Using EMV cards for Single Sign-On

Andreas Pashalidis and Chris J. Mitchell

Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, United Kingdom,
{A.Pashalidis, C.Mitchell}@rhul.ac.uk,
WWW home page: http://www.isg.rhul.ac.uk

Abstract. At present, network users have to manage a set of authentication credentials (usually a username/password pair) for every service with which they are registered. Single Sign-On (SSO) has been proposed as a solution to the usability, security and management implications of this situation. Under SSO, users authenticate themselves only once to an entity termed the 'Authentication Service Provider' (ASP) and are subsequently logged into disparate network Service Providers (SPs) without necessarily having to re-authenticate. The information about the user's authentication status is handled between the ASP and the desired SP in a manner transparent to the user. In this paper we propose an SSO scheme where user authentication is based on payment cards conforming to the EMV industry standard. The card itself, in conjunction with the EMV architecture, takes the role of the ASP. The associated SSO protocol does not require online card issuer participation, preserves user mobility and does not put user's financial data at risk.

Keywords: single sign-on, EMV, authentication

1 Introduction

Network users typically have to manage a set of authentication credentials (usually a username/password pair) for every Service Provider[1] (SP) with which they are registered. The number of such SPs with which a typical user interacts has grown beyond the point at which most users can memorise the required credentials. The most common solution is for users to use the same password with every SP — a tradeoff between security and usability in favour of the latter.

Single Sign-On (SSO) has been proposed as a solution to the usability, security and management implications of this situation. It is a technique whereby users authenticate themselves only once to an entity called

[1] In the context of this paper a service provider is any entity that provides some kind of service or content to a user. Examples of SPs include messenger services, FTP sites, web sites and streaming media providers.

an *Authentication Service Provider* (ASP) and are logged into the SPs they subsequently use, without necessarily having to re-authenticate. This seamless experience increases the usability of the network as a whole but introduces a number of security requirements. It is obvious that, under SSO, SPs require some kind of notification from the ASP about the user's authentication status. These notifications are termed *authentication assertions*. The SP, based on the authentication assertions provided by the ASP, determines whether or not to grant access to a protected resource to the specified user.

EMVCo[2], an organisation formed by Mastercard[3] and Visa[4], has developed a set of Integrated Circuit Card (ICC) specifications for Payment Systems [1–4] that focus on the card/terminal interactions that take place at the Point of Sale between a cardholder and a merchant during a financial transaction.

In this paper we present a scheme in which EMV-compliant cards provide user (i.e. cardholder) authentication, and propose an associated protocol that facilitates SSO at disparate network SPs. In this scheme the cardholder's network access device itself, in conjunction with the card, acts as the ASP. The scheme can be regarded as an alternative to other smartcard-based authentication schemes, for example schemes that rely on Subscriber Identity Module cards (used in cellular telephony). The paper is organised as follows. The next section is a review of relevant EMV architectural components and security services. Section 3 describes the proposed authentication method and SSO protocol, while section 4 analyses the associated security threats. Section 5 discusses advantages and disadvantages, and sections 6 and 7 give an overview of related work and conclude the paper.

2 Review of EMV security services

This section introduces those components of the EMV specification that are relevant to this paper. For a full description see [1–5].

In the EMV payment system there are four major interacting entities, namely the cardholder, the merchant, an acquiring bank (the Acquirer) and the card issuing bank (the Issuer). The specifications focus on the interactions between card and merchant terminal. When the card is inserted into the terminal, the steps that occur include the following.

