The Banksys Signature Transport (BST) Protocol

Michel Dawirs and Joan Daemen

Proton World Int.l, Rue du Planeur 10, B-1130 Brussels
dawirs.m@protonworld.com, daemen.j@protonworld.com

Abstract. This document describes a new cryptographic protocol that allows the efficient authentication of an unlimited amount of data coming from an IC Card by an off-line terminal where the IC Card's cryptographic capability is limited to computing one-way functions and MACs. The authentication of the IC Card and its data are based on a Challenge-Response protocol using one-way functions. The protocol is especially suited for use in off-line electronic purse payments in an interoperable environment, where sharing of secret keys between different institutions is not allowed. It is also suited for off-line debit/credit payment applications. The protocol, when applied for purse payments or debit/credit allows transaction timings of the order of 1 second if implemented on currently mass-produced IC Cards.

Keywords. Electronic payment system, purse authentication, cryptographic protocols for IC cards, interoperable purse schemes.

1 Introduction

Interoperability between different Purse schemes is the new challenge for Purse Issuers. Several standards have been issued to achieve this goal. The European Draft Standard prEN1546 [EN1546] defines the Purse interface and transaction flow. However, Purse schemes which comply with prEN1546 are usually not interoperable. The most important reason is that in order to have interoperability between different schemes based on symmetric cryptography, the Purse Issuers must share their secret keys, something they are very reluctant to accept. As a consequence, the European Committee for Banking Standards (ECBS) has initiated the creation of a new standard [ECBS110] focused on interoperability of Purse schemes. This standard is based on EN1546 and EMV'96 [EMV96] and makes use of asymmetric techniques for data authentication. As a consequence, the transaction data are secured without the need of sharing secret keys.

We propose to base the data authentication on the BST protocol that makes use of one-way functions and Issuer-signed payment forms. The advantages of this solution compared with an active public key signature based solution such as proposed in [ECBS110] are twofold:

- The one-way function computations do not require a crypto-coprocessor in the IC Card;

- For parameters that result in comparable security, the execution of the BST protocol on currently available IC Cards is faster than RSA-based dynamic data authentication even if an IC Card with crypto-coprocessor is used for the latter.

2 One-Way Functions

The described protocols make use of so-called *one-way functions* [MOV97]. A one-way function converts an input and a diversification parameter to an output. This is written as:

$$Output \xleftarrow{Parameter} Input$$

The protocol makes use of a one-way function that map an integer number of fixed-length blocks to a single-block output. The security of the protocol relies on the following cryptographic properties of the one-way functions used:

- pre-image resistance: given an output (and diversification parameter) it is infeasible to find a corresponding input;
- 2nd pre-image resistant: given an input, diversification parameter and corresponding output, it is infeasible to find another input for the same output (and diversification parameter).

Collision-resistance is **not** a requirement and for the block length it is advised to take 12 bytes.

3 Card Entity Authentication

Card authentication by the Terminal is based on a challenge-response mechanism: the IC Card proves the possession of a value, the *secret initial value* (*SIV*), consistent with a previously sent *witness*, the *payment token* (*PT*). The protocol assures that an eavesdropper has only a very low probability of success in an attempt to impersonate the Card with respect to a Terminal using the Card response. Figure 1 specifies how *PT* is derived from *SIV* according *to the payment token tree*.

