Electronic transaction security: an analysis of the effectiveness of SSL and TLS

Vorapranee Khu-smith and Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
Surrey, United Kingdom, TW20 0EX
E-mail: {V.Khu-Smith,C.Mitchell}@rhul.ac.uk

Abstract— **In this paper, security requirements for an electronic transaction are examined. An overview of how SSL and TLS work and their major differences are subsequently provided. The aim of the paper is to investigate how effective these protocols are in securing electronic payments. This is achieved by considering how well they satisfy the identified security requirements. The main finding is that, although SSL and TLS are used widely as a means to secure transactions, they do not provide sufficient security. Since they were designed to protect information while it is being transmitted, the e-commerce transaction data is stored in clear on both the client and merchant machines. This can be a threat in some circumstances. Non-repudiation and authentication are also not satisfactorily addressed. To be more precise, only Web server authentication is provided over SSL/TLS links. Therefore it is possible for both client and merchant to deny making a transaction. Although SSL/TLS provides protections against third party replay attacks, replaying transaction details by merchants and clients remains possible.**

Key Words: SSL; TLS; Electronic transaction security; E-commerce security; Web security

I. Introduction

Electronic commerce is growing in significance. Many products, tangible and intangible, are sold over the Internet, with payments typically made by debit or credit cards. Therefore, there is an increase in concerns associated with the security of the payment systems used to process online transactions. Probably the main concern of most Internet users relates to the confidentiality of payment card information. However, security for online transactions is not limited to data confidentiality, but also includes other security services such as authentication, identification, non-repudiation and data integrity.

In a typical debit/credit card payment system, there are four parties involved namely a client, a merchant, an acquiring bank and a card issuing bank. A client, i.e. the cardholder, makes a payment using a card issued by the card issuing bank (issuer) for something purchased from a merchant. The acquiring bank (acquirer) is the financial institution with which a merchant has a contractual arrangement for receiving (acquiring) card payments. The underlying payment model is shown in figure 1.

While the security of the financial network can be assumed, it is certainly not safe to assume the security of the Internet. At the time of writing, SSL and TLS are the most common means of providing security for the connection between merchants and clients. Consequently, it is the aim of this paper to investigate how effectively SSL and TLS serve this purpose, also bearing in mind security requirements for information handling at the client and merchant sites. It is important to note that the analysis in this paper is based on a business to customer (B2C) transaction using a debit/credit card.

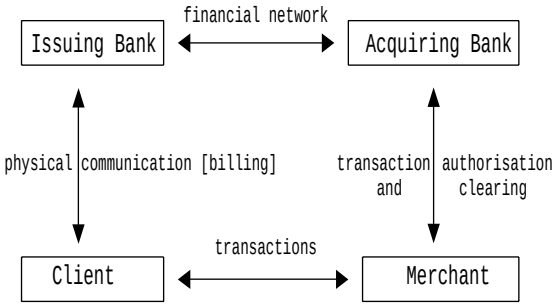

Fig. 1. Debit/credit card payment system.

The paper begins by examining the security requirements for an electronic transaction. An overview of how SSL and TLS work and their major differences are then provided. We subsequently examine the effectiveness of the protocols by considering how well they satisfy the security requirements. The final section summarises and concludes the paper.

II. Security Requirements

As shown in figure 1, a typical card payment system involves four parties namely a card issuer, an acquirer, a merchant and a client. The security requirements for each party vary and hence they will be examined individually. However, the requirements for acquirers and issuers are combined since they are both financial institutions, they are both contractually obliged to abide by the rules of the relevant payment system, and it can reasonably be assumed that they have a similar risk model.

A. Issuers and Acquirers

1. **Non-repudiation:** Issuers and acquirers need to ensure that neither clients nor merchants can deny their participation in a transaction (where the transaction may

involve a refund from merchant to client). In order to achieve non-repudiation, identity authentication may also be needed. Client authentication is required to prove that it is the client who authorised the payment and he/she is a legitimate cardholder. Otherwise, a client can deny making a transaction and the issuer may end up being liable for refunding the amount to the client. On the other hand, if an electronic transaction is found to be fraudulent, merchants are liable for 'card not present' chargebacks. Therefore, it is important for the acquirer to ensure merchant non-repudiation to prevent them challenging their liability.