[2] http://www.emvco.org
[3] http://www.mastercard.com
[4] http://www.visa.com

1. The terminal selects the appropriate EMV application by issuing a SELECT command [1, p.65] to the card.
2. The terminal initiates 'Application Processing' by issuing a GET PROCESSING OPTIONS [3, p.19] and a number of READ RECORD [3, p.23] commands. The purpose of this step is for the card and the terminal to exchange the necessary data for the rest of the transaction.
3. The terminal performs 'Processing Restrictions' [3, p.48]. This mandatory step does not involve communication with the card — its sole purpose is to provide assurance of application compatibility between terminal and card.
4. The terminal issues an INTERNAL AUTHENTICATE [3, p.21] command to the card. This optional step initiates 'Offline Data Authentication', which can be either Static Data Authentication [2, p.15] or Dynamic Data Authentication [2, p.24]. The purpose of this step is to verify the card's authenticity.
5. The terminal performs 'Cardholder Verification' [3, p.50]. During this optional step the cardholder's Personal Identification Number (PIN) is verified, either offline to the card (using the VERIFY command [3, p.25]), or online to the Issuer.

We next focus on the Dynamic Data Authentication (step 4) and the Cardholder Verification (step 5) that are defined in the EMV specifications. These steps are of particular interest since, in the authentication/SSO scheme proposed below, the card reader is under the control of the cardholder.

2.1 Dynamic Data Authentication (DDA)

DDA is supported by a Public Key Infrastructure (PKI), as specified in [2]. In particular, every DDA-capable card has its own asymmetric key pair which is generated and loaded on the card by the Issuer. While the private key cannot be extracted from the card, its public counterpart can be retrieved with a READ RECORD command. This public key is embedded in a public key certificate which is signed by the Issuer. The Issuer's public key certificate, signed by the Payment System's top-level Certification Authority (CA), is also stored in the card and can be retrieved by the terminal. As a result, the merchant terminal only needs to maintain an accurate copy of the CA's trusted root public key in order to verify the Issuer's, and hence the card's, public key certificates, and finally any data signed by the card itself. CA public key management principles and policies are defined in [2].

A simplified description of Dynamic Data Authentication (DDA) is given below:

1. The terminal retrieves the Issuer and card public key certificates from the card. The latter is verified using the former and the former is verified using the appropriate trusted root public key of the Payment System's top level CA.
2. The terminal issues an INTERNAL AUTHENTICATE command to the card. The command requires a number of parameters, including a nonce.
3. The card computes a digital signature over the terminal-provided data (including the nonce) and 'card Dynamic Data', which is data generated by and/or stored in the card [2, p.35]. The card outputs the signature and the card Dynamic Data.
4. The terminal reconstructs the signed data structure and verifies the signature using the card's public key retrieved and verified in step 1.

DDA provides data integrity and freshness. Assuming tamper resistance of the card and the soundness of the Issuer's security procedures, DDA also provides card authentication and card counterfeiting prevention. It should be noted that not all EMV-compatible cards are DDA-capable.

2.2 Cardholder verification

The identity of the cardholder is verified based on a PIN. The PIN is entered into the terminal and may then either be verified online to the Issuer or offline to the card. In the latter case the terminal issues a VERIFY command to the card which takes the PIN as a parameter. It may or may not be encrypted using the card's public key. The card checks whether the supplied PIN matches the one stored inside the card and responds accordingly. If the number of unsuccessful offline PIN verification attempts exceeds a certain limit, the PIN on the card is blocked and can only be unblocked using a script sent to the card by the Issuer.

3 Using EMV cards for SSO

This section describes the proposed EMV-based SSO scheme.

3.1 System entities

The entities involved in the authentication/SSO scheme are the cardholder system, the card itself, and the SPs.

As briefly mentioned above, the system requires the cardholder system and the card to collectively act as the ASP. Instead of directly authenticating to every SP, the cardholder is authenticated by the card, and the card then vouches for the identity of the cardholder to every SP. The fact that the CS would typically consist of a 'standard' PC, PDA or mobile phone equipped only with a special SSO application and an EMV card means that this offers an inherently mobile SSO solution; the SSO application could be downloaded from a trusted source when required, and the EMV card is inherently mobile.

