What You Enter Is What You Sign: input integrity in an online banking environment

Sven Kiljan*,†,‡, Harald Vranken*,†, Marko van Eekelen*,†
* Faculty of Management, Science & Technology - Department of Computer Science
Open Universiteit
Postbus 2960, 6401DL Heerlen, The Netherlands

† Faculty of Science - Digital Security
Radboud University Nijmegen
Postbus 9010, 6500GL Nijmegen, The Netherlands

‡ Economy & Management - Lectureship Cybersafety
NHL Hogeschool
Postbus 1080, 8900CB Leeuwarden, The Netherlands

***Abstract*—One problem with most currently used transaction authentication methods is that they depend on the customer's computer for integrity of the information flow between customer and bank. This allows man-in-the-middle attacks to be conducted using malware for financial fraud. Some banks are implementing new authentication methods that allow customers to verify transactions received by a bank without depending on the customer's computer to provide information integrity. These new methods are more complex compared to traditional authentication methods and need the customer's attention to be effective, since it is up to the customer to verify the information that was received by his or her bank. By examining the intrinsic problems of traditional and new transaction authentication methods as used by banks, we designed an alternative authentication method named 'Entered Single Transaction Authentication'. Our method ensures that the bank receives information as the customer entered it without requiring further verification by the customer. We introduce the concept 'What You Enter Is What You Sign', which ensures the digital integrity of information as soon as it is entered. Our proposal is theoretical and high-level, but opens the way for secure transaction authentication methods that rely less on the authenticating party to provide correct information, thereby reducing errors and improving user friendliness.**

***Index Terms*—transaction; authentication; online banking**

I. Introduction

The use of online banking continues to grow in many countries. For instance, in the European Union the use of online banking by individuals aged 16 to 74 increased from 25% in 2007 to 42% in 2013.[1] Examples of growth in individual countries where Internet banking is being used by a large part of the population include the Netherlands (65% in 2007 and 82% in 2013) and Denmark (57% in 2007 and 82% in 2013). Opposite examples of countries where Internet banking is slowly being more accepted include Greece and Turkey (each 4% in 2007 and 11% in 2013). The trend is that the use of online banking continues to grow in most countries.

With this relatively new type of banking comes a new type of fraud. Instead of interacting directly with a bank (i.e. by talking to an employee at a bank's local office), more and more banking customers rely on electronic devices to create wire transfers. Criminals follow suit. Instead of robbing a bank directly (e.g. by threatening an employee or by breaking into a vault), criminals that commit online banking fraud often focus on deceiving customers instead.

Types of attacks that involve the customer can be distinguished by the actions of an adversary. There are impersonation attacks, with which an adversary obtains authentication information (such as user names, passwords and PIN codes) to create malicious transactions. Impersonation attacks are characterized by an adversary creating a new session with the bank in name of (and thereby impersonating) the customer. Another type is a man-in-the-middle attack, in which the adversary injects information in an existing session between customer and bank. With a successful man-in-the-middle attack, neither the bank nor the customer notice any discrepancies when the adversary makes sure that both parties in the session see what they expect to see.

Man-in-the-middle attacks are often executed through the computers of banking customers [1]. Redhead and Povey's (1998) work states that in the development of online banking applications at the time, general attention was too strongly focused on the issues of network security and not enough on the security of the customer's computer [2]. Their prediction was that the developers of malware would use their skills for financial gain by targeting online banking, which they did.

Several banks are applying Customer Verified Transaction Set Authentication (CVTSA) to prevent man-in-the-middle attacks. With CVTSA, a bank receives transaction information in an insecure way (either from the customer or an adversary)

[1]Eurostat - Individuals using the Internet for Internet banking: http://epp.eurostat.ec.europa.eu/tgm/refreshTableAction.do;?pcode=tin00099&language=en

978-1-4799-7901-1/14 $31.00 © 2014 IEEE
DOI 10.1109/STAST.2014.14

and applies a secure way to make a customer validate the information it received. What You See Is What You Sign (WYSIWYS) is used with CVTSA since the customer has to sign information presented by the bank in a secure way. Former empirical research concluded that 21% of online banking customers do not spot significant changes when comparing critical transaction values [3]. CVTSA leaves room for improvement through the mitigation of insecure user behavior.

Our contribution is a new transaction authentication method with the name Entered Single Transaction Authentication (ESTA) which, like CVTSA, aims to prevent man-in-the-middle attacks. With ESTA, a bank can make a distinction between actions of a customer and those of an adversary. What distinguishes ESTA from CVTSA is a new concept which ESTA applies, named What You Enter Is What You Sign (WYEIWYS). WYEIWYS adds data integrity to digital information as soon as it is created (entered by a human). When data integrity is added as early as possible, customers do not need to verify information that the bank received to detect man-in-the-middle attacks. We introduce ESTA using our research platform, known as the Trusted Entry Pad (TEP).

In Section II, we give a high level view of three different types of transaction authentication: the type that is currently in widespread use (Traditional Transaction Authentication, or TTA), a new type which is being introduced by several banks (CVTSA) and our proposal (ESTA). We note an important limitation of TTA and how CVTSA and ESTA solve this. Finally, we note the difference between CVTSA and ESTA.

