FLAW AND MODIFICATION OF THE *i*KP ELECTRONIC PAYMENT PROTOCOLS

Kazuhiro Ogata[*,**] **Kokichi Futatsugi**[**]

[*] *NEC Software Hokuriku, Ltd.*
[**] *Graduate School of Information Science,*
Japan Advanced Institute of Science and Technology

Abstract: We found that the 2KP/3KP electronic payment protocols as well as the 1KP electronic payment protocol do not possess a probably important property. The property is that if an acquirer authorizes a payment, then both the buyer and seller concerned always agree on it, which is called agreement property in this article. We also propose a modification to have 2KP/3KP possess the property.

Keywords: safety/security in digital systems, electronic payment protocol, agreement property

1. INTRODUCTION

Nobody doubts that security protocols are a key to success of sound development of the Internet, especially success of electronic commerce. But they are subject to subtle errors that are especially difficult to reveal by traditional testing methods and usual operations. Actually Lowe(Lowe, 1995) found out a serious security flaw of the Needham-Schroeder Public-Key authentication protocol(Needham and Schroeder, 1978) 17 years after the protocol was proposed. In addition to the protocol, quite a few security protocols seemingly carefully designed have been found to be insecure so far.

*i*KP (*i*-Key-Protocol, $i = 1, 2, 3$)(Bellare *et al.*, 1995*a*; Bellare *et al.*, 2000) is a family of electronic payment protocols, developed in early 1995 by a group of researchers at the IBM Research labs in Yorktown Heights and Zürich. They have affected the design of well-known SET standard(MasterCard/Visa, 1997).

In this article, we report that the 2KP/3KP electronic payment protocols as well as the 1KP electronic payment protocol do not possess a probably important property. The property is that if an acquirer authorizes a payment, then both the buyer and seller concerned always agree on it, which is called agreement property in this article. We also propose a modification to have 2KP/3KP possess the property. Besides that, we have formally verified that the modified 2KP/3KP actually possess the property with CafeOBJ(Diaconescu and Futatsugi, 1998), an algebraic specification language and system.

The rest of the article is organized as follows. Section 2 provides a summary of *i*KP. Section 3 defines agreement property. Section 4 then shows some counter examples with respect to the property. We propose a modification of 2KP/3KP so that they can possess the property in Section 5. Finally we conclude with Section 6.

2. THE *i*KP ELECTRONIC PAYMENT PROTOCOLS

*i*KP (*i*-Key-Protocol, $i = 1, 2, 3$)(Bellare *et al.*, 1995*a*; Bellare *et al.*, 2000) is a family of electronic payment protocols, developed in early 1995 by a group of researchers at the IBM Research

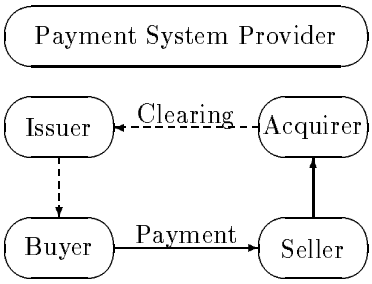

Fig. 1. Generic model of a payment system

labs in Yorktown Heights and Zürich. Afterward it was incorporated into the Secure Electronic Payment Protocols (SEPP), a short-lived standardization effort by IBM, MasterCard, Europay and Netscape. SEPP, in turn, was a key starting point for Secure Electronic Payments (SET), the joint VISA/MasterCard standard for credit card payments(MasterCard/Visa, 1997). In fact, SET retains many of the *i*KP-esque features.

All the *i*KP protocols are based on the existing credit-card payment system. The parties in the payment system are shown in Figure 1. The protocols deal with the payment transaction only (i.e. the solid lines in Figure 1) and therefore involve only three parties called B (Buyer), S (Seller) and A (Acquirer). Note that A is not the acquirer in the financial sense, but a gateway to the existing credit card clearing/authorization network.

The payment system is operated by a payment system provider who maintains a fixed business relationship with a number of banks. Banks act as credit card (account) issuers to buyers, and/or as acquirers of payment records from merchants (sellers). It is assumed that each buyer receives its credit card from an issuer, and is somehow assigned (or selects) an optional PIN as its common in current credit card systems. In 1KP/2KP, payments are authorized only by means of the credit card number and the optional PIN (both suitably encrypted), while, in 3KP, a digital signature is used, in addition to the above. A seller signs up with the payment system provider and with a specific bank, called an acquirer, to accept deposits. Clearing between acquirers and issuers is done using the existing financial networks.