The cardholder system The Cardholder System (CS) consists of the user's (i.e. the cardholder's) network access device and a card reader. A typical configuration would be a PC with an integrated card reader. Whether or not the card reader is equipped with its own (trusted) keypad is optional (see section 4.4). Alternatively, the CS could be a wireless network access device (such as a Personal Digital Assistant (PDA) or a 3GPP[5] mobile phone) capable of communicating with EMV cards. The CS also needs some special software that implements the SSO protocol described in section 3.3 below. This 'SSO agent' might be realised as a process that continually runs on the CS (also known as 'service' or 'dæmon'), or as part of the software that is used to access the SP (e.g. the web browser, instant messenger, e-mail client, etc.). In this latter context the SSO agent could be uploaded to the CS as an applet running within the SP access software, e.g. as a Java applet running within the web browser, or a Java MIDlet that is delivered over-the-air to a mobile phone. The SSO agent is likely to be provided by an EMV card issuer or a trusted third party.

The card The proposed EMV-based SSO scheme imposes certain requirements on the cards, as follows. Cards must be DDA-capable. Unfortunately, from the SSO perspective, the public key certificate used during DDA binds the card's public key to the cardholder's Primary Account Number [2, p.33]. It would constitute a potentially serious security and privacy threat if the Primary Account Number was to be used in an open environment such as the Internet. Therefore, the cards used in the

[5] http://www.3gpp.org

scheme described here need to possess a separate, dedicated EMV application, which we call the *card authentication application* (AA). In the AA, the Primary Account Number (and any other personally identifying information) must be replaced with data that is not linked to the cardholder's identity (and cannot be used for financial transactions). This implies that the Issuer has to provide the AA with an additional certificate for the card's public key. It is required that this certificate does not contain any personally identifying information for the cardholder; thus we call it an 'anonymous certificate' in the sequel. As its serial number can be used for user identification at SPs, it should not contain any other information about the user (e.g. a name). Furthermore, the AA should be able to maintain state within the current session. In particular, it should be able to maintain a data element that indicates whether or not offline PIN verification (via the VERIFY command) has been performed during the current session, and, if so, the data element should also indicate whether or not PIN verification was successful. This card-provided *PIN Verification Data Element* (PVDE) shall be included in the data that is signed by the card during DDA, as part of the card Dynamic Data.

It should be noted here that a card session begins with Application Selection (step 1 in section 2) and ends when the card reading device deactivates the card [1, p.17]. This latter event includes premature removal of the card from the reader.

Service providers In the proposed SSO scheme, SPs are required to accurately obtain and store the root keys of the CAs of the EMV Payment Systems that are to be supported (and trusted). This requirement is exactly the same as that applying to merchant terminals for 'standard' use of EMV cards. The management, distribution and revocation of these root keys is outside of the scope of this paper, but the principles are similar to those specified in [2] for merchant terminals. It is assumed that SPs require a user to be authenticated before granting access to protected resources. Instead of executing an authentication protocol directly with the user, SPs acquire the necessary authentication assertions from the CS, according to the protocol described in section 3.3 below. Moreover, as users also need to authenticate SPs, it is necessary that every SP possesses a *unique, human-readable identifier* (SPID).

3.2 Trust relationships

The SSO scheme depends on the EMV cards offering a level of tamper-resistance, since these cards act as a trusted computing module within

the CS. In addition, cardholders need to trust that SPs will not collude in order to compromise their privacy (see section 4.1). Cardholders and SPs also need to trust that

- the Payment System's top-level CA(s) will not impersonate cardholders,
- card Issuers will not impersonate cardholders.

From the cardholder perspective, authentication/SSO can be achieved with those SPs that choose to trust the Payment System top-level CA corresponding to the cardholder's card. From the SP perspective, authentication/SSO can be facilitated only for those cardholders whose Payment System top level CA a given SP has chosen to trust. The architecture does not provide for explicit trust management at the Issuer level. This feature is inherited from the EMV PKI, which does not allow merchant terminals *not* to trust individual Issuers that have been certified by a trusted CA. This arises from the fact that EMV was designed for use within a closed environment in which all parties (Issuers and Acquirers) have signed an agreement with the brand. Indeed, even in the non-electronic world, merchants are typically required to accept all cards bearing the brand as part of the condition of being an approved merchant.