The following three sections contain use scenarios of each transaction authentication type. Section III gives an example of how a man-in-the-middle attack can be used to work around the security offered by TTA. In Section IV, we explain how CVTSA protects against man-in-the-middle attacks and note how this requires additional attention from the customer compared to TTA. How ESTA is used is noted in Section V. This section also mentions how ESTA offers the same protection as CVTSA without requiring additional attention or actions from the customer during transaction authentication.

Related work to ESTA is noted in Section VI. This includes comparisons with TTA and CVTSA-based authentication devices. We also look at existing technology and concepts which a potential ESTA implementation can use. Section VII follows with possible directions for further research based on our work. Finally, Section VIII contains our concluding remarks.

II. Transaction authentication

Entity and transaction authentication in online banking each apply to different actions initiated by a customer. Entity authentication concerns the customer proving his or her identity to the bank to initiate a new session. Transaction authentication concerns the customer proving the authenticity of transaction requests when the customer asks the bank to approve the requests and create transactions based on them. We distinguish three different types of authentication methods that relate to transaction authentication.

A. Traditional transaction authentication (TTA)

TTA effectively re-applies entity authentication since it misses or has a limited relation with transaction requests. When a bank asks a customer to authenticate a transaction request or a set of transaction requests, the necessary information concerning the transactions is presented to the customer on his or her computer. The computer owned by the customer is a potential man-in-the-middle when it is compromised by malware. When this happens, it cannot protect the information flow integrity from a customer to a bank and vice versa by itself. An adversary can hide newly created illegitimate transaction requests or change characteristics of requests made by the customer before they are sent to the bank. When the bank asks for authentication, the adversary only has to show the original transaction requests to the customer. By hiding the new or modified transaction requests, the customer has no reason for suspicion and continues to authenticate the transaction requests that he or she did not create using TTA.

More information on this flaw in TTA is given in Section III.

B. Customer verified transaction set authentication (CVTSA)

Several banks implement new authentication methods that allow customers to validate transaction requests received by banks without relying on the customer's computer for integrity of information presented to the customer. We refer to these methods as CVTSA. CVTSA applies the concept of What You See Is What You Sign (WYSIWYS) when customers verify (sign) information that can be interpreted in a single semantic context. The information flow of CVTSA is shown in Figure 1.

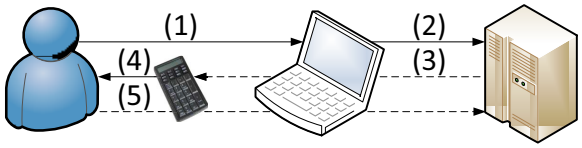

Fig. 1. The transaction information flow in CVTSA methods. A secure device, provided by the bank, is used to present transaction requests received by the bank to the customer for authentication. Dashed lines represent information flows which are optionally forwarded by the traversed devices, depending on the type of authentication device.

The customer enters one or more transaction requests in the client computer (step 1), which are forwarded to the bank (step 2). The bank sends information concerning the transaction requests cryptographically secured to the authentication device in step 3, either through an out-of-band channel or through the customer's computer. The device verifies the authenticity of the received information (i.e. whether it was sent by the bank) and presents transaction request information to the customer in step 4. The customer must confirm that the set of transaction requests entered in step 1 is equal to the set received in step 4. Accepting the transaction requests must only be done if there are no discrepancies. Either the customer does nothing and the transactions are rejected, or the acceptance or rejection is communicated to the bank in step 5. This final step varies in the use of the authentication device and the customer's computer. One example is the use of a verification code (sent

earlier in step 4) to be entered in the customer's computer. Another example is the customer expressing acceptance or rejection on the authentication device, which forwards the customer's decision to the bank either directly or through the customer's computer.

One problem of CVTSA is that it requires more attention from the customer. Not only must the customer check whether information is entered correctly in his or her computer. The customer must also check the information which the bank received. If the customer does not perform the check thoroughly, fraud is still possible. A study of AlZomai et al. (2008) shows that of the test participants, 79% successfully noticed account numbers of which five out of eight digits were replaced in a simulated attack on CVTSA [3]. While this percentage is quite high, it must be noted that the use of the same authentication method over a longer time span was not tested. The risk exists that customers will trade-in security for usability. They can do this by only looking for the validation code that is required for authentication while ignoring the information concerning transaction requests.

An example of the use of CVTSA is given in Section IV.

C. Entered single transaction authentication (ESTA)

We propose a minimalistic approach for transaction authentication which we name ESTA. 'Minimalistic' refers to a minimum of complexity in both usability and technology. Instead of applying WYSIWYS, we apply a new concept that we name What You Enter Is What You Sign (WYEIWYS) for ESTA. This concept is less complex in usability compared to WYSIWYS as applied by CVTSA since the customer does not have to verify data received by the bank. Technical complexity is kept to a minimum by using standard technologies in a simple design.

The information flow of ESTA is shown in Figure 2.

