Boosting usability for Protecting Online Banking Applications Against APTs

Mohannad Alhanahnah
School of Computing
University of Kent
Canterbury, UK
Email: ma481@kent.ac.uk

David W Chadwick
School of Computing
University of Kent
Canterbury, UK
Email: D.W.Chadwick@kent.ac.uk

***Abstract*—With the advent of Advanced Persistent Threats (APTs) and exploits such as Eurograbber, we can no longer trust the users PC or mobile phone to be honest in their transactions with banks. This paper reviews the current state of the art in protecting PCs from malware and APTs that can modify banking transactions, and identifies their strengths and weaknesses. It then proposes an enhanced USB device based on speech and vision. User trials with a software prototype show that such a device is both user friendly and that users are less susceptible to accepting subtly modified transaction with this device than with other vision only USB devices. Since human factors are usually the weakest point in the security chain, and are often the way that APT actors perform their attacks, the focus of the proposed solution is on improving the usability of existing USB devices. However the device is still not failsafe, and therefore may not be as preferable as Sm@rt TAN-plus that is currently used by many German banks.**

***Keywords*-Advanced Persistent Threats, Banking Transactions; Transaction Authentication Number; usability;**

I. Introduction

The primary motivation for non-political cyber attacks is to gain money. In March/April 2016 cyber crime was the source of more than 70% of cyber attacks [21]. Furthermore, the goals for these targeted attacks are not only to steal credit card and bank information, but also to alter online transactions [19]. In 2009, more than half a million dollars was stolen from Patco Constructions online bank account, because there was no transaction verification of the online transactions. The company successfully sued the bank for having commercially unreasonable security procedures [1]. Banks are therefore responsible for providing transaction protection mechanisms. In 2012 the Eurograbber exploit stole more than 36 million Euros from 30,000 customers of multiple European banks [13], even though these banks were using an Out-Of-Band (OOB) transaction authentication method. They were sending the transaction details and a Transaction Authentication Number (TAN) via SMS to the customer's mobile device so that the customer could confirm her online transaction by entering the TAN into her PCs online banking session. Unbeknown to the customer, the Zeus Trojan [9] had previously infected her PCs browser and had asked her for details of her mobile device under the pretense of installing new security software on it. This allowed the attackers to install malware instead on the mobile, so that when Zeus injected false transactions into the web banking session, the malware could intercept the TAN in the mobile, and relay it back to Zeus to confirm the transaction.

Consequently the customer was not aware that fraudulent transactions were taking place in the background. What these examples show is that we can no longer trust any of the banking transactions that a bank receives from a PCs banking application to be genuine, nor can we trust mobile devices that allow executables to be downloaded onto them. The sophistication of todays malware means that attackers can effectively take control of the PC and either modify transactions or inject new transactions almost at will. We cannot be sure which components, such as the operating system, firmware, or application software, have been infected. Furthermore, Advanced Persistent Threats (APTs), in which the malware is controlled by a remote human controller, is sent encrypted, and can be modified and dynamically tailored to suit the victims PC, means that antivirus software will always be behind the curve in detecting their presence and effectively removing them. We must therefore look for an alternative means of protecting online banking transactions sent from untrustworthy devices.

We consider Eurograbber to be an APT because Eurograbber 1) is a targeted attack, which targets only banking users 2) controllers aim to keep a foothold on the victims machines 3) is a sophisticated attack, not only does it infect victims PCs but also their mobile devices 4) controllers utilized phishing attacks to initially infect victims machines.

The rest of this paper is structured as follows. Section II reviews the current state of the art and identifies the strengths and weaknesses of the existing mechanisms. Section III discusses the requirements for an ideal solution and describes our proposed mechanism. Section IV describes the software implementation of a prototype of the new proposed mechanism. Section V describes the user trials with the software prototype, whilst section VI concludes the article.