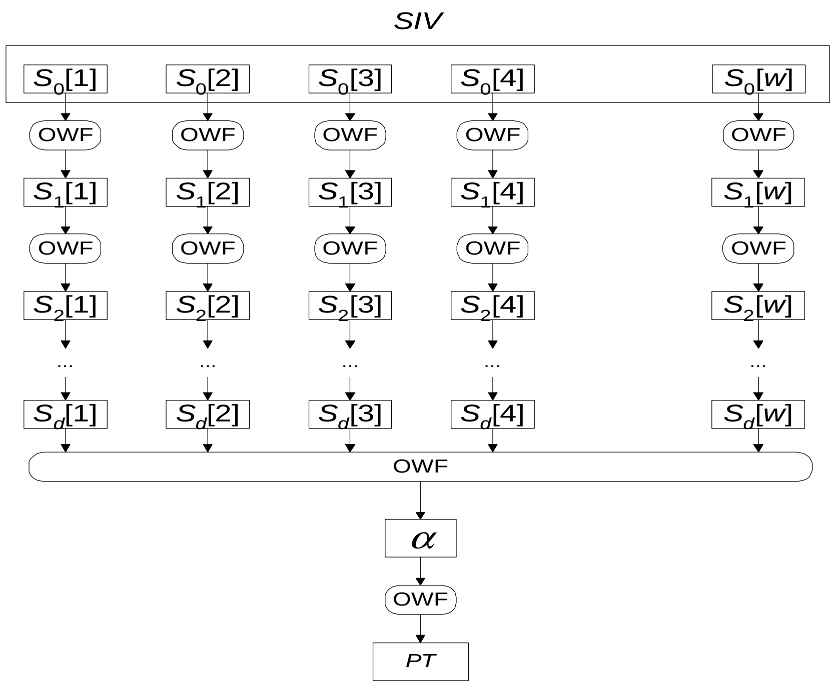

Fig. 1. Payment token tree.

SIV consists of *w* values $S_0[1..w]$. Every intermediate computation value $S_j[i]$ is computed from the value of the previous level $S_{(j-1)}[i]$ by the application of a one-way function diversified by the position in the payment token tree and a unique ID of the Payment token (ID_{PT}):

$$S_j[i] \xleftarrow{i,j,ID_{PT}} S_{j-1}[i] .$$

The Payment Token is derived from the final computation values $S_d[1..w]$ via an intermediate value α (the use of which will be explained later):

$$PT \xleftarrow{ID_{PT}} \alpha \xleftarrow{ID_{PT}} \left(S_d[1], S_d[2], \mathrm{K}\ , S_d[w]\right) .$$

In a first stage of the protocol the Card sends *PT* to the Terminal. In the second stage, the Terminal sends a Challenge to the Card. The Challenge specifies an intermediate level for every column of the payment token tree. A valid Card Response consists of the *w* intermediate values specified in the Challenge. Clearly, the Card knows *SIV* and can therefore compute any intermediate value.

Example: let $w = 8$ and $d = 7$. Let the Challenge specify following intermediate levels: (7, 4, 3, 6, 4, 3, 1, 0). This results in Response: $S_7[1]$, $S_4[2]$, $S_3[3]$, $S_6[4]$, $S_4[5]$, $S_3[6]$, $S_1[7]$, $S_0[8]$.

The Terminal has received *PT* previously and can verify that the Response maps to *PT* as follows:

- for each of the received intermediate values, the terminal computes the final value $S_d[i]$.
- the final values $S_d[1..w]$ are used to compute *PT* (via α).

Obviously, the Card Response to Challenge (0,0,0,0,0,0,0,0) would reveal the secret value *SIV* completely. An eavesdropper could use this value to construct the Response to any Challenge and therefore to successfully impersonate the Card. This is avoided by limiting the valid Challenges to a subset of all possible Challenges such that the Response to one Challenge does not allow to reconstruct the Response to any other one. This is done by imposing the following linear condition:

A Challenge is valid if the sum of the specified intermediate levels is equal to $wd/2$.

The card verifies that that a Challenge is valid prior to generating the Response. It can easily be proven that a valid Challenge has a higher value than any other valid Challenge in at least one column of the payment token tree. Hence, using an observed Response to construct the Response to another Challenge requires the inversion of a one-way function and this is assumed to be infeasible.

If Challenges are generated in a uniform and unpredictable way, the probability of success in impersonation using an observed Response equals 1 divided by the total number of valid Challenges. This number depends on the depth and the width and can be calculated exactly using a recursive procedure. For a width w=8 and depth d=7, it is 1,012,664. This can be increased by augmenting width and/or depth, e.g., for w=12 and d=10, there are 4,577,127,763 valid challenges.

The above argument is only valid if for a given *PT* only a single Response is known to the eavesdropper. It is in fact essential for the security that:

A *PT*, its associated *SIV* and payment token tree are only used once.