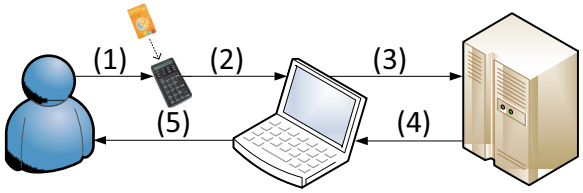

Fig. 2. The transaction information flow of our proposed WYEIWYS-based authentication method for single transaction requests.

ESTA was conceived using our research platform named 'Trusted Entry Pad' (TEP), which represents a device with a display, keypad, smart card slot and a connection to a customer's computer. A smart card is used for storing and applying cryptographic resources. The customer enters one critical value of a single transaction request into the TEP (step 1). After the customer confirms his or her entry, a digital signature of the value is made. Both the value and the digital signature are sent to the customer's computer (step 2), which forwards it to the bank (step 3). The bank checks whether the received value and signature match and sends a confirmation back to the customer's computer (step 4), which shows it to the customer (step 5). The process is repeated for each critical value necessary to complete the transaction request. Examples of critical values include the destination account number (i.e. where the money will go to) and the amount (i.e. how much money will be removed from the customer's account to be sent to the destination account number).

An example of how ESTA can be applied to prevent the manipulation of critical information is given in Section V.

III. Traditional transaction authentication (TTA)

In this section, we clarify why the dependence on an untrusted device to provide information integrity is a shortcoming of TTA. We also point out why aggregated data should not be used as verification information using the sum of a transaction set as an example. First, we will give a scenario in which TTA is used successfully. After that, we demonstrate an attack using the same scenario in which a legitimate transaction is turned into a fraudulent transaction.

Note that the scenario we give for TTA is not fictional. Several banks apply challenge-response authentication for transaction authentication [1].

In the scenario, entity authentication already took place and the customer is logged in the secure banking environment.

The customer is Alice (A). She wants to send money to both Bob and Charlie, and creates two transaction requests in the bank's (B) online environment using her computer (C). The critical values of the transaction requests are the destination account number and the amount (respectively D_1 and S_1 for Bob, and D_2 and S_2 for Charlie). After entry, the bank returns an overview of all prepared transaction requests.

$$D_1 = 123456789, S_1 = 500$$
$$D_2 = 987654321, S_2 = 100$$
$$S_a = \sum S_{1,2} = 600$$

(1) $A \rightarrow C : D_1, S_1$	(7) $B \rightarrow C : N_b, S_a$
(2) $C \rightarrow B : D_1, S_1$	(8) $C \rightarrow A : N_b, S_a$
(3) $A \rightarrow C : D_2, S_2$	(9) $A \rightarrow T : PIN, N_b, S_a$
(4) $C \rightarrow B : D_2, S_2$	(10) $T \rightarrow A : E_K\{N_t, N_b, S_a\}$
(5) $B \rightarrow C : D_1, S_1, D_2, S_2$	(11) $A \rightarrow C : E_K\{N_t, N_b, S_a\}$
(6) $C \rightarrow A : D_1, S_1, D_2, S_2$	(12) $C \rightarrow B : E_K\{N_t, N_b, S_a\}$

Alice enters the critical values of each transaction request in her computer, which sends it to the bank (steps 1 to 4). The bank returns an overview of all entered transactions for Alice to verify on her computer (steps 5 and 6).

Before the wire transfers are created, the bank authenticates Alice's transaction requests using challenge-response authentication. Alice receives a random nonce generated by the bank (N_b) and the rounded down total amount of all prepared transaction requests (S_a) through her computer (steps 7 and 8). These two values form the challenge. S_a is first used by Alice to verify that the total sum of all transactions as received by the bank is the same total sum of the transaction requests she entered in steps 1 and 3. After the verification, Alice unlocks the functionality of an electronic token (T) using a Personal Identification Code (PIN) and enters the challenge (step 9).

The token creates a response by encrypting the current time stamp from its local clock (N_t) and the challenge with a

symmetric key (K) that is only known to the token and the bank. Alice enters the result in her computer, which in turns sends the encrypted message to the bank (steps 10 to 12).

To verify the correctness of the transaction requests, the bank must first decrypt the received response using K. Of the decrypted values, N_t and the current time must both be in an accepted time frame to prevent replay attacks. N_b and S_a must be equal to the challenge that was sent. If verification is successful, wire transfers are created for the transaction requests that Alice made in steps 1 and 3.

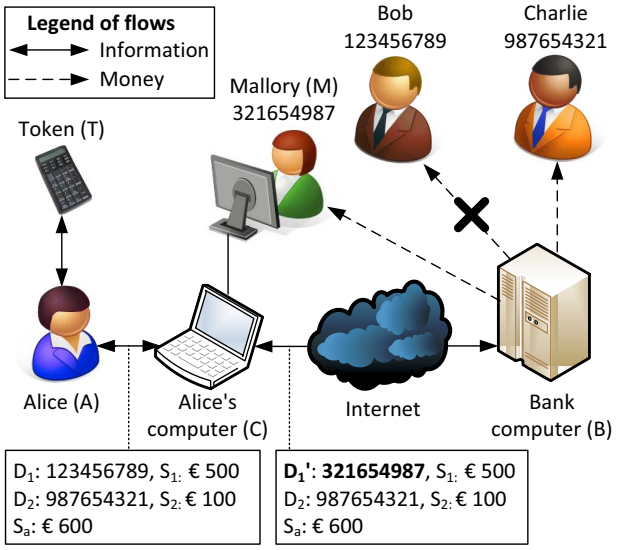

Fig. 3. Alice fails to transfer money to Bob.