II. State of the Art

Oppliger et al. [20] identify three different types of attack against banking applications, which they term: credential

stealing, channel breaking and content manipulation. The latter two are also known by their more populist terms as: Man in the Middle (MitM) and Man in the Browser (MitB) attacks. The Zeus Trojan used in the Eurograbber is a MitB attack, since it injects malicious code into the browser that is activated when the user starts an online banking session. Shujun Li et al. [17] also identify a fourth category that they term Man in the Computer (MitC), which is best exemplified by an APT that takes full control of a users PC. This is more pernicious, since it can control all the input and output channels including the keyboard, screen, network, filestore etc., meaning that nothing in the PC can be trusted.

Mitigating solutions have typically comprised either multi-factor user authentication or transaction authentication. The former is designed to combat credential stealing, so that even if the users password is stolen, the attacker still does not have access to the other authentication factor(s). Many different form factors have been employed including mobile phones, time based or challenge/response based OTP fobs, smart cards, and paper based grids. Some banks are still using a second set of security questions and so are doubling up on something you know. Nevertheless, however good multi-factor user authentication schemes are, they don not address the real problem, which is to stop fraudulent transactions. Transaction authentication schemes on the other hand are designed to do this by having a second trusted device for the user to either enter or confirm the veracity of a transaction. The other device, which also acts as a second factor in user authentication (something you have), could be a home telephone, mobile phone or bespoke equipment provided by the bank. A number of different schemes have been deployed by banks or proposed in the literature and these are critically reviewed below. They either rely on secure hardware for sending the transaction, or an OOB channel for confirming it.

One of the first OOB schemes that the banks implemented was to look for unusual or suspicious transactions submitted to a customers account, and then to telephone the customer (usually at home) asking them to confirm that it was genuine. Unfortunately this has been successfully attacked in a couple of ways, both of which involve the fraudsters hosting a call reception center to take the banks calls. In the first variant, the fraudster phones the customer at home, pretending to be from the telephone company, and gets the customer to reveal their sensitive account information. The fraudsters then use this to masquerade as the customer and to ask the telephone company to transfer incoming calls to their call center whilst the fraudulent transactions are being carried out. They then transfer calls back again once the transactions have been verified. A more recent variant of this has modified the Zeus Trojan to automatically collect telephone account information from customers when they are on their PCs [16]. Consequently we find this scheme to be vulnerable to attack.

Another popular OOB scheme used by many banks, is m-TAN, in which the bank sends an SMS to the users mobile phone. This contains details of the transaction and a TAN which the user must return to the bank via his PC banking session. As previously noted, this has been successfully exploited in Eurograbber [13]. But more worryingly, AlZomai et al. [7] found that even if the mobile device was not infected, obviously modified transactions, in which the transaction account details in the SMS differed significantly to those entered into the PC, were not spotted in 21% of cases in their user trials, whilst subtly modified transactions, in which just one digit of an account number was modified, were not spotted in 61% of cases. Consequently we find this scheme to be both vulnerable to attack, and lacking in usability.

Due to the inherent weaknesses in m-TAN, many German banks are now moving towards the Sm@rt TAN-plus system (e.g. [2], [3]). This is a handheld device with a keypad, a display and a smart card reader. The user inserts their ATM chip and PIN card into the device, reenters (some subset of) the transaction details (that they have previously entered into their web banking session) and their smart card computes the TAN which the user must then copy back to the web session. If the transaction received by the bank does not match the TAN, then the bank knows that it is different from the one input to the Sm@rt TAN-plus device, and can abort it. The advantage of this system is the low cost of the device to the bank as the cryptography computations are done by the existing smart card. The disadvantage is in terms of its usability, since users must enter (some subset of) the transaction details twice, and then copy the TAN back to their PC. An optical variant of this, Sm@rt-TAN optic, addresses the first usability issue, as it reads the transaction details directly from rapidly flickering images on the PC screen, using 5 optical sensors in the back of the device. However this is not recommended for epilepsy suffers as the flickering images could cause seizures.

An earlier transaction verification scheme designed specifically with security in mind, is the 10 year old FINREAD CEN standard [19], [11]. FINREAD devices are connected to the PC and have an inbuilt keypad, smart card reader and display. They will only execute digitally signed Finlets downloaded from the PC. However, few banks, if any, have adopted this PKI based system, as the end user devices cost over €200 each [4]Consequently we find this scheme to be prohibitively expensive.

