”Tap2smart”: a New Lock-screen Option for Modern Smartphones

Maria Shelest, Maxim Grankin, Evgeny Bakin, Grigory Evseev
St.Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia
{maria.shelest, m.grankin, jenyb, egs}@vu.spb.ru

***Abstract*—Nowadays, besides being used as phones, smartphones involve a lot of such important additional functions as being organizer, valet, ID etc. That is why the users are to be sure, that their personal data kept in a smartphone is reliably protected in case of device loss or theft. One of the basic security mechanisms is a lock screen and modern smartphone has a lot of different lock screen options (e.g. PIN, pattern etc.). In this paper we consider a new user hybrid scheme based on tapping the rhythmic sequence on a lock screen. For the scheme we propose a decision making algorithm based on statistical analysis of tapped sequence which provides a high steadiness and a continuous adaptation to a user.**

I. Introduction

In modern society in which almost each person has at least one gadget, protection of information stored on it is an important issue. The first level of gadget protection is a lock screen on which a user is offered to enter some keyword. Most part of popular types of keywords can be divided into two parts: keywords based on secret data and keywords based on individual user features.

The first type includes different alpha-numeric keywords (PIN, secret word etc) and graphical keywords (e.g. pattern-based or picture-based). There are also a few approaches which combine alpha-numeric part and graphical part, such as for example piano-passcode [1], or Knock Code developed in LG smartphones [2]. The disadvantage of this method is that a malefactor obtains a full access to a device if he or she knows a secret data (e.g. as a result of eavesdrop or peeping).

The second type is based on consideration of user individual features. These authentication methods are based on human physiological or behavioral characteristics, such as for example face, voice, or fingerprints.

Face Verification / Face Authentication is the most common because it does not require any special equipment. In [3] authors found out that mobile face and voice recognition had high performance, but were not usable in all situations. Face verification methods require good lighting around and absence of foreign objects on the face. Also, this method of authentication requires a good front camera which is available only on expensive devices. In addition, there is a possibility of unlocking the system by photos of the legal user. And still need to provide a backup authentication option in case the device cannot recognize your face. Recently, some top models started being equipped with a fingerprint reader. Nevertheless, their prevalence encounters high cost of such devices and impossibility of their usage with wet hands or in gloves.

Also there are a few alternative biometric methods such as based on human heart sound signals and based on gait recognition. Authentication using human heart sonic signals is described in [7]. The use of heart sonic signals for authentication is possible due to the fact, that heartbeat melody is unique for every specific person. This feature was revealed when studying the heartbeats sonic signal spectrum. Another previous studies proved, that a human gait has a very distinctive patterns that can be used for identification and verification purposes by using accelerometers of mobile phones [4], [6]. Recent advances in microelectronic chip development make possible user authentication based on gait, using small, light, and low-cost sensors implemented in every mobile smartphone. However, the need to move during authentication considerably narrows the area of algorithm usage.

Generally, authentication based on individual user features requires either specific smartphone equipment, or sophisticated processing algorithms.

In this work we propose a method which is ready to be implemented in low-cost phones (doesn’t have any specific requirements to hardware) and combines both mentioned types of methods. The core of proposed method is a so-called rhythmic keyword. Rhythmic keyword is a keyword based on comparison of a characters input rhythm (sequence of time intervals between entered characters) to the reference pattern. For the first time the concept of such keyword was apparently published in work [8]. In it this technology was described in details, but no decision-making rules were mentioned. One algorithm of decision-making were proposed and analyzed in paper [9]. Authors of the paper proposed simple approach in which time intervals between input characters were considered to be independent identically distributed Gaussian random variables. Hence, a candidate sequence were accepted as a keyword if every interval was in range $\pm 3\sigma$ around reference duration. We propose to increase an efficiency of this algorithm by considering a dependency of the intervals entered by user. This dependency reflects a specific user sense of rhythm and ear of music and improves an the accuracy of authentication.

The rest of the work is organized as follows. In section II we introduce a general description of an authentication method. Section III is devoted to development of a statistical model of user keyword entering process. In Section IV we propose a decision making rule based on developed model. Implementation of authentication algorithm for Android gadgets is described in Section V. The paper is finalized with method steadiness evaluation and conclusions.