2. **Integrity:** It is also important to ensure that once details of a transaction have been confirmed, no one can maliciously modify them. Merchants must not be able to alter the amount that a client has agreed to pay. To be more specific, it should not be possible for a merchant to change the amount after it has been authorised by the card issuer. Similarly, a client must not be able to change the amount that has been authorised.

3. **Replay protection:** A malicious merchant should not be able to use a once authorised transaction to obtain a repeat payment. Additionally, merchants should not be able to use an old transaction to request a new payment authorisation no matter how many similar transactions the client has made with them. Issuers and acquirers need a mechanism to detect if a transaction has been replayed so that they do not authorise an illegitimate transaction.

B. Merchants

1. **Non-repudiation:** A merchant needs evidence that a client has agreed to pay the amount associated with a transaction. A merchant also needs to verify that the client is the legitimate cardholder; otherwise, the merchant can be liable for chargebacks. This occurs when a client tells his/her issuer that a particular transaction was not made. The card issuer then immediately submits a chargeback to the acquirer to recover the amount from the account of the merchant in question. Within a predefined period of time, the merchant can dispute the chargeback by providing evidence of, for example, purchase or delivery. Therefore, it is important for merchants to have non-repudiable evidence of the transaction, i.e. to have client non-repudiation. Furthermore, an issuer should not be able to deny having authorised a payment.

2. **Authentication:** As stated before, merchants need client authentication to make sure that the client is the legitimate cardholder. Moreover, they need to be sure that they are communicating with the genuine acquirer. Otherwise, an adversary may masquerade as an acquirer and authorise an illegitimate transaction.

3. **Integrity:** No one should be able to change the details of a transaction once they have been agreed upon. A merchant will not wish to be credited with payment for less than the amount agreed. In addition, an acquirer or issuer should not be able to modify a transaction that has been authorised.

4. **Replay protection:** A malicious client should not be able to present an old proof of purchase to claim for repeat delivery of goods. Likewise, it should not be possible for an acquirer to claim that a merchant has obtained a payment using an old transaction.

C. Clients

1. **Confidentiality and privacy:** Transaction confidentiality, especially card information confidentiality, may be the security service of most concern to users. It is important that cardholder account details are kept secret since they are the main basis on which Internet payments are made. Moreover, some users may require confidentiality protection for the nature of their transactions.

2. **Integrity:** As for the other parties, transaction integrity is important to the client. No one should be able to maliciously modify the transaction details once they have been confirmed. Clients will not want an adversary to change a delivery address, the price or the description of the merchandise after they have agreed a payment.

3. **Authentication:** A client needs to be sure that he/she is dealing with a trustworthy merchant. When shopping on the Internet, it is relatively easy to be lured into visiting a site which appears to sell something but is actually simply collecting card details. Even though a client may have made a purchase from a site before, it is not always obvious whether the page that is being fetched is authentic.

4. **Replay protection:** Clients need a mechanism to ensure that a malicious merchant or an adversary will not be able to reuse previously authorised payments to make a repeat charge.

III. An overview of SSL and TLS

In order to examine the effectiveness of SSL and TLS in securing electronic transactions, it is important to understand how they work. Therefore, in this section we briefly describe how SSL and TLS operate. More detailed specifications can be found in [1] and [2].

A. Secure Sockets Layer (SSL)

The Secure Sockets Layer (SSL) protocol was launched in 1994 by Netscape, with the primary goal of providing secure communications between web browsers and web servers. Security services provided include server authentication, data encryption, (optional) client authentication and data integrity. The following description of SSL operation is based on SSL 3.0, the current version at the time of writing.

SSL is divided into two layers, namely the SSL handshake protocol and the record layer. The handshake protocol, which is the upper layer, is responsible for initialising and synchronising cryptographic state between the communicating parties. The record layer provides confidentiality and authentication, including protection against replay attacks.

In the most typical case, there are five main steps required to establish an SSL connection.