Matthew Johnson designed a USB device for authenticating transactions, which he called the dongle [12]. For usability, the dongle simply consists of a display and two buttons, one to accept the transaction, the other to reject it. Like most previous mechanisms, the dongle works in synchronous mode, meaning that the user submits his transaction via his PC, then confirms or denies it when it is displayed on the device, before he can submit another transaction. Cost was taken into account by using symmetric cryptography

(AES recommended) instead of PKI, with long term secret keys shared between the dongle and the bank. Session keys are dynamically generated for encrypting the transactions. MACs are used for message integrity. A software prototype was built to simulate the dongles functionality, but no user trials were carried out. Adham et al. proposed an almost identical system in [5], but with fewer worked out details. This scheme suffers from the same usability problems as m-TAN.

hPIN/hTAN [17] is similarly designed with cost and usability in mind, but this is a mechanism for securely sending a transaction to the bank, rather than an OOB confirmation of it. This USB device only uses HMAC as the core cryptographic method and requires neither a second trusted channel nor a secure keypad nor a trusted third party nor encryption, so it is very cheap to manufacture, the estimated cost of this USB device €(3-5). The device comprises a display and a single OK button. hPIN is used to authenticate the user and hTAN to authenticate each transaction. After inserting the device into the PC, the user presses OK, enters her userID into the PC which is relayed to the device, whereupon the device displays the digits 0 to 9 and beneath each digit a randomly generated code (letter or digit) that the user must substitute when entering her PIN into the PC. The code characters are relayed back to the device allowing it to authenticate the user.

To verify a transaction, the sensitive information is entered to the PC but displayed on the device, whilst the PC simply shows asterisks. This is to ensure that the user cannot misread the transaction information (as is the case with m-TAN). After each item of sensitive information is entered, such as account number or amount, the user must press OK on the device, wait a few seconds, then press OK again. The device then sends the transaction information to the banking server along with a HMAC hash. In user trials with 20 students, 91% managed to successfully login, and the mean usability score was 3.65 on a 5 point Likert scale. Consequently this scheme is still not ubiquitously user friendly.

IBMs Zone Trusted Information Channel (ZTIC) [23] is another USB device for securely sending transactions to the bank. It comprises a display, two buttons and optionally a smart card reader for authenticating the user. ZTIC acts as a secure pass through proxy between the PCs web browser and the bank and sets up a TLS connection between itself and the banking server. It intercepts all banking transactions between the web browser and itself and displays the vital information to the user, allowing her to press either the confirm or abort buttons. In this way the user can see the exact details of each transaction before it is relayed to the bank. Only a careless user would allow modified transactions to pass, but as the research in [7] shows, this is more common than one would hope for, and therefore it is still as lacking in usability as m-TAN.

III. Analysis and Design

Analysis of the above mechanisms shows that transaction authentication, as opposed to only multi factor user authentication, is essential. To be secure against MitM, MitB and MitC requires a separate hardware device that is immune to being infected with viruses or APTs, otherwise it will be susceptible to attacks such as Eurograbber. This effectively means that the device should not be capable of accepting software downloads, which rules out mobile phones. One might concede that, from a security point of view, accepting only digitally signed downloads, for example as per the FINREAD specification is also acceptable. But this adds considerable complexity to the device, making it no longer a cost effective solution. At over €200 per FINREAD device, the banks have already decided it is more cost effective to use an alternative cheaper device that cannot accept software downloads, for example, the Sm@rt TAN-plus device that retails for €15 [3].

Even this device might be considered expensive compared to known transactions losses. Although Eurograbber stole €36 million, the population of the Eurozone countries was 332 million in 2011. If only 10% of these perform online banking, which is a conservative estimate[1], this still only represents a loss of €1 per online banking customer. Consequently, any mechanism we propose should not retail for more than Sm@rt TAN-plus, should be at least as secure as Sm@rt TAN-plus and better than it on usability grounds, if it is to have any chance of being accepted in the market place. (We use Sm@rt TAN-plus as the benchmark here since it is already used by many banks).