Mallory (M) is a malicious adversary and has control over Alice's computer, as shown in Figure 3. Her goal is to gain the money that is meant for Bob.

$$D_1' = 321654987$$

(1) $A \rightarrow C : D_1, S_1$
$C \rightarrow M : D_1, S_1$
$M \rightarrow C : D_1'$
(2) $C \rightarrow B : D_1', S_1$
(3) $A \rightarrow C : D_2, S_2$
(4) $C \rightarrow B : D_2, S_2$
(5) $B \rightarrow C : D_1', S_1, D_2, S_2$
$C \rightarrow M : D_1', S_1, D_2, S_2$
$M \rightarrow C : D_1$
(6) $C \rightarrow A : D_1, S_1, D_2, S_2$

Before the destination account number and amount of the first transaction request are sent to the bank, Mallory replaces the original destination account number D_1 with account number D_1', which is under her control (between steps 1 and 2). Mallory intervenes again when the bank sends the transaction requests back to Alice's computer for review. She makes sure that Alice sees the transactions as she entered them in her computer. The modified destination account number that was sent to the bank is kept hidden from Alice due to Mallory's second intervention (between steps 5 and 6).

Alice proceeds with the challenge-response scheme, in which Mallory does not have to intervene (steps 7 to 12). Alice does not get suspicious when she checks S_a (the total sum of all transaction requests) as part of the challenge in step 8 since it has not changed. S_1 and S_2 were not modified, and therefore S_a stays the same. The attack is a success.

Mallory could also change both S_1 and S_2 for her benefit in a more advanced attack. She only needs to make sure that S_a stays the same to avoid suspicion by Alice (e.g. S_1' as € 599 and S_2' as € 1), since S_a is part of the challenge. If S_a would be changed, Alice would enter the wrong challenge in T (the bank's token), which creates an invalid response that will be noticed by the bank.

The flaw in the design of the used TTA method is that the integrity of critical information provided by Alice was not safeguarded, which allowed Mallory's attack to succeed. Alice's computer acts as a man-in-the-middle, which allows it to change unverified information between Alice and the bank.

The bank offers its customers the possibility to verify the total sum that is received by the bank. Because this concerns aggregated information (S_a in steps 7 and 8), the semantic content of the message (the set of transaction requests from steps 1 to 6) can be changed.

The example applied challenge-response authentication, but an adversary also has the same opportunity if other multi-factor authentication methods are used in which the integrity of the customer's input is not protected. This includes one-time passwords and digital signatures over information which a customer can only verify on the display of his or her computer.

We have explained why TTA does not protect the information integrity of transaction requests between customer and bank with this scenario. An opening for man-in-the-middle attacks is present when a bank depends on a customer's computer for information verification by a customer.

IV. Customer Verified Transaction Set Authentication (CVTSA)

In the previous section we have shown an example of a legitimate banking session being turned into an illegitimate session by a man-in-the-middle, represented by the banking customer's computer. In this section, we show an example of how CVTSA methods that are currently being introduced by banks cope with this. As noted in Section II, the concept of WYSIWYS is applied by signing a set of transaction requests. See Figure 1 on page 2 for an information flow overview.

The start of the scenario is similar to the scenario given in the previous section. We assume that some form of entity authentication already took place. Alice (A) is ready to use her computer (C) to transfer money from her account at her bank (B) to both Bob and Charlie. The difference in this scenario is that Alice now possesses a different authentication device provided by the bank with the name 'Reader' (R). Reader is an electronic device with a relatively large display, a keypad and a camera. It can be used to capture barcodes with information sent by the bank through the display of Alice's computer. The information includes critical values of transaction requests and the verification code to be read by Alice from the display of Reader. If Alice deems the transaction requests received by the bank valid, she can enter the verification code in her computer to be forwarded to the bank.

(1) $A \rightarrow C : D_1, S_1$
(2) $C \rightarrow B : D_1, S_1$

(3) $A \rightarrow C : D_2, S_2$
(4) $C \rightarrow B : D_2, S_2$
(5) $B \rightarrow C : D_1, D_2, S_1, S_2, E_K\{\ H(D_1, D_2, S_1, S_2, N), N, V\ \}$
(6) $C \rightarrow R : D_1, D_2, S_1, S_2, E_K\{\ H(D_1, D_2, S_1, S_2, N), N, V\ \}$
(7) $R \rightarrow A : D_1, D_2, S_1, S_2, V$
(8) $A \rightarrow C : V$
(9) $C \rightarrow B : V$

Alice starts by entering the required transaction request information in her computer, which forwards this to the bank (steps 1 to 4). The bank returns a barcode, which contains a message. The barcode is projected on the display of Alice's computer (step 5). The message contains both plain and cipher text. The plain text part consists of destination account numbers (D_1 and D_2) and amounts (S_1 and S_2) of the transaction requests. The cipher text $E_K\{\ H(D_1, D_2, S_1, S_2, N), N, V\ \}$, encrypted with shared secret key K [2], contains a digest (hash) of the transaction requests' critical values concatenated with a nonce ($H(D_1, D_2, S_1, S_2, N)$), the nonce itself (N) and a verification code (V).

