we mean is an authenticatable identity but an authenticatable identity which is that of a permission class (group) rather than of an instance (individual).

Role based access control has a rich literature. It stands somewhere between discretionary and mandatory access controls.[4] Since "authority" can be reduced simply to access to credentials, the expressive power of the X.509v3 certificate extensions can trivially accommodate roles (as v3 certs). Simply put, our bet is that roles will win.

- They will win because pure authentication technology can be reused for authorization, because roles lend themselves to the expression and enforcement of corporate and organizational policies, because roles reflect the structure of business organizations, and because they maximize security impact while minimizing costs, especially design costs.
- A capability is a token that in and of itself is sufficient to gain access to a resource, *e.g.*, a bearer bond. Capabilities are too hard to define and too hard to protect. It can be argued that money is the purest capability; it certainly is the most widely applicable form of authorization. Electronic capability systems have been built, including anonymous electronic cash mechanisms. [Ch92]
- Classic ACL solutions[5] — Central authentication with per-function ACLs synchronized across all servers offering the function are explicitly not scalable and will lose. ACL solutions that attempt to convert corporate policy into explicit enumeration of authorized principal identities suffer from both inconsistent conversion and important synchronization difficulties with respect to the changing pool of authenticable principals.

Putting it differently, roles win because:

	Access-Control Mechanisms		
Costs	ACLs	Capabilities	Roles
hard/easy to manage?	hard(-)	easy(+)	easy(+)
hard/easy to think?	easy(+)	hard(-)	easy(+)
hard/easy to steal?	hard(+)	easy(-)	hard(+)
Net Worth	+1	-1	+3

[4] Under discretionary access control the resource-owner's authority is sufficient to share information, whereas under mandatory access control information is shared only according to its own classification. Role-based access control strips the resource-owner of the authority to share information, *per se*, preferring rather to have the authorization authority be a separate security service, rather than the owner or the information itself.

[5] Clients provide authentication credentials to servers which look up clients in local permission databases.

7. Loose Ends

A further elaboration: For the time being, the easiest way for a consumer to receive purchased data might be to offer a secured FTP service. That is, the consumer sets up an FTP server that only accepts connections from clients bearing certs that the consumer's own smartcard has issued. Initially, this might be most useful for bulky entertainment media and for subscriptions to asynchronous news delivery.

It's also worthwhile for merchants to use smartcards to issue certificates. A merchant's CA-smartcard would issue authorization certs to customers. A customer's merchant-issued certificate constitutes proof-of-payment and, at the same time, the cert relieves the merchant of having to manage an ACL database. In essence, the circulating certs are the ACL database, in a distributed form. This use of CA-smartcards is similar to the current model of Web commerce, except that a merchant with a CA-smartcard doesn't have to manage an ACL, making it easier to allow (or require) a considerable time to pass between the customer's payment and the server's fulfillment. (Futures trading, anyone?)

One detail that we cannot neglect is revocation. Most CA-smartcards will issue short-lived certs, but, even so, revocation will sometimes be necessary. For example, if Alice issues a cert that authorizes a publisher to download stuff to her for a year and the publisher loses control of the certificate's private-key, she will need to be able to tell her machine to reject that certificate. It will not be necessary for her to manage CRLs, though; the merchant can check the cert it holds against the CRL service before each use and will receive a signed, time-stamped "cert OK" message. The merchant can send the "cert OK" message along with the cert itself at each use. Of course, it may be preferable to either issue a dozen certs at the outset, to re-issue the next cert with each cert used, or even to re-issue a counting cert (in an S-Key chaining style). Each of these represents introducing opportunities to revoke as well as moving in the direction of micro-certs.

