Formal Analysis of Card-based Payment Systems in Mobile devices

Vijayakrishnan Pasupathinathan Josef Pieprzyk Huaxiong Wang
Joo Yeon Cho

Centre for Advanced Computing - Algorithms and Cryptography
Division of Information and Communication Sciences,
Macquarie University, Sydney, Australia.
mail: {krishnan, josef, hwang, jcho} @ics.mq.edu.au

Abstract

To provide card holder authentication while they are conducting an electronic transaction using mobile devices, VISA and MasterCard independently proposed two electronic payment protocols: Visa 3D Secure and MasterCard Secure Code. The protocols use pre-registered passwords to provide card holder authentication and Secure Socket Layer/ Transport Layer Security (SSL/TLS) for data confidentiality over wired networks and Wireless Transport Layer Security (WTLS) between a wireless device and a Wireless Application Protocol (WAP) gateway. The paper presents our analysis of security properties in the proposed protocols using formal method tools: Casper and FDR2. We also highlight issues concerning payment security in the proposed protocols.

Keywords: Electronic Payments, Mobile Payment, Card-based Systems, Formal Verification.

1 Introduction

Card based payments is the most popular method of payment for purchasing of products on the Internet. But the lack of consumer confidence in security of electronic transactions has been a major issue preventing wider acceptance. Some of the consumer concerns in relation to Internet purchases are: unauthorised distribution of private information to third parties, theft of information kept by the merchants, transmission of credit card numbers in clear transmission of personal data and receiving unwanted emails[1].

But the main concern of merchants and service providers is authentication of their customers/card holders. Currently, card based payment systems do not provide adequate authentication of card holders, as it is always possible for an unscrupulous user to enter credit card numbers stolen from a valid card holder and make a purchase that will be charged to a valid user's account.

To provide better security Visa and MasterCard, the two largest credit card companies, independently proposed methods for authentication of card holders, Visa 3-D Secure (Verified by visa) (VISA 2002, VISA 2002, VISA 2003) and MasterCard Secure Code (MasterCard 2003).

To verify the security properties of the protocols, we propose a set of generic security goals applicable to electronic payment systems and analyse the proposed protocols using the formal method tools Casper (Gavin Lowe 1985) and FDR2 (Formal Systems (Europe) Ltd).

1.1 Organisation

This paper is organised as follows: Section 2 provide an overview of Visa 3D Secure and MasterCard Secure Code payment protocols. Section 3 presents an overview of model checking tools: Casper/FDR and the method used to analyse protocols using those tools. In Section 4 we present the results of our analysis and we conclude in Section 5.

2 Overview

MasterCard Secure Code and Visa 3-D Secure are authenticated payment methods that provide authentication of card holders during an electronic payment. The protocols depend on transport layer security (e.g., SSL,TLS,WTLS) to provide confidentiality and integrity for data transmission between entities. Operation of Visa 3-D Secure and MasterCard Secure Code are technically different but both models depend on passwords as a primary form of card holder authentication. The models also have the option of providing a second layer of authentication using an Subscriber Identity Module (SIM) toolkit, Europay-MasterCard-Visa (EMV) compliant *chip card*, voice authentication, and Public Key Infrastructure (PKI) based authentication using WAP Identity Module (WIM) (VISA 2003).

Before any purchase, both models require their card holders to register a password with their issuing banks. This phase is known as card holder enrolment (Refer 1). The card holders also need to select a Personal Assurance Message (PAM) which is displayed by the issuer every time the card holder is authenticated. The PAM provides an additional layer of security as they protect the card holder against any web spoofing attacks. When the card holder sees the messages he/she is assured that the password page displayed is from the issuer and is not a "shadow web page" by an attacker.

One of the major advantages of MasterCard Secure Code and Visa 3-D Secure over SET (MasterCard and VISA) is that card holders are not required to hold digital certificates, as authentication is carried out using passwords. The newly proposed models reduce complexity and provide security from card theft and card skimming attacks, as the malicious user not only has to obtain card details but also the pre-registered password to make an electronic purchase.

[1] Source: Ipsos Public Affairs, December 2003

2.1 Visa 3-D Secure vs MasterCard Secure Code

A summary of the distinctive features in the two payment systems are:

- Visa 3-D Secure uses a centralised structure with inclusion of Visa Directory Servers (DS). The directory servers are focal points for transfer of the card holder participating information between issuer's Access Control Server (ACS) and merchants. However, this adds to the complexity and increases the number of messages in the protocol. MasterCard's approach is to provide a distributed environment for its payment system. It introduces an "e-wallet" like system, a card holder applet available through the card holder's issuer. The applet resides on the card holders device and scan web pages for predefined "hidden fields"[2]. An inherent problem with MasterCard's implementation is that the applet download can be a hindrance for providing payment at multiple locations, as the card holder is required to download it every time when he/she uses a different terminal.
- In Visa 3-D Secure payment authorisation is carried out separately after card holder authentication. This not only increases the number of messages in the protocol but also has the potential to cause network connection problems, as each message will require a new session to be established. MasterCard Secure Code carries out authorisation and authentication in the same step, thus fewer protocol messages are transmitted across the Internet.
- MasterCard Secure Code has the option of "one-time authentication" for multiple purchases. For an open authenticated session, the card holder is not required to re-authenticate to the issuer. In Visa 3-D Secure, because the merchant initiates the card holder authentication, the card holder is required to authenticate every time he/she makes a purchase.

