Serdica J. Computing **10** (2016), No 1, 49–62

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

EFFECTIVENESS OF A CONCEPTUAL MODEL FOR INCREASED MOBILE BANKING SECURITY

Bonimir Penchev

Abstract. Despite the advantages that mobile banking has for both banks and customers, its security level is one of the barriers that have a negative influence on its large-scale adoption. In our previous research we identified a certain number of security attacks against mobile banking security. Along with them, we found out that not all of the protection strategies and best practices are effective enough. Therefore we proposed a conceptual model for increased mobile banking security, which consists of five modules. In this report we aimed at researching the effectiveness of each of the proposed modules. In order to achieve our objective we conducted five different experiments, each covering a module. The results confirmed the effectiveness and applicability of each of the five modules that are part of the proposed conceptual model for increased mobile banking security.

1. Introduction. For more than 40 years one of the main objectives of financial institutions is related to the provision of easy access and convenience for their customers in the realization of banking operations. Although ATMs and

ACM Computing Classification System (1998): K.4.4.

Key words: mobile security, mobile banking, mobile device, mobile applications, mobile operating system.

Internet banking are effective channels for providing traditional banking products, a relatively new type of banking – mobile banking – has a significant impact on the market. Its continuous development is attested by various studies conducted in different regions of the world, covering both developed and developing countries [4].

The continuous development and the widespread usage of mobile banking are associated with the advantages that it provides to both banks and their customers [3]. This channel allows users to perform financial operations anywhere, anytime, at a lower price and without the need to visit a bank branch. On the other hand mobile banking offers different strategic advantages to the banks. It can be used as an opportunity to reach new customers, can improve the organization's reputation and its products, or can be used to conduct marketing campaigns.

Despite these advantages, the use of mobile phones and tablets, with the intention of performing banking transactions or gaining access to financial information is not as widespread as expected [5]. This indicates that there are certain factors which have a negative impact on the large-scale adoption of mobile banking. In order to identify them we conducted another research. Its results not only clearly indicate the existence of various factors that have a negative impact on customers, but also prove that the various risks related to the security of this type of channel have a significant influence in the decision making process for the use of mobile banking [8].

The above mentioned results provoked us to conduct another research whose primary objective was to examine the most frequently occurring vulnerabilities and security threats to mobile banking. The study was focused on the user, who is often defined as the weakest link in terms of security. As a result, we identified four major problem areas — mobile device, mobile web browser, mobile operating system and mobile banking application. For each one of them we studied the existing vulnerabilities and threats and we identified the most common attacks on mobile banking: eavesdropping attack, cross site request forgery attack, unauthorized physical access to the device, phishing attacks (vishing attack, smishing attack, tabnabbing attack and phishing applications) and mobile malware [6].

In the scientific literature for each of these attacks, there is a wide range of best practices and strategies for protection. However, we found that not all of them are effective enough. This in turn identifies a need of introducing some improvements in this area. As a result, in another research [7] we proposed a conceptual model for increased mobile banking security (see Fig. 1).