3.3 The SSO protocol

This section describes the protocol of the SSO scheme. The protocol starts when the user requests a protected resource from the SP. As we see below, the same protocol can also be used for initial user registration at an SP.

The protocol assumes that the SP has already been authenticated to the user (cardholder) by some mechanism outside the scope of this paper. Specifically, it is assumed that, as part of that mechanism, the user manually verifies the SP's unique identifier and, ideally, a cryptographically protected session is set up between the SP and the CS. A suitable mechanism for SP authentication is, for example, an SSL/TLS channel with server-side certificates[6] [6]. In this case the SP's unique identifier, the SPID, would be a field in its SSL/TLS certificate (typically its unique URL). Here it is worth noting that, if SSL/TLS is used for SP-to-user authentication, our scheme essentially uses the commercial PKI established on the Internet for SSL/TLS connections to facilitate SP-to-user authentication, and the EMV PKI established for credit/debit card payments to

[6] Since the user requests a protected resource, it is likely that an SSL/TLS connection will be required anyway.

facilitate user-to-SP authentication[7]. A detailed description of the SSO protocol follows.

1. The SP sends an authentication request message to the CS. This message contains a freshly generated nonce and an indication saying whether or not PIN verification is required.
2. The CS selects the card's AA, initiates application processing and performs processing restrictions, as explained in steps 1-3 in section 2. If this step fails, SSO also fails.
3. If PIN verification was required in step 1, the CS performs offline PIN verification with the card, as explained in section 2.2.
4. The CS performs DDA with the card. The main difference from the 'standard' DDA (as explained in section 2.1) is that the nonce used with the INTERNAL AUTHENTICATE command is the SP-provided nonce from step 1. The SP's Identifier (acquired during SP-user authentication, as explained above) is also included in the data passed to the card for signing. It should be noted that the CS cannot verify the card's and the Issuer's public key certificates as it does not have the root CA's public key.
5. The CS sends an authentication assertion message back to the SP. This message includes the following data structures obtained from step 4.
   - The card's anonymous certificate.
   - The card Issuer's public key certificate.
   - The card's signature produced as part of DDA. This signature covers the nonce of step 1, the SPID and the PVDE, as explained in section 3.1.
   - Any other data that is input to the card signature calculation.
6. The SP verifies the Issuer and card public key certificates, as explained in section 2.1. The SP also makes sure that the card has not been blacklisted and that the aforementioned certificates have not been revoked. If this step fails, SSO also fails.
7. The SP reconstructs the data structure that was signed by the card in step 5 and verifies the signature using the card's public key. If verification is unsuccessful, SSO fails.
8. The SP assesses the data used to compute the card's signature. In particular, the SP checks the SPID and makes sure that it indeed

[7] From this perspective, the paper could be re-titled 'Integrating EMV certificates and SSL'. However, SP-to-user authentication schemes other than SSL could also be used.

represents this SP (and not any other). Furthermore, the SP assesses the PVDE. If the SP's requirements are met, SSO succeeds and access to the protected resource is granted. Otherwise, SSO fails.

The CS's response (step 5) does not contain any personally identifying information about the cardholder. The SP may, however, differentiate between users based on the unique (Serial Number, Issuer Identifier) pair included in the card's anonymous certificate. Furthermore, the protocol can be used for initial registration of a user at a SP; the SP creates a new user account for a newly encountered anonymous certificate.

The CS can achieve SSO at disparate SPs by running the protocol whenever needed. Of course, the card needs to be in the card reader of the CS during the protocol run. If PIN verification has been performed, the card needs to remain in the reader between protocol runs so that the session state is maintained within the card.

