Bringing Confidence to the Web –
Combining the Power of SET and Reputation Systems

Michael Kinateder
Institute of Parallel and Distributed Systems (IPVS)
Universität Stuttgart
Universitätsstr. 38
70569 Stuttgart, Germany
+49-711-7816-230
kinateder@informatik.uni-stuttgart.de

Kurt Rothermel
Institute of Parallel and Distributed Systems (IPVS)
Universität Stuttgart
Universitätsstr. 38
70569 Stuttgart, Germany
+49-711-7816-434
rothermel@informatik.uni-stuttgart.de

***Abstract*— Reputation systems suffer from easy copying of recommendations and recommenders attaching themselves to *trustworthy* recommenders to benefit from their good reputation. Electronic commerce in general and electronic payment systems in particular suffer from the uncertainty of potential customers about the reputation of online merchants and the quality of the offered goods or services. In this paper we address these issues to a certain degree by creating an *originality statement* in the payment process that is included in recommendations to prove, that a particular recommendation is indeed linked to a real world transaction. We present the initial protocol and two variations and discuss their distinct features. Although the protocol is described working in conjunction with the SET payment scheme, it is easily applicable to other payment systems with the features outlined in this paper.**

I. Introduction

Analysts agree, that lack of trust is among the most prominent inhibitors of successful B2C electronic commerce. Consumers are still wary to entrust their payment data to online merchants. The credit card companies Visa and Mastercard have introduced the payment standard Secure Electronic Transaction that allows confidential payment processing without the merchants having access to the payment data.

However, providing confidentiality for payment data is essential but not enough. What people are interested in before making an electronic purchase are the experiences of other persons with the targeted product or service and in addition to that with the merchant or service provider. This is what a tightly integrated reputation system can do that transfers the real world "word of mouth" trust building to the electronic world. This allows consumers to gain access to a whole variety of recommendations and reviews from other users, and reputation mechanisms allow to judge the quality of these reviews and to personalize the list of preferred recommenders.

Organization:

In the following section we will cover background information necessary to understand the achievements of our work, namely a brief overview about *Secure Electronic Transaction* (SET), the *UniTEC Reputation System* and the notation used for describing the following protocols. In Sect. III we will describe the enhanced features to be gained for payment systems and reputation systems by combining the two and especially how this can be done. A basic version of our protocol extension is explained and evaluated in Sect. III and we then move on to two variations in Sect. IV and V.

The work of Michael Kinateder has been funded by Hewlett-Packard Limited.

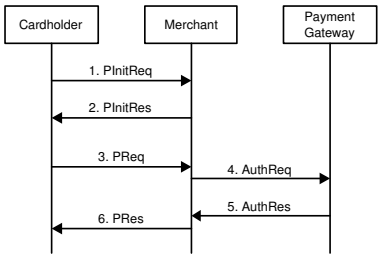

Fig. 1. Typical SET Message Flow.

II. Background

A. Secure Electronic Transaction

The secure electronic transaction protocol (SET) is a payment protocol developed in a joint effort by Visa, Mastercard and several other companies in 1997. *Objectives* of this protocol are among others to provide confidentiality for payment and order information, to ensure the integrity of all transmitted data and to provide cardholder- and merchant authentication. The *key players* are cardholder, merchant, payment gateway (acquirer) and certificate authorities. The cardholder is using a SET wallet software on his or her computer to invoke SET after having selected the goods in the merchant's online store. Figure 1 describes one typical SET message flow performed for a purchase transaction.

The optional *Initiate Request* and *Initiate Response* message pair (*PInitReq*, *PInitRes*) is used among others for the cardholder to obtain the payment gateway's certificate and certificate revocation list (CRL). The main purchase is initiated by the cardholder sending the *Purchase Request* (*PReq*) to the merchant. This mainly contains two distinct parts, namely

the order information and payment information whereas the payment information is encrypted with the payment gateway's public key, hiding its content therefore from the merchant. The merchant forwards this payment information in the *Authorization Request* message (*AuthReq*) to the payment gateway. In case of an authorized payment transaction the payment gateway sends the confirmation to the merchant in *AuthRes* and the merchant sends the confirmation *PRes* to the cardholder so that fulfillment of the order can take place.

