SCA as Mobile Security Threat

Alla Levina, Pavel Borisenko, Roman Mostovoy
ITMO University Saint Petersburg, Russia
levina@cit.ifmo.ru, borisenkopp@yandex.ru, ramostovoy@corp.ifmo.ru

***Abstract*—Recent time the mobile phones became the main device for the modern society members. Of course, it means that they also became the main target of the attackers. Our team is working on the issue of the mobile security regarding the side-channel attacks. We have faced with a number of challenges which the attacker has to overcome to obtain the sensitive data, but at the same time our research shows that the severity level of the side-channel attacks threat for the mobile security is even more high then we can assume. The instrument available for the any potential malefactor, the consumer-level device, and realy simple technique of the trace gathering allow to distinguish the cryptographic operations in dependency on the used input data. Although on this equipment level the obtaining of the key data is not possible, the attacking algorythms designed during the our research will allow to reach this goal on the more detailed signal records.**

I. Introduction

Side channel attacks has changed the people's mind who involved into security issue researches. Although the first works related to the side channel attacks were published more than twenty years ago, this topic became a realy popular in a recent time. And the vulnerabilities which were found in a context of the side-channel attacks becomes more and more significant. As an example, last year the well-protected electronic safe locks were compromised by usage of the rather simple techniques such as timing attack and power analysis [6]. Thus, the device which is robust enough against the regular attacker's actions turned up completely vulnerable to the side-channel attacks. It shows how the side-channel attacks changed the accents in the information security area.

Regarding the mobile security, there are three trends of the malefactor interest. They are discribed below. In general, mobile phone is a key for the different areas of the owner's life. It is being used to store, process and transmit a significant volume of the sensitive data, and to getting access to different external information systems as well. And unexpected vulnerabilities such as the described above and other already discovered weaknesses [2] can lead to the tragic security accidents.

At first, an attacker may be potentially interested in capturing voice, text or video transmission services' data. If the encryption apparatus performs high number of computational operations being the same type (atomic for text messaging; continuous for data streaming), thus sufficient number of parasitic signal traces may be obtained in order to determine the key. If the key is updated infrequently enough, critical information would be disclosed to the attacker.

The second type of potentially vulnerable applications are ones related to shared data access. Cloud storages utilize the so-called *user-controlled cryptography*, i.e. the encryption of data is made on the client side before transmission to the server. Here, the host system is allowed to utilize weaker stored data protection methods at the same time creating a possibility for side-channel attack. Non the less, even if data is secured on the device side, the authorization and transmission data may be still compromised by the attacker [5].

The third risk group is related to the concept of "smartphone as part of Internet of Things" and a variety of services associated with the management of other devices, processing of the status messages, and making certain decisions. The main goal for the attacker would be to capture the authorization data in order to get the private data access in the future, but in addition to privacy of the data [4], an important role is played here by their integrity and availability, so that the task of ensuring the security of these applications is even more difficult and complex, and in addition of cryptographic protection in such applications should be implemented communication high-quality protocols, precluding, for example, a control command transmission to any device without proper authorization.

But we cant ignore the fact that there are many different power consumers excluding processor in the mobile phone: screen and various communication modules (GSM-module, Bluetooth, WiFi, NFC). For obvious reasons, these consumers usually work at the same time with the applications described above. It creates additional difficulties in the registration of traces, as all of these energy consumers can also make a noise in the side channels, but this noise is little informative, for example, in terms of the search of the encryption key. As we will show further, attacker is forced to take it into account.

The rest of the paper is organized in the following way. The second part contains the testing environment description. Our methods of trace recording and the results of the trace analysis are represented in the third part. Section four detailed the proposed neural network solution for key derivation and lessons learned experience. There are several advice regarding the mobile security in the fifth section. Paper is finalized with the overview of the current results and further improvement of the work.