Alice uses her Reader to scan the code (step 6). The Reader decrypts the cipher text information (digest, nonce and verification code) and creates a digest using the received plain text information and the nonce. A check is made whether the plain text was modified in-transit using the created and received digests. If both are equal, the plain-text values were not modified. In this example, the values are equal and authentication can therefore continue. If this would not be the case, the Reader would show an error and not allow authentication to proceed.

The Reader shows the transaction requests as received by the bank with a verification code generated by the bank to Alice (step 7). Alice checks whether the transaction requests are as she entered them. If they are, she reads the verification code from the Reader and enters it in her computer (step 8). The verification code is then sent to the bank (step 9).

Where this scenario differs with TTA as applied in Section III is that the bank relies on its own infrastructure to provide integrity of information that must be verified by Alice. The message in steps 5 and 6 only relies on Alice's computer to provide availability. While part of the message is plain text, its integrity is protected by a mandatory check by the Reader using the digest and nonce from the cipher text.

Mallory is again intervening. Her attack is similar to the used approach in Section III.

(1) $A \rightarrow C : D_1, S_1$
$C \rightarrow M : D_1, S_1$
$M \rightarrow C : D'_1$
(2) $C \rightarrow B : D'_1, S_1$
(3) $A \rightarrow C : D_2, S_2$
(4) $C \rightarrow B : D_2, S_2$
(5) $B \rightarrow C : D'_1, D_2, S_1, S_2, E_K\{\ H(D'_1, D_2, S_1, S_2, N), N, V\ \}$
(6) $C \rightarrow R : D'_1, D_2, S_1, S_2, E_K\{\ H(D'_1, D_2, S_1, S_2, N), N, V\ \}$
(7) $R \rightarrow A : D'_1, D_2, S_1, S_2, V$

[2]For this scenario, we assume that symmetric encryption is used and that secret key K is known by both B and R. An alternative would be the use of asymmetric encryption in which B uses a public key and R a private key from the same keypair.

Mallory changes the destination account number of the first transaction from D_1 to D'_1 between steps 1 and 2. It is not possible for Mallory to change D'_1 back to D_1 between steps 5 and 6 without the Reader reporting an error, since she cannot change the cipher text containing the information used to verify the plain text values. Alice can see that D'_1 is received by the bank, which allows her to abort the authentication and transaction requests by not entering the verification code in her computer.

This demonstrates both the strength and the weakness of CVTSA. The customer has the opportunity to check whether the bank received the correct information, unlike with TTA. Unfortunately, nothing stops the customer from skipping this check. The customer can just read the verification code from the display of the Reader and enter it in his or her computer without taking a look at the transaction request information.

Humans cannot be treated as machines. They take actions that may seem irrational, although they are perfectly justifiable from cognitive and social perspectives. With CVTSA, a customer can use validation information without paying any attention to transaction request information. This nullifies the added security of CVTSA and therefore seems irrational, but from cognitive and social perspectives it can make sense because skipping the validation is the shortest and easiest route to what the customer needs to accomplish (conduct payments).

V. Entered single transaction authentication (ESTA)

We noted how TTA methods which depend on the customer's computer for information integrity are flawed in Section III. We also noted in Section IV how CVTSA mitigates this flaw by relying on the attention span of the customer. In this section, we demonstrate how the use of What You Enter is What You Sign (WYEIWYS) protects against session modifying attacks in a similar way to CVTSA without requiring additional effort from the customer.

See Figure 2 on page 3 for an overview. Alice (A) again wants to use her computer (C) to transfer money from her account at her Bank (B) to Bob and Charlie. The bank provided her with a Trusted Entry Pad (TEP) and a smart card (SC). We also assume for this scenario that Alice is already logged in the secure banking environment by previously applying entity authentication. Therefore, the bank already knows that Alice authenticated the session with her smart card.

(1) $A \rightarrow TEP : PIN$
(2) $TEP \rightarrow SC : PIN$

Alice starts the first transaction request. Because this is the first use of the 'Pay' function, her smart card must be unlocked. She inserts her smart card in the TEP, chooses the function 'Pay' and enters her PIN on the device (step 1). The TEP forwards the unlock request with the PIN to the smart card (step 2). The PIN is valid and therefore the required functionality is unlocked.

(3) $A \rightarrow TEP : D_1$
(4) $TEP \rightarrow SC : D_1$

(5) $SC \rightarrow TEP : E_{PrK(SC)}\{\ H(SC, D_1, N_a), N_a\ \}$
(6) $TEP \rightarrow C : D_1, E_{PrK(SC)}\{\ H(SC, D_1, N_a), N_a\ \}$
(7) $C \rightarrow B : D_1, E_{PrK(SC)}\{\ H(SC, D_1, N_a), N_a\ \}$

The first transaction request Alice wants to enter is for Bob. The TEP asks Alice to enter a destination account number for the transaction request (D_1). Alice enters this using the TEP's keypad and reads her entry on the TEP's display while she is typing. Any typographical mistakes can be corrected using the keypad. Alice confirms her entry with a push on the OK button on the device (step 3). After Alice's confirmation, the TEP sends D_1 to the smart card (step 4).