II. Preliminary considerations

In the proposed algorithm a keyword is a melody, known only to a smartphone user. The procedure of authentication is based on tapping of this melody on a smartphone screen which displays a keyboard of musical instrument (for example, a piano). Hence, for accessing to his or her smartphone a user periodically enters the keyword which can be represented as a set of notes $\boldsymbol{n} = [n_1, n_2, ...n_N]$ and a set of durations of these notes $\boldsymbol{\Delta t} = [\Delta t_1, \Delta t_2, ...\Delta t_N]$ (time intervals between taps).

The algorithm has three levels of defense:

1) Only notes are checked. In this case, it is a variant of alpha-numeric keyword.
2) Only time intervals are checked. In this case, it is a pure rhythmic keyword.
3) Both notes and intervals are checked. In this case, a full defense is provided.

One of the most challenging problems in such a scheme is a provision of possibility of adaptation to a particular user sense of rhythm. For solving this problems we propose the following general scheme (see Fig.1). I.e. after every successive entering of a keyword, the reference is updated. This allows adaptation of the authentication system to slow age-related and health-related changes of user individual features.

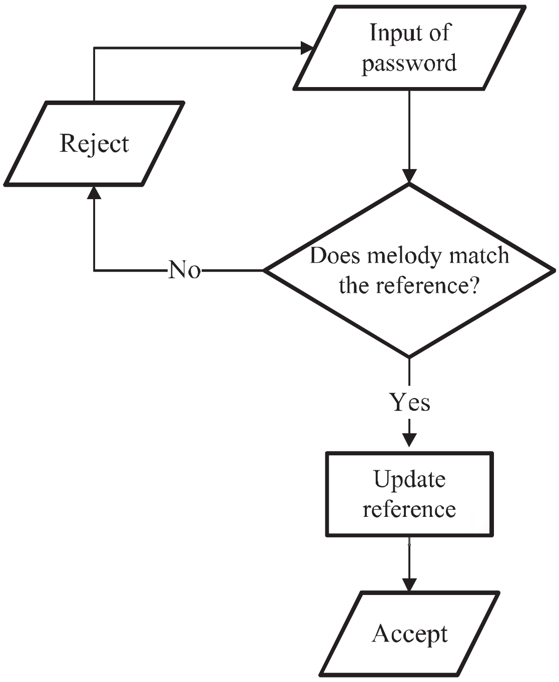


Fig. 1. General scheme of authentication algorithm

III. User model

Even in a case of keyword knowledge tapped intervals exhibit a random behavior (due to natural reasons, non-ideal user sense of tempo, errors in tap detection etc.). Hence, a special statistical analysis is required for studying of these random numbers features. During our experiments we found out that these random variables have a distribution reasonably close to multidimensional normal distribution. In the Fig.2 histograms of registered intervals of a 8-notes keyword are given (see section VII for details). It is worth noting, that our results are similar to those, published in [9] and [10].

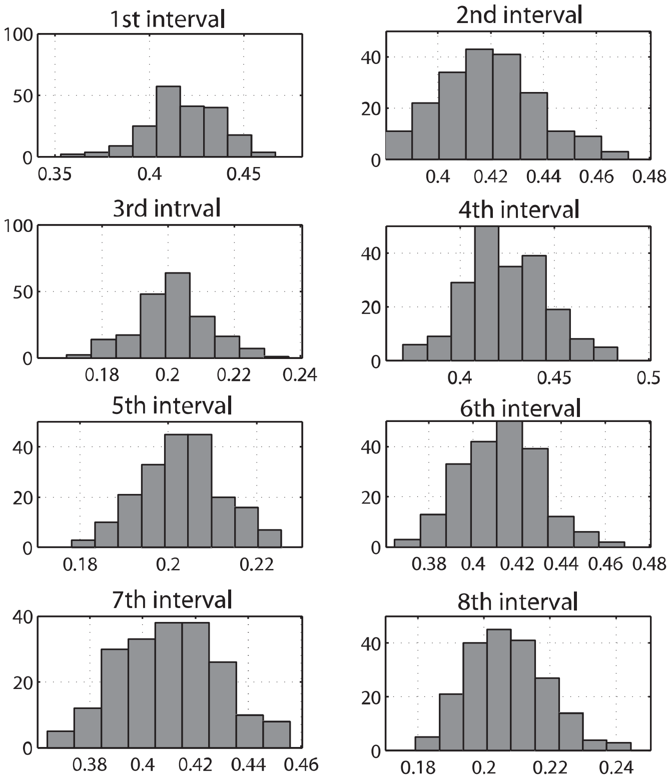


Fig. 2. Histograms of 8 intervals of a keyword

Thus, it was proposed to use a normal approximation for a joint distribution of tapped intervals of the entered rhythmic keyword:

$$f(\boldsymbol{x}) \approx \frac{1}{(2\pi)^{N/2}\sqrt{|\boldsymbol{R}|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{R}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})^T\right], \tag{1}$$

where N - keyword length (number of notes in a keyword), $\boldsymbol{x}$ - a vector of durations of tapped intervals, $\boldsymbol{\mu}$ - vector of mean values of intervals durations, $\boldsymbol{R}$ - a covariance matrix of durations of keyword intervals. Thus, for a system of authentication each user is characterized with unique function (1), which components $\boldsymbol{\mu}$ and $\boldsymbol{R}$ reflect the melody chosen by the user as a keyword and specific features of user tapping of this melody.

IV. Authentication algorithm

A processing of entered notes $[n_1, n_2, ...n_N]$ is trivial: if all the notes match to a reference, then this part of checking is considered to be passed. For intervals $[\Delta t_1, \Delta t_2, ...\Delta t_N]$ checking we propose the following decision-making algorithm. After tapping the keyword, an algorithm calculates a "pseudo" likelihood metric for the interval sequence and compares it with a threshold (see equation 2). We use a prefix "pseudo", because function (1) is just an approximation of a real distribution.

$$Accept = I\left\{f(\boldsymbol{\Delta t}) \geq h\right\} \tag{2}$$

, where $I\left\{\cdot\right\}$ denotes indicator of event.

Considering (2) the following equivalent decisision-making rule can be ontained:

$$Accept = I\left\{(\boldsymbol{\Delta t} - \boldsymbol{\mu})\boldsymbol{R}^{-1}(\boldsymbol{\Delta t} - \boldsymbol{\mu})^T \leq h_0\right\}. \tag{3}$$

A threshold h_0 is chosen by user according to his or her preference to a trade of between keyword steadiness and false rejection rate. As one can see a means vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{R}$ are used during authentication process. They can be estimated by means of preliminary training procedure which is to be fulfilled before starting of authentication system usage. During this training procedure a user enters a desired keyword K times, thus generating K reference sequences:

$$\begin{aligned} \boldsymbol{\Delta t}^{\langle 1\rangle} &= \left[\Delta t_1^{\langle 1\rangle}, \Delta t_2^{\langle 1\rangle}, \ldots, \Delta t_N^{\langle 1\rangle}\right] \\ \boldsymbol{\Delta t}^{\langle 2\rangle} &= \left[\Delta t_1^{\langle 2\rangle}, \Delta t_2^{\langle 2\rangle}, \ldots, \Delta t_N^{\langle 2\rangle}\right] \\ \ldots \\ \boldsymbol{\Delta t}^{\langle K\rangle} &= \left[\Delta t_1^{\langle K\rangle}, \Delta t_2^{\langle K\rangle}, \ldots, \Delta t_N^{\langle K\rangle}\right] \end{aligned} \tag{4}$$

After the training procedure keyword components are calculated as follows:

$$\begin{aligned} &\boldsymbol{\mu}_i = \tfrac{1}{N}\sum_{k=1}^{K} \Delta t_i^{\langle k\rangle}, i = \overline{1,N} \\ &\boldsymbol{R}_{i,j} = \tfrac{1}{N-1}\sum_{k=1}^{K} \left(\Delta t_i^{\langle k\rangle} - \boldsymbol{\mu}_i\right)\left(\Delta t_j^{\langle k\rangle} - \boldsymbol{\mu}_j\right), i,j = \overline{1,N} \end{aligned} \tag{5}$$

Moreover after every successive keyword entrance the components $\boldsymbol{\mu}$ and $\boldsymbol{R}$ are to be updated for tracking of changes in user keyword entering manner (see Fig.3). For the update the oldest reference sequence 4 is erased and changed to the latest successfully accepted one.