2.2 Card holder enrolment

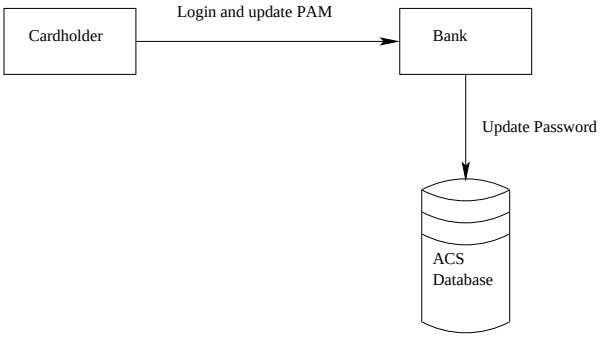

Figure 1: Card holder Enrolment Protocol

Both Visa and MasterCard require their card holders to enrol and register their passwords before they can participate in the payment system (Fig. 1). Payment authentication is carried out only if the card holder has previously enrolled with his/her Issuer. To register their participation in the payment protocol card holders must visit their issuer's enrolment server and provide the relevant information like card number, expiry date and other issuer specified data. The registration process is generally carried out by issuers using their Internet banking web-page thus authenticating the card holder before the registration. Each time the card holder makes a purchase, the registered password is verified, thus providing card holder authentication. The card holders also select a "personal assurance message", which is displayed by the issuer every time the card holder is authenticated.

The enrolment server maintains a record for participating card holders and passes the information to issuer's access control server during payment authentication.

[2]MasterCard has defined a set of HTML hidden fields to collect and transfer information. The merchant specifies hidden fields on payment pages and on successful authentication; the applet populates the fields with authorisation information.

2.3 Payment Protocols

2.3.1 3D secure protocol

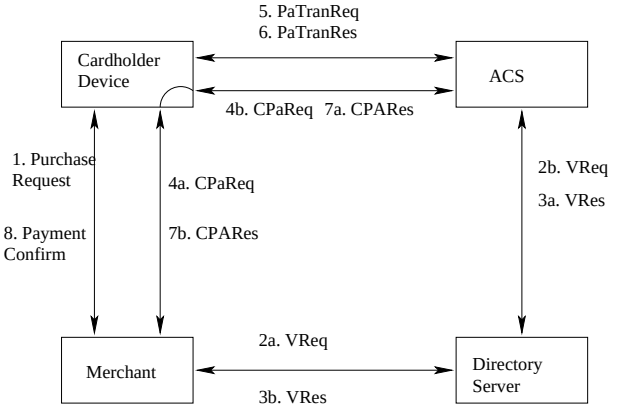

Figure 2: Visa 3D secure purchase protocol

The Visa 3D Secure protocol (Fig. 2) consists of 12 message that are exchanged between the card holder, merchants, directory server and the access control server.

1. *Purchase Request* : The card holder browses participating merchant website to make a purchase. After selecting the items he proceeds to checkout and enters his card details (number, expiry date).
2. *Verify enrolment Request (VRreq)*: The merchant plug-in, which is installed at the merchant's e-commerce web site, gathers customer's information and forwards it to the Visa directory server to confirm if the account is valid for participation in the 3D secure protocol. The Visa directory server authenticates the merchant (based on password or certificates) and contacts the card issuer based on the account number supplied by the merchant plug-in and passes the message from the merchant plug-in for the confirmation of card holder enrolment.
3. *Verify enrolment Response (VRres)*: On receiving the request, the issuer's access control server validates the participation of the card in the 3D secure protocol. For participating cards, an acknowledgement, and URL of the issuer's access control server is sent to the merchant. The visa directory server passes the information back to the merchant.
4. *Condensed Payment Authorisation Request (CPAReq)*: On receiving the acknowledgement for card participation, the Merchant plug-in generates an authorisation request message and contacts the issuer's access control server via the

card holder's browser to send its authorisation request message[3]. The merchant plug-in also includes the merchant's URL for response messages to be sent.

5. *Payment Authentication Transaction Request (PATranReq)*: On receiving a payment authorisation request the access control server displays the pre-registered personal assurance message and requests authentication information.