This is similar to properties of the one-time digital signatures of e.g. Merkle [ME87]. It implies that the Card contains one *SIV* value per transaction to be conducted. To limit the storage, in practice the *SIV* values are derived from a common secret value by use of a diversified one-way function.

4 Transaction Data Authentication

4.1 With Respect to the Terminal

In the first stage of the Protocol, α is used by the Card as a key to MAC transaction data. This data and the corresponding MAC (denoted by S3 [EN1546]) are sent to the Terminal.

In the second stage of the protocol, the Challenge is sent to the Card that replies with the Response to the Terminal. The Terminal obtains α in the course of authenticating the Card and so can verify the validity of S3. Only the Card (possessing *SIV*) could have generated the MAC as α was private to the Card at the moment the MAC was presented and α maps to *PT* by a one-way function.

4.2 With Respect to the Card Issuer

After the Protocol has been conducted, α is revealed to the Terminal, and so the MAC S3 cannot be used to authenticate the data with respect to some third party. In Payment Transactions (electronic purse, debit/credit) it is important that the *Transaction Trace*, that remains in the Terminal after the transaction and is sent to the Card Issuer afterwards, contains a “proof” of the transaction verifiable by the Card Issuer.

For this purpose, an additional MAC (S6 [EN1546]) is computed in the first stage of the protocol with a secret key known by the Card Issuer. It is included in the data authenticated with S3, but only sent to the Terminal in the second stage of the protocol. It is included in the Transaction Trace and can be verified by the Card Issuer afterwards.

Note: Observe that the amount of authenticated transaction data is not limited by the protocol.

5 Acknowledgement and Cancellation of Purse Payment

For purse payments acknowledgement and cancellation can be performed with *transaction tokens* linked by one-way functions. The basic principle is that the authenticated data in the transaction trace include a so-called *witness token* T_{WITN}, generated in the following way:

$$T_{\text{WITN}} \xleftarrow{ID,TN} T_{\text{CHAL}} \xleftarrow{ID,TN} T_{\text{ACKN}} \xleftarrow{ID,TN} T_{\text{CANC}} .$$

ID and *TN* denote respectively the Terminal unique ID and unique Transaction Number. The Purse payment acknowledgement and cancellation can only be performed by the Terminal that actually conducted the payment transaction.

- Prior to the first stage, the Terminal presents T_{WITN} to the Card. The Card includes it in the data authenticated by S6 and S3.

- In the second stage, the *challenge token* T_{CHAL} is presented to the Card. The card verifies that it comes from the same entity as T_{WITN} by applying the one-way function. If so, it derives the Challenge from the challenge token T_{CHAL} and returns the appropriate Response.
- After the verification of the Response, the Terminal may send the *acknowledgement token* T_{ACKN} that can be verified by the Card.
- After the transaction, the Terminal may cancel it by sending a *cancellation token* T_{CANC}.

The authorisation of a cancellation requires the knowledge of T_{CANC} corresponding to the T_{WITN} in the Transaction Trace (and authenticated by S3 and S6).

6 Incremental Payments

In the basic BST protocol, there is a one-time authentication of data resulting in a transaction trace with a MAC S6 verifiable by the Card Issuer in the Terminal. For repeated electronic purse payments of small fixed amounts (e.g., payphones, photocopier,…) it is useful to have a mechanism to do incremental payments, resulting in only a single transaction trace. For this purpose a simple scheme using one-way function is used.

In the data authenticated by S6 and S3, the Card includes a so-called *initial tick token* TT_0.

A chain of tick tokens TT_i is computed in advance with:

$$TT_{i-1} \longleftarrow TT_i$$

For every incremental payment, the Card supplies the subsequent tick token TT_i. For every Tick Token received, the Terminal:

- verifies that it maps to the previous tick token by applying the one-way function;
- if valid, updates the Transaction Trace with the most recent tick token TT_i, and its index i.

The value associated with the Transaction Trace corresponds with the number of ticks performed, indicated by the tick token index i in the Transaction Trace. The Card Issuer can verify that it is valid by applying the one way function i times to retrieve TT_0 and verify S6 with this value.

