Authentication by Mapping Keystrokes to Music: The Melody of Typing

Amith K. Belman^{*}, Tirthankar Paul[†], Li Wang[‡], S. S. Iyengar[‡], Paweł Śniatała[§],

Zhanpeng Jin^{||}, Vir V. Phoha^{*}, Seppo Vainio[†] and Juha Roning[†]

*Syracuse University, {akamathb,vvphoha}@syr.edu

[†]University of Oulu, {tirthankar.paul,seppo.vainio,juha.roning}@oulu.fi

[‡]Florida International University, {lwang059,iyengar}@cs.fiu.edu

[§]Poznań University of Technology, pawel.sniatala@put.poznan.pl

^{II}University at Buffalo, SUNY, zjin@buffalo.edu

Abstract—Expressing Keystroke Dynamics (KD) in form of sound opens new avenues to apply sound analysis techniques on KD. However this mapping is not straight-forward as varied feature space, differences in magnitudes of features and human interpretability of the music bring in complexities. We present a musical interface to KD by mapping keystroke features to music features. Music elements like melody, harmony, rhythm, pitch and tempo are varied with respect to the magnitude of their corresponding keystroke features. A pitch embedding technique makes the music discernible among users. Using the data from 30 users, who typed fixed strings multiple times on a desktop, shows that these auditory signals are distinguishable between users by both standard classifiers (*SVM*, *Random Forests and Naive Bayes*) and humans alike.

Index Terms-Keystroke, Mapping, Music, Authentication

I. INTRODUCTION

Behavioral biometrics has seen an upsurge in research and applications in the recent past. Behavioral biometrics such as keystrokes [1]–[3], touch and swipe [4]–[6], gait patterns [7]–[9] and wrist movement patterns [10], [11] have been shown to be good second-factor authentication techniques. As humans have well developed auditory sense, representation of visual information in sound and vice versa has been of great interest to researchers [12]–[15]. But such alternative interfaces to convey biometric information have not been explored. Mapping of information to sound can lead to deeper interpretations of user biometrics which motivates our work. But various issues like varied feature space, keystroke latency timings and human interpretability of the music complicate this mapping.

A. Contributions

Our key contributions are; *a*) *Keystroke signature to musical signature:* We present a method to map keystroke features to music notes which can be used as a musical signature. Using two modified functions to compute a note's duration and pitch, we are able to derive the musical equivalent of a keystroke signature. *b*) *Analysis of inter-user and intrauser distances between music samples:* We analyze the efficiency of music files for user verification using inter-user and intra-user distances between two key vectors; note-pitch and note-duration. *c) Human discretion results:* Results from human subjects with little to no formal background in music, performing verification based on the music files of user are presented. Human classifiers were trained by listening to music files to verify users. User-wise accuracies, Type 1 errors (false rejects) and Type 2 errors (false accepts) are presented. *d) Extendable to other biometrics:* Our designed procedure to map keystroke features to musical signatures for a user is portable to other forms of behavioral biometrics, such as gait, swipes and wrist movements, with some modifications.

II. RELATED WORK

The benefits of transforming information between the visual and aural senses has been studied in various contexts other than biometrics. Meijer [14] designed and evaluated a system that represented image information in form of sound. Inverse mapping (sound-to-image) mapping experiments showed convincing evidence for the preservation of visual information through the transformation. Kim [12] explored the other direction of information mapping by presenting techniques to represent sound in form of visual images. A few studies use such mapping techniques and extend existing systems to be more accessible to the visually impaired. Matta et al. [13] proposed a theoretical system that provided auditory image representations as an approximate substitute for vision, whereas Rigas and Memery [15] used both audio-visual stimuli to communicate information in browsing e-mail data. Both studies found auditory representation of data to be useful not only for the blind but also to maximize the volume of information communicated.

The association of keystrokes to sound has been explored by few researchers, only in context of the acoustic emanations that occur while a user is typing. Zhuang et al. [16] show how keyboard acoustic emanations from 10 minute recordings can be used to attack and recover up to 96% of typed characters. Roth et al. [17] proposed keystroke sound as a modality of authentication of users in a continuous authentication scenario and discuss the shortcomings and possibility of better features in their work. In another attack focused

This work was supported in part by the US National Science Foundation (US NSF) under Grant SaTC-1527795 and US Army Research Office under Grant W911NF-15-1-0572.

work by Zhu et al. [18], off-the-shelf smartphones are used to record keystroke emanations. The authors use Time Difference of Arrival (TDoA) method and show that more than 72% of keystrokes can be recovered without any context-based information. A similar study by Liu et al. [19] performed better by recovering 94% of keystrokes with acoustic emanations and discrimination of mm-level position differences that help locate origin of keys on a keyboard. Another work by Roth et al. [20] investigated the discriminative power of these keystroke emanations, with an EER of 11% they conclude that there is promising discriminative information in the keystroke sound to be further explored. These works primarily focus on acquiring sound at the point of typing, which might not be audible or easily understandable to make meaningful interpretations by humans.

In a similar musical mapping work by Paul et al. [21] proposed a method to generate personalized music from DeoxyriboNucleic Acid (DNA) signatures of users. The number of Short Tandem Repeats (STRs) and the STR sequences were used as the units mapped to musical elements.