Each acquirer A has a private key K_A that enables signing and decryption. In this article, for brevity, we assume that its public counterpart K_A^{-1} that enables signature verification and encryption is securely conveyed to every buyer and seller participating the protocols via any of a number of key distribution mechanisms. Each seller S in 2KP/3KP and each buyer B in 3KP has a private/public key-pair $(\mathrm{K}_S, \mathrm{K}_S^{-1})$ and $(\mathrm{K}_B, \mathrm{K}_B^{-1})$ respectively. We also assume that each seller's public key is securely conveyed to every acquirer and buyer in 2KP/3KP, and that each buyer's public key is securely conveyed to every acquirer and seller in 3KP.

Cryptographic primitives used in the protocols are as follows:

- $\mathcal{H}(\cdot)$: A one-way hash function.
- $\mathcal{H}_k(K, \cdot)$: A keyed one-way hash function; the first argument K is the key.
- $\mathcal{E}_X(\cdot)$: Public-key encryption with K_X^{-1}.
- $\mathcal{S}_X(\cdot)$: Signature computed with K_X.

Figure 2 shows the three *i*KP protocols from which quantities that are irrelevant to agreement property are hidden. Parts enclosed by $[_{2,3}\ldots]$ and $[_{3}\ldots]$ are ignored for 1KP and 1KP/2KP respectively. The main difference between 1, 2 and 3KP is the increasing use of digital signatures as more of the parties involved possess a private/public key-pair.

Quantities occurring in the protocols are as follows:

- PRICE : Amount and currency.
- NONCE_S : Seller's nonce (random number) used for payment replay protection.
- ID_S : Seller ID.
- R_B : Random number chosen by B to form ID_B.
- BAN : Buyer's Account Number such as credit card number.
- ID_B : A buyer pseudo-ID computed as $\mathrm{ID}_B = \mathcal{H}_k(\mathrm{R}_B, \mathrm{BAN})$.
- RESPCODE : Response from the clearing network: YES/NO or authorization code.

Composite fields used in the protocols are as follows:

- Common : $\mathrm{PRICE}, \mathrm{ID}_S, \mathrm{NONCE}_S, \mathrm{ID}_B$
- Clear : $\mathrm{ID}_S, \mathrm{NONCE}_S, \mathcal{H}(\mathrm{Common})$
- SLIP : $\mathrm{PRICE}, \mathcal{H}(\mathrm{Common}), \mathrm{BAN}, \mathrm{R}_B$
- EncSlip : $\mathcal{E}_A(\mathrm{SLIP})$
- Sig_A : $\mathcal{S}_A(\mathrm{RESPCODE}, \mathcal{H}(\mathrm{Common}))$
- Sig_S : $\mathcal{S}_S(\mathcal{H}(\mathrm{Common}))$
- Sig_B : $\mathcal{S}_B(\mathrm{EncSlip}, \mathcal{H}(\mathrm{Common})))$

We are about to describe how the *i*KP protocols work. Before each protocol starts, each party has the following information:

- B : $\mathrm{PRICE}, \mathrm{BAN}, \mathrm{K}_A^{-1}, [_{2,3}\mathrm{K}_S^{-1}], [_{3}\mathrm{K}_B]$
- S : $\mathrm{PRICE}, \mathrm{K}_A^{-1}, [_{2,3}\mathrm{K}_S], [_{3}\mathrm{K}_B^{-1}]$
- A : $\mathrm{K}_A, [_{2,3}\mathrm{K}_S^{-1}], [_{3}\mathrm{K}_B^{-1}]$

Initiate: B forms ID_B by generating a random number R_B and computing $\mathrm{ID}_B = \mathcal{H}_k(\mathrm{R}_B, \mathrm{BAN})$. B then sends Initiate to S.

Invoice: S retrieves ID_B from Initiate and generates a random quantity NONCE_S that is used later by A to uniquely identify this payment. S forms Common and computes $\mathcal{H}(\mathrm{Common})$. In

Initiate:	B	$\longrightarrow$	S	:	ID_B
Invoice:	S	$\longrightarrow$	B	:	Clear, $[_{2,3}\mathrm{Sig}_S]$
Payment:	B	$\longrightarrow$	S	:	EncSlip, $[_3\mathrm{Sig}_B]$
Auth-Request:	S	$\longrightarrow$	A	:	Clear, EncSlip, $[_{2,3}\mathrm{Sig}_S]$, $[_3\mathrm{Sig}_B]$
Auth-Response:	A	$\longrightarrow$	S	:	RESPCODE, Sig_A
Confirm:	S	$\longrightarrow$	B	:	RESPCODE, Sig_A

Fig. 2. The iKP protocols

2KP/3KP, S also computes Sig_S. Finally S sends Invoice to B.