Several state of the art designs are based on USB devices [17], [12], [23]. These have the advantages over Sm@rt TAN-plus of not needing replacement batteries, and are much more compact to carry around. Depending upon the design, they may be easier or more difficult to use than Sm@rt TAN-plus. Whilst hPIN/hTAN wins on cost, at €3-5 for the components [23], in our opinion it fails on usability as 1 in 10 users failed to login and the mean usability score was not high at only 3.65/5.0. Whilst ZTIC is better than Sm@rt TAN-plus for usability (users only have to visually compare transaction details and press the confirm button once vs. enter (part of) the transaction again, compare it visually and enter a TAN to the PC), it fails on cost. Swiss bank UBS are retailing ZTIC to their customers at €50 [15]. Admittedly this does contain a smart card reader to authenticate the user, but then so does Sm@rt TAN-plus. The remaining USB contender is Matthew Johnsons dongle, which we asses to be more usable than Sm@rt TAN-plus (since it has the same usability as ZTIC) and it should be able to retail for

[1]Over a third of Internet users accessed online banking sites in Europe in 2012 (http://www.comscoredatamine.com/2012/06/1-in-4-internet-users-access-banking-sites-globally/) whilst over two thirds of Europeans use the Internet (http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013.pdf)

about the same price as Sm@rt TAN-plus. Whilst Johnson does not actually build his dongle, he reasons that USB Internet phones are comparable in complexity to the dongle and these typically retail from €13-39. The main difference between the USB devices and Sm@rt TAN-plus, is that if the user makes a mistake when accepting the transaction, Sm@rt TAN-plus will abort the transaction, whereas the USB devices will accept it. This is because if the user inputs wrong information to the Sm@rt TAN-plus device, the TAN will not match the received transaction causing the bank to abort it; whilst if a user makes a mistake matching the transaction displayed on the USB device with the one he intends to submit, and presses the accept button, then the transaction displayed on the device will be accepted by the bank. Sm@rt TAN-plus is therefore failsafe, whilst the USB devices are not. Our design builds on Matthew Johnsons dongle with the objective of reducing the incidence of users who mistakenly accept modified transactions, thereby endeavoring to make the device more failsafe.

We know from [7] that users failed to spot 21% of significantly modified transactions and 61% of subtly modified ones. This user error could be for several reasons. Firstly people see what they expect to see [24], meaning that if they expect a transaction to show xyz they will tend to see xyz even if it did not show exactly that. Secondly many users lack security awareness. We also know from a survey conducted by B2B International and Kaspersky Lab [14],that more than half of the 8,605 respondents from different regions use no security on their Android phones. Furthermore, in a recent large scale study of users clicking through security warnings in browsers, a tenth clicked through Mozilla Firefoxs malware and phishing warnings, a quarter through Google Chromes malware and phishing warnings, and a third through Mozilla Firefoxs SSL warnings [6]. We therefore need a mechanism where security is built in and cannot be switched off, and can jolt people sufficiently so that they do not see what they expect to see, but rather what is there. They need to be made more aware when a transaction is not what they were expecting it to be, so that they do not click through it.

It is well known that we learn best if we stimulate several senses at once [18]. We can significantly increase a users awareness by using multiple modes of communication. Speech is a fundamental means of human communication. Cognitive research and tests show that verbal communication can sometimes be the greatest means for transferring information, and the end-user does not need to pay as strict attention in contrast to vision communication [22]. [22], [10] describe situations where using audio is better than using a display, such as when the message is simple, short, will not be referred to later, or calls for immediate action. All of these situations apply to the proposed device. The audio message is simple and short since it contains only the beneficiary IBAN and the amount of money to be transferred, and the user should take immediate action after hearing the audio message. By combining voice with a digital display in the USB device, this should be able to better communicate to a user that a transaction is or is not the same as they expected. Since a programmable voice chip costs approximately 50 US cents to buy in bulk then it will not add significantly to the overall cost of the USB device.