6. *Payment Authentication Transaction Response (PATranRes)*: The card holder enters the password and if necessary the smart card issued by the Issuer.

7. *Condensed Payment Authorisation Response (CPARes)*: If card holder authentication was successful, ACS then forwards a digitally signed authentication response to the merchant plug-in through the card holder's browser.

8. *Payment Confirmation*: The merchant plug-in on receiving payment authorisation from the ACS generates a receipt for the customer and passes the authorisation token to merchant's acquirer for settlement.

2.3.2 Secure Code protocol

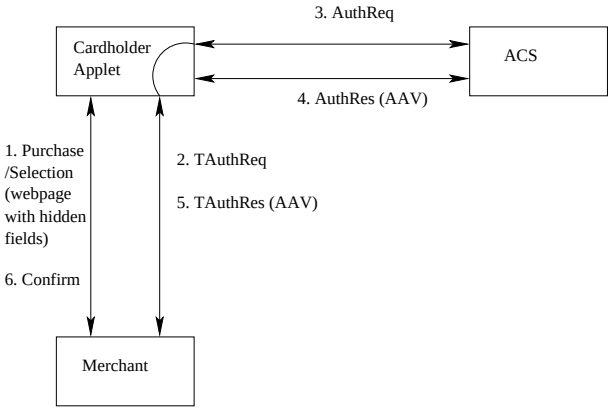

Figure 3: MasterCard secure code purchase protocol (PC Authentication)

MasterCard Secure Code protocol (Fig 3) consists of 8 message that are exchanged between the card holder, merchant and access control server.

1. *Purchase Request*: The card holder browses participating merchant website to make a purchase. After selecting the items, he/she proceeds to checkout and enters the card details (number, expiry date).

2. *Transaction Authorisation Request*: Merchant sends transaction details using hidden field in the final checkout page. The card holder's plugin (Secure Code PC authentication (SPA) applet) which is installed at the customer's device detects the presence of hidden fields on merchant's webpages and makes a secure connection to issuer's Access Control Server (ACS). On establishing a secure connection with the ACS, the SPA applet forwards transaction information to the issuer.

3. *Card holder Authentication Request*: On receiving transaction information, the ACS makes a Payment Authentication Request.

4. *Card holder Authentication Response*: The issuer's access control server authenticates the card holder using previously registered password. If successful, it generates a Account holder Authentication Value (AAV).

5. *Transaction Authorisation Response*: The card holder SPA applet enters the AAV in one of the merchants hidden field, which is then forwarded to the merchant.

6. *Payment Confirmation*: The merchant on receiving AAV generated by the Issuer and optionally he/she can also choose to verify the agreed amount and other merchant details. The merchant issues a receipt to the customer and passes the authorisation token to merchant's acquirer for settlement.

3 Analysing protocols using Casper/FDR

This section provides an overview of Casper and Failure Divergence's Refinement, since they are used in verification of 3D Secure and Secure Code payment systems.

Casper (Gavin Lowe 1985) developed by Lowe, is a compiler which converts a high-level notation of the protocol to a Communicating Sequential Processes (CSP) (Hoare 1985) script. The CSP script can then be run on a model checker like FDR (Formal Systems (Europe) Ltd), to verify if the protocol achieves specific security goals.

A formal model does not cover all aspects of a protocol. Normally the underlying functions are assumed to be true. An apparent limitation of this approach is the verification of the simplified protocols does not necessarily mean the complete version of the protocol is secure against attacks but only suggest the protocols requirements are satisfied. Nevertheless it does provide an assurance to users and designers about the relevant security goals that are met by the protocol.

3.1 Modelling Protocols in Casper

Each agent (users, TTP's, CA's) and intruders who can interact in a protocol are modelled as a CSP process. The resulting system is tested against specifications representing desired security goals. The FDR searches the state space to investigate whether any insecure state (sequence of messages) can occur. If the FDR finds a specification that cannot be met then it returns a trace of the system that does not satisfy the specification. This trace corresponds to an attack upon the protocol.

The modelling of CSP description of the protocol is time consuming and error prone. The aim of Casper is to simplify this process by letting the user specify the protocol at an abstract level. This script when compiled in Casper outputs a CSP script, which is then run through the model checker software FDR2[4].

The Casper script is divided into two distinct parts: a definition of the way the protocols operates and a definition of the actual system to be checked. Each part further consists of four sections specifying variables, processes, protocol description, specification, actual variable, functions, system, and intruder. The first part can be thought as a function that returns a model of a system running the protocols and contains free-variables, processes, protocol descriptions and specifications section. The second part can be thought of as defining a particular image of that function, by instantiating the parameters of

[3]The merchant plug-in uses HTTP POST method for sending information to the ACS's URL

[4]FDR2 software is a refinement checker developed by Formal Systems (http://www.fsel.com/)

the protocol and contains actual variable, functions, system, and intruder section.

