iKP and SET: a comparison

Mark van Cuijk

April 12, 2009

Abstract

Starting in the 1990's, the Internet started to be used for electronic commerce, including electronic payment transactions using credit cards. The available technologies were advanced enough to allow these transactions to happen, but they failed to incorporate security features to protect against threats that were introduced in an Internet environment. Both, iKP and SET address these threats. This paper gives a brief introduction on the iKP and SET protocols and compares the protocols at a high level.

1 Introduction

While the Internet at first was developed as an academic communication medium, at the end of the twentieth century a new usage scenario started to emerge: electronic commerce. An entire electronic commerce scenario might entail a buyer visiting the website of a seller, viewing available products, making a selection of products to form an order, querying or even negotiating a total price for the transaction, performing a payment and delivery of the ordered products. Except for the actual delivery of goods, existing Internet technology at the time allowed all of these steps to be performed.

However, the hostile nature of the Internet introduces security threats when performing an electronic credit card payment that don't exist when performing the payment 'in person'. In particular, exchanging transaction information over a public network like the Internet allows adversaries to eavesdrop or even alter the message flow. The computer environment also allows attackers to selectively filter interesting message and to automate the process of initiating fraudulent transactions. Several electronic payment transaction protocols were developed to secure transactions against these new threats. An overview of the protocols that were developed in this period can be found in [1].

Three of these are 1KP, 2KP and 3KP, which together make up the iKP protocol set [2][3] designed by IBM in 1995. All iKP protocols follow a common message flow, but with a different level of security. In fact, the number $i \in 1, 2, 3$ in the protocol name denotes the number of parties that need a keypair.

Based on the iKP protocol set, MasterCard and VISA have jointly developed the SET Secure Electronic Transaction protocol [4] and released a specification in 1997. The SET protocol is based on the iKP protocol set, augmented with additional messages to allow the protocol to be used in a broader set of usage scenarios.

1.1 Credit card payments

In a classic credit card payment model, a buyer opens an account with an issuing bank and receives a credit card that allows buyer identification during payment transactions. Likewise, a seller opens an account with an acquiring bank. During a payment transaction, the buyer presents the credit card to the seller, that creates a payment slip containing the credit card number of the buyer and the amount of money the transaction encompasses. The buyer signs the payment slip to authorize the transaction and returns it to the seller that stores it in a secure place.

Periodically, the seller forwards a bundle of stored payment slips to his acquiring bank. For each payment slip, the acquiring bank will contact the respective issuing bank to perform the transaction clearing, which means that the issuer debits the specified amount from the account of the buyer and the acquirer credits the account of the seller with the same amount.

A modern model changes the method of authorization from the buyer. In the classic model, this authorization is performed by placing a signature on the payment slip. In the modern model, an electronic device reads a magnetic stripe on the credit card and requests the buyer to enter a secret PIN. The device is capable to verify the correctness of the entered PIN using the information on the magnetic stripe.

1.2 Overview

After this short introduction on the topic, chapters 2 and 3 will respectively introduce the *i*KP protocol set and the SET protocol, outlining the design goals and main protocol requirements; these chapters also briefly introduce the message flows of the protocols. Chapter 4 will compare the three *i*KP protocols amongst each other. A comparison between SET and *i*KP will be presented in chapter 5. The paper will finish with a conclusion.

2 *i*KP

The *i*KP protocol set assumes the credit card payment model introduced in section 1.1 and describes what messages must be transferred between the buyer and the seller and between the seller and his acquiring bank to meet certain security requirements. *i*KP makes use of the existing financial network to perform clearing between acquirers and issuers.

Several security requirements are described in chapter IV of [3]. Summarized, *i*KP was designed to provide:

- proof of transaction authorization by the buyer, the seller and the acquirer;
- the impossibility of unauthorized payments;
- certification and authentication of the seller; and
- a receipt from the seller for the buyer.

In an *i*KP transaction, the number i represents the number of parties that have a public certificate. For 1KP, only the acquirer has a keypair, that allows messages to the acquirer to be encrypted and allows the acquirer to sign messages that it returns. In 2KP, also the seller has a keypair and in 3KP all parties have a keypair.

Before an *i*KP protocol transaction starts, the buyer and seller are assumed to have an agreement on a description about the order and the total amount for payment. From that point, the buyer and seller can start the *i*KP message flow, depicted in figure 1. The messages are described in detail in chapter V of [3], the following description is a simplified summary.