1. The client's browser first sends a ClientHello message to the web server. This message consists of a list of the cipher suites the browser supports, the version of SSL it uses, the

data compression methods it can employ, and a challenge string (a random number and a session ID).
2. The server sends back a ServerHello message consisting of the SSL version number, a challenge string, and the selected cipher suite and compression method. Then the server sends a ServerKeyExchange message containing the server's public key information. The server can optionally request the client's certificate for user authentication by sending a CertificateRequest message. Finally the server sends a ServerHelloDone message to indicate that it has finished with its initial negotiation messages.
3. The client sends its certificate (if requested by the server) in a Certificate message. This is followed by a ClientKeyExchange message which contains key information, i.e. the 'premaster secret' that will be used as a seed to generate the master secret and keys subsequently used for encryption. The key information is encrypted with the server's public key. If client identification is required, a CertificateVerify message must be sent to prove that the client has the private key corresponding to the public key in the certificate. The CertificateVerify message essentially contains a signed hash of the key information and all previous SSL handshake messages exchanged so far.
4. The client sends a ChangeCipherSpec message to indicate the starting point of a protected channel followed by a ClientFinish message which contains a hash of the handshake messages exchanged by the systems and the key information. The ClientFinish message is encrypted and authenticated using the algorithms in the negotiated cipher suite. Note that ChangeCipherSpec messages are not considered as handshake messages and thus are not included in the hash.
5. The server sends back a ChangeCipherSpec message and a ServerFinish message which are similar to the messages with corresponding names sent by the client.

The next section briefly explains how TLS operates and the differences between SSL and TLS.

B. Transport Layer Security (TLS)

In 1995, the IETF introduced a similar protocol called Transport Layer Security (TLS) version 1.0 [2]. Operationally, SSL and TLS work in a very similar way. However, there are some significant differences, as follows.

- The protocol version appearing in SSL messages is 3.0 while for TLS it is 3.1.
- TLS offers 11 more alert message types than SSL.
- For message authentication, SSL combines key information and application data in an SSL-unique fashion. By contrast, TLS employs a widely used and standardised method for computing a Message Authentication Code (MAC), i.e. the HMAC technique [3], to provide message authentication.
- TLS employs a simpler CertificateVerify message. The signed information includes only the handshake messages exchanged so far. However, in SSL, the information consists of two-round hash of the handshake messages, the master secret and the padding.
- TLS employs a pseudorandom function (prf) to generate key materials using a master secret, a label in which the name of the key is specified, and a seed as initial inputs. SSL, by contrast, uses a complex and rather ad hoc procedure to generate key materials.
- The Finish message of SSL is created in an ad hoc way whereas it is generated by a pseudorandom function in TLS.
- The cipher suites offer in SSL includes Fortezza, while in TLS it does not.

The differences are summarised in Table 1 [4].

TABLE I
DIFFERENCES BETWEEN SSL AND TLS

Attributes	**SSL v3.0**	**TLS v1.0**
Protocol version in messages	3.0	3.1
Alert message types	12	23
Message authentication	ad hoc	standard
Key material generation	ad hoc	prf
CertificateVerify message	complex	simple
Finished message	ad hoc	prf
Baseline cipher suites	Fortezza	no Fortezza

Although these differences between the two protocols do exist, in the remainder of the paper both protocols will be referred to as TLS unless explicitly stated otherwise.

IV. Analysis

This section analyses the effectiveness of TLS as a method for securing electronic payments. This is achieved by examining how well it satisfies the security requirements described in Section II. Since TLS was designed to protect communications between Web clients and Web servers, the analysis will only address interactions between clients and merchants. Clearly, TLS cannot, by itself, address any security issues relating to interactions between other pairs of parties. In any event, interactions between issuers and cardholders (mainly relating to card issue and billing) are outside the scope of this paper. Similarly, we can assume that interactions between issuers and acquirers are addressed in the context of securing the financial network, and hence are again outside the scope of this paper.

As far as the acquirer/merchant interactions are concerned, security services can be provided by separate security mechanisms operating to protect communications between the merchant server and the acquirer host. Such mechanisms would typically be managed by the acquirer. Note, however, that unlike TLS, the use of SET[1] does offer a level of protection for merchant/acquirer interactions.