ISSN 2305-7254

II. Testing environment

In scope of this research we decided to concentrate on attacker's model with low-level opportunities regarding equipment. It also has defined requirement to software which was examined. It was important to perform restricted well-controlled computation operations. We are analyzing self-designed application for Android operating system and several models of mobile devices.

The normal Android applications have a number of features which make the attacker's task more difficult:

- Any application seeks to interact with the user in many ways and the most basic one is graphic user interface (GUI). Working with GUI elements leads to active power consumption from the phone screen, making it difficult to identify parasitic signal directly related to cryptographic operations.
- Users often become distracted by the actions that do not lead to the implementation of cryptographic operations on the device. Their attention is distracted by other applications, external stimuli, etc. It leads to noisy parasitic signal and a lot of data which the attacker is not interesting in.
- Even if the user is fully concentrated on the actions that lead to qualitative parasitic signal (for example, sending messages to other users via a secure channel), he/she makes them slow, and hence, with a limited period of time you are able to get a very small volume of required data.

Due to the large number of constraints imposed on side-channel attacks by production applications, custom application has been developed, which is a "sandbox" of cryptographic primitives. Its concept is to minimize GUI, which is not used at all during the cryptographic operations, easily extensible set of cryptographic primitives and controllable set of secret encryption keys used. In addition, the application allows you to perform the necessary cryptographic operations at a high frequency for the rapid accumulation of sufficient data to carry out your attacks. The full list of features is the following:

- 3DES, AES, RSA and DES encryption are supported;
- The list of secret keys and initial vectors is controllable;
- Time-stamps and detailed logging are supported because it is very important to keep synchronization between the data and attacking tool;
- High frequency of cryptographic operations with controllable delay. The same functionality was implemented for production application to facilitate data gathering;
- Random plaintexts and keys generation;
- Both CBC and ECB encryption schemes;
- Number of equal starting signatures is configurable.

It is possible to make several equal cryptographic operations (with the same plaintext and key) before processing of random plaintexts and/or keys. The goal is make it easier to synchronize attacking tool and the data: if you have several equal signatures inside a trace you are able to find them and detect further encryptions starts and finishes. This functionality is very useful in case of noisy hardware inside a target device;

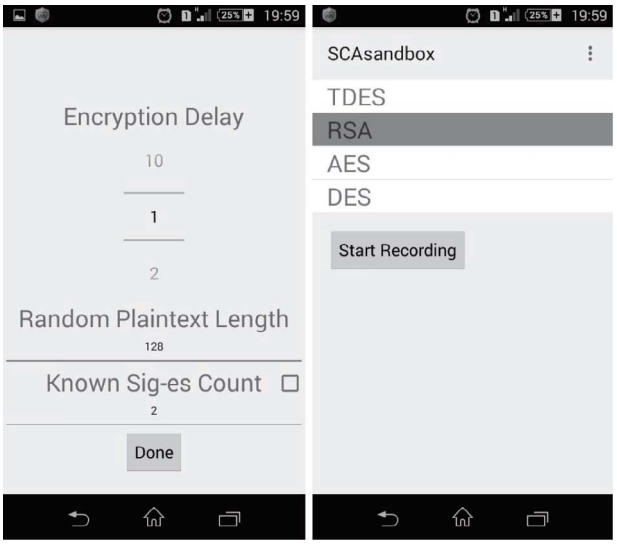

Fig. 1. Sandbox Android application

Based on the fact that the attacker does not use expensive equipment to carry out attacks only two easily accessible tools were used for traces recording - external sound card and jack-jack cable:

Fig. 2. Test installation

III. Trace gathering and analysis

In scope of this research, we decided to concentrate on attacker's model with low-level opportunities regarding equip-

ment. Similar requirement was applied to the examined software as well. In this section, we perform restricted well-controlled computation operations with further analyzis of results for two different models of mobile devices (Alcatel POP3 and Xperia M2).