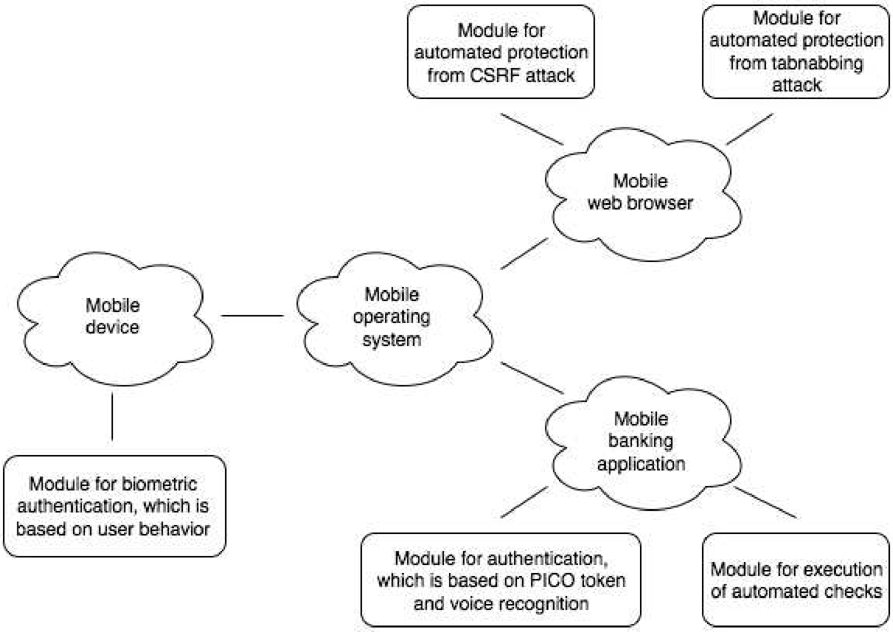

Module for automated protection from CSRF attack
Module for automated protection from tabnabbing attack
Mobile web browser
Mobile device
Mobile operating system
Mobile banking application
Module for biometric authentication, which is based on user behavior
Module for authentication, which is based on PICO token and voice recognition
Module for execution of automated checks

Fig. 1. A conceptual model for increased mobile banking security

In the proposed conceptual model we defined the following five modules:

- Module for biometric authentication, which is based on user behavior – It is designed as a tool for identifying the user of mobile banking. This identification is based on behavioral characteristics which are generated as a result of user interaction with the touch screen of the mobile device.
- Module for automated protection from tabnabbing attack – It is designed as a tool for detecting changes that occur on a certain tab of the mobile web browser when this tab was out of the focus of the user. As a result the user receives a visual warning that identifies the changed content, which helps him distinguish between legitimate changes from those used for the implementation of a tabnabbing attack.
- Module for automated protection from CSRF attack – It is designed as a tool that protects the integrity of the authentication session from malicious cross-site requests. To achieve this in the mobile web browser of the user we use an algorithm which exactly distinguishes between malicious and non-malicious cross-site requests.

- Module for authentication, which is based on PICO token and voice recognition – It is designed as a tool for identifying the mobile banking application user by combining a PICO token which is software embedded in the mobile device and a biometric mechanism for voice recognition.

- Module for execution of automated checks – It is designed as a tool which implements a set of automated checks defined by the providers of mobile banking services, and which helps users take certain actions that will lead to an increased level of mobile banking security.

Although we designed the presented modules as a result of a security analysis of the effectiveness of existing protection strategies and best practices against the existing attacks and vulnerabilities, it is still not clear to what extent they are effective and applicable in practice. As a result we defined the objective of this paper — to research the effectiveness of each of the five modules involved in the conceptual model for increased mobile banking security.

2. Materials and methods. In order to achieve the objective of this research we conducted five different experiments, each covering a module.

The first experiment was designed to investigate the efficiency and to determine the most appropriate algorithm for machine learning that can be used in the module for biometric authentication, which is based on user behavior.

To implement it we chose 30 participants (17 men and 13 women) aged 18 to 50 years. Each one of them has his own mobile device with a touch screen and the necessary knowledge to work with it. In participant selection we used the stochastic (or random) selection method.

For every participant we provided a mobile device LG Nexus 5 working with the most widespread mobile operating system Android and its most used version KitKat 4.4. Prior to the experiment we installed on the chosen device a mobile application developed by us. The application was used to gather input data from the touch screen of the mobile device. The application gave the user the possibility to implement standard actions while surfing in the internet. Meanwhile, as he interacted with the device, the developed mobile application was gathering input data and saved it to a file on the device.

After all participants finished working with the device we used the data saved in the file to manually produce two files in .arff format, which are suitable for the software product KNIME. The first .arff file was used as a training set in order to implement pre-training for classifying the users according to the collected input data. The second .arff file was used as a test set and helped us to determine if the selected algorithm for machine learning properly classifies the user according to the test data.

The second experiment was designed to measure the performance of the detection algorithm used in the module for automated protection from a tabnabbing attack. Under performance we should understand the time required for information processing and for alerting the user about the occurred changes. To this end we implemented measurements of the time required for: snapping a picture of a section of the mobile web browser, splitting a picture into parts, comparing two pictures, presenting the differences between two pictures.

To measure the time required to snap a picture of a section of the mobile web browser we decided to create an add-on for Firefox Mobile. We chose this mobile web browser because among the most widely used at this moment only it allows the installation of add-ons. During the development of the add-on we encountered a problem. At this moment Add-on SDK (software development kit for Firefox Mobile) does not support this functionality for the mobile version of the web browser. As a solution we chose and installed 6 add-ons for the standard web browser Firefox (46.0.1), which provide the ability to snap a picture of a section of a web browser. The objective of our decision was to receive an indicative result of the time required to implement this functionality, despite the imposed restrictions.