Brief evaluation: According to SETCo[1] merchants can expect increased sales due to the increased confidence of the buyers in SET-compliant merchants and increased savings through a reduction of exception handling and reduced costs associated with fraud. From the perspective of the cardholders we see that SET offers increased protection of their privacy by keeping the payment information (credit card data) and order information separate from each other and only visible to the organization with a need-to-know.

On the downside, however, cardholders do have to install the wallet software and obtain the certificates which is somehow burdensome compared to alternative technologies like *Secure Sockets Layer* (SSL) and its successor *Transport Layer Security* (TLS) that are becoming more and more accepted due to seemingly "sufficient" security features and their integration in modern web-browsers. Merchants are relatively slow at adopting the standard due to its complexity and the involved cost.

The mechanisms we introduce in this paper are aimed at increasing the value and usefulness of SET for consumers and credit card companies alike, although our set of protocols can be applied to other payment systems as well, as long as a payment gateway is used that is available for direct communication with consumers and can perform basic cryptographic functions.

B. Reputation Systems

We will give a brief overview about reputation systems in general using the UniTEC[2] system developed at IPVS, Universität Stuttgart as an example. The protocols proposed in this paper however are generally applicable to any reputation system whose recommenders own certificates that bind *pseudonymous identities* to public keys, whose information items are digitally signed and which allows the building of trust in those pseudonyms. More information on UniTEC can be found in [1], [2] and on reputation systems in general in [3] and [4].

The UniTEC system is a distributed reputation system that allows users to estimate their trust in an (information-) source over a link of mutually-known intermediate entities in a certain category. UniTEC is based on a *peer-to-peer system model* with distributed nodes that reside on desktop computers or mobile devices with communication capabilities. Each of these nodes hosts one or more reputation agents each running in the context of a certain entity, mostly the end-user of the system. Each entity uses a certain pseudonym (which one is chosen depends on the category of the exchanged information) when communicating with another entity. Each entity stores a trust and expertise model that contain entries for each pseudonym that it has been in contact with. Whenever experiences with another pseudonym are made, the entries in these two models are updated accordingly.

A user who is requesting trusted information items or recommendations submits the query to the reputation system together with its associated category. For this query and category a set of trusted and knowledgable potential recommenders is chosen from the list of known entities in the trust and expertise model. For each member of this set an information request including the trust in that specific member is built and sent.

Upon receipt of the request the contained trust chain is evaluated and if a fitting response is available it is sent back to the requester. If the trust chain is still strong enough, the request is formed anew and sent on to other potential recommenders again including the own trust in the next recipient (hereby forming the aforementioned trust chain). At some point this dissemination of requests stops, either due to the trust chain being too weak, too many hops, too much time passed or no fitting further recipients being available.

The requester receives the responses to a posed request each with an associated trust chain that are evaluated. The responses are condensed and accumulated where possible and presented to the user that originally sent the query.

At a later point in time, the user might have made an own experience with the queried information (e.g. she has bought the recommended book) and can judge, which recommending pseudonym has given out a fitting and which one a non-fitting recommendation. This information is fed into the reputation system which updates the trust values of the recommenders in the appropriate category accordingly.

C. Our Notation

We will describe here briefly the notation used for the following description of the protocols:

- Symmetric cryptographic keys are denoted by $K_{symmetric}$ whereas private respectively public keys are denoted by $K_{private,Owner}$ respectively $K_{public,Owner}$.
- $Hash(Data)$ includes *just* the *hash* of the mentioned data, not the data itself.
- $Enc(Key, Data)$ refers to the data being encrypted by the appropriate key with a fitting encryption algorithm, e.g. RSA for a public or private key, Triple-DES for a symmetric key.
- $Sign(Key, Data)$ however includes the data to be signed and a digital signature with the appropriate key. This translates to:
  $Sign(Key, Data) = Data, Enc(Key, Hash(Data))$

[1] *http://www.setco.org*
[2] *http://unitec.informatik.uni-stuttgart.de*

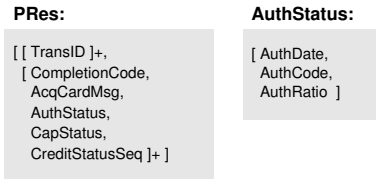

Fig. 2. Simplified Content of the SET PRes Message.

III. The Extended SET Protocol

In this section we will describe how payment systems (respectively SET) and reputation systems (respectively UniTEC) can be combined by generating an *originality statement* in the payment process and integrating it in recommendation messages, thereby linking a recommendation to a real purchase.