We present a *clean* run using ”sandbox” application, i.e. most of the device activity is lowered, background processes shut down and screen is turned off. This is done due to the nature of user input pseudo-random behavior, i.e. registered signals strongly depend on the mobile phone user actions. Therefore, cryptographic operations executed in ”silent” mode have more or less clear traces, as it is shown in Fig. 3 and Fig. 4.

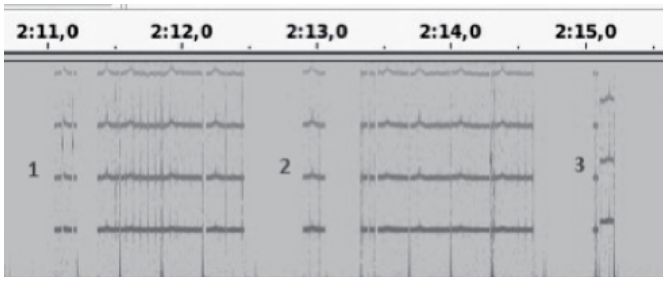

Fig. 3. Clear trace for Alcatel POP3: each time of encryption operation is numbered

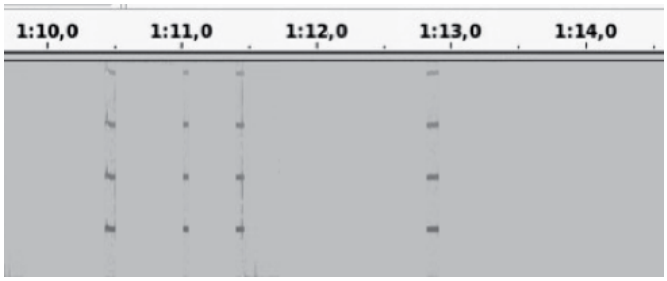

Fig. 4. Clear trace for Sony Xperia M2

On this step it is visible that hardware differencies lead to different signal profile: for Xperia M2 cryptographic operations were not clearly defined. Then, comparison of the acoustic parasitic signal from two phones was performed using the same application but with variety of side processes, enabled Wi-Fi, Bluetooth and other common modules. User interaction was perfomed also. And again there is a dramatic difference in the definition of the data. In Fig. 5 and Fig. 6 we see totally undiagnosed flow noise for Alcatel and almost empty (from the signal level point of view) trace for Xperia.

Based on the results collected it was decided to continue with analysis if the traces only from Alcatel POP3 due to data insufficiency in the traces gathered from Xperia M2.

The next step is parsing the raw data to make it possible for nural network work with it in form of vectors where each vector - is a signal recorded during a particular cryptographic operation. If the synchronization is perfectly achieved for a

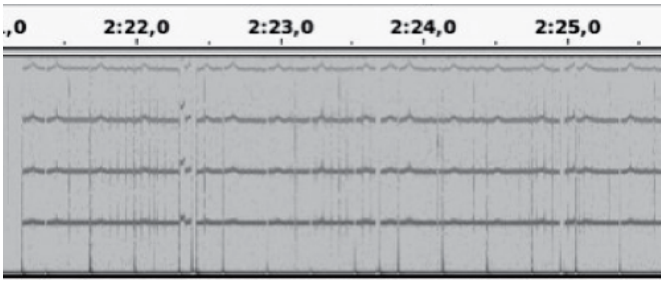

Fig. 5. Noisy trace for Alcatel POP3 during user interaction

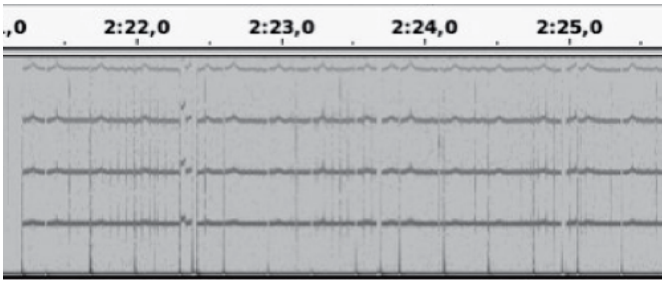

Fig. 6. Trace for Sony Xperia M2 during user interaction

trace with detectable encryption operations, i.e. by knowing the first N signatures of the trace, attacker is able to predict the next signature starting and ending points. The accuracy of this mechanism strongly depends on the sound card sampling and time periods required for the cryptographic operations execution on the device side.