A nonce (N_a) is generated by the smart card. The smart card concatenates its own identifier SC, D_1 (received earlier from the TEP) and N_a. A digest (hash) is computed over the concatenated values, represented by $H(SC, D_1, N_a)$. The smart card encrypts the digest and nonce with its private key $PrK(SC)$ and returns encrypted message $E_{PrK(SC)}\{\ H(SC, D_1, N_a), N_a\ \}$ to the TEP (step 5).[3] The TEP sends both values (the destination account number in plain text and the encrypted message) to Alice's computer (step 6), which forwards both to the bank (step 7).

(8) $A \rightarrow TEP : S_1$
(9) $TEP \rightarrow SC : S_1$
(10) $SC \rightarrow TEP : E_{PrK(SC)}\{\ H(SC, D_1, S_1, N_b), N_b\ \}$
(11) $TEP \rightarrow C : S_1, E_{PrK(SC)}\{\ H(SC, D_1, S_1, N_b), N_b\ \}$
(12) $C \rightarrow B : S_1, E_{PrK(SC)}\{\ H(SC, D_1, S_1, N_b), N_b\ \}$

The TEP asks for the next value, which is the amount of money associated with the transaction (S_1). Use of the TEP by Alice and the communication between TEP and Alice's computer in steps 8 to 12 is similar to steps 3 to 7. Note that the digest of the message is calculated over SC, D_1, S_1 and the new nonce N_b.

The transaction request for Bob is now received by the bank. For Charlie, Alice repeats steps 3 to 12 using Charlie's account number and the amount of money she wants to send to Charlie.

The bank performs an integrity check to determine whether received messages are valid. This is done after each critical transaction request value is received (after steps 7 and 12). Before the integrity check is started, the bank decrypts the signature using public key $PuK(SC)$. The bank knows which public key is appropriate based on earlier performed entity authentication, which identified the smart card.

For the integrity check, the bank starts by computing a digest. When a destination account number is received, the digest is based on the known identifier SC, the received D_1 and nonce N_a from step 7. When an amount is received, the computed digest is based on the known identifier SC, the previously received (step 7) D_1 and the received (step 12) plain text S_1 and nonce N_b. This binds the amount of the transaction to the destination account number and ensures that the messages from steps 7 and 12 cannot be used independently. The message is valid if the digest of the received message is equal to the digest that the bank computed.

Mallory (M) is again an attacking party and has full control over Alice's computer. She can see that Alice is transferring money to bank accounts of Bob an Charlie since D_1, S_1, D_2 and S_2 are forwarded as plain text by Alice's computer from TEP to bank.

(6) $TEP \rightarrow C : D_1, E_{PrK(SC)}\{\ H(SC, D_1, N_a), N_a\ \}$
$C \rightarrow M : D_1, E_{PrK(SC)}\{\ H(SC, D_1, N_a), N_a\ \}$
$M \rightarrow C : D_1'$
(7) $C \rightarrow B : D_1', E_{PrK(SC)}\{\ H(SC, D_1, N_a), N_a\ \}$

It is possible for Mallory to change each plain text value before it is sent to the bank (in this example, D_1 to D_1' between steps 6 and 7). This attack fails when the bank decrypts the encrypted message and notices that the digest does not match the received input. Alice is not allowed to continue.

(11) $TEP \rightarrow C : S_1, E_{PrK(SC)}\{\ H(SC, D_1, S_1, N_b), N_b\ \}$
$C \rightarrow M : S_1, E_{PrK(SC)}\{\ H(SC, D_1, S_1, N_b), N_b\ \}$
$M \rightarrow C : S_1'$
(12) $C \rightarrow B : S_1', E_{PrK(SC)}\{\ H(SC, D_1, S_1, N_b), N_b\ \}$

If Mallory would only change the amount instead (e.g. she works together with Bob to get more money than Alice intents to give), then the intervention would look as shown between steps 11 and 12. Similar to the previous attack, the bank notices that the signature does not match the received value S_1' and will not allow the transaction to continue.

To summarize, applying ESTA protects against man-in-the-middle attacks that modify critical transaction request information in a customer's session. This is similar to the added information integrity of CVTSA when compared to TTA, but differs in that the use of WYEIWYS does not introduce a dependency on the customer to perform the required validation, unlike WYSIWYS as applied by CVTSA.

VI. Related work

Several characteristics of the TEP are already represented in existing authentication methods. We note several examples and how they relate to the TEP.

A. Devices that apply keyboard emulation

Keyboard emulation can be used to transfer information from one electronic device to another by applying a hardware interface and a protocol used by keyboards (e.g. PS/2 or USB HID). The receiving device does not require changes to hardware or device drivers to facilitate the communication. An example of an existing entity authentication device which utilizes this is Yubico's YubiKey [4].

