Augmenting Internet-based Card Not Present Transactions with Trusted Computing: An Analysis

Shane Balfe and Kenneth G. Paterson

Technical Report
RHUL–MA–2006–9
24 October 2006

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

Contents

Abstract

In this paper, we demonstrate how the staged roll out of Trusted Computing technology, beginning with ubiquitous client-side Trusted Platform Modules (TPMs), can be used to enhance the security of Internet-based Card Not Present (CNP) transactions. This approach can be seen as an alternative to the proposed mass deployment of unconnected card readers in the provision of CNP transaction authorisation. Using TPM functionality (and the new PC architecture that will evolve around it) we demonstrate how TPM-enabled platforms can integrate with SSL, 3-D Secure and server-side SET. We highlight how the use of TPM functionality, as is currently being deployed in the marketplace, is not a panacea for solving all the problems associated with CNP transactions. In this instance, a more holistic approach requiring additional Trusted Computing components incorporating Operating System, processor and chipset support is required to combat the threat of malware.

1 Introduction

The Internet as an avenue for card-based commerce has seen something of a popularity explosion in recent years. In the United Kingdom alone, on-line shopping has become a multi-billion pound industry and in 2004 accounted for nearly 11 pence out of every £1 spent using credit cards. However, this particular form of commerce, typically referred to as Card Not Present[1] (CNP) transactions, whilst ubiquitous, is currently far from secure.

A recent report by the Association for Payment Clearing Services (APACS) on card fraud [2] showed that Internet-based CNP transactions and their associated chargebacks[2] accounted for nearly 27% of all card fraud perpetrated in 2005 in the UK. This translated into £117 million in losses for card issuers and merchants.

This proliferation of Internet-based commerce (and the increasing level of fraud associated with it) has resulted in a great deal of effort in developing protocols for securing these transactions. However, the vast majority of Internet-based payments are secured using a single protocol suite, namely SSL, to protect card account information.

Unfortunately, this usage of SSL is not a panacea for enabling secure Internet-based CNP transactions. SSL was not designed as a payment proto-

[1]For the remainder of this paper all references to CNP transactions refer to Internet-based CNP transactions.

[2]A chargeback is a term used to refer to the situation in which a genuine cardholder reports an unknown and possibly fraudulent transaction to their card issuer.

col but instead adopted as a de facto standard for securing CNP transactions. Indeed, the use of SSL in CNP transactions has a number of short-comings. These 'flaws' in SSL can largely be attributed to the marriage of convenience that exists with current CNP-based card processing and are not necessarily intrinsic to the protocol itself. For example, SSL is used only in relation to securing the payment channel; there is no guarantee that the customer owns the account number being proffered in a particular payment transaction. In this regard, transaction processing is reliant on a Mail Order Telephone Order (MOTO) based system whereby demonstrating knowledge of a card's Personal Account Number (PAN) and corresponding Card Security Code (CSC) are deemed a sufficient form of transaction authorisation.

To address some of these inadequacies other proposals for securing CNP transactions, such as the *i*KP protocols and their successor, the Secure Electronic Transaction (SET) protocol have been proposed. However, whilst offering additional security benefits over an SSL-based approach, neither protocol suite has seen wide-spread adoption. One relatively new proposal, however, namely 3-D Secure, appears to becoming widely deployed. 3-D Secure is an optional adjunct the SSL-based approach and attempts to provide cardholder authorisation for CNP transactions by requiring customers to authenticate themselves prior to transaction processing. This authentication forms an ancillary step to regular merchant checkout processing where, after receiving a customer's PAN and CSC, a merchant site redirects its customer to a 3-D Secure Access Control Server (ACS) to which the customer authenticates. If successfully authenticated, the ACS informs the merchant who then proceeds with regular transaction processing based upon the previously supplied account details. This approach aims to tackle the fraudulent acquisition of card account details for use in CNP transactions by providing a delineation between card authentication data and customer authentication data. However, this approach has only limited security benefits in the face of the omnipresent threat of malware such as trojans and keystroke loggers, a threat which is increasing at an unprecedented rate [25]. In this setting a piece of malicious software residing on a customer's platform could capture user authentication credentials and manipulate transactions (including possibly instigating new transactions).

To address this issue there has been a recent development to strengthen 3-D Secure's authentication process through integrating with EMV[3] chip cards. This approach involves the use of "unconnected" card readers which, when interacting with a customer's physical card, generate a one-time passcode on a per-tranaction basis [22]. This passcode would then be used instead of

[3]http://www.emvco.com/

a customer-supplied password for 3-D secure authentication. However, this approach suffers from the costs associated with distributing card readers to end-users, and as yet there are no publicly available specifications detailing the precise operation of such a system. Additionally, there have been recent reports of time-of-check to time-of-use attacks on similar two-factor authentication schemes [21].

This paper examines how Trusted Computing can be used to enhance the security of existing protocols (SSL, SET and 3-D Secure) in the provision of secure CNP transactions. In doing so, we highlight a number of well known weaknesses in their (unmodified) deployment and show how they can be addressed using Trusted Computing. In particular, we will examine the role of client-side certification in the context of Trusted Platform Module (TPM) enabled platform ubiquity. The idea of using Trusted Computing to enable client-side certification has previously been discussed in [10, 1, 6] as well as in the as-yet-unpublished TCG's (Trusted Computing Group) TLS extensions for carrying attestations. However, none of the work presented thus far takes into consideration the threat posed from malware nor the infrastructural requirements necessary to support client-side certification. Other related work includes the use of Trusted Computing as an adjunct to securing connected card readers for generating digital signatures, presented in [26] and [5]. However, both approaches, much like the unconnected card reader proposal outlined above, suffer from costs associated with the provision of card readers to end users. Additionally, both proposals assume the presence of trusted software to interact with the readers.

Our proposal centres around the use of a TPM to provide a small amount of trusted cryptographic functionality to bind a platform, and by extension its owner, to a particular card. One of the most salient issues in the provision of such functionality is the problem of customer enrollment, during which a customer/card binding is established. We examine different system architectures and discuss the pros and cons of their associated enrollment procedures. The real world applicability of this approach is demonstrated by the number of TPM-enabled platforms currently in deployment. Currently available sales figures for 2005 [9] showed estimates of 32% of all notebook systems shipped that year being TPM enabled. This figure is expected to nearly triple by 2007 with similar growth expected in other device types.