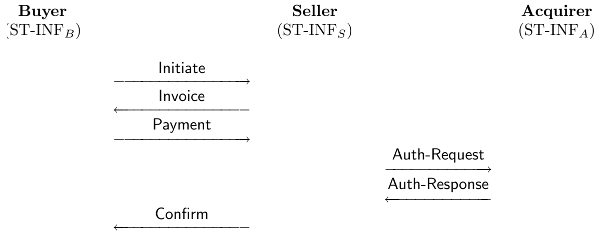

Figure 1: *i*KP message flow.

The "Initiate" message informs the seller that the buyer wants to start an *i*KP transaction. The seller responds with a "Invoice" message, transmitting his identity and transaction identification values to the buyer. Both messages also contain a nonce to prevent replay attacks. The buyer verifies the identity of the seller and creates a slip, containing the price, his credit card details and a hash value computed over all relevant values. The slip is encrypted with the public encryption key of the acquirer and transmitted to the seller using the "Payment" message.

At this point, the seller requests authorization from the acquirer, by forwarding the encrypted slip to the acquirer, together with the transcation identification values it sent in the "Invoice" message and a hash value computed over the same values as the buyer did when creating the slip. The acquirer extracts these values, decrypts the slip from the buyer and computes the hash value over the received values. The acquirer verifies that the three hash values are equals to ensure that the buyer and seller agree on the transaction and that the received values are the ones that are intended by both parties.

When everything checks out, the acquirer performs the transaction clearing with the issuer. The result of the clearing process is signed using the private sigining key of the acquirer and sent to the seller using the "Auth-Response" message and is forwarded to the buyer using the "Confirm" message. The signature makes sure that both the buyer and the seller are certain that the transaction was authorized by the acquirer.

For a more detailed description of the message flow and an analysis of how the protocol meets the security requirements, the reader is invited to study chapter V of [3]. The described message flow is common for all *i*KP protocols. The 2KP and 3KP protocols introduce additional signatures in the messages; these are described in section 4.

3 SET

The SET Secure Electronic Transaction protocol is a payment transaction protocol, that incorporates a broad range of features, including electronic payment, credit processing for returned or defective goods, separation of authorization and capture processing, split payment, installment and recurring payments. The short introduction in this section will only focus on electronic payment. The interested reader is referred to book 2 of [4].

The protocol assumes the credit card payment model, introduced in section 1.1 and describes the message flow between buyer (called cardholder in SET), seller (called merchant) and acquirer and prescribes communication requirements for managing the PKI involved in a SET system. SET makes use of the existing financial network to perform clearing between acquirers and issuers.

In [4], book 1, sections 2.1 and 2.2 and [4], book 2, part I, chapter 2, section 1, several security requirements are described. Summarized, the security requirements of SET are to provide:

- confidentiality of payment information;
- authentication of the cardholder, the merchant and the payment gateway; and
- integrity of protocol messages;

In a SET transaction, at least the payment gateway and the merchant have a certificate with a keypair. Optionally, the cardholder also has a certificate with keypair. SET describes the structure of the PKI involved in [4], book 2, part II, including certificate management details, like how parties can obtain certficates from a CA.

SET can be used in several different ways, depending on preferences of cardholder and merchant. The message flow summary in this section only introduces a basic payment scenario, as depicted in figure 2.

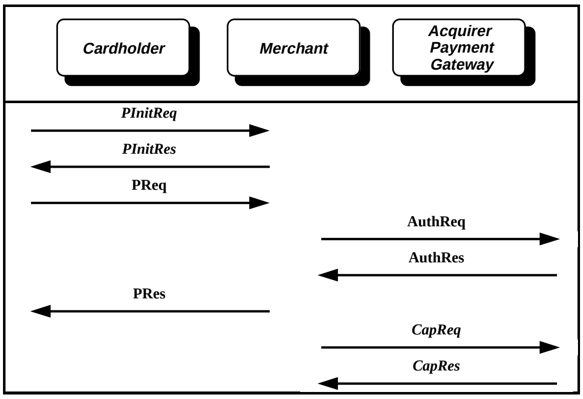

Figure 2: Basic SET message flow. Italic message names indicate optional message.