The modified design of the USB device is to have three buttons instead of two, with the extra button being to replay the speech of the transaction in case the user mishears it first time. When the user connects the USB device to the PC, it autonomously establishes a secure connection with the banking server (independently of the web browser) and polls the server to see if there are any pending transactions. If there are it downloads the first one, displays the details (account number and amount) and simultaneously speaks it out. If the user is happy with the details he presses the accept button, otherwise the decline button. If another transaction is pending, this will then be downloaded. Our design allows the user to submit as many transactions as he wishes to the banking site, independently of the USB device. However, each transaction is placed in the pending state until the user either confirms or aborts the transaction via the USB device. This is an improvement over all previous schemes that have been synchronous in nature. Our design allows the user to submit transactions even if he does not have his USB device with him. He can subsequently confirm them when he does have his device.

IV. The Implementation

Similar to Matthew Johnson, we only had the resources to implement a software prototype device. But since the main objective of the research is to determine the users increased ability in spotting subtly modified transactions, we decided that comparing Johnsons software prototype with ours would be sufficient to do this. We built a dummy banking web site that has a login page followed by two main pages: submit transaction and transaction status. The former contains fields that allow the user to enter a new transaction, whilst the latter contains the list of recent transactions, each marked as either Confirmed, Cancelled or Pending. At the bottom of the page is a Connect USB device button which brings up one of the USB devices in a pop up window (Fig. 1 and Fig. 2).

All the web pages were created using Java Server Pages (JSP). The DB comprises three tables: account, transaction and users. There is a many-to-one relation between the account table and users table, which allows a user to possess several bank accounts.

Two software prototype devices were built as separate pages on the banking web site, one based on Johnsons design and one on our enhanced design (see Figures 1 and 2 respectively). The chosen device displays a pending transaction in a pop up window in the main browser window, to simulate viewing it on a USB device. The devices were

built in the Java language consisting of three parts: the speaking part (optional), the GUI part, and existing Java functions and methods for sending and receiving message and establishing a secure connection. The FreeTTS project [2] was used to accomplish the speech function. However FreeTTS v1.2 does not speak some thousand numbers in the correct way, for example, the string 1500 is spoken as fifteen hundreds instead of one thousand five hundred. We found that by inserting a comma separator between the thousand and hundred digits, it then spoke the number correctly.

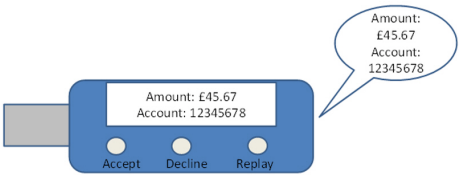

Figure 1: Software prototype for Speech and Vision USB device

Figure 2: Software prototype for Vision only USB device

When the user has submitted his transactions he can view the transaction status and click the Connect USB device button which will POST the users account number to the device page. This page can then fetch the outstanding pending transactions one by one and display them to the user.

V. The User Trials

A. First experiment

10 students were recruited to test the two devices. All were male Masters level students (since the research was performed during the summer vacation, no undergraduate students were available for recruitment). The experiment consisted of two parts. In each part participants submitted five transactions that were given to them on a printed piece of paper. At any time during the part experiment they could check and confirm that the submitted transactions were correctly received by the bank. In the first part they used the vision device, and in the second part the speech and vision device. The participants had the freedom to check and confirm the transactions whenever they wanted to. After completing both parts the participants answered a brief questionnaire.

The objective of the experiment that was disclosed to the participants was that they were to test and comment upon the usability of two alternative USB device designs. In fact, the main objective was to find out which of the devices can best help the participant to detect a subtly modified transaction, because one transaction in each part experiment was intentionally modified by the banking server altering just one digit of one account number.

The questionnaire comprised 6 questions. The first question asked How easy was the vision device to use, with the answer being selected from a 5 point Likert scale ranging from very difficult to very easy. The second question asked for any free form comments about the usability of the vision device. Questions 3 and 4 repeated these for the speech and vision device. Question 5 asked which device the user preferred, whilst question 6 asked why they preferred this device.