In order to distinguish signatures, a special parsing software developed in Python by our team is utilized. The parser gets a trace recorded during cryptographic operations and corresponding file with time stamps which was generated by ”sandbox” application. The goal of the parser is to distinguish all cryptographic operations from the signal, avoiding noise, and unload them as a separate vectors for further loading into neural network. The parser consists of two similar methods for this purpose:

- *Convolution function*: Here the parser considers the whole trace and signatures inside it as functions. Previously it was mentioned that to simplify the process several N signatures can be set equal during recording of traces. The parser goes through all the hypothesis of the first signature and calculates the resulting convolution function between the hypothesis and the whole trace. Peaks of the resulting functions show what hypothesis are the most similar and how much signatures matching this hypothesis exist in the trace. After the first signature has been found, the parser goes through the trace and based on starting point of the signature and time stamp file detects starting and ending points of other signatures. Finally, every signature is indicated in the trace and can be uploaded as a separate vector. The work results of

this method are represented in Fig. 7.

- *Convolution function with neighbourhood*: Algorithm is very similar to the previous one.Nnow the parser uses information from time stamp file (how much time left between and during cryptographic operations) as a 'lattice' for detection of signatures. Convolution function is calculated between every hypothesis of the furst signature and corresponding hypothesis of other signatures, that are detected just based on timing: we assume that we know correct starting point of the first signature and knowing durations from time stamp file we can assume where every other signature is located in the trace. In this way, analyzing maximums of convolution calculations the parser detects what hypothesis of the first signature is the best one. After that it marks staritng and ending points of other signatures and unloads them. The work results of this method are represented in Fig. 8.

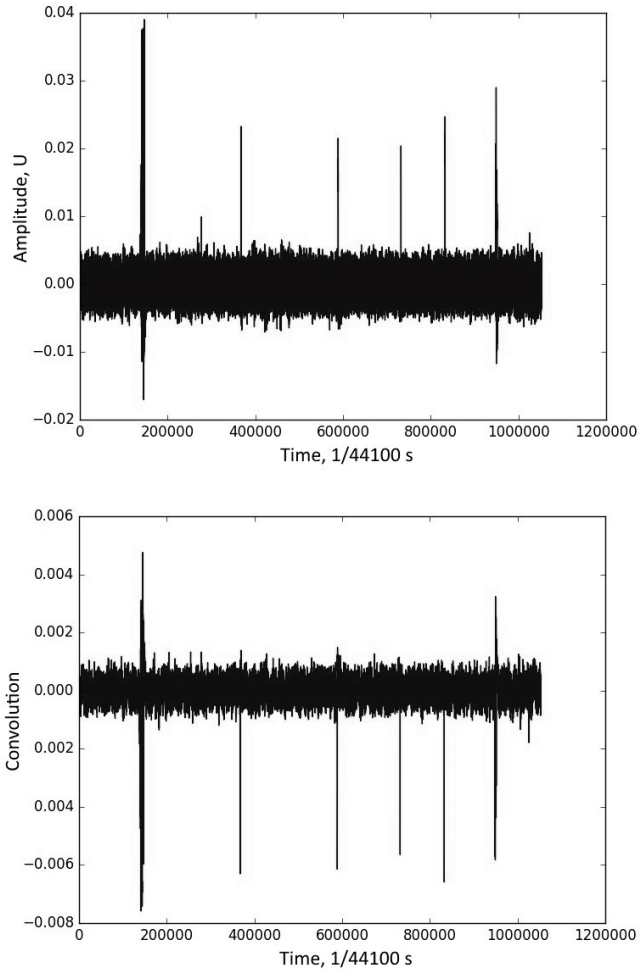

Fig. 7. Signatures detection with the first method

Two facts are noticable after parcing has been done:

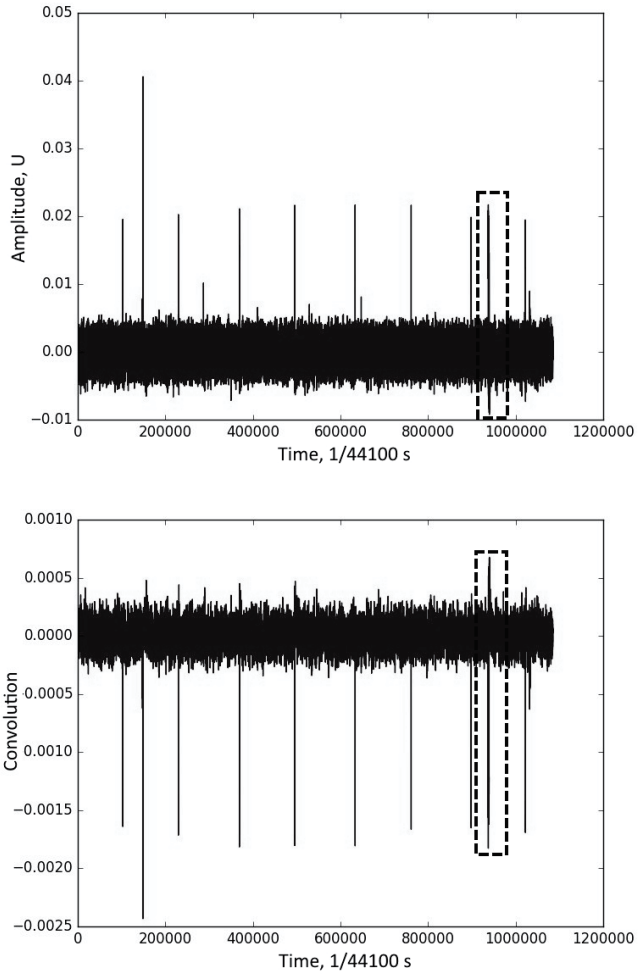

Fig. 8. Signatures detection with the second method

- *Signatures are not well-detailed because of scarce sampling of audiocard. Detalization is very important for further analysis;*
- *On the second example it is visible that some signatures were detected incorrectly. Similar result was achieved using the first method on the same trace. From the current point of view, the reason such inaccuracy also relates to bad detalization of signatures.*

The utilization of more expensive and capable data capturing equipment is a solution to both of the above challenges.

IV. Derivation of the key

Current tests prove that performing detailed traces analysis utilizing simple sound card is a complex task. This is due to tremendous difference between sampling rates of the attacker device (44100 Hz) and encryptions ones. A lot of details

desappear due to scarce sampling of sound card - only every 44100th point of the original signal is available for analysis. According to that inconsistence, even the whole computation process, if it is executed pretty fast, can be invisible for the measuring tools. Otherwise, the resolution of the obtained traces is still very low. It means that a little information about the sensitive data can be derived from such traces. It is planned to utilize more powerful sound card with higher sampling rate for future experiments. In this work, we utilize the cheapest attacking device.

We performed an attack on two specific devices and exact encryption algorithms. Probably such sampling can be insufficient for some other use cases.Besides the developing of the parser, we are also working on analysis of the trace to derive the keys.

Our current goal is to utilize the versatile approach for the analysis of parasitic signal based on artificial neural networks for side channel attack execution. This approach does not imply an absolute identification of a secret device key, but bring an opportunity to determine the most probable state for each of its bits.

Overall, the model of the attacking system consists of several functional modules, which you can see in Fig. 9. The model is based on the iterative approach and allows to optimize the attacking process.