V. Implementation

For the presentation and testing of the algorithm a demo application has been implemented for a smartphone based on OS Android 4.1.

The application consists of one screen (main activity) where two octaves of piano keys are shown. Also there are a menu buttons which are used for registration, authentication of users and deleting of reference data (see Fig.4).

Keyword sequence recording starts with the tap of the first note and ends after pressing the "save" button. You must enter the melody several times to register a new user until the message on successful registration appears (see Fig.5). Notes identifiers and the interval between tapping (for the first note timing is initialized to zero) are stored in the user template after each attempt. If the sequence of notes is not the same as the previous attempt then an error message will be displayed. Attempt is reset after 5 seconds pause (a similar effect is produced after pressing "cancel"-button).

After playing a melody the decision of a successful or unsuccessful authentication is made based on the proposed algorithm. If the authentication is successful the reference data ($\boldsymbol{\mu}$ and $\boldsymbol{R}$) is updated by new statistics (see Fig.6).

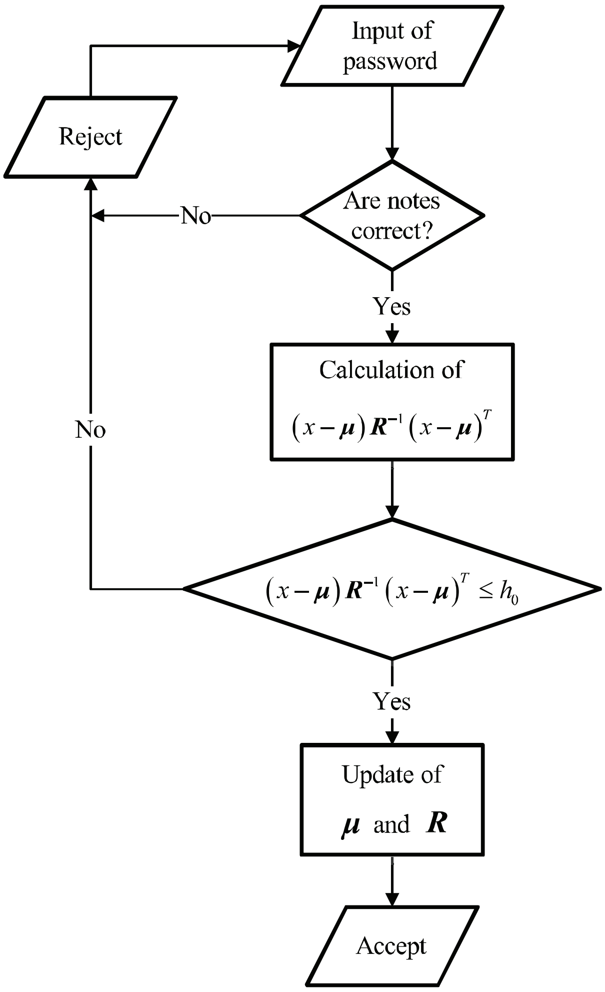

Input of password
Reject
No
Are notes correct?
Yes
Calculation of
$(x-\boldsymbol{\mu})\boldsymbol{R}^{-1}(x-\boldsymbol{\mu})^T$
No
$(x-\boldsymbol{\mu})\boldsymbol{R}^{-1}(x-\boldsymbol{\mu})^T \leq h_0$
Yes
Update of
$\boldsymbol{\mu}$ and $\boldsymbol{R}$
Accept