4 Threat Analysis

In this section threats to the scheme are evaluated.

4.1 SP collusion

If a number of SPs collude, they can trivially compromise user privacy by correlating the unique identifying (Serial Number, Issuer Identifier) pairs found in the card anonymous certificates. The scheme does not address this threat.

However, as also pointed out in [7], complete prevention of a 'SP collusion' attack is difficult as SPs may also be able to correlate users based on other profile information they may maintain (such as names or telephone numbers). As stated in the Liberty specifications [7, p.71], 'The only protection is for Principals [users] to be cautious when they choose service providers and understand their privacy policies'.

4.2 Reflection attack

An attacker could forward the authentication request message (step 1 of the protocol) received from an SP as part of the SSO process to a victim user, while masquerading as the SP to that user (maybe by spoofing the SP's interface and SPID). Forwarding the user's valid response (step 5) to the SP might result in successful impersonation.

This attack is prevented as long as the SP authentication method is secure, it results in a cryptographically protected session and it is correctly performed by the user. In the case of SSL/TLS this involves the user inspecting the SP's URL and making sure that it indeed represents the desired SP.

As the CS's authentication response contains the SP's unique identifier, which is digitally signed by the card, intermediaries (such as an attacker) cannot change it without being detected. At the same time the SP is given assurance that the response is indeed meant for this particular SP (and not any other).

It should be noted that the attack is *not* prevented if launched by a dishonest SP. As explained in section 4.1, users should be cautious when they choose SPs.

4.3 Traffic analysis

An attacker capable of monitoring network traffic between the CS and SPs could compromise the user's privacy in that the attacker will learn which SPs the user is communicating with. The attack cannot be prevented by encrypting traffic (using SSL/TLS, for example), as packet headers typically need to be available in the clear for routing purposes. This threat is outside the scope of the scheme described here, but could be addressed separately using anonymising techniques, such as those described by Chaum [8].

4.4 Attacks using a malicious Cardholder System

The EMV specifications make no provision for cards to authenticate merchant terminals prior to releasing information. As a result, when the card is inserted into the CS, it may be possible for malicious software in the CS to extract private information (such as the cardholder's Primary Account Number which is likely to be stored in the card) and to disclose it to unintended parties. Similarly, if the card reader does not have its own (trusted) PIN pad, the CS could collect cardholders' AA PINs[8]. Furthermore, a malicious CS could spoof local and remote user interfaces and abuse the user's authentication status at SPs by modifying traffic or hijacking the entire session, even if communications are 'cryptographically protected'. Thus, the SSO agent has to be trusted by the user not to

[8] The PIN used by the AA should be separate from the PIN(s) used by EMV applications that may coexist on the card.

engage in such behaviour and its integrity has to be protected. Having it signed by a party trusted by the user (e.g. the card Issuer) might address the threat, but risks remain if other malicious software is executed on the CS.

However, despite these threats, the scheme provides two-factor user authentication (proof-of-possession of the card and, if required, proof-of-knowledge of the PIN), even in the presence of a malicious CS: firstly, it requires the EMV card to be present in the CS; without it user authentication (and therefore illegitimate impersonation) will fail. Secondly, the scheme protects against the CS falsely pretending that PIN verification took place; the PIN verification status maintenance is managed by the trusted card itself, as explained in section 3.1. This protects against PIN compromise by a malicious CS as the PIN never needs to be inserted into the device (for SPs that do not require PIN verification).

4.5 Stolen EMV card

Stolen EMV cards allow attackers to impersonate users to SPs that do not require PIN verification. The obvious countermeasure is for SPs to require PIN verification. In this case the attacker will not be able to impersonate the legitimate cardholder, even by using a maliciously modified CS. Of course, if the attacker also has access to the user's PIN, then impersonation will be successful.