70% of participants failed to spot the subtly modified account number with the vision token. This is comparable with the 61% of failures recorded in [7], which had a much larger sample of users (92 participants compared to our 10). Disappointingly, 40% of participants still failed to spot the subtly modified account number with the speech and vision token. Obviously this represents the worst case attack scenario, as an attacker would have to be very lucky or devious to obtain a mule account with just one digit different to the intended account number. However the results do show that adding speech to the USB device has a positive beneficial effect on users being able to detect a subtly modified transaction, but this effect is not large enough to be 100% failsafe. Clearly further experiments and improvements are needed.

80% of participants said they preferred the speech and vision token. The reasons given were because the Speech/Vision token helps them to be more alert to the transaction details, and its easier and more accurate to verify the transactions. The two users who preferred the vision device said that this was better because it was faster than the voice and vision device, and one of them spotted the subtly modified transaction using it (so had no need for the voice confirmation). In terms of usability, both devices were liked by the users, but the voice and vision device scored slightly higher (all users gave it a 4 or 5).

B. Second experiment

We designed the second experiment to attract a large number of users. Consequently it was performed through Amazon Mechanical Turk (AMT), a system that allows users of the Internet to enroll for AMT experiments and be paid small sums of money for successfully completing them. In our experiment, each participant only tested one of the devices and this was assigned to them dynamically when they enrolled. (Even enrollment numbers were assigned one device and odd enrollment num-bers the other device). As before, the participants were allowed to enter the 5 transactions displayed to them, in any order. The banking server

[2]http://freetts.sourceforge.net

Table I: Summary of submissions

	Modify 1 digit	Modify 2 digits	Modify 3 digits	Modify 4 digits	Modify 5 digits	Modify 6 digits	Modify 7 digits	Modify 8 digits	Modify 9 digits	Total
No. of vision token submissions	18	25	26	50	8	20	26	13	6	192
No. of speech token submissions	31	31	32	40	7	12	41	14	13	221
Total submissions	**49**	**56**	**58**	**90**	**15**	**32**	**67**	**27**	**19**	**413**
Correctly spotted vision	2	3	2	12	0	1	3	0	0	23
Correctly spotted speech	1	13	14	18	2	1	17	3	5	74
% spotted using vision	11%	12%	8%	24%	0%	5%	12%	0%	0%	12%
% spotted using speech	3%	42%	44%	45%	29%	8%	41%	21%	38%	33%

was configured to replace the first occurrence of the number 1 with the number 7 (or replace 2 with 3 if there was no occurrence of 1), in just one account number of the five. This was done for the first hundred participants of each device. Then the server randomly changed 2 account digits of one transaction for the next 100 users etc. until the account number of one transaction was completely changed (or until a lower number of changes were spotted by everyone). In this way we should have been able to measure at what point all users noticed that an account number had been changed, for both a vision device and a speech and vision device. This should indicate at what point the devices become failsafe (if at all).

We received 2249 submissions, but after checking the data, only 413 (18%) were deemed to be valid. Many participants did not complete the experiment as requested in the instructions. They either did not submit all 5 transactions, or submitted one or more incorrect transactions (either a wrong account number or monetary value). Many other participants denied all 5 transactions, These speedies presumably could not be bothered to verify the transactions and were only interested in finishing the experiment as quickly as possible in order to be paid. These submissions were reject-ed, and some of these participants complained to us afterwards when they had not been paid. The remaining 413 submissions input 5 correct transactions and either accepted them all, or denied, at most, one of them . It is highly likely that a number of these participants accepted all 5 transactions without checking them first, but it is not possible for us to differentiate between these speedies and a conscientious participant who missed spotting one or more incorrect digits. The results indicate there were a significant number of these speedies, due to the low spotting rate.

Table I shows the distribution of the submissions. The total number of valid sub-missions using the speech and vision device was 221, whilst using the vision only device was 192. The valid submissions were divided into 9 groups based on the number of modified digits in the corrupted transaction. Figure 3 shows the percentage of participants who discovered the transaction manipulation in each group. The percentage of participants who spotted the manipulation using the speech and vision device was higher for all groups except the first one. The manipulation discovery across all groups was 12% for the vision only device and 33% for the speech and vision device.