This paper is structured as follows. In Section 2 we introduce the steps involved in a CNP payment clearing process as well as introducing a number of protocols used to protect CNP transactions. In Section 3 we introduce some of the core concepts of Trusted Computing that we will later apply to securing CNP transactions. In Section 4 we examine the issue of customer enrollment with particular emphasis on the establishment of customer-centric

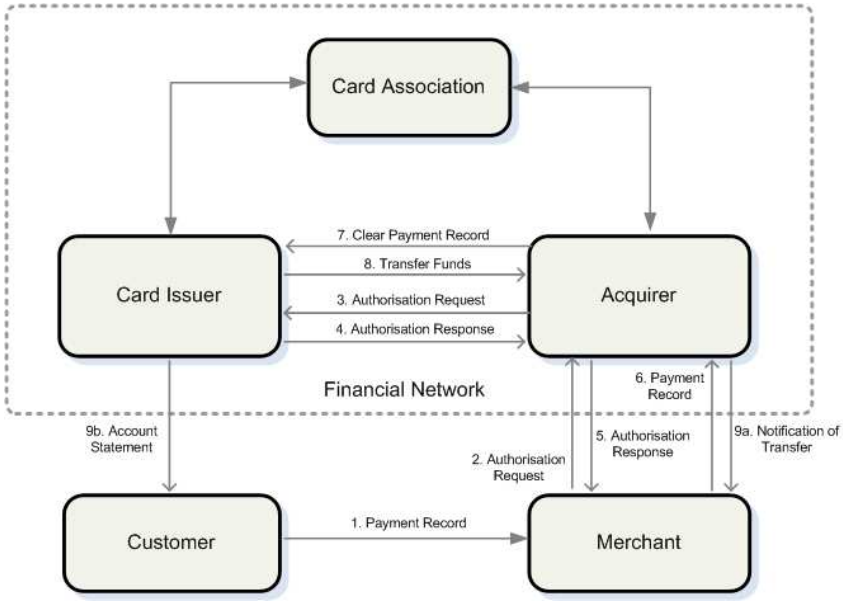

Figure 1: Generic model for card processing.

credentials within a TPM-enabled platform. Section 5 examines the role these TPM-enabled customer credentials can play in supplying additional security to the protocols outlined in Section 2. Finally, we conclude with Section 7. For the remainder of this paper we use terms cardholder, client and customer interchangeably.

2 CNP transactions and the Internet

This section begins with an overview of the generic four corner model used in card payment systems before moving on to discuss some of the more significant protocols used for securing CNP transactions. In describing this model (also referred to as a pull model) a number of steps are necessary to complete a given transaction (see Fig. 1).

- **Step 1**: The process begins with a customer signaling their intent to purchase goods by forwarding a payment record to a merchant. In this instance, the actual characteristics of a payment record differ depending on the environment in which it was created. For an on-line purchase, a payment record typically includes the information embossed on the customer's physical payment card in conjunction with certain merchant supplied information (such as the invoiced amount).

- **Steps 2-5**: These steps occur immediately after receiving the customer's payment record. They consist of a merchant submitting the transaction details to their acquirer which will either authorise or reject the transaction based on their interactions with the customer's card issuer. After this, the merchant will either confirm payment or inform the cardholder that their transaction has been rejected.

- **Steps 6-9**: Based upon the transaction being approved, either as a result of a successful outcome from steps 2-5 or merchant risk management routines, steps 6-9 represent the account settlement process through which funds are debited from a customer's account and credited to the merchant's.

Perhaps the most surprising feature of this model, is that a positive transaction authorisation (step 5) does not guarantee payment for a merchant. It is merely an indication that the card account details being proffered have not been reported stolen and that the customer has sufficient funds to cover the transaction amount. Indeed, unless the card has been reported stolen, it is impossible for a card issuer, and by extension a merchant, to ascertain whether a particular transaction is fraudulent or not.

In this regard, the merchant trusts (hopes) that the customer is the valid account holder (or at least a delegate of the primary account holder) for the presented payment record. This trust, or lack thereof, is largely underpinned by the level of indemnity offered by card issuers to their customers in the case of lost or stolen cards being used in illegitimate transactions. However, the level of indemnity afforded to merchants is dependent on their adherence to their acquirer supplied Merchant Operating Guidelines (MOG). The MOG lays out the procedures that should be followed when processing CNP transactions. An example of such a procedure would be a requirement to use an Address Verification Service (AVS) which compares the billing address, as entered by the customer, to that of the card issuer's records. If they match this is seen as an indication that the customer owns the card being used. In many cases a merchant may be held liable for chargebacks associated with a transaction if they do not properly perform cardholder verification. This verification is more difficult to do in a CNP setting.

2.1 SSL

The Secure Socket Layer (SSL) protocol was first introduced in 1994 by the Netscape corporation. The protocol itself was designed to provide end-to-end security services to connections running over TCP/IP and has since become

the de facto standard for the secure transmission of CNP transaction information. However, this use of SSL can be seen as more of a highjacking of an existing technology rather than a systematic approach to securing CNP transactions. In this regard, SSL establishes a session between a customer and a merchant and acts as a facilitator for the secure transfer of account details, of which, quintessentially, the PAN, CSC and relevant billing information are all requisite elements. SSL's primary advantage, and perhaps the main reason for its pervasive deployment, is that it requires no additional equipment for a cardholder and not much additional inconvenience for a merchant. However, what happens outside of an established transfer session is not within the scope of SSL's protection remit.

In this respect, the confidentiality and integrity afforded by SSL only protects against attacks from parties attempting to eavesdrop on a transaction between a customer and a merchant. It says nothing as to validity of the data emanating from either end-point. Potentially the biggest deficiency in the use of SSL for CNP payments is the lack of customer authentication. Even though SSL provides a provision for client (customer) authentication, it is seldom, if ever used. This stems from the inconvenience and cost associated with distributing and managing client certificates. A further issue relevant to client certificates, as mentioned in Section 1, is the problem of the perpetual increase in malware-affected platforms. If the private component of a key bound to a client SSL certificate is exposed to malicious software on a platform, then it becomes impossible to attest with any certainty that an entity purporting to be certified is as claimed.

2.2 SET

SET differs from SSL in that it was designed explicitly as a payment protocol and addresses a number of the deficiencies found in the SSL-based approach for facilitating on-line card-based commerce. However, despite improvements over an SSL-based approach, SET is no longer being deployed for use in CNP transactions. A number of theories have been put forward to explain why SET never became a success. These range from ease of use to the cost and difficulty of maintaining a stable PKI. For a more through treatment of SET than the one presented here, we refer readers to [24, p.100-123].

SET allowed every entity that was party to a transaction to be authenticated. SET used a certification authority hierarchy in which all participants were required to enroll. Certificates were then exchanged allowing authentication to occur. When it came to making a purchase within SET, a purchase order message would be constructed in such a way that only the merchant could see the Order Information (OI) and only the payment gateway could

see the Payment Information (PI). This was accomplished though what was termed a 'dual signature', whereby messages intended for the merchant and messages intended for the payment gateway could be linked without simultaneously revealing both. In this instance, the PI comprised transaction related data as well as a transaction ID which were then carried in a "digital envelope". A one time session key was created for bulk encryption which then encrypted the PI, the dual signature and a hash of the OI. This key was then enciphered with the public key of the payment gateway. Only the gateway could decrypt the said envelope and obtain the key to decrypt the enciphered PI.