Payment: B retrieves Clear from Invoice. B computes $\mathcal{H}$(Common), and checks it matches the corresponding value in Clear. In 2KP/3KP, B also validates the signature retrieved from Invoice using K_S^{-1}. Next B forms SLIP and encrypts it using K_A^{-1} (EncSlip $= \mathcal{E}_A$(SLIP)). In 3KP, B also computes Sig_B. Finally B sends Payment to S.

Auth-Request: In 3KP, S validates the signature retrieved from Payment using K_B^{-1}. S forwards EncSlip (and also Sig_B in 3KP) along with Clear (and also Sig_S in 2KP/3KP) as Auth-Request to A.

Auth-Response: A extracts Clear and EncSlip (and also Sig_S in 2KP/3KP and furthermore Sig_B in 3KP) from Auth-Request. A then does the following:

(1) Extracts ID_S, NONCE_S and the value h_1 presumably corresponding to $\mathcal{H}$(Common) from Clear. A checks for replays, i.e. makes sure that there is no previously processed request with the same $(\mathrm{ID}_S, \mathrm{NONCE}_S)$.
(2) Decrypts EncSlip. If decryption fails, A assumes that EncSlip has been altered and the transaction is therefore invalid. Otherwise, A obtains SLIP and, from it, extracts PRICE, h_2 (corresponding to $\mathcal{H}$(Common)), BAN and R_B.
(3) Checks that h_1 and h_2 match.
(4) Rebuilds Common, computes $\mathcal{H}$(Common) and checks that it matches h_1.
(5) In 2KP/3KP, validates Sig_S using K_S^{-1}.
(6) In 3KP, validates Sig_B using K_B^{-1}.
(7) Uses the credit card organization's existing clearing and authorization system to obtain on-line authorization of this payment. This entails forwarding BAN, PRICE, etc. as dictated by the authorization system. Upon receipt of a response RESPCODE from the authorization system, A computes a signature on RESPCODE and $\mathcal{H}$(Common).

Finally A sends Auth-Response to S.

Confirm: S extracts RESPCODE and the A's signature from Auth-Response. S then validates the signature using K_A^{-1} and forwards both RESPCODE and the signature as Confirm to B.

3. AGREEMENT PROPERTY

There are several properties that electronic payment protocols such as the iKP protocols should have. For example they should make it impossible for intruders or malicious sellers to launch replay attacks. The property that we deal with in this article is as follows:

If an acquirer authorizes a payment, then both the buyer and seller concerned always agree on it.

The property is called *agreement property* in this article.

In iKP, that an acquirer authorizes a payment implies that she/he receives the valid Auth-Request corresponding to the payment. Moreover that the buyer and seller concerned agree on the payment (namely the valid Auth-Request) can be stated as they have actually sent the Initiate and Payment, and the Invoice and Auth-Request corresponding to the valid Auth-Request, respectively. Therefore agreement property can be restated as follows:

If an acquirer receives valid Auth-Request *stating that a buyer pays a seller some amount, no matter who has sent the valid* Auth-Request, *then the buyer has always sent the* Initiate *and* Payment *corresponding to the valid* Auth-Request *to the seller and the seller has always sent the* Invoice *and* Auth-Request *corresponding to the valid* Auth-Request *to the buyer and the acquirer respectively.*

The designers of iKP consider eight security requirements that the protocols should satisfy. They conclude that 3KP satisfies all, while 1KP/2KP do not all(Bellare *et al.*, 1995*a*; Bellare *et al.*, 2000). Two of the requirements are closely related to agreement property: A1 – Proof of Transaction Authorization by Buyer and A2 – Proof of Transaction Authorization by Seller. A1 means that when an acquirer debits a certain credit card account by a certain amount, the acquirer must be in possession of an unforgeable proof that the owner of the credit card has authorized this payment, and A2 means that when an acquirer authorizes a payment to a certain seller, the acquirer must be in possession of an unforgeable proof that this seller has asked that this payment be made to her/him. The designers claim that 2KP/3KP satisfy the both, while 1KP does A1 but not A2.