7 Outline of a Payment Transaction

Figure 2 gives an overview of a payment transaction based on the BST protocol. In this figure the following conventions are used:

- a MAC computation on *data* using a *key* is denoted by:

$$\text{MAC} \xleftarrow{\textit{data}} \textit{key}$$

- the verification of a MAC or a one-way mapping is denoted by a question mark above the arrow.
- ✐ denotes the recording of information.
- ➲ denotes an irreversible operation.

TD’ denotes transaction data originating from the terminal, *TD*” transaction data originating from the IC Card

	IC Card		Terminal
1	➲ TN_I ++ $S6 \xleftarrow{T_{WITN}, TD} K_I$ $S3 \xleftarrow{T_{WITN}, TD, S6} \alpha$	$\Leftarrow T_{WITN}, TD'$	➲ TN_T ++
		$TD'', \langle PT \rangle, S3 \Rightarrow$	$\langle PT \rangle$ OK ?
2		$\Leftarrow T_{CHAL}$	
	$T_{CHAL} \xleftarrow{?} T_{WITN}$ ➲ $\langle PT \rangle$ set to “used” $Response \xleftarrow{T_{CHAL}} S$		
		$Response, S6 \Rightarrow$	$PT \xleftarrow{?} \alpha \xleftarrow{T_{CHAL}} Response$ $S3 \xleftarrow{T_{WITN}, TD, S6?} K_R$ ✐ *Transaction Trace*
3	$T_{ACKN} \xleftarrow{?} T_{CHAL}$	$\Leftarrow T_{ACKN}$	
4		$TT_i (i = 1,2,3,\ldots) \Rightarrow$	✐ *replace* $TT_{(I-1)}$ *by* TT_i *in Transaction Trace*
5	$T_{CANC} \xleftarrow{?} T_{ACKN}$ ✐ Trace = (T_{ACKN} ,TN_I)	$\Leftarrow T_{CANC}$	Invalidate *Transaction Trace*

Fig. 2. Outline of payment transaction conducted with the BST protocol.

8 Payment Token Authentication

In the protocol the Terminal uses *PT* to verify the response of the Card. As a consequence, a proof must be presented to the Terminal that the *PT* value originates from a Card issued by a genuine Card Issuer. This can be done by a simple certification of the payment token: *PT* is signed with the Card Issuer private key and the resulting signature can be verified by the Terminal. This signature on the payment token is sent along with the payment token in the first stage of the protocol as denoted by $\langle PT \rangle$.

Due to memory restrictions, it is not possible to store a signature for each token. By using authentication trees [ME87], many *PT* values may be signed by a single signature, called a *Payment Form*. An example of such an authentication tree is given in Figure 3. To authenticate *PT*(12) using a signature computed on *IM*, the values *PT*(13), *CV*(7) and *CV*(2) need to be sent along.

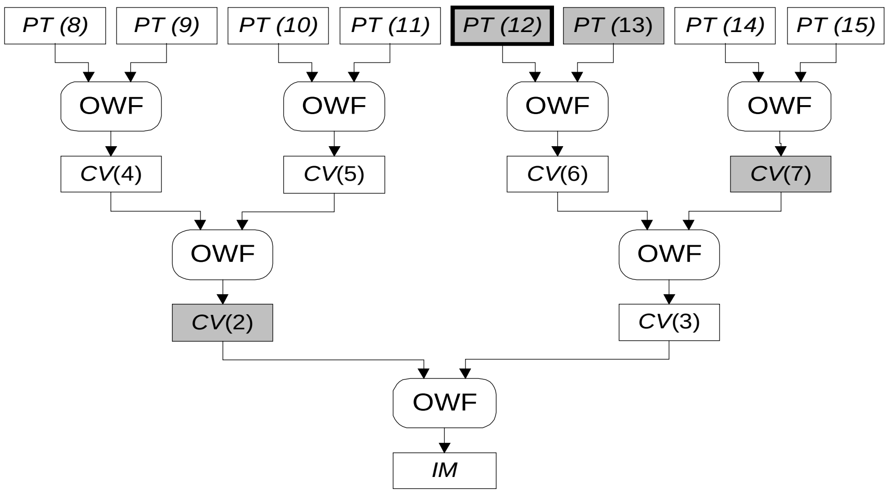

Fig. 3. Example of authentication tree: three-level balanced binary tree.