Authorisation was more complex insofar as a payment needed to be authorised by both the customer and their bank. The merchant forwarded the encrypted PI information to the payment gateway along with their own authorisation information. This information comprised, amongst other things, an authorisation block containing a transaction ID encrypted with a session key and signed with the merchant's private key. This allowed the payment gateway to verify both parties by their respective signatures as well as confirming that they were referring to the same transaction by comparing ID values. One of the nice features of SET was that it did allow for non-repudiation of transactions — assuming the transaction was authorised by the card issuer.

2.3 3-D Secure

3-D Secure and both Visa's [3, 4] and MasterCard's [20] proposals, Verified by Visa (VbV) and SecureCode respectively, attempt to provide cardholder authorisation for Internet-based CNP transactions, and in this respect, can be seen as an adjunct to the SSL-based approach outlined in Section 2.1. Both proposals are designed solely to provide cardholder authorisation and both require customers to preregister their account with their card issuer prior to using the system. During the registration procedure the cardholder chooses a secret password that will later be used to authorise subsequent CNP transactions. These authorisations may later act as non-repudiable evidence in case of a dispute. Both the VbV and SecureCode proposals provide equivalent functionality (as they are both derivations of 3-D Secure), so we will concentrate our discussion on Visa's proposal as an illustrative example.

In the VbV approach, during the payment phase of a transaction a customer's browser is redirected by a merchant plug-in component to an appropriate ACS for their account. The customer authenticates to this ACS by providing their username and password, as established in the registration

phase. Based upon the correctness of the supplied username/password combination, the ACS formulates its response (authenticated/not authenticated) and signs it. This signature is then passed through the customer's browser and onto the merchant plug-in. The plug-in then verifies the ACS signature and decides if it wishes to proceed with the transaction. A validated response can later be used as evidence to show the customer authorised a particular payment. If the customer account number is not registered with any ACS, a visa directory server informs the merchant plug-in and normal MOTO-based authorisation procedures are attempted.

The use of 3-D secure (and its derivatives) can be seen as forcing an additional customer authentication prior to the completion of a transaction. However, given the nature of current implementations, especially with regard to the static nature of current authentication information (based on passwords), it is difficult to see how authoritative this authentication would be, and how non-repudiable the evidence of transaction authorisation would be. Indeed, there are various threats that affect the security of any CNP proposal, most notably spyware and phishing attacks. However, 3-D Secure's real benefit comes in reducing the economies of scale possible with card skimming attacks: an attacker obtaining a customer's card details, possibly by means of a compromised POS terminal, will no longer be able to complete a fraudulent purchase using the obtained information as a PAN and CSC are no longer sufficient to authorise a CNP transaction authorisation. Unfortunately, in this instance the use of a static authenticator may prove no less of a barrier to obtaining card account details. Perhaps the greatest threat to such a scheme would be that of an automated attack script that compromises cardholder platforms and installs malware that monitors keyboard activity and generates new transactions using the observed authorisation data. Additionally, a phishing site that purports to provide a 3-D secure plug-in capability could potentially dupe cardholders into revealing authentication data.

2.4 Disposable Credit Cards

Disposable credit cards are cards that limit the exposure of account details by generating a new PAN/CSC for each new transaction. The majority of these schemes are proprietary but ostensibly follow the same *modus operandi*. A customer downloads an application which, when requested, establishes a connection to the customer's card issuer. Based upon customer supplied information, such as an upper value limit, the card issuer generates a new PAN/CSC combination from the range allocated globally to them. These new account details are sent back to the customer who may present them to

a merchant server during checkout. As the newly generated account details are indistinguishable from normal account details the merchant can proceed with conventional processing behaviour. Such schemes have been piloted by various card issuers and banks, such as: American Express (Private Payments), Discover (Discover Deskshop), Citibank (Virtual Account Numbers) and more recently in the form of a TSB and Visa collaboration: 3V Transactions Services. The latter scheme takes a slightly different approach whereby account details for CNP use are purchased at POS terminals. However, all these schemes, just like the ones presented earlier, suffer with respect to platform subversion attacks as outlined in Section 1.

3 Trusted Computing

This section highlights two important sets of specifications that are germane to our discussion, namely the Trusted Platform Module (TPM) [15, 16, 17] and the Trusted Network Connect (TNC) specifications [12, 13, 14]. Trusted Computing as discussed here, relates directly to the type of system espoused by the TCG (Trusted Computing Group).

3.1 TPM Specification

The TPM forms the core of all efforts in instantiating the TCG's definition of a trusted system. The TPM itself comes in the form of a microcontroller with Cryptographic Co-processor (CCP) capabilities that resides on a platform's motherboard. The TPM, as well as offering a secure storage area for keying material, is capable of providing the following functionality:

- **Protected capabilities and shielded locations**: The TPM provides secure areas in which a platform can operate on sensitive data.
- **Integrity Measurement and Storage**: The TPM is assumed to be adept at making (and storing) intrinsically reliable integrity measurements pertaining to a platform's current state.
- **Reporting and Attestation**: The TPM has the ability to faithfully recount a platform's current state to third parties. The mechanism through which this is achieved is referred to as 'remote attestation'.

In providing this functionality there are two cryptographic keys in particular that hold a special meaning. These keys are the Endorsement Key (EK) and the Attestation Identity Key (AIK). Within a TCG-conformant

platform, AIK key pairs act as aliases for the EK and are responsible for attesting platform states. AIK pairs are used because an EK pair is unique per TPM instance and this is considered a possible risk to user privacy should the EK pair become connected with personally identifiable information. As there is no prescribed limit on the number of AIKs that can be used within a platform, this provides an anonymity mechanism, whereby the TPM can use different AIKs each time it attests to platform integrity metrics.

However, in order for an AIK to have meaning outside of the confines of a particular platform, it is necessary for the platform to obtain a credential for an AIK from a trusted third party. How this credential is obtained differs between version 1.1b and version 1.2 of the TCG specifications. Version 1.1b uses what is referred to as the "Privacy CA" model whilst version 1.2 introduced a new model in the form of Direct Anonymous Attestation (DAA) whilst retaining the Privacy CA model for backward compatibility. However, for the remainder of this paper we will concentrate our discussion solely on the Privacy CA model.

Within this model, credential acquirement is achieved as follows: a Collate Identity Request command [19, pp.111] is issued by a platform prior to the generation of an AIK key pair, this command gathers all the required information necessary for a Privacy CA to examine the requestor's platform. This information includes various credentials that vouch for the trustworthiness of the TPM itself. Provided the evidence presented by a user's platform is validated by the Privacy CA, the Privacy CA will encrypt the newly generated AIK credential with a symmetric key, which in turn is encrypted with the EK of the requesting platform. In this way only a specific platform is capable of decrypting the credential and performing the TPM_ActivateIdentity command [17, pp.151]. This then allows an AIK private component to be used to generate signatures over platform integrity metrics.