A. Confidentiality

TLS protects transaction confidentiality by using symmetric encryption. The encryption algorithm to be used in any particular connection depends on the cipher suite negotiated in the handshake protocol. Although TLS protects the confidentiality of transferred data against interception

[1] `http://www.setco.org`

attacks, there remain some risks which need to be examined.

Since TLS was designed to provide confidentiality between Web clients and Web servers, transaction information is protected only while it is being transmitted. Therefore, information such as clients' account details and addresses are exposed to the merchant. The users thus have to rely on the security of the merchant's Web server. If someone succeeds in penetrating the merchant server, potentially large numbers of user account details could be compromised.

Another issue is that the US federal regulations have severely restricted the export of strong encryption technology. Until recently, this meant that popularly available TLS implementations only used relatively short key lengths unless both the communicating parties were within the US or Canada [5].

In October 2000 the US export restrictions were relaxed to allow TLS to use longer key lengths when the parties are in the EU or one of eight other countries [5]. However, risks clearly remain for clients and merchants in countries outside the scope of this new exemption. Moreover, this new exemption still only permits 56-bit secret keys, for which exhaustive key searches can be performed [6]. However it is probably hard to imagine a circumstance where it would be worth the effort of breaking such a key given that it will only reveal the details of a single transaction.

TLS also protects the confidentiality of information regarding the nature and value of the transaction whilst this information is transmitted across the Internet. Of course, TLS cannot offer any protection for the confidentiality of this data whilst it is stored at the merchant — although such protection is probably meaningless since the merchant will clearly need to know this information. However, unlike in SET, when using TLS for security the merchant will also know the account details of the purchaser, and hence can use these to link transactions and build profiles of user purchasing behaviour. If required, consumer anonymity could possible be achieved by using alternative payment mechanisms — see, for example, [7]. However, if the merchant will need the shipping address to deliver the purchased goods, then achieving purchaser anonymity will be rather difficult!

B. Integrity

As for confidentiality, TLS provides integrity protection for transferred data only. Consequently, if an adversary succeeds in compromising either the merchant server or the client PC, it would be possible for them to modify the information stored. As a result, such information will not be helpful if there is a dispute. Moreover, for the same reasons, TLS offers no protection against modification of transaction information by corrupt merchants or clients.

C. Authentication

We next consider how TLS ensures the required authentication services — we subdivide this discussion into considerations of merchant authentication, client authentication and acquirer authentication.

C.1 Merchant authentication

The TLS protocol uses the server certificate as the basis of server authentication. The client verifies the server by verifying its ability to decrypt information encrypted using the server's public key. Nevertheless, there remain some risks of server masquerade. One possibility is by means of a 'man in the middle attack'. Such an attack can be launched relatively easily by using a sniffing application such as dsniff[2] to intercept the communications between two entities at the stage of TLS initialisation. An alternative means of launching this attack would be to use 'Web spoofing' instead of a sniffing application. However, in this latter case, the user must be lured into visiting the attacker's page first [8].

Briefly, the man in the middle attack operates as follows. After successfully inserted themselves in the middle of the communication, the attacker simply fetches the page requested by the client from the genuine server. Upon receipt of the requested page, the malicious server returns the spoofed page to the client. The spoofed page is the page containing rewritten URLs of the links on the page. This enables the attacker to maintain a compromised link between the client and whichever server is visited, since if the client clicks any links on the page, the request will go through the attacker and the process repeats [9], [8].

If a TLS connection is in use, the attacker simply establishes two secure connections, one with the client and the other with the server. Thereby, he/she can read and modify the information sent between the two parties as well as convince them that they are communicating via a secure channel. However, since TLS requires server authentication, the attack should be prevented by the client examining the certificate or the URL of the page, since the certificate will show the URL of the attacker instead of the genuine server. However, the attacker can control the appearance of the URL to the client by using scripting techniques — moreover, users will often neglect to check such details, since Web browsers tend to be designed to make things as easy as possible and minimise the work for the user. Hence, although the server authentication in TLS prevents such attacks in theory, the practical situation is rather different.