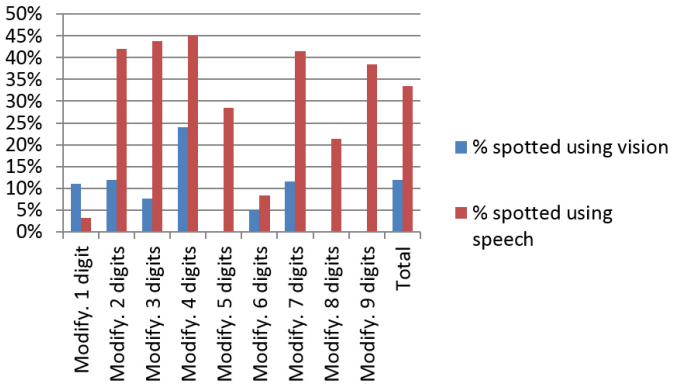

Figure 3: Spotting out the modifications in each group

We additionally conducted a hypothesis analysis to validate our assumption that the speech and vision device improves the users ability to spot a modified transaction. This statistical analysis approach aims to prove or disapprove a hypothesis [8]. This can be accomplished by computing the P-value, which represents the lowest level of significance at which the observed value of a test statistic is significant. This value is compared to the identified significance level, which reflects the probability that observed results occurred due to chance or error. Finally, null hypothesis is rejected if the P-value is lower than the significance level.

Hence, in our analysis the null hypothesis is *modification discovery using the speech and vision device is the same as modification discovery using the vision only device*. After performing the computation, the P-value was 0.0153, where the significance level was 0.05 (i.e. 5%). Hence, we can disapprove the null hypothesis. In other words, this proves the speech and vision device improves the users ability to spot a manipulations in transaction details.

VI. Conclusion

Human factors are often the entry point for malicious attacks, and are considered the weakest point in the security chain. Moreover, according to our analysis of APT attacks (including the Eurograbber incident), APT actors usually infect victims' devices, and keep a foothold there, by exploiting human factors, using phishing, social engineering and human frailty attacks. This inspired us to focus on designing a usable device that can boost the users ability to recognize abnormal transactions.

Accordingly, we conducted two experiments for validating the usability of the proposed solution. The 1st experiment was a controlled one with only a few participants, whilst the 2nd experiment was uncontrolled but with a large number of participants. Both experiments showed that the addition of speech improved the users recognition of manipulated banking transactions. However, both experiments suffered from weaknesses and neither showed the improvement we expected. Consequently, we recommend performing a large

scale experiment in a controlled environment to validating the effectiveness of our approach based on different scenarios.

We conclude that designing a cost effective, secure, user friendly transaction authentication device is a very difficult task. We think that German banks have made a good choice in opting for the existing Sm@rt TAN-plus device. Whilst this is not the most user friendly of devices, nevertheless it is cheap to produce and it effectively aborts modified transactions. One downside is that it also aborts transactions in which the user has made a mistake e.g. by mistyping the transaction number into the device or the TAN into the PC, but this is a failsafe route which does minimum harm to the user. The speech and vision device that we have designed is easier to use, should be comparable in price to produce, but still suffers from the problem that users do not always spot subtly modified transactions, and this could lead to their bank accounts either being fraudulently debited or money going mistakenly to the wrong recipient.

ACKNOWLEDGMENT

This project was funded by $(ISC)^2$ Foundation graduate scholarship.

REFERENCES

[1] http://www.bankinfosecurity.co.uk/interviews/patco-owner-on-fraud-settlement-i-1726. Accessed: 2016-05-31.

[2] www.raiba-aburg.de/smartTan/index.cfm. Accessed: 2016-05-31.

[3] http://www.gallinat.de/fragen-antworten/smrt-tan-plus-mobiletan/. Accessed: 2016-06-02.

[4] http://www.motechno.com/cardman-trust-finrea.0.html. Accessed: 2016-05-31.