A recent addition to the concept of remote attestation has been the introduction of the Subject Key Attestation Evidence (SKAE) X.509 extension [10]. This extension provides a standard mechanism through which a verifying party can be assured of the security properties of a private key within a TPM. The security properties of a private key include both key type, which indicates whether a key is migratable or not, and attribute designation, which indicates what the key can be used for: signing, storage or both. After obtaining an AIK credential (following the method outlined above), a user signs the public component of either a non-migratable key pair (a key which is not allowed leave a TPM in an unencrypted form) or a Certified Migration Key pair (CMK, a key which is allowed to leave a TPM but only under strict conditions). The signature on the public component is produced using the private component of an AIK. The user then applies to an SKAE CA for

certification of the corresponding TPM-controlled (non-migratable or CMK) public key. If the CA is satisfied as to the AIK/public key binding, then a public-key certificate is issued by the CA to the platform. Here the certificate not only includes the public key which has been cryptographically bound to a TPM but also includes enough information for the relying party to validate this binding.

3.2 TNC Specification

The Trusted Network connect (TNC) specification forms a expatiated subclass of the Infrastructure Work Group (IWG) interoperability specification [11] and deals predominantly with enabling the enforcement of operator controlled policies for endpoint security in determining network access.

TNC can be seen as an enhancement to the IETF's AAA authorization frameworks [27, 28, 29] in offering a way of assaying an endpoint's integrity to ensure it complies with a particular predefined policy. A particular instance of this would be ensuring that a certain software state exists on a platform prior to the platform being granted network access, for example, requiring anti-viral or software patch updates to be installed. The means through which this is achieved follows a three phase approach of assess, isolate and remediate which we briefly discuss next.

The assess phase deals predominantly with an Access Requestor (AR) wishing to gain access to a restricted network. In this phase the Integrity Measurement Verifier (IMV) on a Policy Decision Point (PDP) examines the integrity metrics coming from the Integrity Measurement Verifier (IMC) on the AR's platform and compares them to its network access policies. From this process of reconciliation the PDP informs a Policy Enforcement Point (PEP) of its decision pertaining to an AR's access request. The PEP is then responsible for enforcing the PDP's decision. As an extension to the assessment phase, in the event that the AR has been authenticated but failed the IMV's integrity-verification procedure, a process of isolation may be instigated whereby the PDP passes instructions to the PEP which are then passed to the AR directing it to an isolation network. The final phase, remediation, is where the AR on the isolation network obtains the requisite integrity-related updates that will allow it to satisfy the PDP's access policy.

4 Applications of Trusted Computing to CNP Transactions

In this section we will look at the issue of customer enrollment with a view to obtaining certification of a TPM-controlled (non-migratable or certified migratable) key. We present a number of different system architectures through which enrollment may occur and discuss the issues of client-side certification in the face of the omnipresent threat of malware. Related work pertaining to TPM-enabled platforms enrolling with Trusted Computing aware CAs can be found in Section 6.

4.1 Enrollment

This section aims to explore different architectural options for enrolling a platform, and by extension its owner (cardholder), using a card-issuer-controlled Trusted Computing CA. The goal here is for a cardholder to obtain an X.509 certificate incorporating both card account details as well as a cardholder's public key, with the corresponding private key being inextricably bound to the cardholder's TPM. This certification by the card issuer will effectively bind a cardholder's hardware platform to a particular card. The cardholder can later demonstrate this binding when authenticating himself to a merchant during a CNP transaction. Thus the TPM acts as both a secure storage area for the cardholder's private key as well as providing a means by which the use of the private key can be controlled.

In order for a card issuer to provide an enrollment facility for their customers' platforms, it will be necessary for the card issuer to provide some form of CA functionality. This functionality can take the form of either a Privacy CA, an SKAE CA or possibly both, and will allow cardholders to enroll their platforms with their card issuers. As we saw in Section 3.1, in order for a platform to obtain an X.509 certificate for a TPM resident key it is necessary to go through a number of steps. A platform at the behest of its owner (the cardholder) first makes a request to a Privacy CA to certify an AIK public key. The corresponding AIK private key is then used to sign the public key of a non-migratable TPM key pair. This signed non-migratable public key is then sent to an SKAE CA who certifies that the private portion satisfies certain key type and attribute designation constraints (as evidenced in the TPM_Certify_Info structure) before issuing an X.509 certificate on the non-migratable (public) key.

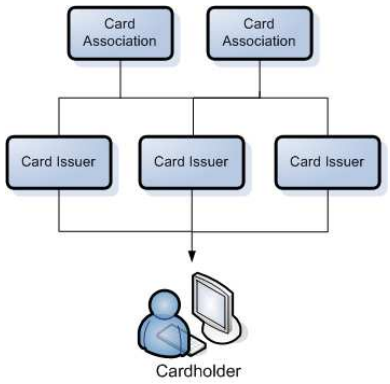

Figure 2: Certificate Enrollment Hierarchy

4.1.1 Deciding on an Architecture

Figure 2 shows the general certificate enrollment hierarchy in which customers can enroll with multiple card issuers who in turn can enroll with multiple card associations. The cardholders themselves have no direct dealing with the card association but instead interact with the enrollment interfaces exposed by their card issuers. In defining these interfaces there are various design decisions related to a card issuer providing Privacy/SKAE CA functionality. These can be broken down as follows:

Privacy CA

In many Trusted Computing settings, the Privacy CA approach appears to be impractical as there is no clear business case in offering such a service. However, in the case of CNP transactions there is a natural party, in the form of a card issuer, who can fulfil this role. By acting as a Privacy CA a card issuer can issue AIK certificates to its customers' TPM-enabled platforms. Unfortunately, the usefulness of this approach is limited by the fact that an AIK is only allowed to sign integrity metrics and non-migratable/CMK keys, but not information generated outside a TPM. Additionally, there is a potential privacy concern for customers in disclosing a platform's EK public component to a non-manufacturing entity. As an EK is unique per platform instance it may act as a 'super-cookie' in identifying subsequent platform actions across multiple domains.

SKAE CA

By acting as an SKAE CA a card issuer can issue X.509 certificates on non-migratable/CMK keys to customers' TPM-enabled platforms.