3.2 Interpreting FDR output

FDR is a model-checking tool for state machines, with foundations in the theory of concurrency based around Hoare's Communicating Sequential Processes (CSP) (Hoare 1985). The verification technique is based on the method of establishing whether a property holds by testing for the refinement of a transition system and the ability to check the determinism of a state machine that is primarily used for checking security properties. FDR is designed to mechanise the process of carrying out refinement checks.

Casper generates refinement assertions to check for all specifications. It generates one assertion for all secret specifications and one assertion for each agreement and aliveness specification. A CSP script file includes statements making assertions about refinement properties. These statements will typically have the following form:

```
assert Abstract [X= Concrete
```

Example: Secret specification:

```
Secret(B, ban, [A])
```

Assertion generated:

```
SECRET_M::SECRET_SPEC[T=SECRET_M::SYSTEM_S
```

The selected assertion is submitted for testing by choosing the *Run* option from the *Assert* menu in FDR. FDR then attempts to prove the conjecture by compiling, normalising, and checking the refinement. When a test finishes the symbol associated with the assertion is updated to reflect the result. The symbols projected by the FDR are:

- Tick ($\surd$): Indicates that the check completed successfully. i.e., the stated refinement holds.
- Cross (X): Indicates that the check completed, but the refinement does not hold. The FDR debugger is then used as to explore the reasons for the failure.
- Exclamation mark (!): Indicates that the check failed to complete for some reason: either a syntax or type error was detected in the scripts, some resource was exhausted while trying to run the check, or the check was interrupted.
- Zig-zag (Z): Indicates that FDR was unable to complete a check because of a weakness in the currently coded algorithms.

If we find a refinement is not satisfied, then there might be a weakness in the protocol. To examine the weakness, the FDR debugger is invoked. This will open a new window allowing the behaviour of the processes involved to be examined. The information presented by the debugger is represented as two parts: a hierarchical view of the structure of the process represented as a tree, and a series of windows showing the contribution of a selected part of the process to the overall behaviour. The root node represents the process as a whole and when the leaf nodes are expanded, branches are added according to the number of sub-components of that node. For example, a node labelled with a parallel composition symbol ([|..|]) will expand to have two children representing the sub-processes which are combined in parallel. Each child is associated with its own contribution to the overall erroneous behaviour being examined.

When a node in the process structure view is selected, information about the currently selected node is displayed in the behaviour window. The information displayed depends on the nature of the counterexample being examined and the contribution made to it by the selected component. The following types of information may be displayed for each type of counterexample behaviour:
Successful refinement: no information displayed.
No direct contribution: a non-erroneous trace.
Refusal/acceptance failure: a non-erroneous trace, plus the illegal refusal/acceptance.
Divergence: the trace leading to divergence.
Divergence (internally): the trace leading to divergence, plus a trace of repeated events.

The weakness in the protocol is examined by observing the trace leading to divergence.

3.3 Modelling 3D Secure and Secure Code

A major challenge modelling 3D Secure and Secure Code was to carefully simplify the protocols but retain the important protocol mechanisms. This section presents a simplified version of both the protocols and assumptions made during modelling.

In Visa 3D Secure each agent except the card holder holds a public key-secret key pair. And in both protocols each message in the protocol is encrypted using a session key obtained by agents after a SSL handshake protocol.

Visa uses the technique of URL forwarding for transfer of messages from merchant to access control server and back. In message 6 (PAReq) the merchant sends an authorisation request to the access control server. To do this the merchant redirects the card holder's web browser to the URL of the access control server by creating a secure SSL session between the card holder and the access control server. The forwarding of information is be modelled in Casper using the "%" operator.

Casper representation of the Visa 3D secure and MasterCard Secure Code is presented in Figure 4 and 5.

SSL Representation in Casper: Both Visa 3D secure and MasterCard secure code use SSL (Netscape Communications) (Secure Socket Layer) to provide security for data transmission. SSL protocol uses a combination of public key and symmetric key ciphers to establish a secure communication channel between a server and a client. It uses public key encryption system to provide authenticated key exchange and symmetric key cipher system for data encryption. For protocol analysis using Casper/FDR, we assume the following:
1. Both the client and the server successfully negotiated the SSL handshake protocol to establish a symmetric session key.
2. The underlying cryptographic algorithms used in SSL's public key and symmetric key ciphers are secure.

Certificates: All parties unconditionally trust the certification authority and public keys signed by it. The certification authority certifies public key for all entities except the customer who is not required to hold a certified public key. In modelling the protocol, we ignore the process of distribution of certificates and assume that the certification authority has validated all certificates held by the protocol participants.