To measure the time required to split a picture into parts we designed a JavaScript function that receives as an input parameter a pre-made picture, splits it into parts and stores them in an array. As input parameters we created 8 different pictures with the help of the previously installed add-on Easy Screenshot. The pictures differ on the following criteria: content of the picture (executed on two different websites), resolution (tested on 2 different devices — LG Nexus 5 and Vonino Sirius QS), and dimensions of one part (10×10 pixels and 15×15 pixels). Here as well as in the next measurement we used JavaScript code because in that way we avoid the limitations that could result from the use of Add-on SDK.

To measure the time for comparing two pictures we developed a JavaScript function that receives as input parameters two pictures and implement a comparison between each pixel on one picture with each pixel on the other. As input parameters we created 5 different pictures. They differ only by the percentage of the changes that were made on the original picture (0%, 25%, 50%, 75%, 100%).

To measure the time for presenting the differences between two pictures we decided to create an add-on that visually presents them to the user. Due to the limitations in the Add-on SDK, we created an add-on that displays only a message with the percentage of the occurred changes.

The third experiment was designed to examine the effectiveness of the proposed algorithm used in the module for automated protection from CSRF attack.

To implement this we produced initially a formal description of the algorithm. This was accomplished with the help of the modelling language Alloy. As a basis for this formal description we used the model of web infrastructure developed by Akhawe [1]. Where it was necessary for the purposes of the experiment, we expanded the model.

As our second step we created also a formal description of the CSRF attack again using the aforementioned model and modelling language.

Based on the formal descriptions we implemented a formal inspection of the proposed algorithm as we used the latest stable version 4.2 of the Alloy Analyzer software for testing models, which is used to violate some security properties and in this way to prove or disprove the effectiveness of a model.

The fourth experiment was designed to assess the module for authentication, which is based on PICO token and voice recognition.

Our first step was to determine the approach which would be used to assess the module. As a basis we used an approach for assessing web-based authentication mechanisms developed by Bonneau [2]. It is called UDS and consists of 25 properties, which are divided into 3 main categories: usability, deployability and security. The assessment of an authentication mechanism is implemented by checking whether certain properties are satisfied using a three-level rating scale — yes, almost, no.

Since Bonneau's approach was not fully compatible with our module for authentication we modified it in order to improve its applicability. As a result we defined a new assessment approach, which includes a subset of the properties defined in the UDS approach. To the subset we added two additional properties:

- Periodical-authentication – the property is satisfied when the authentication mechanism provides periodical verification of the identity of the user. This process should not be a drawback for the user in the terms of usability.
- Different-levels-of-permissions – the property is satisfied if the authentication mechanism provides access to different functionalities depending on the degree of confidence that the user is the one that he claims to be.

The last step of the experiment was first to assess the module for authentication, which is based on the PICO token and voice recognition, and second to compare it with the following alternative mechanisms — PIN, password, biometrics and token devices.

The fifth experiment was designed to determine how users perceive the module for execution of automated checks.

To implement it we chose 30 participants (17 men and 13 women) aged 18 to 50 years. Each one of them has his own mobile device with a touch screen and the necessary knowledge to work with it. In participant selection we used the stochastic (or random) selection method.

For every participant we provided a mobile device LG Nexus 5 working with the most widespread mobile operating system Android and its most used version KitKat 4.4. Prior to the experiment we installed on the chosen device a mobile application developed by us. The application was used to execute the following automated checks:

- UWN — if the mobile device is connected to an unencrypted public wireless network.
- EOS — if the encryption of mobile operating system is activated.
- AM — if an authentication mechanism on the mobile device is activated.
- AL — if the automatic lock of the mobile device is activated.
- BT — if the Bluetooth interface of the mobile device is activated.
- UOS — if the version of the mobile operating system is up-to-date.
- ROS — if the mobile operating system is rooted.
- AV — if a mobile antivirus application is installed on the mobile device.

At the beginning of the experiment, the participants started the application and each one of them received as a result that none of the automated checks did pass successfully. In addition users saw which checks are obligatory (AM, AL, UOS) and which recommended (the rest) as well as the value of the security indicator that in this case was “0 of 11”.

The security indicator is calculated according to the following rules:

- If the check is obligatory and has passed successfully — 2 points.
- If the check is recommended and has passed successfully — 1 point.
- If the check does not pass successfully — 0 points.

As a second step of the experiment we provided the users with the opportunity to fix the unsuccessfully passed checks with the option to use or not some instructions, depending on their competence and desire. We prepared these instructions in paper format before the beginning of the experiment.