The SET protocol may terminate with the merchant sending the *PRes* message to the cardholder as described in Sect. II-A and in much more detail in [5]. The entries in this message correspond to one or more current credit card transactions identified by their transaction identities *TransID*. Each transaction is associated with a completion code and data that further explains the code. In case of a successful credit card transaction the message component *AuthStatus* contains the *AuthCode approved* and *AuthRatio* equals 1 as it refers to the ratio of authorized amount to required amount of the transaction. Fig. 2 illustrates the simplified content of this message.

After the order has been fulfilled and the cardholder has made experiences with the product or service that the transaction was about, she or he initiates the protocol extension as can be seen in Fig. 3 by forming the *Recommendation Signature Request* message (*RecSigReq*) and sending it directly to the payment gateway. The payment gateway processes the request and answers with a *Recommendation Signature Response* message (*RecSigRes*) that contains an *Originality Statement* (*OStat*) that the cardholder can include in his or her recommendation to prove its originality.

We acknowledge the fact, that the information communicated in *RecSigReq* and *RecSigRes* could be included in the original SET messages (Messages 3 to 6 in Fig. 3) as well. Since the recommendation has not been formed yet at that point in time, we can use an encrypted identifier instead of the recommendation hash to be signed by the payment gateway.

A. Recommendation Signature Request RecSigReq

In order to understand the structure of this message it is important to notice the content of UniTEC recommendations that are digitally signed and contain the following components:

- *Recommendation Identifier: RID*
- *Target Identity: TID*
- *Rating: RData*
- *Recommender Certificate (pseudonymous and self-signed): RCert*

After the cardholder has formed the recommendation concerning the purchased product or service the *RecSigReq* message is created with the following structure:

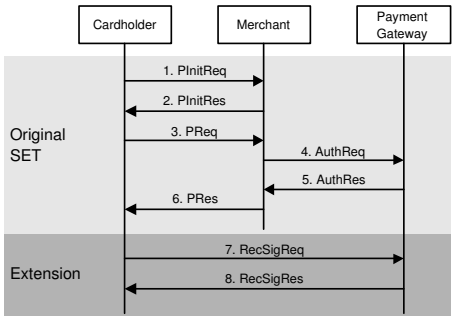

Fig. 3. Message Flow of the Extended SET Protocol.

$$
\begin{aligned}
RecSigReq = \{\ & Enc(\ K_{public,PaymentGateway},\\
& \quad K_{symmetric}),\\
& Enc(\ K_{symmetric},\\
& \quad Sign(\ K_{private,Cardholder},\\
& \quad\quad \{\ TransID,\\
& \quad\quad\quad Enc(\ K_{private,Pseudonym},\\
& \quad\quad\quad\quad Hash(\{RID, TID, RData\})\ )\ \}\ )\ )\ \}
\end{aligned}
$$

Firstly several important parts of the recommendation, the recommendation identifier RID, the target identity *TID* and the rating *RData* itself, are hashed and the hash is encrypted with the private key of the pseudonym that the cardholder intends to use for the recommendation in question. Obviously, this signed hash could only have been built by the recommender since only he has access to the mentioned private key. To allow the payment gateway to process the request and to link this extension to the preceding credit card transaction the field *TransID* from the *PRes* message is added.

This combination is signed with the signature key of the cardholder for the payment gateway to authorize this transaction. To protect the link between *TransID* and the *encrypted hash* from eavesdroppers, this part of the message has to be encrypted. For efficiency reasons we put it in an digital envelope instead of encrypting it with an asymmetric algorithm and the payment gateway's public key. If this part was not encrypted, it would be easier for an attacker to break the link between the cardholder's real identity and his pseudonym as will be seen later. This so-formed message is sent to the payment gateway.

B. Recommendation Signature Response RecSigRes

Upon receipt of the *RecSigReq* message, the payment gateway retrieves the symmetric key by decrypting it with its private key. The symmetric key is used to gain access to the signed statement. In case of an invalid cardholder signature the request is discarded.