Pre-Calculation of Authentication Tree Nodes

PT values are the leaf nodes of the authentication trees. To allow the terminal to verify that a *PT* value maps to the imprint in a payment form, a number of nodes of the authentication tree need to be sent along. For authentication trees of realistic size (200-1000), the number of nodes is too large to store them all in the Card. Still, all nodes are needed in using the Payment Tokens. Leaf nodes need to be calculated during the life of the payment form. New authentication tree topologies and algorithms have been devised to allow for large trees with a reasonable amount of pre-calculation and storage. These are based on so-called binary Fibonacci Trees, that are defined recursively. An example of such a tree is given in Figure 4.

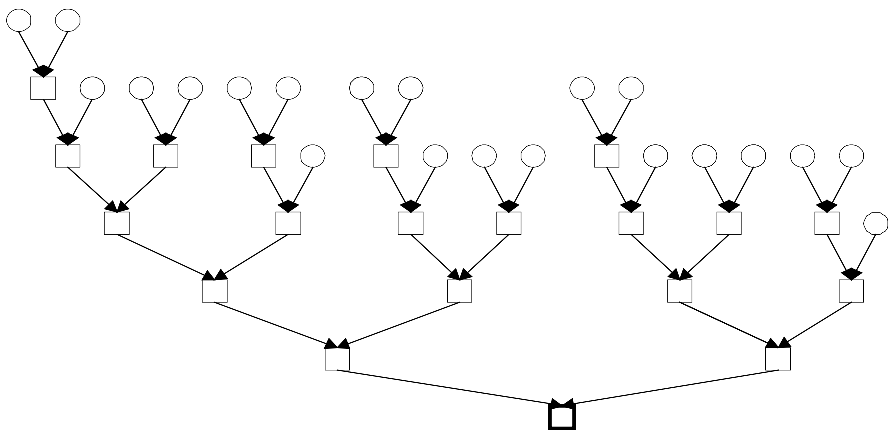

Fig. 4. Fibonacci tree of order 7. It is composed by joining a Fibonacci tree of order 6 at the right and a Fibonacci tree of order 5 at the left.

9 Payment Form Generation and Loading

The generation of a Payment Form consists of the generation of the *SIV* values of all payment tokens, the subsequent computation of the *PT* values themselves, their combination in the authentication trees to an imprint, and the signing of this imprint, combined with payment form specific data (such as expiration date) using the Card Issuer Private Key.

Payment forms can be loaded in the Card at personalisation time or loaded into IC Cards during on-line transactions (e.g. Purse load). Next to the payment form itself, it is essential that also the secret value from which the *SIV* are constructed be loaded in a secure way guaranteeing confidentiality.

10 Conclusions

A new signature transport protocol has been presented that is well suited for the use in IC Card-based electronic purse and debit/credit payment applications.

References

[ME89] R. C. MERKLE, “A certified digital signature”, Advances in Cryptology - Crypto ’89 (LNCS 435), 218-238, 1990

[ME87] R. C. MERKLE, “A digital signature based on a conventional encryption function”, Advances in Cryptology - Crypto ’87 (LNCS 293), 369-378, 1988

[LA79] L. LAMPORT, “Constructing digital signatures from a one-way function”, Technical report, CSL-98, SRI International, Palo Alto, 1979

[EMV96] EMV ‘96, Europay International S.A., Master-Card International Incorporated, VISA International Service Association: IC Card Specification for Payment Systems, version 3.0, June 1996, including the Errata Version 1.0 (31 January, 1998)

[EN1546] prEN 1546 Identification card systems -Inter-sector electronic purse:

- prEN 1546-1: Definitions, concepts and structures, March 1995
- prEN 1546-2: Security architecture, January 1996
- prEN 1546-3: Data elements and interchanges, December 1996

[ECBS110] ECBS TCD 110-1, 2, 3, 4, 5, draft European Banking Standard for Interoperable financial sector Electronic Purse, April 1998

[MOV97] Handbook of Applied Cryptography, A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, CRC Press, 1997