[5] Manal Adham, Amir Azodi, Yvo Desmedt, and Ioannis Karaolis. How to Attack Two-Factor Authentication Internet Banking. pages 322–328. 2013.

[6] Devdatta Akhawe and Adrienne Porter Felt. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 257–272, Washington, D.C., 2013. USENIX.

[7] Mohammed Alzomai, Bander Alfayyadh, Audun Jøsang, and Adrian Mccullagh. An Experimental Investigation of the Usability of Transaction Authorization in Online Bank Security Systems. In *Sixth Australasian Information Security Conference (AISC 2008)*, pages 65–73, 2008.

[8] David R. Anderson, Kenneth P. Burnham, and William L. Thompson. Null hypothesis testing: problems, prevalence, and an alternative. *Journal of Wildlife Management*, 64(4):912–923, October 2000.

[9] N. Etaher, G. R. S. Weir, and M. Alazab. From zeus to zitmo: Trends in banking malware. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 1386–1391, Aug 2015.

[10] Christine Faulkner. *The Essence of Human-Computer Interaction*. Prentice Hall, 1997.

[11] A. Hiltgen, T. Kramp, and T. Weigold. Secure internet banking authentication. *IEEE Security Privacy*, 4(2):21–29, March 2006.

[12] Matthew Johnson. A new approach to Internet banking. Technical Report UCAM-CL-TR-731, University of Cambridge, Computer Laboratory, September 2008.

[13] Eran Kalige and Darrell Burkey. A Case Study of Eurograbber: How 36 Million Euros was Stolen via Malware. Technical report, 2012.

[14] Kaspersky Lab. Careless majority more than half use no security software on their Android-based devices. http://www.kaspersky.com/about/news/press/2013/Careless_majority_more_than_half_use_no_security_software_on_their_Android_based_devices, 2013. Accessed: 2016-05-31.

[15] Jeremy Kirk. UBS to Deploy IBM Secure Banking USB Device. www.pcworld.com/article/189938/article.html, 2010. Accessed: 2016-05-31.

[16] John Leyden. New Trojan routes your bank's calls to CROOKS. www.theregister.co.uk/2012/02/02/ice_ix_trojan_social_engineering_trickery/, 2012. Accessed: 2016-05-31.

[17] Shujun Li, Ahmad-Reza Sadeghi, Sören Heisrath, Roland Schmitz, and Junaid Jameel Ahmad. hpin/htan: A lightweight and low-cost e-banking solution against untrusted computers. In *Proceedings of the 15th International Conference on Financial Cryptography and Data Security*, FC'11, pages 235–249, Berlin, Heidelberg, 2012. Springer-Verlag.

[18] John Medina. *Brain Rules: 12 Principles for Surviving and Thriving at Work, Home, and School*. Prentice Hall, 2008.

[19] Roger Meyer. Secure Authentication on the Internet. Technical report, SANS Institute, 2007.

[20] R. Oppliger, R. Rytz, and T. Holderegger. Internet banking: Client-side attacks and protection mechanisms. *Computer*, 42(6):27–33, June 2009.

[21] Paolo Passeri. Cyber Attacks Statistics, 2016. Accessed: 2016-07-07.

[22] Derek Scott. *Human-computer interaction: a cognitive ergonomics approach*. Ellis Horwood, 1991.

[23] Thomas Weigold, Thorsten Kramp, Reto Hermann, Frank Höring, Peter Buhler, and Michael Baentsch. *Trusted Computing - Challenges and Applications: First International Conference on Trusted Computing and Trust in Information Technologies, Trust 2008 Villach, Austria, March 11-12, 2008 Proceedings*, chapter The Zurich Trusted Information Channel – An Efficient Defence Against Man-in-the-Middle and Malicious Software Attacks, pages 75–91. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[24] Susan Weinschenk. 100 Things You Should Know About People: #58 People See What They Expect To See — The Team W Blog. https://www.blog.theteamw.com/2011/02/10/100-things-you-should-know-about-people-58-people-see-what-they-expect-to-see/, 2011. Accessed: 2016-05-31.