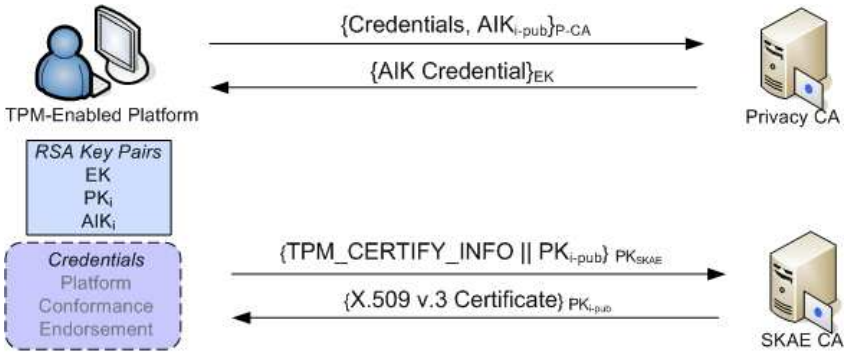

Figure 3: Card Issuer Controlled Privacy/SKAE CA

Once this certificate is received it can be used in future transactions, either in support of a 3-D Secure authentication (see Section 5.2) or during an SSL handshake (see Section 5.1). In providing this service a customer's card issuer does not need to provide Privacy CA functionality. This can be provided by an entity that is in the best position to do so, typically a TPM manufacturer. However, a card issuer would need to trust the outcome of the Privacy CA AIK credential issuance procedure that precedes a customer's SKAE application. This solution can be seen to offer additional anonymity to a customer's platform as it breaks the link between an EK and an AIK by having a Privacy CA outside of the bank's domain handle this mapping.

Hybrid CA

The final option is to have a card issuer act as a dual Privacy/SKAE CA. This is perhaps the most pragmatic solution for customer enrollment as it avoids the assumption that Privacy CAs are widely available. It also has the added benefit of shortening the customer enrollment procedure. Instead of making two separate CA requests, a customer generates an AIK and a non-migratable/CMK key pair and signs the public component of the non-migratable/CMK key using their private AIK key. The AIK/SKAE certificate request package is then bundled and sent to the Hybrid CA which processes each component individually before issuing an AIK credential for the AIK and an X.509 certificate for the non-migratable/CMK key.

Figure 3 shows the most generic case where a Privacy CA and SKAE CA are distinct entities. Obtaining a X.509 certificate for a TPM-bound non-migratable key is a result of the following process:

1. The cardholder instructs their TPM to create an AIK key pair, AIK_{i-pub} and AIK_{i-priv} for the public and private components respectively.

2. The cardholder instructs their TPM to generate a certificate request package for their card issuer's Privacy CA in order to obtain an AIK credential for their newly generated AIK key, AIK_{i-pub}.

3. The Privacy CA validates the cardholder's request and issues an AIK Credential to the cardholder's TPM.

4. The cardholder's TPM receives the AIK Credential and the cardholder instructs their TPM to activate their AIK, AIK_{i-priv}.

5. The cardholder instructs their TPM to generate a key pair K_{i-pub} and K_{i-priv} with K_{i-priv} having the following properties: its type should be non-migratable, its attribute designation should be signing only, and the use of the key should always require authorisation which the cardholder now supplies.

6. The cardholder instructs their TPM to certify (sign) K_{i-pub} generated in Step 5 using AIK_{i-priv} generated in Step 1. This creates a signed TPM_CERTIFY_INFO structure [16] describing the security properties K_{i-priv} from step 5.

7. The cardholder instructs their platform to create an SKAE extension. This extension acts as a receptacle for a TPM_CERTIFY_INFO structure [16] from the preceding step.

8. The cardholder instructs their platform to create a certificate request package incorporating the SKAE extension from the previous step. During this process the cardholder authenticates themselves to their card-issuer-controlled SKAE CA. This authentication would involve demonstrating knowledge of their payment card's PAN, CSC, address as well as a secret Personal Identification Number (PIN) or password[4].

9. If the card issuer SKAE CA is satisfied with the above information then the SKAE CA issues an X.509 v.3 certificate containing a customer's PAN with an SKAE extension incorporating the K_{i-pub} of the non-migratable key pair generated in Step 5. The inclusion of the PAN in the certificate provides a mechanism through which a card can be demonstrably bound to a platform and by extension the platform's

[4]We assume a secret PIN or password would be provided to cardholders using an out-of-band mechanism, similar to that currently used in Internet banking.

owner (cardholder). The exclusion of the CSC from the certificate removes certain security issues with respect to backward compatibility. Without a CSC/PAN combination, an adversary cannot engage in traditional MOTO-based payment authorisation. Thus the absence of the CSC from the X.509 certificate effectively neuters the value of the PAN to an adversary. Additionally, a subject can be identified using X.500 systax which can be used directly in an AVS system (see Section 2). Finally, setting a validity period can further constrain a card's usage, as is common in physical deployments.

10. The cardholder's platform receives the certificate from their issuing bank.

Whilst it may appear that the burden for a cardholder is exorbitant in the above protocol, in reality an application such as a card issuer supplied applet that interacts with a platform's Trusted Software Stack could perform the majority of the cardholder's interactions with a TPM. The cardholder would only need to select and enter an authorisation string at Step 5 and a PIN/password at step 9.

4.2 Client-Side Certification and Malware

The concept of client-side certification, as outlined in Section 4.1, works well if we assume an attack model that centers around external threats. However, as we have seen in Section 1, a model which only considers external threats is not always appropriate in CNP transactions. In order for a cardholder to generate a signature using the private component of the key referenced in the X.509 certificate, the cardholder needs to send authorisation data to their TPM to activate their signature key. It is important that we secure the "channel" over which this authorisation travels. However, in the absence of additional Trusted Building Blocks (TBBs), such as Intel's La Grande[5] or AMD's Pacifica[6], this authorisation information may be observed and replayed by malware to obtain access to the private key and generate signatures on unauthorised transactions.

Both the AMD and Intel initiatives aim to provide a number of hardware features which can be exploited by next generation Operating Systems to provide security properties to an executing process:

- No interference: Ensuring that the program is free from interference from entities outside its execution space.

[5]http://www.intel.com/technology/security/
[6]http://www.amd.com/us-en/Processors/

- Trusted path: Provides a trusted path between a program and an input device.
- Secure inter-process communication: Enabling one program to communicate with another, without compromising the confidentiality and integrity of its own memory locations.
- Non-observation: An executing process and the memory locations it is working upon should be free from observation.

As we saw in Section 1, malware is beginning to target personal computers in order to obtain user credentials. The mechanisms used to achieve this are diverse and can be seen to target the absence of each of the four properties that new AMD and Intel hardware aim to provide. Our proposal thus far does nothing to prevent such attacks. Indeed, given the current absence of (AMD and Intel provided) building blocks in the market it is difficult to prevent a sufficiently motivated attacker from obtaining user credentials within a platform.