At the end of the experiment, after all participants finished trying to fix the checks, we investigated how many of the checks are fixed and what is the value of the security indicator.

Meanwhile we observed each user in his interaction with the module. During that process we accomplished the following observations:

- Is the user trying to fix the unsuccessfully passed checks or does he just refuse to use the mobile banking application?
- Does the user know how to fix the unsuccessfully passed checks without receiving instructions?
- Do the prepared instructions assist the user to more easily fix the unsuccessfully passed checks?
- Is the user willing to ignore the recommended checks?
- To what extent does the user improve the security indicator at the end of the experiment?

3. Results and Discussion. After the first experiment we received the results presented in Table 1.

Table 1. Error rate of the algorithms for machine learning

Algorithm	**Error rate**
Back-Propagation Neural Networks	5.32%
C4.5	19.5%
Naive Bayes	17.16%
Particle Swarm Optimization Radial Basis Function Network	1.8%
Radial Basis Function Network	8.9%
Repeated Incremental Pruning to Produce Error Reduction	7.84%

The results show that the error rate achieved using the algorithm Particle Swarm Optimization Radial Basis Function Network is 1.8%, which gives us reason to conclude that this algorithm for machine learning is appropriate for the examined task. As a result we consider that the module for biometric authentication, which is based on user behavior, will execute successfully its protection functions against unauthorized access to the mobile device.

After the second experiment we received the results described below.

The necessary time for each of the 6 add-ons to snap a picture on one and the same web site is presented in Table 2.

Table 2. Time in milliseconds(ms) to snap a picture

Easy Screenshot	**Awesome Screenshot Plus**	**Screengrab**	**Nimbus Screen Capture**	**Lightshot**	**Abduction**
133 ms	107 ms	122 ms	103 ms	111 ms	100 ms

Table 2 helps us calculate that the average time to snap a picture with the help of the application programming interface of the web browser is 113 ms.

The necessary time for the created JavaScript function to split the picture into parts is presented in Table 3.

Table 3. Time in milliseconds (ms) to split a picture into parts

Picture	**Resolution**	**Part dimensions**	**Time**
Picture 1	1024 × 768	10 × 10 pixels	59 ms
Picture 1	1366 × 768	10 × 10 pixels	67 ms
Picture 1	1366 × 768	15 × 15 pixels	39 ms
Picture 1	1024 × 768	15 × 15 pixels	29 ms
Picture 2	1024 × 768	10 × 10 pixels	63 ms
Picture 2	1366 × 768	10 × 10 pixels	71 ms
Picture 2	1366 × 768	15 × 15 pixels	42 ms
Picture 2	1024 × 768	15 × 15 pixels	31 ms

The results in Table 3 help us calculate that the average time to split a picture into parts with the created JavaScript function is 50 ms. Furthermore, we found out that the time for implementation of this operation depends on one hand on the resolution, and on the other on the dimensions of a single part. It does not depend on the actual content of the picture.

The necessary time for the created JavaScript function to perform a comparison between two pictures is presented in Table 4.

The results in Table 4 help us find out that much time is consumed by the algorithm that implements pixel by pixel comparison of two pictures. The time used by it is closely related to the percentage of the changes that have taken

Table 4. Relationship between the amount of changes to the original picture and the necessary time for the algorithm to compare

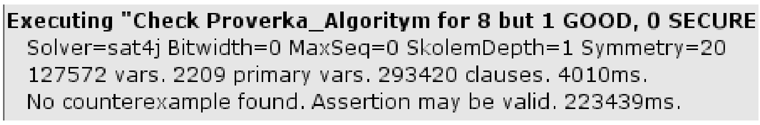
Changes to the original picture	**Time spent to compare**
0%	119 ms
25%	90 ms
50%	58 ms
75%	29 ms
100%	5 ms

place on the picture, because the algorithm is implemented in such a way that if it detects a difference at the first pixel, it does not compare the remaining pixels. Therefore, if a tabnabbing attack is performed more changes will be detected and the comparison algorithm should be executed even faster.

The necessary time for the created add-on to display a message representing the percentage of the occurred changes shows that this operation is executed quickly — 1 ms.

The results obtained from the implementation of the four measurements help us determine that the average time required by the module to perform its function is 230 ms. Almost half of it (113 ms) is used by the application programming interface of the web browser, which is beyond our control. Measuring the performance in the described limitations gives us reason to say that the module manages fast enough to process the information and to alert the user about the occurred changes. This determines its effectiveness in terms of the tabnabbing attack. As a guideline for future work, we can point out that the measurements should be repeated after the removal of the limitations associated with its complete implementation.