Fig. 3. Detailed scheme of authentication algorithm

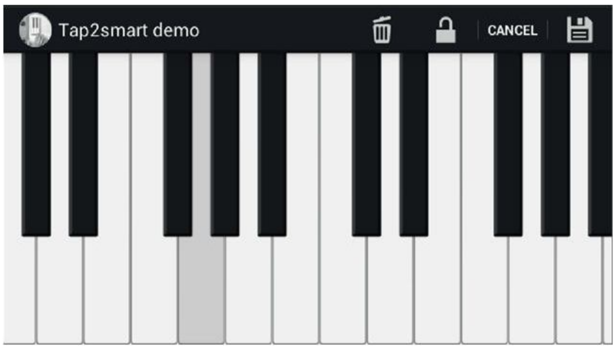


Fig. 4. Application interface

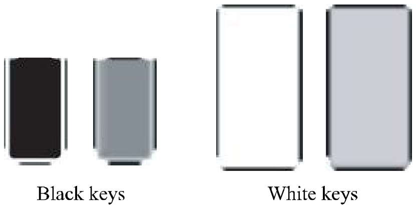
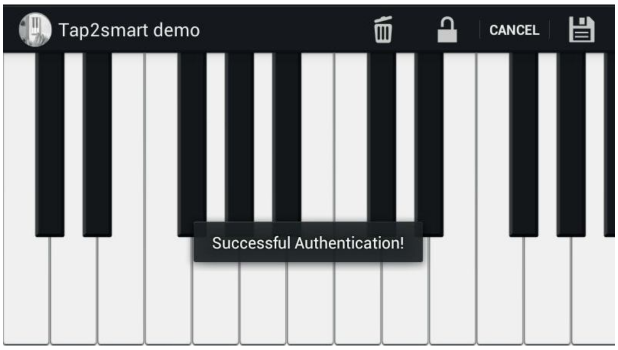


Fig. 6. Images of keys



Successful Authentication!

Fig. 5. Examples of information messages

One of the challenges in creating an application is drawing a piano keyboard [12], [11]. To create an application a derived class of a View class has been used. Standard Android tools have been used for drawing the keys: selector, nine-patch drawable.

For xml-selector images of the keys were prepared in two states: pressed and unpressed (see in Fig.6).

Array of objects ”Key” is initialized. Each object stores the following information:

- two coordinates of key which also determine a size of the keys;
- type of key (black or white);
- status (pressed, unpressed);
- identifier which is used for keyword sequence or for playing sound. Key identifier is calculated by index of objects in the array.

At the beginning the first octave is built then is cloned and moved along the X axis for drawing the second octave. Drawing of each octave takes place in 2 stages to avoid potential overlap: first the white keys then the black keys.

After drawing keys, the user gestures are tracked. GestureDetector class is used to handle ”clicks”. Method onDown sends the pressed-event coordinates to the keyboard which find the pressed key. If the key is found that it is marked as pressed.

The method getTime of Date class is used to measure the time between the pressing events. getTime returns the number of milliseconds since January 1, 1970, 00:00:00, than the difference in time between the events is calculated.

This application have been designed for demonstration and testing the proposed algorithm. Further the development of lock screen application of Android smartphone is planed. Also musical arrangement is planned for each key. This option may be more attractive for potential users but adds the danger of ”eavesdropping”.