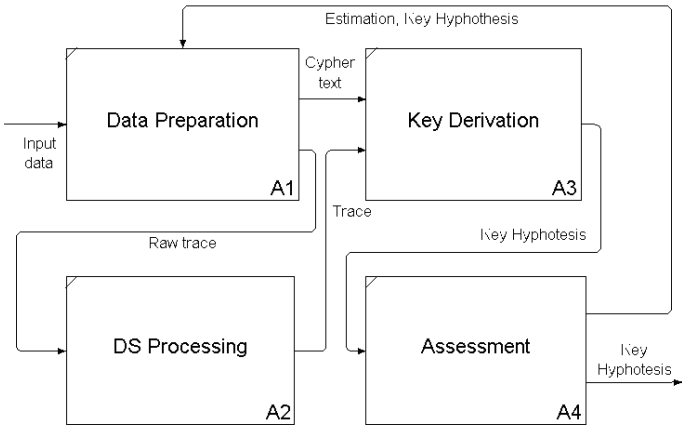

Fig. 9. Functional model of the attacking system

During the first phase, it is necessary to validate if obtained traces are key- and plaintext-dependent. Multilayer perceptron is one of the possible solutions for this purpose. In case such dependence exists, it is possible to learn multilayer perceptron to distinguish two keys which differ by only one particular bit.

In the next stage, we utilize convolutional neural network (CNN) with the corresponding matrices/weights prepared during the previous stage. It allows to compute so-called “feature maps” to distinguish each bit of the key. Normally, each output neuron provides estimation of the probabilities either input data is consistent to particular class or not. Since a number of classes is vast for cryptographic keys, it is not convenient to utilize neural network. Hereby, neural network would provide information regarding the meaning and confidence of each key bit by relevant output neuron.

This is done due to a set of factors. First of all, to decrease the number of output neurons and, consequently, the number of connections between neurons as well, i.e. output layer of the network would contain as many neurons as many bits are contained in key.

Moreover, the increasing of the key length leads to the growing of the network more slowly than in case of classical approach (each bit of the key increase the amount of the neurons in the output layer twice in case of classical approach, we have only one output neuron perbit of the key).

Another advantage of the suggested approach is to couple the probability and value for each bit on the output. It allows to utilize various indicators for the error estimation and confidence of the results. Error vectors could be represented either by precise difference or by comparing binary hypothesis with actual key.

The attacking system has to distinguish three types of the hypothesis estimation for configuring the exit conditions. The worst type of the error to be completely eliminated is the confident-wrong hypothesis of the particular bit. In case of this error we have wrong result and cannot localise the position of the mistake. Whereas if we have the limited amount of the unconfidently-defined bits, we can try to switch each of them from zero to one or vice versa. Thus, these few bits of the key can be derived by brute force. In case of uncertainty, it does not matter if hypothesis is correct or not, i.e. in both cases the overall result would contain uncertain relevant bit and two probable keys would have to be checked. Hence, each uncertain bit increases the number of probable keys two times and thus the sum of such errors has to be limited.

During the learning process, vectors of the error are utilized for the data input relevance estimation. After the first stage, all the input data has the corresponding associated weights further involved in data preparation algorithm. Besides the input data, the correlation (XOR) between it and actual key (the hypothesis of the key in case of real attack) is being estimated.

On the stage of data preparation and when the part of the key bits in the key hypothesis are already defined confidently, estimated correlation vectors could be utilized for optimization. The system chooses the most applicable vector of the correlation to select the set of compliant input vectors. From this set, the vector with the highest weight would be selected for this iteration as input data.

Neural network is being utilized as a set of binary classifiers, and also, actually, only a part of the trace has correlation with particular bit of the key. The topology of the artificial CNN allows to find such search correlations.