After the third experiment we received the results presented in Figure 2.



```
Executing "Check Proverka_Algoritym for 8 but 1 GOOD, 0 SECURE
   Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
   127572 vars. 2209 primary vars. 293420 clauses. 4010ms.
   No counterexample found. Assertion may be valid. 223439ms.
```

Fig. 2. Result of the formal model checking of the algorithm used in the module for automated protection from CSRF attack

Figure 2 shows that no counterexample is found which would indicate that a malicious user can generate a malicious cross-site request using a mobile web browser. This in turn specifies the effectiveness of the proposed algorithm and the possibility to use it in the module for automated protection from CSRF attacks.

In Table 5 are presented the results of the assessment of the module for authentication, which is based on PICO token and voice recognition (fourth experiment). Simultaneously, in the table we made a comparison with alternative mechanisms, whose assessment was already implemented in Bonneau's publication [2].

Table 5. Comparison of the different authentication mechanisms

Property	**Module**	**Token**	**PIN**	**Bio-metrics**	**Pass-word**
(U) Memorywise-effortless	**yes**	**yes**	–	**yes**	–
(U) Nothing-to-carry	*almost*	–	**yes**	**yes**	–
(U) Efficient-to-use	*almost*	*almost*	**yes**	**yes**	**yes**
(U) Infrequent-errors	*almost*	*almost*	*almost*	–	–
(U) Easy-recovery-from-loss	*almost*	–	**yes**	**yes**	**yes**
(D) Negligible-cost-per-user	*almost*	–	**yes**	**yes**	**yes**
(D) Mature	–	**yes**	**yes**	*almost*	**yes**
(S) Resilient-to-physical-observations	**yes**	**yes**	–	**yes**	–
(S) Resilient-to-targeted-impersonation	**yes**	**yes**	*almost*	–	*almost*
(S) Resilient-to-throttled-guessing	**yes**	**yes**	**yes**	**yes**	–
(S) Resilient-to-phishing	**yes**	**yes**	**yes**	–	–
(S) Resilient-to-theft	**yes**	*almost*	**yes**	**yes**	**yes**
(S) Unlinkable	–	**yes**	**yes**	–	**yes**
(S) Periodical-authentication	**yes**	*almost*	–	–	–
(S) Different-levels-of-permissions	**yes**	–	–	–	–

The results presented in Table 5 help us find out that the proposed authentication mechanism based on PICO token and voice recognition has the highest score for the properties from the category "security" (marked with "(S)"). On the other hand regarding the properties of the other two categories (usability — marked with "(U)" and deployability — marked with "(D)") it gives away to the alternative authentication mechanisms. As a positive feature we could emphasize that the module has a higher assessment in these categories than token authentication, which is ranked in the second place in the category "security". In addition, the module satisfies the property "Memorywise-effortless" and almost satisfies "Nothing-to-carry", which are among the main problems that users experience in terms of convenient authentication.

After all we can conclude that the proposed module for authentication increases the security of mobile banking. As a guideline for future improvement we will point out the necessity to work towards enhancing the values of the properties of the categories “usability” and “deployability”.

After the fifth experiment we received the results described below.

Still at the beginning of the experiment and after the tested users understood that in order to use the mobile banking application it is necessary to meet certain requirements, five of them said that they prefer to use the mobile web browser to implement mobile banking. As a reason they pointed out that they use mobile banking only for reports and therefore they believe that there is no need to spend time to meet the aforementioned requirements.

In Table 6 are presented the results of the other 25 participants.

Table 6. Number of users, who have fixed a certain check

Check	**Fixed without instructions**	**Fixed with instructions**	**Total attempts to fix a check**
UWN	1	4	8
EOS	2	4	8
AM	12	10	25
AL	6	16	25
BT	5	2	8
UOS	8	14	25
ROS	0	0	8
AV	1	2	8

The results in Table 6 help us find out that 22 (88%) of the users were able to fix all three obligatory checks, while 3 of them were not able to fix any of the checks, even after using the instructions. As a reason they pointed out that the instructions are not clear enough. So we can conclude that the instructions need further improvement.

One notes the number of the users who tried to fix the recommended checks. They are only 8 since the others clearly said they would try to fix only the obligatory checks. So we can conclude that the user responds better to the obligatory checks and they can be better applied in this module.