4 Verifying Visa 3D Secure and MasterCard Secure Code

Visa and MasterCard does not explicitly specify any formal security goals. To verify the protocol we pro-

```
#Protocol description
0.  -> C : M
-- Purchase Request
1. C -> M : {pan,expiry}{keyMC}
-- VRreq
2. M -> DS : {pan,macqbin,mid,mpasswd}{keyDSM}
2a. DS -> ACS : {pan,macqbin,mid,mpasswd}
    {keyACSDS}
-- VRres
3. ACS -> DS : {panyes,acctid,url,proto}
    {keyACSDS}
3a. DS -> M : {panyes,acctid,url,proto}{keyDSM}
-- CPAreq
4. M -> C : {{macqbin,mid,mname,murl,xid,
   pdate,pamt,expiry,acctid}{SK(M)}
   % pareq }{keyMC}
4a. C -> ACS : { pareq % {macqbin,mid,mname,
    murl, xid,pdate,pamt,expiry,acctid}
    {SK(M)} }{keyACSC}
-- PATranReq
5. ACS -> C : {mname,pamt,pdate,panshort,
    expiry}{keyACSC}
-- PATranRes
6. C -> ACS : {password}{keyACSC}
-- CPARes
7. ACS -> C : {{macqbin,mid,xid,pdate,pamt,
    panshort,datetime,transtatus,cavv,eci,
    cavvalg}{SK(ACS)} % pares }{keyACSC}
7a. C -> M : { pares % {macqbin,mid,xid,pdate,
    pamt,panshort,datetime,transtatus,cavv,
    eci,cavvalg}{SK(ACS)} }{keyMC}
-- Payment Confirm
8. M -> C : { {transtatus}{keyMC} }{SK(M)}
```

Figure 4: Visa 3D Secure protocol representation in Casper

```
#Protocol description
0.  -> C   : M
-- Purchase Request
1. C -> M  : {pan,expiry}{keyMC}
-- Authorisation Request
2. M -> C  : {city,con,curr,amt,mts,brand,pan,
   expiry,mname}{keyMC}
2a. C -> ACS  : {city,con,curr,amt,mts,brand,
   pan,expiry,mname}{keyAC}
-- Authentication Request
3. ACS -> C : {mname,amt,pan,mesg,expiry}
   {keyAC}
-- Authentication Response
4. C -> ACS : {securecode}{keyAC}
-- Authorisation Response
5. ACS -> C : {{aav}{SK(ACS)} % avalue}{keyAC}
5a. C -> M : {avalue % {aav}{SK(ACS)}}{keyMC}
-- Payment Confirm
6. M -> C : {cconfirm,mname,amt}{keyMC}
```

Figure 5: MasterCard Secure Code protocol representation in Casper

pose a generic set of security goals that are defined in subsequent subsections. The security goals are categorised into four sections namely Data security, Payer security, Payee security and Transaction security. We present our analysis of the protocols with respect to each security goal.

4.1 Data Security

4.1.1 Third party

Req. Definition: *In an electronic payment system consisting of registration, payment and deposit protocols, any third party not involved in the payment system should not obtain access to participant's transactional data or their secret keys that will lead to a successful execution of a payment (or deposit) protocol.*

The third party security requirement for Visa 3-D Secure can be interpreted as: $\mathcal{C}$'s values `pan` and `edate` should be known only to the parties involved in the protocol $\mathcal{ACS}$, $\mathcal{M}$ and $\mathcal{DS}$ and no other third-party. The Casper specifications for this requirement is represented as:

```
StrongSecret(C,pan,[ACS,M,DS])
StrongSecret(C,edate[ACS,M,DS])
```

For MasterCard Secure Code to satisfy third party data security requirements, the card holder $\mathcal{C}$'s values *pan* and *expiry* should be known only to ACS and merchant $\mathcal{M}$ and no other third party not involved in the protocol. The Casper specification is represented as:

```
StrongSecret(C,pan,[ACS,M])
StrongSecret(C,expiry,[ACS,M])
```

The check for the refinement generated for the above Casper specifications using FDR was successful, which implies the third-party security requirements holds.

4.1.2 Privacy

Req. Definition: *In an e-payment system, from the view of the payer, the payee should not have access to payer's payment information and the bank should not have access to payer's order (invoice) information.*

Under Visa 3D Secure the privacy security requirement can be interpreted as:

From the view of card holder $\mathcal{C}$ - $\mathcal{C}$'s password `password` should be known only to $\mathcal{ACS}$, the personal assurance message `pam` displayed by the $\mathcal{ACS}$ should be known only to $\mathcal{C}$ and $\mathcal{C}$'s accepted payment amount `pamt` should be known only to $\mathcal{M}$ and $\mathcal{ACS}$. The Casper specifications for this requirement are represented as:

```
StrongSecret(C,password,[ACS])
StrongSecret(ACS,pam,[C])
StrongSecret(C,pamt,[ACS,M])
```