If the signature is correct the payment gateway checks whether the included transaction identifier fits to a transaction that this cardholder has performed, whether this transaction has been performed successfully and whether no previous *RecSigRes* message with a different encrypted hash has been sent to the cardholder for this transaction. This ensures, that one *OStat* can be created for a single recommendation corresponding to a real transaction if and only if this transaction really took place. If these tests are successful, the *RecSigRes*

message is formed.

$$RecSigRes = \{\, TransID, \\ Enc(\, K_{symmetric}, \\ Sign(\, K_{private,PaymentGateway}, \\ \{\, Enc(\, K_{private,Pseudonym}, \\ Hash(\{RID, TID, RData\})), \\ PGCert\}\,)\,)\,\}$$

The encrypted hash that has been received in *RecSigReq* and the digital certificate of the payment gateway *PGCert* are signed with the private key of the payment gateway. We will refer to this signed item from now on as *originality statement OStat*. Again in order to protect the link between the real identity and the pseudonym of the cardholder *OStat* is encrypted with the symmetric key used in the previous message. To enable the cardholder to link request to response the *TransID* is included and the message sent.

C. Integration of Originality Statement OStat in Recommendation

Upon receipt of the *RecSigRes* message the cardholder takes the symmetric key corresponding to *TransID* to decrypt *OStat*. The digital signature on *OStat* is checked and in case of a correct payment gateway ought to be correct. The cardholder can now insert the originality statement in the recommendation and publish it via the mechanisms offered by the used reputation system, e.g. UniTEC:

$$Recommendation = Sign(\, K_{private,Pseudonym}, \\ \{RID, TID, RData, RCert, OStat\}\,)$$

Requesters receiving recommendations including an originality statement will perform several tests that all have to succeed in order to accept the recommendation as valid.

The validity of the recommender's signature on the recommendation is checked. If the signature is valid, the payment gateway's signature on *OStat* is verified by using the included certificate (which should be a trusted SET certificate). In case this signature is valid as well, it is certain that *OStat* originated from the payment gateway and a transaction really took place. The *RID*, *TID* and *RData* are hashed. The encrypted hash contained in *OStat* is decrypted with the key contained in *RCert*. If both hashes match, this serves as proof that the recommendation is linked to a real transaction performed at the payment gateway.

D. Evaluation

From a reputation system's point of view, the most important gain is that a recommendation can only be created if a real transaction concerning the recommendation target took place. This also means that identity switching is hindered. A pseudonymous identity will become more valuable, since it is not possible to simply take over the recommendations to a newly created identity. Since copying recommendations is no longer possible without indeed having bought the product or service that the recommendation is about, it is harder respectively more expensive to attach oneself to a well reputable recommender and gain a good reputation by copying the recommendations from this expert. Obviously, even a valid recommendation is not necessarily trusted. The question of whether or not to trust the recommendation and its recommender depends on the trust mechanisms in the used reputation system.

For the financial institutions that are operating the payment gateways one possible gain from such a combination is the possibility to offer their customers a better service. This is a differentiator from other payment gateway providers that is not to be underestimated. Furthermore the participation in a reputation service is a motivation for all participants in payment transactions to behave properly.

On the downside, there is a certain privacy loss through the possibility of the payment gateway to learn the link between the real identity and the pseudonym. If it stores the whole *OStat* instead of just marking completed transactions (with sent *RecSigRes*) and receives recommendations, it can follow the link from the encrypted hash in *OStat* in the recommendation to the encrypted hash received through the *RecSigReq* messages (signed with the real-identity SET cardholder certificate) and find out the link. However, it is quite likely that protection of this data is covered by the current banking confidentiality legislation already and besides some measure of trust in those institutions that handle our bank accounts might be in order. For those readers that are not as trusting we will address this privacy loss in variation 2.

IV. Variation 1: Include Transaction Value

We will now present a minor variation of the protocol presented in Sect. III in order to solve a common problem of reputation systems that suffer from malicious entities building a good reputation with *low-value transactions* and consequently use this reputation for *dishonest high-value transactions* until they are discovered.

The *RecSigReq* message stays the same as before. However instead of the payment gateway including only the encrypted recommendation hash from *RecSigReq* in the originality statement it inserts the *transaction value TValue* as well. This is obviously known from the related SET transaction. The new *RecSigRes* message looks as follows:

$$RecSigRes = \{\, TransID, \\ Enc(\, K_{symmetric}, \\ Sign(\, K_{private,PaymentGateway}, \\ \{\, Enc(\, K_{private,Pseudonym}, \\ Hash(\{RID, TID, RData\})), \\ TValue, \\ PGCert\}\,)\,)\,\}$$

The new *OStat* is defined as the payment gateway-signed component and this time includes the transaction value. As before, the cardholder inserts *OStat* in the recommendation to be published by UniTEC.