In Figure 3 are presented the results concerning the percentage of the successfully fixed checks by distinguishing the ones made with or without instructions.

The results in Figure 3 help us find out that 75% of the checks were fixed by more than 60% of the participants, that stated a desire to do the fixes. One part of the users had the necessary knowledge to fix unsuccessfully passed checks. However, at 63% of the checks it is seen that the success is due to the provided

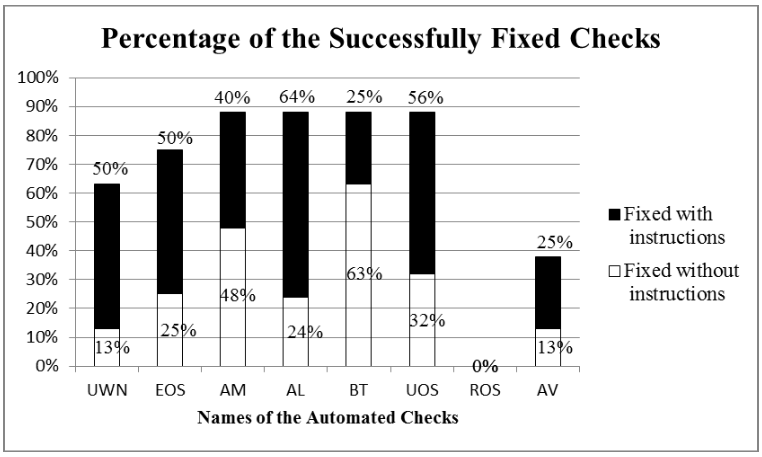


Fig. 3. Percentage of the successfully fixed checks

instructions, which is an evidence for their benefits to the users.

We can observe serious differences in the check connected with the rooted operating system. In this regard, the users said that their systems are rooted and therefore they decided not to fix this check.

The results show that 74% of all users (30 participants) improved by at least 55% their security indicator, as this is the percentage of those who have fixed the obligatory checks. For those of them who decided to fix the recommended checks this percentage is even with a higher value. This gives us a reason to say that if the module for automated checks is going to be developed in its full functionality it would increase the level of security in mobile banking.

4. Conclusions. The results of this experimental research confirmed the effectiveness and applicability of each of the proposed five modules included in the conceptual model for increased mobile banking security, which was presented in the introduction. However, these results are only preliminary in nature and therefore at a later stage it will be necessary to conduct further and more comprehensive research, which will provide statistically more significant results. This can be outlined as a guideline for a future work.

REFERENCES

[1] Akhawe D., A. Barth, P. E. Lam, J. C. Mitchell, D. Song. Towards a Formal Foundation of Web Security. In: Proceedings of the 23rd IEEE

Computer Security Foundations Symposium (CSF), Edinburgh, 2010, 290–304.

[2] Bonneau J., C. Herley, P. Oorschot, F. Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, 2012, 553–567.

[3] Esmaili E., M. Desa, H. Moradi, A. Hemmati. The Role of Trust and Other Behavioral Intention Determinants on Intention toward Using Internet Banking. *International Journal of Innovation, Management and Technology*, **2** (2011), No 1, 95–100.

[4] Global Mobile Statistics 2014 Section G: Mobile Banking and M-money; Section H: Venture Capital (VC) Investment in Mobile. `http://mobiforge.com/research-analysis/global-mobile-statistics-2014-section-g-mobile-banking-and-m-money-section-h-venture-capital-vc-inve`, 30 April 2016.

[5] Juniper Research. Mobile Banking Handset & Tablet Market Strategies 2013–2017. `http://www.juniperresearch.com/reports/mobile_banking`, 1 May 2016.

[6] Penchev B. Security Issues in Mobile Banking. In: Proceedings of International Conference Human Systems Integration Approach to Cyber Security, Sofia, 2016, 135–144.

[7] Penchev B. Conceptual Model for Increased Mobile Banking Security. In: Proceedings of the IV International Scientific Technical Conference, Vol. **2**, Veliko Tarnovo, 2016, 50–53.

[8] Penchev B. Factors Negatively Affecting Consumers in Mobile Banking Adoption. *Bulletin of the Union of Scientists—Varna, Economic Sciences Series*, Varna, 2015, 150–155 (in Bulgarian).

Bonimir Penchev
University of Economics
Varna, Bulgaria

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
e-mail: `bonimir@gmail.com`

Received June 20, 2016
Final Accepted August 29, 2016