From the view of Merchant $\mathcal{M}$ - $\mathcal{M}$'s password `mpasswd` should be known only to $\mathcal{DS}$ and the Casper specification is:

```
StrongSecret(M,mpasswd,[DS])
```

From the view of Access control server $\mathcal{ACS}$ - the transaction status `transtatus` issued by $\mathcal{ACS}$ should be known only to $\mathcal{C}$ and $\mathcal{M}$ and the Casper specification is:

```
StrongSecret(ACS,transtatus,[C,M])
```

Privacy requirement in MasterCard Secure Code can be represented as:

From the view of the card holder $\mathcal{C}$ - $\mathcal{C}$'s password *securecode* and the assurance message *mesg* should be known only to ACS. Casper specification for $\mathcal{C}$'s privacy requirement is:

```
StrongSecret(C,securecode,[ACS])
StrongSecret(C,mesg,[ACS])
```

There is no privacy requirement for Merchants. Values like merchant name, brands accepted, city, etc., used by merchant in MasterCard Secure Code protocol are publicly available.

From the view of the ACS there is also no privacy requirement. The authentication and authorisation value *aav* need not be secret as the *aav* sent by the ACS in an encrypted form and can be read only the ACS. Master card provides two methods: *comparative* - use of random numbers to generate AAV and stored in a database for verification and *cryptographic* - use of encryption to generate AAV. Thus the message transmitted to merchant is in encrypted form and does not rely on certificates and PKI.

The check for the refinement generated for the above Casper specifications using FDR was successful, which implies the privacy security requirements holds.

4.1.3 SSL keys

Both Visa 3-D Secure and MasterCard Secure Code depend on transport layer security to provide confidentiality and integrity for its messages. Every message is encrypted using a shared SSL session key between the two participating agents.

To verify the secrecy and agreement of SSL Keys the following Casper specifications are used:

```
StrongSecret(C,keyACSC,[ACS])
StrongSecret(C,keyMC,[M])
StrongSecret(M,keyMC,[C])
StrongSecret(M,keyDSM,[DS])
StrongSecret(ACS,keyACSC,[C])
StrongSecret(ACS,keyACSDS,[DS])
StrongSecret(DS,keyDSM,[M])
StrongSecret(DS,keyACSDS,[ACS])
Agreement(C,ACS,KeyACSC)
Agreement(ACS,C,KeyACSC)
Agreement(M,ACS,KeyACSM)
Agreement(ACS,M,KeyACSM)
Agreement(C,M,KEYMC)
Agreement(M,C,KEYMC)
```

The check for the refinement generated for the above Casper specifications using FDR was successful, which implies that SSL session keys remain secure and the entities agree on the session key during any complete or incomplete run of the protocol.

4.2 Payer Security

4.2.1 Authentication

Req. Definition: *In an e-payment system, the payer should obtain unforgeable proof of other participant's authenticity before it engages in a protocol with that participant.*

Visa 3-D Secure uses certification authorities, to certify the authenticity of public keys held by the Merchants, Access control servers and Directory servers.

The payer's conversations are only with the Merchant and the ACS and their authenticity is proved by verifying the certificates during the initial SSL handshake. We were able to prove the authenticity of an entity based on the successful proofs obtained from data security requirements that the data encrypted using the SSL session key remain secure and our initial assumption about validity of SSL certificates.

Weakness: An apparent weakness in the protocol exists due to the use of SSL certificates as the only means of authentication. Most clients do not actually view the certificates to verify them. Even though the software tools (e.g. web browsers) check the validity and certification hierarchy of the SSL certificates produced by the server, the client is responsible to actually view the certificate to confirm the identity of the server. This weakness can be exploited in case of mobile payment as the payment protocol relies on a local cache of SSL certificate for validation with pre-existing clients. A possible attack on the system is presented in section 4.6.

In MasterCard Secure Code the card holder directly contacts the ACS using the MasterCard SPA applet which was obtained from his/her bank, the card holder can be assured of the authenticity of the ACS as the card holder implicitly trusts his/her bank. For merchants in MasterCard Secure Code, similar to visa 3D secure, authentication of the merchants is carried out using SSL certificates.

4.2.2 Authorisation and Acceptance

Req. Definition: *In an e-payment system, the payer should obtain unforgeable proof of transaction authorisation by the bank and transaction acceptance by the payee for a particular transaction.*

In Visa 3D secure the Card holder does not receive any proof of transaction authorisation from its Issuer at the end of the transaction. Visa 3-D Secure protocol documentation specifies that the proof is to be provided during the monthly card statement from its issuer. Thus the protocol does not make any improvement on proofs provided to card holders for authorisation that are currently used in existing credit-card based payment system. This is not necessarily a weakness but an inconvenience to the card holder, due to the delay that can occur between a transaction and receiving a monthly statement.