On the recommendation requester's side, the same tests for signatures and hashes are performed as already described with the basic variant. This time however, the requester is able to weight the impact of this recommendation against other received recommendations with the transaction value if he or she wishes to do so. In addition to that, the update of trust of the requester in the recommenders, which is performed after own experiences have been made and the quality of

recommendations can be judged, can be weighted with the transaction value as well.

Evaluation

In addition to the points raised in the evaluation of the basic protocol, we gain the ability to weight recommendations depending on the values of the corresponding transactions. This solves up to a certain degree the problems that modern reputation systems such as the one at EBay[3] and other online auction sites face with malicious sellers that first build up a reputation by performing lots of successful but very small-value transactions and then start causing havoc with few (until they are discovered) high-value transactions with missing fulfillment. Building up a good reputation with this weighted scheme should be too expensive to risk loosing the good reputation again by showing malicious behavior.

There is more information provided that could in theory be used for profile building e.g. by linking the transactions of one pseudonym to construct a financial profile. However through the use of pseudonyms the danger of detailed profile building is still kept at bay and the additional value of the provided data for the participants outweighs (in the authors' view) the slightly increased privacy concerns.

V. Variation 2: Credit Card Companies as Trust Enabler

In this variation we address the privacy concerns raised in both aforementioned protocols. Instead of making both recommendation and payment information available to the payment gateway we divide those responsibilities between a "trust server" operated by a credit card company and the payment gateway. The trust server is responsible only for the reputation information part whereas the payment gateway processes the payment data.

Instead of sending the recommendation signature request message *RecSigReq* to the payment gateway, the cardholder sends this to the trust server and receives *RecSigRes* from this server. Since the trust server provider is not directly involved in the credit card transaction performed between the cardholder and the payment gateway, information from the corresponding SET messages is needed to authorize the signature request.

The purchase amount is not included in any signed message *received* by the cardholder during the SET transaction. Thus the cardholder cannot prove the correctness of a certain transaction value to the trust server. If the transaction value – as presented in Sect. IV – should be included in *OStat*, it is necessary to introduce a query-response message pair *Transaction Value Request* (*TValReq*) and *Transaction Value Response* (*TValRes*) as can be seen in Fig. 4 between the trust server and the payment gateway which ensures that the transaction value that the cardholder mentioned to the trust server is correct.

[3] *http://www.ebay.com*

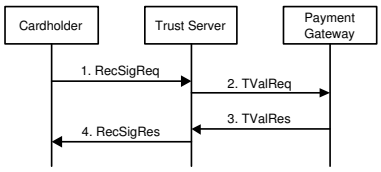

Fig. 4. Message Flow to receive the Originality Statement.

A. Recommendation Signature Request RecSigReq

The *RecSigReq* message is extended to contain an authorization for the trust server to request the transaction value from the payment gateway. This information however could be used to link real and pseudonymous identity of the cardholder and is hidden from the trust server by encrypting it with the payment gateway's public key. *TransID2* is created and inserted instead of the real *TransID*. The encrypted hash is used to confirm to recommendation requesters later, that the underwriter of the recommendation is indeed the one who performed the mentioned transaction.

$$
\begin{aligned}
RecSigReq = \{\ & Enc(\ K_{public,TrustServer},\\
& \quad K_{symmetric}),\\
& Enc(\ K_{symmetric},\\
& \quad Sign(\ K_{private,Pseudonym},\\
& \quad\quad \{\ Enc(\ K_{public,PaymentGateway},\\
& \quad\quad\quad Sign(\ K_{private,Cardholder},\\
& \quad\quad\quad\quad \{\ TransID,\\
& \quad\quad\quad\quad\ \ TValue\}\ )\ ),\\
& \quad\quad Enc(\ K_{private,Pseudonym},\\
& \quad\quad\quad Hash(\{RID, TID, RData\})\ ),\\
& \quad\quad TransID2,\\
& \quad\quad TValue,\\
& \quad\quad PGCert\}\ )\ )\ \}
\end{aligned}
$$