(1) In 1KP Clear′ and EncSlip′ are Clear and EncSlip replaced PRICE with PRICE′ respectively.

Initiate:	IB	$\longrightarrow$	S	:	ID_{IB}
Invoice:	S	$\longrightarrow$	IB	:	Clear
Payment:	IB	$\longrightarrow$	S	:	EncSlip
Auth-Request′:	$IS(S)$	$\longrightarrow$	A	:	Clear′, EncSlip′
Auth-Request:	S	$\longrightarrow$	A	:	Clear, EncSlip
Auth-Response:	A	$\longrightarrow$	S	:	RESPCODE, Sig_A

(2) In 2KP/3KP

Initiate:	IB	$\longrightarrow$	S	:	ID_{IB}
Invoice:	S	$\longrightarrow$	IB	:	Clear, Sig_S
Auth-Request:	$IS(S)$	$\longrightarrow$	A	:	Clear, EncSlip, Sig_S, $[_3\mathrm{Sig}_{IB}]$
Auth-Response:	A	$\longrightarrow$	S	:	RESPCODE, Sig_A

IB and IS stand for the intruder acting as a buyer and a seller respectively. $IS(S)$ means that IS fakes a message seemingly sent by S and sends it.

Fig. 3. Counter examples

4. COUNTER EXAMPLES

As the designers of iKP point out, 1KP does not possess the property. Although you can easily imagine counter examples for 1KP, one of the interesting counter examples is shown in Figure 3 (1). We assume that there exists an intruder that can also act as a legitimate principal in the protocols. The intruder can eavesdrop any message flowing in the network and, from it, glean any quantity except those cryptographically processed (namely it is assumed that the cryptosystem used cannot be broken). Based on the gleaned information, the intruder fakes messages to attack and/or confuse the payment system.

In the counter example shown in Figure 3 (1), the intruder fakes Auth-Request′ seemingly sent by S and sends it to A before A receives Auth-Request from S. Since the intruder knows all the quantities to compute Auth-Request′, she/he can generate and send it to A, and then A receives it as valid. If PRICE′ is smaller than PRICE, the payment would be disadvantageous to S. Although S will notice that this payment transaction is not valid by checking Sig_A, she/he cannot prove it invalid to others.

One interesting point of this counter example is to show that even if two hash values (corresponding to $\mathcal{H}$(Common)) extracted from Auth-Request match, both the buyer and seller concerned do not always agree on the payment. This contradicts the designers' claim that the equivalence between the two hash values ensures that the buyer and seller concerned agree on the order information such as price(Bellare *et al.*, 1995*a*; Bellare *et al.*, 2000).

How about 2KP/3KP? They seemingly possess the property as the designers claim, but there exists a counter example shown in Figure 3 (2). What advantage can the intruder get from the counter example? We can imagine several. S might want to cancel IB's payment request due to some reason if S received Payment from IB, although the cancellation is outside the scope of iKP. In the counter example, A accepts Auth-Request regardless of S's intention. Besides that, S cannot show that she/he has never sent the Auth-Request to A, which also means that it is possible for S to repudiate transmission of Auth-Request even if S has actually sent it.

The intruder may just want to confuse the payment system. S receives Auth-Response from A even if S has never sent the corresponding Auth-Request to A, and gets aware that something, no matter what it is, that does not follow the protocol has occurred. S might decide not to use the payment system because S cannot believe the payment system anymore. Getting worse, the media covers this unexpected behavior of the payment system, and more people stop making use of the payment system. This is clearly disadvantageous to the payment system.

If possible, don't you think that electronic payment protocols should possess agreement property? In the next section, we propose a possible modification to have 2KP/3KP possess the property.

5. MODIFICATION OF THE iKP PROTOCOLS

The reason why the counter example shown in Figure 3 (2) can occur is that IB receives Invoice and gains all the quantities to generate valid Auth-Request. If S newly computes another signature when it sends Auth-Request, not reusing Sig_S used for sending Invoice, then the counter example cannot occur. Therefore the modification is computing a different signature for sending Auth-Request than that used for sending Invoice.