The messages “PInitReq” and “PInitRes” are used to exchange certificates and nonces. They are optional, but a cardholder risks having out-of-date certificate data and reduces the protection against replay attacks if he chooses to omit this step. To start the actual transaction, the cardholder sends a “PReq” message, that consists of an Order Information (OI) and a Payment Instruction (PI). The OI is meant for the merchant and allows him to verify that the cardholder agrees on the order. The PI is encrypted under the public key of the payment gateway, such that the merchant is unable to read it.

Now, the merchant forwards the encrypted PI to the payment gateway using an “AuthReq” message, requesting authorization for the payment transaction. The acquirer communicates with the issuer using the existing financial network to obtain authorization for the transaction. In SET, this doesn’t mean that the transaction actually occurs and that clearing is performed; this may be postponed to a later moment. However, the merchant may include a capture request in the “AuthReq” message to request the clearing process to be performed directly. The payment gateway responds with a “AuthRes” message to notify the merchant about the result of the authorization request. The merchants forwards the result to the cardholder using a “PRes” message. If the merchant didn’t include a capture request in the “AuthReq” message, an additional “CapReq” / “CapRes” message pair is exchanged to request capturing of the payment.

If the cardholder has a keypair, he creates a dual signature for the OI and PI to be included in the “PReq” message. The dual signature is made from $\mathcal{H}(\mathcal{H}(OI)\,||\,\mathcal{H}(PI))$, where $\mathcal{H}$ is the SHA-1 hash function and $||$ the concatenation operator. Although only the merchant will receive OI and only the payment gateway is able to extract PI, the cardholder sends $\mathcal{H}(PI)$ to the merchant in the “PReq” message and the merchant sends $\mathcal{H}(OI)$ to the payment gateway in the “AuthReq” message, so they can both verify the dual signature. A payment gateway may reject requests that are not signed by the cardholder.

4 *i*KP protocols comparison

As already mentioned in chapter 2, the main difference between the protocols in the *i*KP protocol set is the number of parties in a transaction that have a keypair and certificate. The result of this difference creates a balance between ease of implementation and meeting more security requirements.

In the 1KP protocol, only the acquirer has a certificate. Although this results in meeting only a small subset of the security requirements, the designers argumented that the protocol can find a good use in a transitional phase, before a full deployment of 2KP or 3KP. The acquirer produces a digital signature for the "Auth-Response" message, which is forwarded in the "Confirm" message, so both the buyer and the seller can verify the transaction authorization given by the acquirer.

Using the 1KP protocol, no proof is generated that the merchant has authorized the transaction. Authorization from the buyer is only verified by presenting his credit card number to the acquirer and optionally a secret PIN. This doesn't form a undeniable proof, as the credit card number might be known to other parties, for example to another seller that has performed a transaction with this buyer in a real life setting, not using the *i*KP protocol set. The secret PIN is a weak proof, as it has only a limited entropy and may be even more guessable if it has been chosen by the buyer.

Besides the acquirer, in the 2KP protocol also the seller has a keypair and certificate. The 2KP protocol augments the "Invoice" message — introduced in chapter 2 — with a signature, allowing the buyer to authenticate the seller. The "Auth-Request" message sent to the acquirer is also augmented with a signature, giving the acquirer proof of transaction authorization by the seller.

The 2KP protocol still doesn't provide undeniable proof of transaction authorization from the buyer, but it does provide this from the seller. Although implementing the 2KP protocol requires more effort from the seller, it allows buyers to authenticate sellers, which might in turn lead to more trust in the protocol and therefore a better public acceptance.

In a 3KP protocol setting, the acquirer, the seller and the buyer all have a certificate. With respect to the 2KP protocol, 3KP augments both the "Payment" and "Auth-Requeset" messages with a signature from the buyer. This means that both the acquirer and the seller obtain undeniable proof of transaction authorization from the buyer. In fact, only the 3KP protocol meets all requirements that are defined for the *i*KP protocol set.

A difficult problem with a 3KP protocol deployment is the fact that every buyer has a keypair and must securely store a private key. The problem lies in the fact that most users of such a system are no security experts and might not know how to accomplish this correctly. Tamper-resistant hardware modules might be used to reduce these risks.

Figure 3 summarizes the security requirements that individual *i*KP protocols meet.