The proof of transaction acceptance from the merchant is provided in the last message which is a signed receipt from the merchant. We can confirm the authenticity of the message by verifying the merchant public key that was used to sign the receipt.

In MasterCard Secure Code the proof of transaction authorisation by the ACS is provided by *aav*. The card holder SPA applet notifies the card holder of authorisation and stores all *aav*'s processed in its internal data base.

The proof of transaction acceptance by the merchant with the confirm message (Message 6) of the protocol. Because MasterCard Secure Code does not use any PKI's the message is not signed but verification is carried out by validating the SSL certificate during that conversation.

4.3 Payee Security

4.3.1 Authentication and Authorisation

Req. Definition: *In an e-payment system, the payee should obtain unforgeable proof of other participant's authenticity before it engages in a protocol with that participant, an undeniable proof of transaction authorisation from the payer and an unforgeable proof of transaction authorisation from the bank.*

Visa 3-D Secure protocols provides proof authentication and authorisation to merchants during transac-

tion authorisation response (PARes). The merchant obtains an unforgeable proof and the proof can be checked by validating the digital signature of the bank on the message. For card holders the merchant relies on the confirmation obtained from ACS about card holder's authenticity. Authentication of directory servers and access control servers can also be done by verifying the SSL certificates during the session setup.

One of the main advantages of MasterCard Secure code over Visa 3d Secure is that the merchant is guaranteed of payment from the card holder. The proof obtained by the merchant in the form of *aav* is preauthorised for payment. The ACS verifies the card holder and also for availability of funds before the *aav* is issued. The AAV obtained is validated by the merchant by verifying the digital signature on the value. The AAV also implicitly provides card holder authentication as the ACS authenticates the card holder before issuing.

4.4 Bank Security

4.4.1 Authentication

Req. Definition: *In an e-payment system, the bank should be presented with an unforgeable proof, certifying the authenticity of the other participants.*

Authentication of the both the card holder and the merchant is by passwords. Security of passwords can be proved by proving that the password remains secure and that the entities agree on the password. The Casper representation for proving authentication of card holder and merchant by the bank and directory server respectively is represented as:

```
Agreement(C,ACS,[password])
Agreement(ACS,C,[password])
Agreement(M,DS,[mpasswd])
Agreement(DS,M,[mpasswd])
StrongSecret(ACS,password,[C])
StrongSecret(DS,mpasswd,[M])
```

In MasterCard Secure code authentication of the card holder is by passwords. The security of the password can be proved by proving the password remains secure and the entities agree on the password. The Casper specification for proving authentication of card holder is:

```
Agreement(C,ACS,[securecode])
Agreement(ACS,C,[securecode])
StrongSecret(ACS,securecode,[C])
```

The check for the refinement generated for the above Casper specifications using FDR was successful, which implies that the bank and directory server can successfully authenticate both the card holder and the merchant.

4.4.2 Authorisation

Req. Definition: *In an e-payment system, the bank before it authorises a transaction should obtain unforgeable proof from the payee, certifying that the payee has agreed to the transaction details and authorised to proceed with the transaction.*

The bank obtains an authorisation proof for transaction from the card holder based on the successful authentication of card holder and proof of agreement on amount and merchant. The Casper representation for proving authorisation by the card holder is represented as:

```
Agreement(C,ACS,[mname])
Agreement(C,ACS,[pamt])
Agreement(C,ACS,[pdt])
Agreement(C,ACS,[pam])
```

In MasterCard Secure Code the proof for card holder's authorisation for payment processing is based on the successful authentication of the card holder and agreement between ACS and card holders on merchant details. The Casper specification is:

```
Agreement(C,ACS,[amt])
Agreement(C,ACS,[mname]
Agreement(C,ACS,[curr])
```

The check for the refinement generated for the above Casper specifications using FDR was successful, implying the bank obtains proof of transaction authorisation from the card holder.

4.4.3 Processing Request

Req. Definition: *In an e-payment system, the bank should obtain unforgeable proof, certifying that the payee has requested a specific transaction to be processed.*

The bank obtains proof of transaction processing request from the merchant, when bank is able to successfully validate the digitally signed payment authorisation request (PAReq) from the merchant.

In MasterCard Secure Code the processing of transaction by the merchant is placed outside the system. MasterCard provides merchant with options to delay processing. This could be useful, as the merchant may be able to collect a number of *aav*'s from various transaction during the day and proceed with a bank settlement once for all transaction. The risk of bank declining a payment is not present as the payments are pre-authorised.