In order to guard against 'stolen card' attacks, SPs should follow the same procedures as merchant terminals. In particular they should periodically contact card Issuers and/or Payment System CAs to obtain Certificate Revocation Lists (CRLs) and/or blacklisted card information. Step 6 in of the SSO protocol (section 3.3) provides for checking of these CRLs and blacklisted card information.

4.6 Service denial attacks

The scheme requires the SPs to check whether the signature returned by the CS is computed using the correct nonce, i.e. it requires the SP to maintain state while waiting for the response from the CS. This potentially opens the door to service denial attacks. However, we have assumed prior SP-to-user authentication, which ideally results in a secure session. This means that, before the SSO protocol is executed, the SP has established that it is talking to an existing client for whom it has already created some state. The fact that the SP is required to remember a nonce for

each user-to-SP authentication attempt, is thus not likely to significantly increase the SP's exposure to service denial attacks.

4.7 Signature oracle attacks

The SSO scheme, as described in sections 3.1 and 3.3 (step 5), involves the card signing a data string containing a nonce supplied by the SP. Thus the protocol involves the card signing a message, part of which is provided by an external party. There exists the possibility that this could be used as part of an 'oracle attack', where the card is persuaded to sign a string that could be used in another application using the same key pair. The reason why this is not a significant threat in the case of the EMV payment application is that the signed string is different in format from the one expected by an EMV application.

5 Advantages and Disadvantages

This section discusses the advantages and disadvantages of the described SSO scheme.

5.1 Advantages

Advantages of the authentication/SSO scheme described in this paper include the following.

- The scheme reuses the existing EMV PKI which is already established on a world wide basis.
- The scheme does not require a continuous online presence of the card Issuer.
- Once the authentication/SSO protocol has completed successfully, subsequent protocol runs do not necessarily require user intervention. This yields transparent user authentication at subsequently used SPs.
- As user authentication may be transparent, the protocol can be repeated whenever appropriate. An online banking SP, for example, may wish to ensure that the cardholder's card is still present in the CS whenever access to a sensitive resource is requested. Rerunning the SSO protocol during a session increases the achieved level of security without usability implications.
- No identifying information about the user is included in the messages exchanged. This protects the user's anonymity and privacy. Furthermore, no risks of personal information exposure arise at the SP.

- Maliciously acting devices can only compromise the user's current session or impersonate users while the EMV card is present. Furthermore, they cannot falsely pretend that PIN verification took place successfully.
- The scheme does not necessarily require an online third party. SPs need, however, to follow the same principles and policies as merchant terminals with respect to the certificates used.
- The scheme preserves user mobility.
- The scheme can potentially be adapted as a new Liberty Alliance [7] profile.

5.2 Disadvantages

Disadvantages of the authentication/SSO scheme described in this paper include the following.

- Issuers must install a separate EMV application on the card in order to support user authentication. This is a potentially significant cost. This cost is minimised if the AA is installed on the card at the time of issue, and it has to be weighed against the potential benefits gained by the Issuer. These might include new revenue streams from SPs that benefit from the AA.
- The cards used must be DDA-capable. The cost of DDA-capable cards is higher than the cost of cards not capable of DDA.
- It obviously works only for EMV cardholders equipped with card readers. The cost of the card reader (and maintaining the SSO agent), has to be weighed against the convenience offered by SSO.

6 Related Work

Single sign-on architectures within enterprise environments are examined in [9]. Currently deployed or proposed SSO schemes for open environments are based on a continually online ASP [10–12]. The scheme proposed in this paper, on the other hand, does not necessarily require the continuous online presence of any party; it falls into the category of *local* true SSO schemes [13]. Being based on a different trust model, it also constitutes an interesting alternative to the aforementioned schemes.

Other related work includes [14], where a security-enhanced e-commerce transaction scheme based on EMV cards is proposed. The scheme makes use of DDA and offline PIN verification in order to facilitate card and

cardholder authentication respectively. Being a payment scheme, however, it requires online presence of the Issuer and does not aim for user privacy protection.