For a system to be at once browser-independent and require no client-end software other than the browser, the natural solution is to use redirection of specially constructed URLs as a poor-man's RPC. Today we have

• Series of custom interactions between the browser and the merchant
• Redirect of the browser to a transaction manager
• Exchange of the browser's authentication credentials for authorization credentials issued by the transaction manager
• Redirect of the browser to the fulfillment site
• Exchange of both the browser's authentication and authorization credentials for synchronous delivery of the bought goods

Alternately, we propose:

• Series of custom interactions between the browser and the merchant
• Redirect of the browser to a transaction manager
• Exchange of the browser's authentication credentials for authorization credentials issued by the transaction manager
• Redirect of the browser to the fulfillment site
• Exchange of the browser's authorization credentials alone for asynchronous delivery of the bought goods

There are many advantages to simply using certs rather than vendor-specific structures, including adaptive reuse of the authentication technology that will inevitably be certificate-based. None of this is possible without smartcard (cryptographic co-processor) support, and smartcards plus this certificate strategy make the entire fabric of commerce location-independent in both space and time. From the perspective of any vendor in the electronic commerce space, it enables a transactional semantic for nearly any aspect of business involving authority or communication. For the consumer, it is the first way to be a player in the net using security skills commensurate with their prior experience yet not requiring their allegiance to any authority. For the merchant, it is yet another broadening of what constitutes commerce — and it is the successive calculus of these broadenings that explains why electronic commerce is indeed an idea whose time has come.

8. Conclusion

We believe that this notion of smartcard CAs is a new cryptographic primitive offering flexible application. Smartcard CAs are particularly valuable for electronic commerce, where they bring out the best of public key's benefits. We argue that overall, public key's advantages for commerce are less than they seem, because electronic commerce's purpose is cost-reduction, and public-key's security features do little to reduce per-transaction costs. However, smart-card CAs can help solve an open problem in electronic commerce, which has vexed designers of public key and symmetric key security alike. This problem is large-scale, secure authorization. We suggest that most payments problems should really be viewed as authorization problems, and that this approach is profitable precisely because smartcard-issued certs lend themselves especially well to authorization.

9. Bibliography

[AnNe95]
R.J. Anderson and R.M. Needham, "Robustness Principles for Public-Key Protocols," Advances in Cryptology - CRYPTO '95, Springer-Verlag, Berlin, 1995.

[BlFeLa96]
M. Blaze, J. Feigenbaum and J. Lacy, "Decentralized Trust Management," Proc. IEEE Symp. on Security and Privacy, Oakland, May 1996.

[Ch92]
D. Chaum, "Achieving Electronic Privacy," Scientific American, August 1992, pp. 96-101.

[Da96]
D. Davis, "Compliance Defects in Public-Key Cryptography," Proc. 6th USENIX Security Symp., San Jose, July '96. pp. 171-178.

[DaSw92]
D. Davis and R. Swick, "Network Security via Private-Key Certificates," Proc. 3rd USENIX Security Symp., Baltimore, Sept. '92, pp. 239-242. Also in ACM Operating Systems Review, v.24, n.4, Oct. 1990.

[DeDoLeMa93]
T. Denny, B. Dodson, A.K. Lenstra and M.S. Manasse, "On the Factorization of RSA-120," in Advances in Cryptology - CRYPTO '93, Ed. by Douglas R. Stinson, 1994, Springer-Verlag Lecture Notes in Comp. Sci. #773, pp. 166-174.

[El96]
C. Ellison, "Establishing Identity Without Certification Authorities," Proc. 6th USENIX Security Symp., San Jose, July, 1996, pp. 67-76.

[GiPaStTr95]
D. Gifford, A. Payne, L. Stewart and W. Treese, "Payment Switches for Open Networks," Proc. 1st USENIX Workshop on Electronic Commerce, New York City, July 1995, pp. 69-75.

[LaAbBuWo91]
B. Lampson, M. Abadi, M. Burrows and E. Wobber, "Authentication in Distributed Systems: Theory and Practice," 13th ACM Symp. on Operating Systems Principles, Oct. 1991, pp. 165-182.

[NeMe95]
B.C. Neuman and G. Medvinsky, "Requirements for Network Payment: The NetCheque Perspective," Proc. IEEE Compcon '95, San Francisco, March 1995.

[Od96]
A. Odlyzko, "The Future of Integer Factoring," CryptoBytes, RSA Data Security, Inc., 1996.

[VI90]
VISA International and Microsoft Corp., "Secure Transaction Technology Specifications," 1995.

10. Acknowledgements

The authors wish to thank the referees for their comments and Josh Lubarr, Henry Tumblin, Jack Rieden and other members of the Security Study Group at Open Market for their advice and support.

11. Availability

Patent pending; contact first author for queries of any sort.