We do not specify the interface between TEP and customer computer (see Figure 2 on page 3, step 2) in this paper. Since communication is modeled as unidirectional from TEP to the customer's computer, keyboard emulation can provide a software independent bridge between these devices. In this case, the implementation of client-side device drivers is unnecessary, which is beneficial for both banks (lower implementation costs) and customers (less installation time).

[3] We apply asymmetric encryption in our example for information integrity and non-repudiation. For confidentiality, the data can in addition be encrypted with a public key from B or with a symmetric key K shared by B and SC. Alternatively, only symmetric encryption could be used to get confidentiality and integrity, but non-repudiation would be lost.

	Transaction authentication				Requires client-side software
	Type	Secure entry	Verification by user	Data flow	
Nordea	TTA	×	×	↔	✓
ABN-AMRO	CVTSA	×	✓	↔	✓
FINREAD	CVTSA	×[a]	✓[b]	↔	✓
Radboud	CVTSA	×	✓	↔	✓
Rabobank	CVTSA	×	✓	←	×
TEP	ESTA	✓	×	→	×

TABLE I
COMPARISON OF INTERACTIVE SMART CARD TERMINALS.

[a] [b] Based on default use scenarios.

B. Interactive smart card terminals

This category includes devices which apply their own user interface (keypad and display) while connected to a computer and that depend on a smart card for cryptographic functions. An overview of the discussed devices, their similarities and their differences is given in Table I. We note two banks from our former work that apply this to authenticate their customers [1]. Nordea Bank (Nordic countries) allows its corporate customers to connect their card reader to a client computer using USB.[4] The client can use the card reader after software is installed (provided by the bank). Before messages can be signed by the smart card, its functionality must be unlocked by entering a PIN on the reader. Information for the customer to sign is shown on his or her computer. ABN-AMRO (the Netherlands) is a bank that uses a similar card reader, named the E.Dentifier2. This device differs from Nordea's card reader by showing information to sign on the device itself instead of on the customer's computer. An older version of the E.Dentifier2 authentication device had a notable security flaw, which was fixed in a later version [5].

The FINREAD Card Reader is a bank independent example [6]. It also features a smart card reader and a connection to a customer's computer. A difference with the previous examples is that the reader itself also hosts cryptographic credentials and functions together with user-installable applications from different providers. Communication between reader and provider is secured in terms of confidentiality and integrity. One recognized weakness of FINREAD Card Reader is the high cost required to produce the device [7, 8]. The Radboud Reader is another interactive smart card terminal [9], which by itself is less complex compared to FINREAD. Complex functionality and control is instead moved to the smart card.

Rabobank (the Netherlands) announced a new authentication device with the name Rabo Scanner.[5] It allows one-way communication without the installation of additional software by displaying a color code on the customer's computer, which is scanned using a camera on the Rabo Scanner. The color code contains the information to be verified, which is shown on the display of the Rabo Scanner after it is scanned. For confirmation, the customer reads a verification code from the device and enters it in the bank's site on his or her computer.

The TEP has a very minimalistic approach compared to the other discussed authentication methods while still providing the ability to verify and sign transaction requests separately from the customer's computer. It is not required for customers to install client-side software. Unlike the CVTSA examples, it is also not required for the customer to verify information received by the bank. Instead, the process of protecting the integrity of information is initiated by the customer and completed by the bank, which does not require a round trip of the same information for verification.

C. Mobile devices used for out-of-band verification

A customer owned smartphone and its connection to the Internet can be used as an out-of-band channel to allow a customer to verify information from a banking session with a desktop computer.[6] This can apply WYSIWYS if information to sign can be interpreted in a single semantic context. Drawbacks include the previously mentioned limitation of WYSIWYS (customers must perform additional actions adequately for effective security), but also that the customer's smartphone is an untrusted device that is vulnerable to malware [10, 11].

ESTA can potentially be implemented on a smartphone, with the caveat that the platform is not as trustworthy as a bank provided device. While mobile applications can be hardened against malware and other software-based threats[7], malware threats cannot be prevented in an untrusted environment.

D. Authentication solutions in trusted execution environments

A trusted execution environment (TEE) is a collection of resources and controls of those resources that are physically or logically separated from other resources on the same device [12]. Such resources can include volatile memory space, persistent storage space, CPU cycles, security functions and different types of in- and output interfaces. Other resources outside the TEE cannot interact with any of the resources that compose the TEE unless explicitly permitted by the TEE. A TEE can attest its identity and allows authorized remote parties to interact with applications within the secure environment.

The principles of TEEs potentially allow the integration of the TEP's functions into a customer's computer if it has a TEE. Requirements of a TEE to host a TEP are that the user knows whether he or she works within the normal or the trusted environment and that in- and output interfaces are secure against injection attacks.

VII. FURTHER RESEARCH

We give a high-level description of a security protocol in Section V as an example of ESTA's principles. This protocol has not been formally tested against attacks. It can be vulnerable to existing attacks that were not considered and new

[4]About Nordea's card reader: http://www.nordea.com/Our+services/International+products+and+services/Corporate+Netbank/Nordea+card+reader/1079602.html

[5]Rabobank introduces the Rabo Scanner (Dutch): http://www.rabobank.nl/particulieren/servicemenu/nieuws/rabobank_nieuws/rabobank_introduceert_de_rabo_scanner

[6]An example of a software product which provides this is Entersekt's online banking authentication: http://www.entersekt.com/

[7]An example of a framework which allows mobile application hardening is Versafe's MobileSafe: http://www.versafe-login.com/?q=mobilesafe

types of attacks. A further worked out technical definition of the scheme and formal verification can be used to determine whether the protocol is secure.