Given current market constraints, one possible mitigating solution to the malware problem would be to use the current TCG mandatory requirement for TPM-enabled platforms to be able to demonstrate physical presence through a secure attention sequence to a TPM. Physical presence as defined by the TCG is a signal from the platform to the TPM that indicates operator instigated hardware manipulation of the platform. Examples of such manipulation would include "depressing a switch, setting a jumper, depressing a key on the keyboard or some other such action" [15]. The combination of customer provided card account details and evidence of the successful completion of a secure attention sequence can demonstrate that an authorised customer instigated the transaction. Only a person physically present at a computer can demonstrate physical presence and only an individual who knows the correct password for K_{i-priv} can load the key for use in a transaction, see Section 5. If malware were to surreptitiously observe cardholder authentication data, it would be impossible to generate new clandestine transactions as malware would be incapable of generating a corresponding secure attention sequence. Unfortunately, in this instance, user education surfaces as a potential weak link in the security chain. Malware may fool a user into providing a demonstration of physical presence.

Regrettably, the manner in which physical presence functionality is presented to an end-user is entirely dependent on how a manufacturer chooses to implement it. In this way customer education may be a difficult obstacle to surmount given the heterogeneity of physical presence implementations amongst manufacturer devices. In this setting, attesting to physical presence

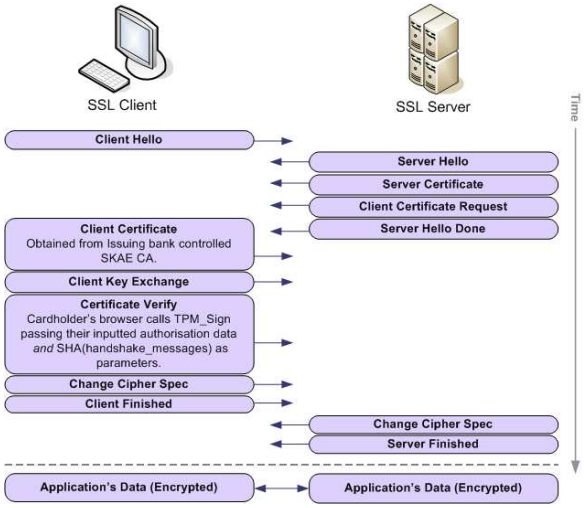

Figure 4: SSL Augmentation Authentication

may be better suited to constrained devices such as mobile phones that conform to the upcoming Trusted Mobile specifications. Here the mechanism used to demonstrate presence would be restricted by a limitation in how such a function can be presented to an end-user whilst still being functional. Failing this, hardware that supports compartmentalised memory that is free from observation and interference coupled with trusted I/O may be the only viable solution.

5 Augmenting Existing Protocols with Trusted Computing

5.1 SSL Augmentation

SSL augmentation involves the addition of client (customer) authentication as provided (but seldom used) in standard implementations of SSL. Under the assumption of ubiquitous Trusted Platforms and the corresponding infrastructure that will be necessary to support them, we can use the enrollment mechanism outlined in Section 4 to provide a bootstrapping mechanism for providing client-side SSL certification.

The SSL process described here is identical to that of a standard SSL handshake in which client (cardholder) certification is requested by the server (merchant). Here the server requests a certificate by sending a list of certificate authorities that it is willing to participate with in accordance with its

MOG. These requested certificate authorities may take the form of one or more root CAs (card associations) or of one or more subordinate CAs (card issuers) depending on the constraints a merchant's acquirer wishes to place on the type of payment cards they are willing to accept.

If the client is in possession of an X.509 certificate that satisfies the merchant's request, then the cardholder's platform forwards this certificate to the merchant along with a certificate verify message. This certificate verify message provides a proof of possession for the private key, K_{i-priv}, corresponding to the public key, K_{i-pub}, referenced in the client certificate. Here a customer's TPM is responsible for performing customer authentication prior to using K_{i-priv} to generate the certificate verify message. The process of generating this certificate verify message requires the authorisation data for K_{i-priv} (as supplied by the cardholder) as well as all the handshake messages exchanged thus far. These two parameters are input to a TPM_Sign command [17]. This command checks to see if the provided authorisation data matches the authorisation data stored with the private component of the requested non-migratable key. If they match then the TPM uses the K_{i-priv} to generate a signature over the provided handshake messages. This signature is then passed to the merchant server for validation, subsequent to which the SSL handshake protocol proceeds as normal.

Assuming the presence of additional Trusted Building Blocks as outlined in Section 4.2, the primary advantage to this approach is that a remote verifier can gain (implicit) assurances about the protection levels associated with a private key used in performing a SSL handshake, something which is currently impossible without expensive customised hardware. In this instance, the key used to authenticate a customer (and authorise a transaction) is not exportable to applications outside of a TPM. Interactions with K_{i-priv} can only be as a result of using protected capabilities to communicate authorisation data to a customer's TPM. In this way, only a legitimate owner of the key can have interactions with it. In addition to this, it makes the targeting of merchant servers and third party processing facilities, as evidenced by [30], redundant. This is because exposed customer account details would be unexploitable without demonstrating possession of a corresponding private key. However, without the presence of additional TBBs this approach suffers with respect to malware-based replay attacks of the type discussed above. A further disadvantage of this approach is the increase in server loading that would result from using client-side authentication. It may be necessary for merchants to provision additional hardware to cope with the increased processing demands. This is especially true when attesting to the presence of TBBs within a platform. A merchant will need to verify the presence of such functionality prior to transaction initiation by a cardholder in order to satisfy

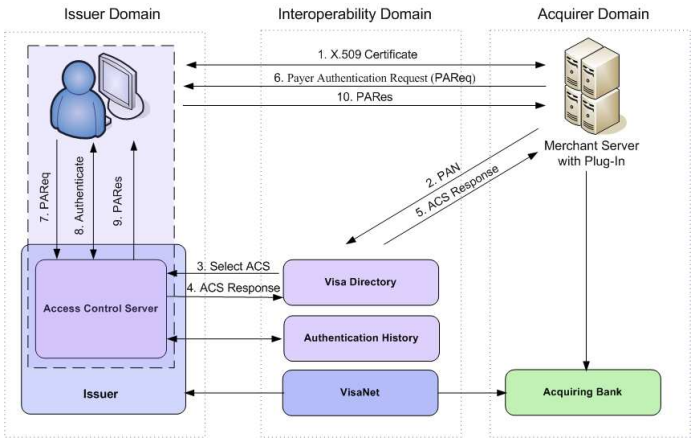

Figure 5: 3-D Secure Authentication

any requirements laid out by their acquirer's MOG.

5.2 3-D Secure Integration

As we saw in Section 1, there has been some movement recently in bringing unconnected card-readers (based on Mastercard's Chip Authentication Program (CAP) proposal) to market.

An alternative approach could be to use Trusted Platforms as a means of authorising transactions. Enrollment into a 3-D Secure like environment would occur as laid out in Section 4. A TPM-enhanced 3-D Secure purchase transaction flow would proceed as follows:

1. **Initiate Purchase:** This stage is representative of the typical 3-D secure initiation procedure revolving around the merchant plug-in component, and can be seen in Steps 1-6 of Figure 5. Pursuant to a customer payment initiation request, the merchant plug-in contacts the Visa Directory Server which provides the address of an appropriate ACS. The ACS response is then forwarded through the merchant plug-in and back to the customer's browser via a Payer Authentication Request (PAReq).