REQUIREMENTS/PROTOCOLS	1KP	2KP	3KP
Issuer/Acquirer			
A1. Proof of Transaction Authorization by Buyer	√	√	√√
A2. Proof of Transaction Authorization by Seller		√√	√√
Seller			
S1. Proof of Transaction Authorization by Acquirer	√√	√√	√√
S2. Proof of Transaction Authorization by Buyer			√√
Buyer			
B1. Unauthorized Payment is Impossible	√	√	√√
B2. Proof of Transaction Authorization by Acquirer	√√	√√	√√
B3. Certification and Authentication of Seller		√√	√√
B4. Receipt from Seller		√√	√√

Figure 3: Comparison between *i*KP protocols.

5 SET compared to *i*KP

A very important difference between the *i*KP protocol set and the SET protocol, is the fact that the *i*KP protocol set is designed with only the scenario of a single payment transaction, while the SET protocol is designed for a much broader range of scenarios. For example, the specification includes the following scenarios that are not present in the *i*KP protocol set:

Credit processing to return money when the cardholder returns the goods;

Split payment to split a single order into multiple transactions, e.g. when some goods are delivered immediately, while others have to be back-ordered;

Recurring payments that allow a cardholder to authorize a merchant to periodically capture a certain amount of money, like a monthly phone or internet fee; and

Installment to allow a number of successive payments to settle the total amount.

When focussing on the differences in the basic electronic payment transaction scenario, it is interesting to notice that in a SET protocol setting, the payment gateway and merchant always have a keypair and certificate, but for the cardholder this is optional. This compares very well to 2KP and 3KP. However, 2KP and 3KP are two distinct protocols, so all parties must either agree on using 2KP — such that no cardholder has a keypair — or on using 3KP — such that all cardholders have a keypair — while SET allows a mixed situation, where some cardholders do, but others don't have a keypair. Payment gateways may decide to reject any payment from a cardholder without certificate.

For all protocols, a PKI must be available. In the case of 1KP, this can be a very simple one, but for 2KP, 3KP and SET it need to be organized better. The *i*KP protocol set only describes certain requirements that the PKI must meet, but the SET specification prescribes exactly how it must be organized. The SET specification also describes messages to allow merchants and cardholders to communicate with a Certificate Authority for certificate management.

This kind difference can also be found in other aspects of the specifications. For example, the specification for *i*KP doesn't make any definite choices for cryptographic algorithms to use, but instead gives a list of requirements that an algorithm must meet, together with an example selection. However, the designers have made specific algorithm choices for a protocol demonstration (called the Zürich iKP Prototype, see [3]). In the SET specification, a selection has already been made.

When looking at the actual message flow of this scenario, a very obvious difference is that the SET protocol allows the merchant to split requesting authorization from the payment gateway and requesting the actual money capture. The choice to do so is for the merchant, as SET allows him to request capturing in the "AuthReq" message. The *i*KP protocols always combine authorization and capture request. Typically, the merchant agrees with his acquirer on a transaction fee, that is a small percentage of the transaction amount, a fixed fee or both, often with discounts when money captures are processed in large batches. Therefore, the SET protocol allows the merchant to request authorization during communication with the cardholder, so he can be sure that he will receive the payment, and postpone the capture request to bundle it in a daily payment capture batch.

6 Conclusion

This paper presents the *i*KP protocol set and the SET protocol. All protocols are based on the credit card payment model, incorporating electronic payments over a public network, like the Internet. The main difference between the protocols in the *i*KP protocol set is the number of parties in a transaction that have a keypair and certificate, resulting in a different balance between meeting security requirements and ease of deployment.

The SET protocol differs from 2KP and 3KP, in that it incorporates a broader range of usage scenarios and is more real-life oriented than *i*KP; it exactly describes how the PKI is organized, how the parties in a SET transaction communicate with a Certifying Authority to manage their certificates and it prescribes which cryptographic algorithms are used.

References

[1] N. Asokan and Phillipe A. The state of the art in electronic payment systems. *IEEE Computer*, 30:28–35, 1997.

[2] Mihir Bellare, Juan A. Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Michael Steiner, and Gene Tsudik. ikp – a family of secure electronic payment protocols. In *in First USENIX Workshop on Electronic Commerce*, pages 89–106, 1995.

[3] Mihir Bellare, Juan A. Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Michael Steiner, Gene Tsudik, Els Van Herreweghen, and Michael Waidner. Design, implementation, and deployment of the ikp secure electronic payment system. *IEEE Journal on Selected Areas in Communications*, 18:611–627, 2000.

[4] MasterCard and VISA. Set secure electronic transaction specification, May 1997.