An annex of [3] describes how to combine the Secure Electronic Transaction (SET) protocol[9] with EMV-compliant cards for electronic transactions conducted over the Internet. The complexity of the scheme is quite high. In addition, it requires the online presence of a 'Payment Gateway' which is connected to the Acquirer's (and Issuer's) legacy network.

Finally, it is worth noting that Subscriber Identity Module (SIM) cards of mobile phones have recently been augmented with EMV-compliant applications (`http://www.oberthurcs.com`) and that mobile equipment with EMV-compliant card readers has been available for some time.

7 Conclusion

In this paper we have proposed a SSO scheme which relies on EMV-compliant cards for cardholder authentication at SPs. These cards need to be able to perform asymmetric cryptographic functions (i.e. DDA) and must have a separate EMV 'Authentication Application' installed on them by Issuers.

The CS itself acts as ASP for relying SPs. The scheme does not require online participation of the Issuer and its security does not depend on CS integrity, as core functions are delegated to the trusted card. It leverages the existing and established EMV PKI and preserves user mobility and privacy, and can be regarded as an alternative to other smartcard-based user authentication mechanisms (such as Subscriber Identity Modules).

The associated SSO protocol only requires minimal interaction, yielding a potentially seamless user experience and allowing several transparent re-authentications to occur within a given user/SP session.

References

1. EMV. *EMV2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 — Book 1: Application Independent ICC to Terminal Interface Requirements*, December 2000.
2. EMV. *EMV2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 — Book 2: Security and Key Management*, December 2000.
3. EMV. *EMV2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 — Book 3: Application Specification*, December 2000.

[9] http://www.setco.org

4. EMV. *EMV2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 — Book 4: Cardholder, Attendant and Acquirer Interface Requirements*, December 2000.
5. Cristian Radu. *Implementing Electronic Card Payment Systems*. Computer Security Series. Artech House, Norwood, 2002.
6. Eric Rescorla. *SSL and TLS*. Addison-Wesley, Reading, Massachusetts, 2001.
7. Liberty Alliance. *Liberty ID-FF Bindings and Profiles Specification version 1.2-08*, April 2003.
8. David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, February 1981.
9. Jan De Clercq. Single sign-on architectures. In George I. Davida, Yair Frankel, and Owen Rees, editors, *Proceedings of the Infrastructure Security Conference (InfraSec 2002)*, volume 2437 of *Lecture Notes in Computer Science*, pages 40–58. Springer-Verlag, Berlin, 2002.
10. Liberty Alliance. *Liberty Architecture Overview*, November 2002.
11. OASIS, http://www.oasis-open.org/committees/security/. *Security Services Technical Commitee Homepage*.
12. Andreas Pashalidis and Chris J. Mitchell. A taxonomy of single sign-on systems. In R. Safavi-Naini and J. Seberry, editors, *Information Security and Privacy, 8th Australasian Conference, ACISP 2003, Wollongong, Australia, July 9-11, 2003, Proceedings*, volume 2727 of *Lecture Notes in in Computer Science*, pages 249–264. Springer-Verlag, Berlin, July 2003.
13. Andreas Pashalidis and Chris J. Mitchell. Single sign-on using trusted platforms. In C. Boyd and W. Mao, editors, *Information Security, 6th International Conference, ISC 2003, Bristol, UK, October 2003, Proceedings*, volume 2851 of *Lecture Notes in in Computer Science*, pages 54–68. Springer-Verlag, Berlin, 2003.
14. V. Khu-smith and C.J. Mitchell. Using EMV cards to protect e-commerce transactions. In A. Min Tjoa K. Bauknecht and G. Quirchmayr, editors, *Proceedings of the 3rd International Conference on Electronic Commerce and Web Technologies (EC-Web 2002)*, volume 2455 of *Lecture Notes in Computer Science*, pages 388–399. Springer-Verlag, Berlin, 2002.