VI. Template Protection

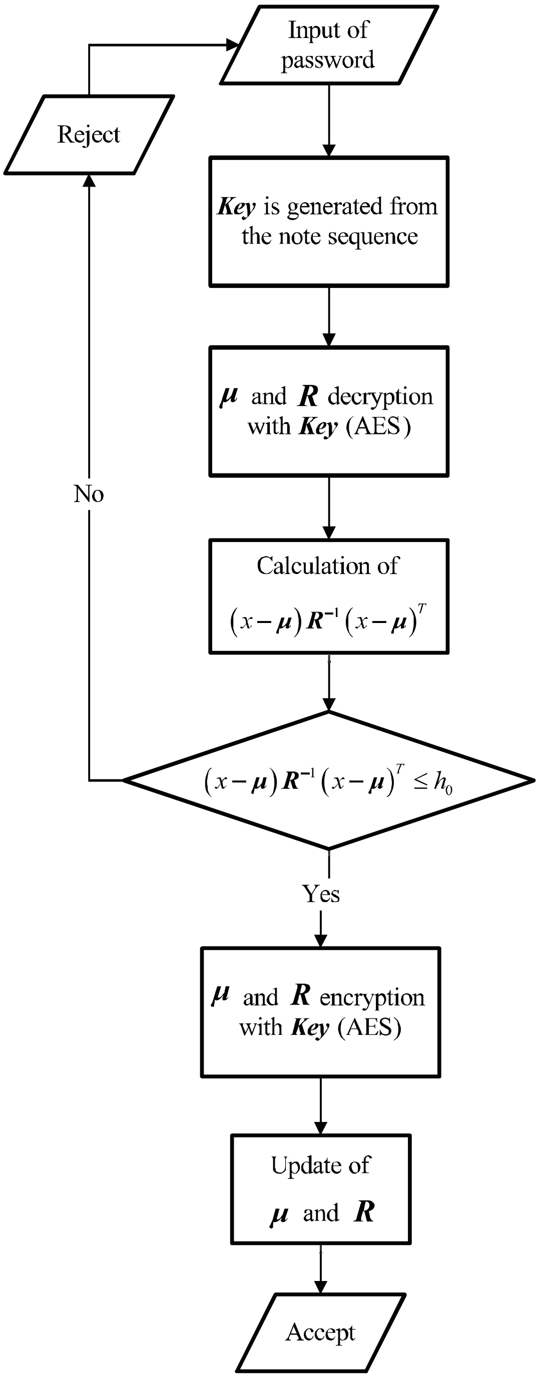

Input of password
Reject
Key is generated from the note sequence
$\boldsymbol{\mu}$ and $\boldsymbol{R}$ decryption with *Key* (AES)
No
Calculation of $(x-\boldsymbol{\mu})\boldsymbol{R}^{-1}(x-\boldsymbol{\mu})^T$
$(x-\boldsymbol{\mu})\boldsymbol{R}^{-1}(x-\boldsymbol{\mu})^T \le h_0$
Yes
$\boldsymbol{\mu}$ and $\boldsymbol{R}$ encryption with *Key* (AES)
Update of $\boldsymbol{\mu}$ and $\boldsymbol{R}$
Accept

Fig. 7. Detailed final scheme of authentication algorithm

The considered algorithm is a keyword system (something the user have). Biometric identifier (template, something the user is) can not be reproduced without direct user intervention [5]. Storing user keywords as a cleartext can result in a security breach if the keyword file is compromised. For a keyword protection a hash function is used. To authenticate a user, the keyword entered by the user is hashed and compared with the stored hash.

Since a hash function is not reversible it cannot be used in proposed algorithm. In application we proposed to store an user template with encryption by AES standard. The keyword notes are used for AES key generation. and the template consist only of the arrays $\boldsymbol{\mu}$ $\boldsymbol{R}$. Thus the final scheme of proposed algorithm is as follows (see Fig.7).

As one can see from the pictures, user data is not stored as a cleartext. Note, that a notes sequence is not stored at all.

VII. Test results

It can be easily shown, that for the proposed method false acceptance rate is a probability of malefactor guessing of notes, intervals and tempo. Hence, it can be estimated in a following way:

$$FAR \leq 24^{-N} 14^{-N} Pr\{\text{Tempo is guessed}\}. \qquad (6)$$

Here we assume, that two octaves are used for a keyword tune, and that only 14 most common intervals present in this tune. The last probability mostly depends on a chosen threshold h_0.

For testing of proposed scheme the dependencies of false acceptance rate (FAR) and false rejection rate (FRR) on threshold value were investigated. In this experiments we simulated a following "worst-case" malefactor:

1) A malefactor knows a melody, used as a keyword (!), but doesn't know a tempo.
2) Trying to get an access a malefactor tries to pick a right tempo choosing it as a uniformly distributed random variable in interval $[40, 200]$ bits per minute (BPM). This range contains all the commonly used today tempos.

The keyword was based on a rhythm of a fragment of a Carmen habanera from G. Bizet opera (see Fig.8).

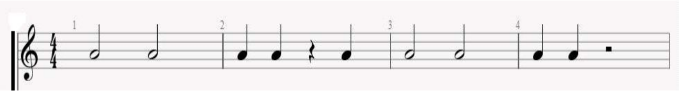


Fig. 8. keyword, used in experiment

The simulation results are given in Fig.9 and Fig.10. As one can see the proposed algorithm provides a level of both probabilities at about 0.1.