B. Transaction Value Request TValReq

Upon receipt of the *RecSigReq* message the signed part is taken out of the digital envelope and the signature is checked. If this check is successful, the component that is encrypted with the payment gateway's public key is copied to the *TValReq* message. The trust server creates a transaction identifier *TransID3* which is added to the message that is then signed and sent.

$$
\begin{aligned}
TValReq = Sign(\ & K_{private,TrustServer},\\
& \{\ Enc(\ K_{public,PaymentGateway},\\
& \quad Sign(\ K_{private,Cardholder},\\
& \quad\quad \{\ TransID,\\
& \quad\quad\ \ TValue\}\ )\ ),\\
& \ TransID3\}\ )
\end{aligned}
$$

C. Transaction Value Response TValRes

After successfully checking the signature of the *TValReq* message, the payment gateway decrypts the authorization information with its private key and checks the cardholders signature and whether that cardholder has indeed successfully performed the SET transaction with the stated transaction identifier and value. In case of successful tests, the *TValRes* message is built.

$$
\begin{aligned}
TValRes = Sign(\ & K_{private,PaymentGateway},\\
& \{TransID3, TValue\}\ )
\end{aligned}
$$

D. Recommendation Signature Response RecSigRes

After having received *TValRes* with a valid payment gateway signature and matching transaction value the trust server builds the originality statement *OStat* by signing the encrypted hash, the transaction value and the trust server's digital certificate *TSCert* with its private key.

$$
\begin{aligned}
RecSigRes = \{\ & TransID2, \\
& Sign(\ K_{private,TrustServer}, \\
& \quad \{\ Enc(\ K_{private,Pseudonym}, \\
& \qquad Hash(\{RID, TID, RData\})), \\
& \quad TValue, \\
& \quad TSCert\}\ ))\}
\end{aligned}
$$

Upon receipt of the *RecSigRes* message the cardholder checks the trust server's digital signature and can now insert *OStat* into its recommendation as shown before.

E. Evaluation

As opposed to variation 1 we have gained improved privacy protection by strictly separating the SET information (with the real cardholder identity and payment data) and the reputation system information (with the recommendation and the pseudonymous identity). This obviously depends on the players keeping their role and sticking to the protocol as it is. In case of the payment gateway and the trust server working together, there is no way of keeping the link of pseudonym to real identity private. Besides the increased privacy protection we have the benefits mentioned in the evaluation of the basic version and variation 1 as well.

VI. Conclusion

In this paper we have proposed a set of protocols that can be used in order to combine payment and reputation systems with gains on both sides.

The payment system world benefits from users being much more at ease with paying electronically due to (hopefully) good recommendations from other cardholders that made already good experiences with certain merchants. Merchants behaving improperly will be identified and loose business whereas reputable merchants will supposedly gain new customers and increase their revenue. Providing this trust enabling service might turn out to be a new business model for credit card companies like VISA or Mastercard and be a differentiator among payment systems and therefore might give a push to make SET take flight.

The gains on the reputation system side are an improved quality of recommendations by linking the recommendations to real transactions and therefore hindering identity switching, copying of recommendations and malicious entities attaching themselves to reputable recommenders.

The mechanisms proposed here are generally applicable and can be applied to other payment protocols and other reputation systems with the properties mentioned in Sect. II.

References

[1] M. Kinateder and K. Rothermel, "Architecture and Algorithms for a Distributed Reputation System," in *Proc. of the First International Conference on Trust Management*, ser. LNCS, P. Nixon and S. Terzis, Eds., no. 2692. Crete, Greece: Springer-Verlag, May 2003, pp. 1–16.

[2] M. Kinateder and S. Pearson, "A Privacy-Enhanced Peer-to-Peer Reputation System," in *Proc. of the 4th International Conference on Electronic Commerce and Web Technologies (EC-Web 2003)*, ser. LNCS, K. Bauknecht, A. M. Tjoa, and G. Quirchmayr, Eds., no. 2738. Prague, Czech Republic: Springer-Verlag, Sept. 2003, pp. 206–215.

[3] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, Dec. 2000.

[4] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Mining and Knowledge Discovery*, vol. 5, no. 1/2, pp. 115–153, Jan. 2001.

[5] Visa International and Mastercard International, "SET Secure Electronic Transaction Specification Book 3: Formal Protocol Definition," May 1997.