Initiate:	B	$\longrightarrow$	S	:	ID_B
Invoice:	S	$\longrightarrow$	B	:	Clear, $[_{2,3}\mathrm{Sig}_S]$
Payment:	B	$\longrightarrow$	S	:	EncSlip, $[_3\mathrm{Sig}_B]$
Auth-Request:	S	$\longrightarrow$	A	:	Clear, EncSlip, $[_{2,3}\mathrm{Sig2}_S]$, $[_3\mathrm{Sig}_B]$
Auth-Response:	A	$\longrightarrow$	S	:	RESPCODE, Sig_A
Confirm:	S	$\longrightarrow$	B	:	RESPCODE, Sig_A

Fig. 4. The modified *i*KP protocols

Figure 4 shows the modified *i*KP protocols. $\mathrm{Sig2}_S$ is newly introduced, which is defined as follows:

- $\mathrm{Sig2}_S : \mathcal{S}_S(\mathcal{H}(\mathrm{Common}), \mathrm{EncSlip})$

It is used for sending Auth-Request instead of Sig_S.

The reason why *i*KP has been designed as a protocol family is that *i*KP can be gradually deployed, first 1KP, secondly 2KP and finally 3KP. Hence 1KP is inherently weaker than 2KP/3KP with respect to security and does not possess agreement property. Therefore we do not propose a modification of 1KP to have it possess the property.

The modified 2KP/3KP most likely possess agreement property. But we cannot be assured that they really do unless it is formally verified. Therefore we have formally verified that they actually possess the property by modeling them as observational transition systems(Ogata and Futatsugi, 2002*b*), describing the models in CafeOBJ, writing proof scores in CafeOBJ based on the CafeOBJ documents and having the CafeOBJ system execute the proof scores(Ogata and Futatsugi, 2002*a*). The way of modeling the intruder or the enemy is similar to Inductive Method(Paulson, 1998).

6. CONCLUSION

We have reported that 2KP/3KP do not possess agreement property [1], and have proposed the modification to have 2KP/3KP possess the property.

The counter example shown in Figure 3 (2) was found while we were trying to verify that 2KP/3KP possessed agreement property using the CafeOBJ system. Although the CafeOBJ system does not directly help us find such counter examples, it may let us deeply understand targets, leading us to finding counter examples if any.

We do not think that the designers of the *i*KP protocols were careless even though the protocols do not possess the property. Security protocols are that sensitive, and should be verified formally.

[1] Another version of *i*KP is proposed in (Bellare *et al.*, 1995*b*) that is probably the earliest paper on *i*KP and the counter example cannot occur in 2KP/3KP of the version. The authors are grateful to Professor Doug Tygar for pointing this out.

ACKNOWLEDGEMENTS

The authors wish to thank anonymous referees who commented on drafts of this article.

REFERENCES

Bellare, M., J. A. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik and M. Waidner (1995*a*). *i*KP – a family of secure electronic payment protocols. In: *First USENIX Workshop on Electronic Commerce*. pp. 89–106.

Bellare, M., J. A. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik and M. Waidner (1995*b*). iKP – a family of secure electronic payment protocols (WORKING DRAFT). `http://citeseer.nj.nec.com/bellare95ikp.html`.

Bellare, M., J. A. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. Van Herreweghen and M. Waidner (2000). Design, implementation and deployment of the *i*KP secure electronic payment system. *IEEE Journal of Selected Areas in Communications* **18**(4), 611–627.

Diaconescu, R. and K. Futatsugi (1998). *CafeOBJ report*. AMAST Series in Computing, 6. World Scientific. Singapore.

Lowe, G. (1995). An attack on the Needham-Schroeder public-key authentication protocol. *Inf. Process. Lett.* **56**, 131–133.

MasterCard/Visa (1997). SET secure electronic transactions protocol. Book One: Business Specifications, Book Two: Technical Specification, Book Three: Formal Protocol Definition (`http://www.setco.org/set_specifications.html`).

Needham, R. M. and M. D. Schroeder (1978). Using encryption for authentication in large networks of computers. *Comm. ACM* **21**(12), 993–999.

Ogata, K. and K. Futatsugi (2002*a*). Formal analysis of the *i*KP electronic payment protocols. JAIST Research Report, IS-RR-2002-020.

Ogata, K. and K. Futatsugi (2002*b*). Rewriting-based verification of authentication protocols. In: *WRLA '02*. Vol. 71 of *ENTCS*. Elsevier Science Publishers.

Paulson, L. C. (1998). The inductive approach to verifying cryptographic protocols. *Journal of Computer Security* **6**, 85–128.