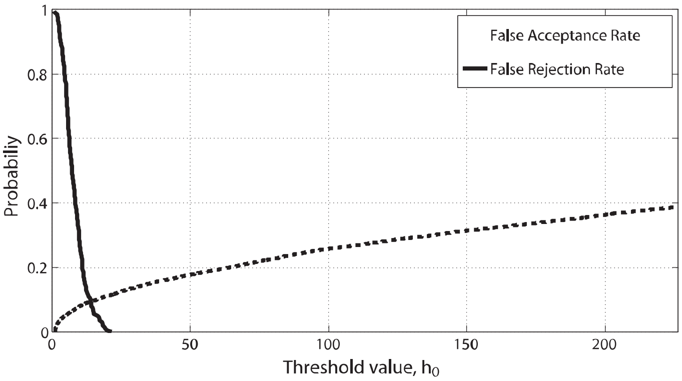


Fig. 9. Dependencies of FRR and FAR on a threshold

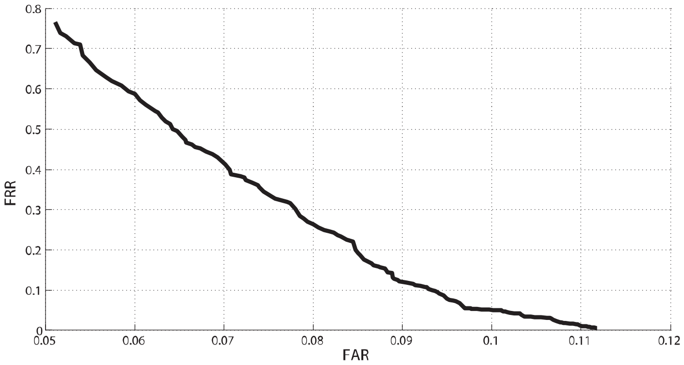


Fig. 10. FRR/FAR locus

VIII. Conclusion

In the paper a novel authentication method for smartphones was proposed and investigated. By means of statistical analysis and simulation it was found, that in a worst case, even when a malefactor knows the keyword tune, the scheme provides $FAR \approx 0.1$ at $FRR \approx 0.1$. In a real case when the tune is unknown FAR level would sufficiently lower. The scheme would be sufficiently effective for the users with a good sense of rhythm, who can use complex rhythmic patterns in a keyword.

As a continuation of this work the following steps can be proposed:

- extensive statistical analysis of the scheme;
- building of more accurate user model and, consequently, more effective decision making algorithms;
- development of more realistic malefactor model for a more accurate FAR estimation;
- implementation of a lock screen application on the basis of the proposed algorithm is planed.

References

[1] iPhone/iPod Piano Passcode: https://www.youtube.com/watch?v=eDW0bbDq2so.

[2] LG G3, User Guide, 2014.

[3] S. Trewin, C. Swart, L. Koved, J. Martino, K. Singh, and S. Ben-David. Biometri authentication on a mobile device: A study of user effort, error and task disruption. *In Proc. ACSAC*, 2012.

[4] Mohammad Omar Derawi. Smartphones and Biometrics. Gait and Activity Recognition. Doctoral Dissertations at Gjovik University College 3-2012.

[5] Richard E. Smith. Authentication: From keywords to Public Keys. Addison-Wesley Professional, 2001.

[6] Maxim Grankin, Elizaveta Khavkina, Alexander Ometov. Research of MEMS Accelerometers Features in Mobile Phone. *Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Travel*, 2012

[7] E. Andreeva, "Secret sharing in continuous access control system, using heart sounds," *IEEE Proceedings of the 13th International symposium on problems of redundancy in information and control system*, 2012

[8] Eric Cheung, Thomas M. Smith. Patent EP 1469372 A2. User authentication using rhythmic keywords, US, 2004.

[9] Jacob O. Wobbrock, TapSongs: Tapping Rhythm-Based keywords on a Single Binary Sensor. *22nd Symposium on User Interface Software and Technology, Canada*, 2009

[10] J. Mates, U. Muller, T. Radil, and E. Poppel, Temporal integration in sensorimotor synchronization. *J. Cognitive Neuroscience 6 (4)*, 1994.

[11] Training for Android developers, sets of lessons, Web: http://developer.android.com/training/index.html.

[12] Creating a custom component from scratch, Web: http://habrahabr.ru/post/176643/