There is a number of other reasons for selecting CNN as a main analyzer for the key derivation task. First of all, particular bits of the key have impact only at particular trace parts thet

could be taken into account by CNN. Hence, there is less amount of configurable links than in multilayer perceptron. Furthermore, convolution computations can be performed by applying higher number of streams in more simple way than most of networks with other topologies. Finally, deep machine learning can examine much more complexity correlations.

The disadvantage of the neural networks as a tool for analysis, is low efficiency for processing data that containing large number of features. Discussing the parasitic signal, the attacker is not able to independently distinguish important features of each trace. To solve this "curse of dimensionality" problem, we use normalized inter-class variance (NICV) method [1]. It allows to distinguish the most vulnerable features of traces, based on the data classification and detection of anomalous dispersion deviations.

The actual topology of the artificial neural network which will be used is always be obtained by the way of the tries and mistakes. Due to specific of the our task the completely applicable best practices to build neural network which we need are absent. We are working on this task now and, as was mentioned above, at this step the neural network is facing with the insufficient quality of the traces regardless of it's topology and characteristics.

In other words, experience of the attempts to use obtained traces in the key derivation process showed the strongly insufficient quality of the traces.

V. Recomendations for mobile security

One of the major weakness which leads to SCA vulnerabilities is the data-dependent execution flow. If the algorithm assume that the sensitive data are used as the arguments for the crucial conditions, and the number and type of the following instructions are depending on the data values, it can affect on the power consumption and thus can be detected by the passive side-channel attacks. Such methods have to be removed or obfuscated by the balancing approaches. So, first of all, the number and the type of the execution operations should not be dependent on the sensitive data values.

From the side of consumer, it is important for mobile phone users to pay attention to the used communication channels (Wi-Fi,Bluetooth etc.), to the quality and the confidence level of the peripheral devices and accessorises (like power-banks, adapters etc.). Communication through insecure channels, or with the compromised devices, and also usage of the non-trusted peripherals and accessorise allows for the attacker to get access to the phone and derive the sensitive data even using the side-channel information.

Of course, the user should not forget about the well-timed upgrades of the operation system and applications, which often contain the security improvements.

VI. Conclusion

The issue of safety in critical world of mobile devices is becoming more crucial every day. Our research has shown that using of low-level equipment for side-channel attacks on the mobile phones is a challenge.

On the other hand, the results indicate that even with low-level equipment attackers can detect signals of cryptographical computations. So minor improvement of the tools can allow to get much more informative traces. Successful analysis of such data is a real danger for the sensitive information stored on device.

We a planning to increase the level of equipment (staying in common consumer segment) to continue detection of minimum necessary hardware level for getting into mobile phone's secrets. Also, the algoritms for the parsing and classifying of traces will be improved to use new informative traces even more effective.

References

[1] Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, Zakaria Najm, *NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage*, vol. 3, 2013.

[2] Daniel Genkin, Lev Pachmanov, Itamar Pipman, Eran Tromer, Yuval Yarom, *ECDSA Key Extraction from Mobile Devices via Nonintrusive Physical Side Channels*, vol. 3, 2016.

[3] Lukasz Romaszko, Demian Battaglia, Isabelle Guyon, Vincent Lemaire, Jordi Soriano, *Signal Correlation Prediction Using Convolutional Neural Networks*, JMLR: Workshop and Conference Proceedings, 2015.

[4] Jan Henrik Ziegeldorf, Oscar Garcia Morchon, Klaus Wehrle, *Privacy in the Internet of Things: threats andchallenges Security and Communication Networks*, 2014.

[5] Garrett S.Rose, Dhireesha Kudithipudi, Ganesh Khedkar, Nathan McDonald, Bryant Wysocki, Lok-Kwong Yan, *Nanoelectronics and Hardware Security Springer Advances in Information Security*, Springer New York, 2013.

[6] Plore, *Side-channel Attacks on High-security Electronic Safe Locks*, Def Con 24, 2016 URL:https://www.defcon.org/