2. **Payment Authentication:** The customer authentication mechanism in this setting can be a straightforward signature of a payment record with an ACS-supplied random challenge incorporated for freshness, Step 8 of Figure 5. This would occur as follows:

- A customer, upon receiving the payment record from the merchant, instigates an exclusive transport session with their TPM. This session is designed to create a sequence of attestable instructions that can be verified by a third party as occurring within a particular session.
- After the session is established the customer activates their private key, K_{i-priv} (corresponding to the public key for which they obtained certification in the enrollment procedure), by sending their authorisation data to the TPM.
- K_{i-priv} is then used to sign the payment record, along with an ACS-supplied nonce.
- The customer demonstrates physical presence on their platform by performing a secure attention sequence.
- The customer instructs their platform to tear down the transport session.
- The customer instructs their platform to perform an attestation of the transport session incorporating the nonce sent from the ACS.
- The signed payment record, along with the X.509 certificate corresponding to K_{i-priv} that signed the payment record and an attestation of the transport session, is forwarded to the ACS.
- The ACS validates both the certificate, the presented signature on the payment record, and the attestation of the transport session.

This approach allows the ACS server to be sure that a valid customer is proffering their valid account details as the customer's PAN forms part of the X.509 certificate. In this instance, the CSC, which is not included in the certificate, could effectively act as a PIN in further establishing a binding between a card its cardholder.

3. **Payment Validation:** Payment validation is a result of an examination, by the merchant plug-in, of the Payment Response message (PARes) generated by the ACS server. If everything is as it should be, that is, the ACS signature validates correctly, then the merchant server can be assured that he is dealing with the valid owner of the presented payment card.

By using a 3-D secure authentication procedure that is augmented by Trusted Computing we can achieve the benefits of an unconnected card reading facility without the need for additional client-side security tokens, under

the assumption of TPM ubiquity. As we saw in Section 4.2, the demonstration of physical presence (see Step 2 – payment authentication) combats the threat posed by malware by requiring the customer to perform a physical action as part of the payment authentication process.

The primary advantages of this approach over an unconnected card reader based approach are its lower cost and its capability to support more flexible deployment. An unconnected card reader, once deployed, is a static device that cannot be updated without incurring the costs of reprovisioning every device. Using Trusted Computing allows a much finer-grained control over the life-cycle process where the security afforded to a CNP transaction can take advantage of additional Trusted Building Blocks as and when they become ubiquitous in the marketplace.

5.3 SET Reinvigoration - Server-Side Wallets

The inclusion of Trusted Computing into the electronic payment world could potentially lend itself to a reinvigoration of SET-like processing using server-side wallets. Subsequent to the initialisation phase (see Section 2.2) in which the customer agrees upon the purchase of certain items there is payment request hand-off to a SET wallet. This hand-off is very much like the hand-off performed in 3-D secure in which the merchant plug-in component transfers control to the ACS for payment authentication.

In server-side SET processing we could use functionality provided by the TNC specifications (see Section 3.2). Here the customer's platform joins a network controlled by its card issuer and goes through a process of assessment, isolation and remediation in order to gain access to the wallet in order to progress with a transaction. We sketch this process next

1. **Assessment Phase** The assessment phase deals primarily with determining if a particular customer (AR) should gain access to its card issuer's wallet network through a process of assaying customer endpoint integrity for compliance with predefined integrity policies. In this phase the IMV on a PDP examines integrity metrics coming from the customer's IMC reconciled against its network access policies. The IMC in this instance would be a a card-issuer-supplied down-loadable application that would monitor executing processes on a customer's platform. The PDP informs a PEP of its decision pertaining to an AR's access request after comparing the customer's supplied IMC metrics against its security policy. In this setting the AR would need to authenticate themselves to the PEP using some form of authentication

protocol, for example Radius with EAP. Using this protocol a customer would communicate authentication information (a signature of a PDP supplied challenge using the certified key private key K_{i-priv} from the enrollment phase) in conjunction with its IMC-collated integrity metrics. Once authenticated the user would be free to access their server-side wallet.

There is, however, one caveat in this approach: the issue of partition to TPM binding as laid out in the Trusted Server specifications [18]. A single TPM can only be bound to a single partition at any given time. If the card issuer wished to provide Trusted Computing facilities to their customers, servers may require trusted OS functionality to ensure proper process isolation between concurrently running wallet applications within a single partition.

2. **Isolation Phase** In the event that the AR has been authenticated but failed the IMV's integrity-verification procedure (possibly as a result of the intrusion of some undesirable third party as evidenced in the IMC reported metrics), a process of isolation may be instigated whereby the PDP passes instructions to the PEP which are then passed to the AR directing it to an isolation network. The customer can then be instructed in the removal of any detected malware.

3. **Remediation Phase** The remediation phase is representative of a successful completion of PEP instructions by the ARs platform where the AR on the isolation network obtains the requisite integrity-related updates that will allow it to satisfy the PDP's access policies, after which the customer may gain access to their wallet application. The wallet application in this respect can act very much like SET with chip-card support as described in [8].

The primary advantage of this approach is that the only modification to a customer's platform is the download of a small IMC application. This allows a customer to be authenticated and given access to their wallet account in a controlled environment. In this setting the card-issuer-controlled network would have little trouble in blocking access to payment cards as the wallet applications would reside on their Trusted Servers. This allows a card issuer to instigate their own policies and procedures for managing risk in a CNP setting. The applicability of this approach is further enhanced by recent moves by large industry players such as Barclays bank in provisioning anti-viral software licences to 1.6 million of their on-line banking customers [23]. These anti-viral checks could be performed during the assessment phase of

a TNC connection in order to assay customer end-point integrity. The main disadvantage of this approach is that it would require additional investment on the part of a card issuer in providing TNC connectivity to its customers. However, under the assumption that the card issuer is a bank that also offers on-line banking services to its customers, a lot of the investment in a TNC infrastructure could be reused in providing a secure on-line banking facility.