4.5 Transaction Security

4.5.1 Uniqueness

Req. Definition: *In an e-payment system, every transaction processed should be unique.*

Every transaction in visa 3D secure is unique. The uniqueness is obtained is due to the fresh generation of *xid* (transaction id) by the merchant, and its verification by the bank. The payment is also linked to *pdt* (current payment date-time) variable which is checked by the card holder before he/she authorises the payment.

In MasterCard Secure Code transaction uniqueness is achieved because the merchant generates a fresh transaction number *mts* for every transaction. The ACS also verifies *mts* and stores the value in its database along with *aav* if issued.

4.6 An Attack on Visa 3D secure

Visa 3-D Secure uses HTTP POST method for transferring connections which makes it prone to man in the middle attack, when the attacker is a malicious merchant. Consider message 5, the payment transaction authorisation request (PATransreq) is sent by the ACS to the card holder. The card holder connects to the ACS's URL based on the URL the ACS provided to the merchant in message 3 (Vrres) and the connection from the card holders web-page to the ACS is initiated by the merchant. This method of transfer can be exploited by a dishonest merchant to obtain a valid card holder's password by substituting the ACS URL with an URL that would transfer

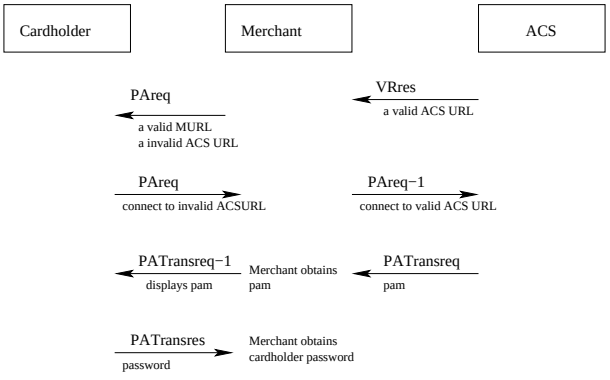

Figure 6: An Attack on Visa 3D Secure

the card holder to a SSL server controlled by the dishonest merchant. The merchant would be able to replicate the authentication dialog by directing connection to the ACS URL and obtaining the personal assurance message *pam* that will be displayed on the authentication dialog. The attack is shown in Fig. 6.

5 Conclusion

Formal methods have become an integral part in verification of protocols. We have used model checking tools, Casper and FDR to verify the security goals of two electronic payment protocols. Both Visa 3-D Secure and MasterCard Secure Code have set out to achieve the same, which is authentication of card holders during on-line-payments. Our representation incorporates SSL into the protocols as both Visa 3D secure and Master Card secure code depend on SSL to provide confidentiality for messages.

Our analysis show, MasterCard has created a more secure protocol with introduction of card holder's applet which includes session management, secure tokens, and more user friendly technology like automatic form-filling. Visa introduces basic but proven solution using PKI's and certificates, which is inefficient way for processing on-line transaction creating network delays and time-outs. Visa also relies heavily on HTTP based browser redirects which can add to further network problems. Visa uses a centralised architecture. Storage of payment and user details in a centralised location can be a concern as they will become a tempting target for hackers, which is avoided by MasterCard's protocol as it is based on a distributed architecture.

Visa 3-D Secure enforces additional responsibility on the card holder to check trustworthiness of merchants. The attack on Visa presented in this paper is based on the merchant being dishonest. Even though currently most customer do on-line-shopping only with known merchants who have a good existing trust relationship with their customers, but this may change in the future with more companies and mainly smaller merchants looking at expanding their market and make their presence on-line. By using Visa 3-D Secure smaller merchants stand to lose because they lack the trustworthiness compared to more well established merchants.

Acknowledgements

We would like to thank Formal Systems (Europe) Limited for providing us with a free educational licence to use their FDR2 software. Josef Pieprzyk was partially supported by ARC grants DP 091484 and DP 0345366.

References

C.A.R Hoare (1985), *Communicating Sequential Processes*, Prentice Hall International.

Netscape Communications (), *SSL 3.0 specification*, http://wp.netscape.com/eng/ssl3/.

Formal Systems (Europe) Ltd (), *Failuers-Divergence Refinement, FDR2 User Manual*, http://www.fsel.com/.

Gavin Lowe (1999), *Casper - A compiler for the analysis of security protocols, User Manual and Tutorial, Ver1.3*

MasterCard (1999), *Master Card Secure code - Merchat implementation guide*, www.mastercardonline.com.

MasterCard & VISA (), *SET Secure Electronic transaction protocol, Book 1,2 and 3*, www.setco.org.

VISA (2002), *3D Secure protocol specification - Core functions*, international.visa.com/fb/main.jsp.

VISA (2002), *3D Secure system overview*, international.visa.com/fb/main.jsp.

VISA (2003), *3-D Secure protocol specification - Extensions for Mobile Internet Devices*, international.visa.com/fb/main.jsp.