C.2 Client authentication

While server authentication is mandatory, user authentication is an optional part of TLS. If client authentication is to be provided, a public key pair and certificate for the client are required. However, most clients do not have key pairs and public key certificates. Even if they do, in most cases the key pair is stored in their PC. This gives rise to a further threat, since anyone who has access to the user's PC may gain the ability to make transactions on behalf of the user. This is especially the case where the merchant uses the client identity to access records containing user personal information including mailing address and account details.

[2] `http://www.monkey.org/~dugsong/dsniff/`

C.3 Acquirer authentication

Since TLS only offers protection for Web server/Web client communications, unlike SET it clearly cannot address any security requirements relating to interactions between merchants and acquirers. Such security requirements will therefore need to be addressed in other ways.

D. Non-repudiation

Although TLS uses signatures for session establishment, all protection of communicated data is achieved using symmetric cryptographic techniques. Hence TLS provides no non-repudiation services; that is, neither client nor merchant has any cryptographic evidence that a transaction has taken place.

E. Replay Protection

TLS provides protection against third party replay attacks by including random numbers in the handshake protocol. However, since TLS simply provides a secure means of communication between clients and servers, and provides no long-term 'evidence' regarding transactions (as discussed in Section IV-D), TLS does not provide any protection against manipulation (including replay) or repudiation of transaction information by merchants or clients.

V. Conclusion

Although each party involved in an electronic transaction has a different risk model, they share some fundamental security requirements. These are confidentiality, authentication, non-repudiation, integrity and replay protection.

Due to the purpose of the protocol, TLS provides confidentiality and integrity only while the information is being transmitted. Once the information has reached its destination, TLS offers no protection, and any security measures depend on the choices of the communicating parties. As a result, there are risks of information being compromised if either side of the communication has been penetrated.

TLS only mandates server authentication. Therefore, if TLS is used to protect electronic transactions, it is possible for anyone who has access to the client's PC to impersonate the client. Moreover, TLS does not provide either clients or merchants with protections against repudiation. This makes transaction information stored by either party of little value in the event of a dispute.

TLS provides only partial protection against replay attacks. It prevents a third party using an intercepted TLS messages. However, it does not prevent corrupt merchants or clients re-using a transaction.

From the analysis, two issues are worth noting. Firstly, since SSL and TLS were not designed specifically to secure payments over the Internet, not surprisingly they do not satisfy all the security requirements for electronic transactions. It is important that e-commerce clients and merchants do not have a false sense of security when using them. Secondly, it would be interesting to see how electronic transaction security could be enhanced by combining use of SSL/TLS with certain additional simple security features, as an alternative to accepting the significant cost of adopting a more complex solution such as SET. This latter issue is the subject of ongoing research.

References

[1] A. O. Freier, P. Karlton, and P. C. Kocher, *The SSL protocol version 3.0*, Netscape, 1996.

[2] T. Dierks and C. Allen, *The TLS protocol version 1.0 — RFC 2246*, IETF, January 1999.

[3] H. Krawczyk, M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication — RFC 2104*, IETF, February 1997.

[4] S. Thomas, *SSL and TLS Essentials — Securing the Web*, John Wiley and Sons, Inc., Third Avenue, New York, 2000.

[5] Bureau of Export Administration, Department of Commerce, "Revisions to encryption items," *Federal Register*, vol. 65, no. 203, October 2000.

[6] Electronic Frontier Foundation, *Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design*, O'Reilly, Sebastopol, CA, 1998.

[7] S. G. Stubblebine, P. F. Syverson, and D. M. Goldschlag, "Unlinkable serial transactions: protocols and applications," *ACM Transactions on Information and System Security*, vol. 2, no. 4, pp. 354–389, 1999.

[8] A. Ghosh, *E-Commerce Security*, John Wiley and Sons, Inc., Third Avenue, New York, 1998.

[9] E. Felten, D. Balfanz, D. Dean, and S. Wallach, "Web spoofing: An internet con game," in *Proceedings of the twentieth National Information System Security Conference, Baltimore, Maryland.* October 1997, pp. 95–104, Computer Security Resource Center.