6 Comparison with Related Work

The issue of client certification with respect to TPM-enabled platform has previously been examined in [10, 1, 6] with Alsaid and Mitchell's approach most closely resembling the one adopted here. However, the primary threat in their model is external attack whereby a credential needs to be extracted from the client's platform in order for it to have any value to an attacker. No consideration is given to the ever increasing threat posed by malware. Also, the SKAE extension, contrary to what the authors of [1] suggest, does not require a general reworking of server SSL/TLS implementations in excess of the reworking required for enabling generic client-side certification. SKAE, as discussed previously, is an extension to X.509 and so SKAE certificates and keys can be used directly in a generic SSL/TLS implementation. Modification to the server is only required in the event that the server wishes to validate the binding between an AIK and the key referenced in the SKAE certificate. Indeed, this need for an additional validation can largely be mitigated by the certificate policy and certification practices documents provided by a particular CA [7]. In this sense, binding of a non-migratable to a TPM can be an implicit adjunct to the enrollment procedure. The fact that a particular cardholder has a valid certificate for a TPM resident non-migratable key can be evidenced by the possession of certificate issued by a card issuer. It is assumed that this card issuer performs the required checks and is responsible for validating the binding.

7 Conclusions and Future Work

The use of the payment cards as an avenue for e-commerce is increasing at an unprecedented rate. In the physical world, the introduction of EMV for card-based payments at point of sale terminals has seen a dramatic reduction in the level of chargeback-related fraud. This is primarily due to the widespread tamper-resistent cryptographic hardware being deployed, preventing the cloning of cards.

Unfortunately, the benefits seen in the physical deployment of EMV for card payment transactions cannot be so easily gained in CNP scenarios. In this setting knowledge of customer account information is all that is required to perform a transaction. This makes it impossible for a merchant or a customer's card issuer to determine if a valid owner of the account details being proffered is the one that actually instigated the transaction.

This paper has attempted to address this imbalance by analysing the role Trusted Computing can play in augmenting three different mechanisms for securing CNP transaction details. We showed how integrating Trusted Computing physical presence signals with 3-D Secure, SET and SSL can thwart the threat posed by malware. In doing so we highlighted how 3-D Secure and SET-based solutions are more amenable to the inclusion of physical presence signals as the merchant plug-in and server-side logic are supplied by the financial network domain and thus can be programmed to verify customer supplied attestations of physical presence. With SSL, much greater heterogeneity of implementations of server logic are possible as it is the merchant and not the financial network domain that decides on the actual implementation. By tieing payment authorisations to Trusted Computing hardware, in the form of a TPM, we provide similar benefits to that of EMV. That is to say, knowledge of a customer's account details is no longer sufficient to complete a transaction. A customer would need to demonstrate possession of a private key which is physically bound to a piece of hardware under their direct control.

As part of on-going work we are looking at new security architectures for securing CNP transactions. In particular, we are examining the role new hardware that provides hardware-based virtulisation (in the form of Intel's La Grande and AMD's Pacifica) can be used to create software-based EMV cards.

References

[1] A. Alsaid and C. J. Mitchell. Preventing phishing attacks using trusted computing technology. In *INC 2006: Sixth International Network Conference*, July 2006.

[2] APACS. Card fraud the facts 2006. `http://www.apacs.org.uk/resources_publications/documents/FraudtheFacts2006.pdf`, April 2006.

[3] Visa International Service Association. 3-D Secure™ Protocol Specification: Core Functions. `http://international.visa.com/fb/paytech/secure/main.jsp`, July 2002.

[4] Visa International Service Association. 3-D Secure™ Protocol Specification: System Overview. `http://international.visa.com/fb/paytech/secure/main.jsp`, May 2003.

[5] B. Balacheff, D. Chan, L. Chen, S. Pearson, and G. Proudler. Securing intelligent adjuncts using trusted computing platform technology. In *IFIP TC8/WG 8.8 4th Working Conference on Smart Card Research and Advanced Applications*, IFIP TC8/WG 8.8, pages 177–195, 2000.

[6] S. Balfe, A.D. Lakhani, and K.G. Paterson. Securing peer-to-peer networks using trusted computing. In C.J. Mitchell, editor, *Trusted Computing*, pages 271–298. IEE Press, 2005.

[7] S. Chokhani and W. Ford. RFC 2527 - Internet X.509 public key infrastructure certificate policy and certification practices framework, March 1999.

[8] EMVCo. *Book 3 - Application Specification*, 4.0 edition, December 2000.

[9] Trusted Computing Group. Trusted computing: Opportunities and challenges. `https://www.trustedcomputinggroup.org/downloads/tcgpresentations/`, 2004.

[10] Trusted Computing Group. *TCG Infrastructure Workgroup Subject Key Attestation Evidence Extension*, 1.0 edition, June 2005.

[11] Trusted Computing Group. *TCG Infrastructure Working Group Reference Architecture for Interoperability (Part I)*, 1.0 revision 1 edition, 2005.

[12] Trusted Computing Group. *TCG Trusted Network Connect TNC Architecture for Interoperability*, 1.0 revision 4 edition, 2005.

[13] Trusted Computing Group. *TCG Trusted Network Connect TNC IF-IMC*, 1.0 revision 3 edition, 2005.

[14] Trusted Computing Group. *TCG Trusted Network Connect TNC IF-IMV*, 1.0 revision 3 edition, 2005.

[15] Trusted Computing Group. *TPM Main: Part 1 Design Principles*, 1.2 revision 85 edition, 2005.

[16] Trusted Computing Group. *TPM Main: Part 2 Structures of the TPM*, 1.2 revision 85 edition, 2005.

[17] Trusted Computing Group. *TPM Main: Part 3 Commands*, 1.2 revision 85 edition, 2005.

[18] Trusted Computing Group. *TCG Generic Server Specification*, 2005 Revision 0.8.

[19] Trusted Computing Group. *TCG Software Stack Specificiation Version 1.2 Level 1*, 2006.

[20] MasterCard International. SecureCode™ Merchant Implementation Guide. `http://www.mastercardmerchant.com/securecode/`, March 2004.

[21] B. Krebs. Citibank phish spoofs 2-factor authentication. `http://blog.washingtonpost.com/securityfix/2006/07/citibank\_phish\_spoofs\_2factor\_1.html`, July 2006.

[22] P. Meadowcroft. Combating card fraud. `http://www.scmagazine.com/uk/news/article/459478/combating+card+fraud/`, January 2005.

[23] BBC News. Barclays banks on anti-virus deal. `http://news.bbc.co.uk/2/hi/technology/5019856.stm`, May 2006.

[24] D. O'Mahony, M. Peirce, and H. Tewari. *Electronic Payment Systems for E-Commerce 2nd edition.* Artech House, 2001.

[25] IBM Global Services. IBM Global Business Security Index Report, February 2005.

[26] A. Spalka, A.B. Cremers, and H. Langweg. Protecting the creation of digital signatures with trusted computing platform technology against attacks by trojan horse programs. In *Proceedings of the IFIP SEC 2001*, pages 403–420, 2001.

[27] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. RFC2904 – AAA Authorization Framework, 2000.

[28] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. RFC2905 – AAA Authorization Application Examples, 2000.

[29] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. RFC2906 – AAA Authorization Requirements, 2000.

[30] K. Zetter. Cardsystems' data left unsecured. `http://www.wired.com/news/technology/0,1282,67980,00.html`, 2004.