While ESTA offers information integrity of transaction requests between customer and bank by applying WYEIWYS, it does not reduce the effectiveness of social engineering attacks on the customer. Further research can answer what would be required to protect against social engineering (e.g. phishing attacks) in addition to the use of ESTA. It might be possible that unambiguous labels for functions (e.g. 'Login' and 'Pay') and requested information (e.g. 'Destination account number' and 'Amount of money') increases the awareness of customers. This is something that can be tested in addition to the behavior of customers when warnings are (repeatedly) observed (e.g. 'The use of this function WILL cost you money').

The TEP is introduced as a technical concept to reduce the effectiveness of malware attacks on banking customers' computers. Injection attacks which add or replace financial transactions can be detected by protecting the integrity of the information flow between customer and bank. While we kept the question about whether banking customers can use this in the back of our minds, the technical concept has not been tested for usability. It is possible that changes have to be made to make WYEIWYS acceptable in everyday use. The use of two input devices (a TEP for the entry of simple but critical values and a regular keyboard for the entry of non-critical and possibly more complex values) might confuse customers.

There are different possible approaches to implement the unidirectional communication between TEP and the customer's computer. Comparing approaches and their (dis)advantages would have to take the possible interfaces of customer devices and their prerequisites for use into account.

Information to be entered in the TEP by the customer itself can also present a challenge if the information next to digits and a decimal separation character also contains letters. A full destination account number can contain letters if it is an International Bank Acccount Number (IBAN) [13]. A common keypad might prove too cumbersome to enter letters. A full keyboard can also present challenges regarding usability on a small form factor. User input methods can be examined for the best fit between user entry and critical information to enter. Also, the possibility of an increase in insecure user behavior through typographical errors on an external device (ETSA) can be compared against the possibility of insecure user behavior when the user is required to compare values (CVTSA).

VIII. Concluding remarks

In this paper we introduced Entered Single Transaction Authentication using the Trusted Entry Pad, which applies the new concept of What You Enter Is What You Sign to verify customer entered data without the need for an extra verification step by the customer. It has a smaller margin for customer errors compared to What You See Is What You Sign-based approaches while still being independent from the customer's computer for information integrity, unlike traditional transaction authentication methods.

References

[1] Sven Kiljan, Koen Simoens, Danny De Cock, Marko van Eekelen, and Harald Vranken. Technical report: Security of Online Banking Systems, February 2014. URL http://portal.ou.nl/documents/114964/523334/TR-OU-INF-2014-01.pdf.

[2] Tim Redhead and Dean Povey. The Problems With Secure On-line Banking. *Proceedings of the XVIIth annual South East Asia Regional Confereance (SEARCC98)*, 1998.

[3] Mohammed AlZomai, Bander AlFayyadh, Audun Jøsang, and Adrian McCullagh. An exprimental investigation of the usability of transaction authorization in online bank security systems. In *Proceedings of the sixth Australasian conference on Information security-Volume 81*, pages 65–73. Australian Computer Society, Inc., 2008.

[4] Robert Künnemann and Graham Steel. YubiSecure? Formal security analysis results for the YubiKey and YubiHSM. In *Security and Trust Management*, pages 257–272. Springer, 2013.

[5] Arjan Blom, Gerhard de Koning Gans, Erik Poll, Joeri De Ruiter, and Roel Verdult. Designed to fail: A USB-connected reader for online banking. In *Secure IT Systems*, pages 1–16. Springer, 2012.

[6] Alain Hiltgen, Thorsten Kramp, and Thomas Weigold. Secure internet banking authentication. *Security & Privacy, IEEE*, 4(2):21–29, 2006.

[7] Ton Slewe and Mark Hoogenboom. Who will rob you on the digital highway? *Communications of the ACM*, 47(5):56–60, 2004.

[8] A Hisamatsu, D Pishva, and GGD Nishantha. Online banking and modern approaches toward its enhanced security. In *The 12th International Conference on Advanced Communication Technology (ICACT 2010)*, volume 2, pages 1459–1463. IEEE, 2010.

[9] Erik Poll and Joeri de Ruiter. The Radboud Reader: a minimal trusted smartcard reader for securing online transactions. In *Policies and Research in Identity Management*, pages 107–120. Springer, 2013.

[10] George Lawton. Is it finally time to worry about mobile malware? *Computer*, 41(5):12–14, 2008.

[11] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, pages 3–14. ACM, 2011.

[12] Amit Vasudevan, Emmanuel Owusu, Zongwei Zhou, James Newsome, and Jonathan M McCune. Trustworthy Execution on Mobile Devices: What security properties can my mobile platform give me? In *Trust and Trustworthy Computing*, pages 159–178. Springer, 2012.

[13] European Committee for Banking Standards. IBAN: Standard Implementation Guidelines (SIG203 V3.2), August 2003.