III. METHODS

A. Data Collection

The typing data was collected from 30 participants at our University after the IRB approval. The participants consisted of 13 females and 17 males, aged from 19 to 28. All participants were right-hand dominant and fluent in English. Twelve participants indicated that they were touch typists while the rest indicated to be visual typists. The participants performed the following activities on a desktop, with a standard QW-ERTY keyboard: multiple brief and interleaved sections of transcription, free-text typing, browsing and online shopping. This is a subset of SU-AIS BB-MAS dataset [22].

We focus on the transcription activities and only extract the data generated from the users while typing the phrase "this is a test" (hereinafter referred to as "test-phrase"). All users typed the test-phrase at many different points in their session, a minimum of 30 occurrences for each user were extracted from the data. We consider each occurrence of the test-phrase as one sample from the user.

B. Keystroke features

For any two consecutive keys K_i and K_{i+1} the following values are recorded: K_iPress and $K_{i+1}Press$, the UNIX standard time at which K_i and K_{i+1} are pressed respectively. $K_iRelease$ and $K_{i+1}Release$, the UNIX standard time at which K_i and K_{i+1} are released respectively. From these recorded time-stamps , the following temporal features are extracted:

- $Keyhold_{K_i}$: $K_iRelease$ K_iPress
- $Flight1_{K_iK_{i+1}}: K_{i+1}Press K_iRelease$
- $Flight_{K_iK_{i+1}} : K_{i+1}Release K_iRelease$
- $Flight3_{K_iK_{i+1}}: K_{i+1}Press K_iPress$
- $Flight_{K_iK_{i+1}}: K_{i+1}Release K_iPress$

The features are in milliseconds and Figure 1 illustrates these temporal features. A dictionary D_{us} where u and s



Fig. 1: Features extracted from the temporal data of keys K_i and K_{i+1} .

are the user and sample numbers respectively, both ranging from 1 to 30. D_{us} consists of multiple Key : Value pairs in form $X_f : v$. In case of the unigraphs (t, h, i, s, a, e, space in test-phrase) in the sample X is the unigraph, f = Keyholdand v is the respective duration (in milliseconds). In case of digraphs ((t, h), (h, i), ..., (s, t)), X is the digraph, $f \in$ $\{Flight1, Flight2, Flight3, Flight4\}$ and v is the respective duration. v is averaged, if unigraph or digraph has multiple occurrences in sample (for example t, h, space in test-phrase).

C. Music features

Keystroke data consists of keyhold times and inter-key (flight) latencies grouped by their keys of origin whereas music is generated from notes of different pitch and duration played in a certain pattern. Mapping data between these two very different modes of information is complicated as the magnitudes of the keystroke features, information of their different keys of origin, repeating key presses and many other such peculiarities cannot be expressed with simple equivalencies in the music domain.

Therefore, we shortlist the elements of music [23] that can be manipulated to create music from the keystroke features. We chose the following; Melody: the tune generated due to successive single notes affected by pitch and rhythm. Harmony: sound produced by two or more notes played simultaneously. Rhythm: combinations of sounds of varied length. Pitch: sound varied with the frequency of vibrations. Tempo: speed at which the music is played. By controlling the pitch and duration of musical notes, we can manipulate these five elements of music.

The MIDI protocol is a message-based communication between computer and equipment. The MIDI protocol was initially made to create polyphonic sound by using multiple musical devices once, linked with cable, in the music industry [24]. To create music that complies with MIDI standard we use *Ken Schutte MIDI Matlab Toolbox* [25]. The *matrix2midi* module takes a N * 6 matrix and converts it to MIDI format which is then written to a MIDI file using the *writemidi* module. N rows of the matrix represent the notes (one for each note) and the 6 columns represent the *track number*, *channel number*, *note number* (midi encoding of pitch), *velocity* (volume), *start time* (seconds), *end time* (seconds)



Fig. 2: Music notes and their placement, generated from a digraph K_i, K_{i+1} using functions T(v) and P(v) for duration and pitch respectively.

respectively. For simplicity, we set track number = 1, channel number = 1(piano), and volume = 75 to be constants. By varying the pitch of a note and its duration we generate MIDI files M_{us} for each D_{us} .

D. Mapping keystroke features to music

To generate M_{us} we compute their MIDI matrices of shape N * 6, we can denote M_{us} as:

$$M_{us} = [t \vec{n}_{us}, c \vec{n}_{us}, \vec{p}_{us}, \vec{v}_{us}, \vec{st}_{us}, \vec{et}_{us}]$$
(1)

Where $t\vec{n}_{us}$ is track number, $c\vec{n}_{us}$ is channel number, \vec{p}_{us} is pitch, \vec{v}_{us} is volume, \vec{st}_{us} is start time, and \vec{et}_{us} is end time and all the vectors are of the same length N (number of notes). As explained earlier, $t\vec{n}_{us} = \vec{1}$, $c\vec{n}_{us} = \vec{1}$ and $\vec{v}_{us} = \vec{75}$. \vec{p}_{us} , \vec{st}_{us} and \vec{et}_{us} are mapped from D_{us} using our embedding technique to enhance the user-specific information held in D_{us} .

Each digraph $\langle K_i, K_{i+1} \rangle$ in D_{us} (in alphabetic order) is mapped to music notes by converting its six associated keystroke values (*Keyhold* of K_i, K_{i+1} and their four *flight* values) with *duration function* T(v) and *pitch function* P(v). T(v) is a simple scaling function, to scale the keystroke feature (in milliseconds) to practical music note duration (in seconds). P(v) is a modified form of MIDI Tuning Standard (MTS) which is specified in the MIDI protocol [26]. The two functions are shown below:

$$T(v) = v/100 \tag{2}$$

$$P(v) = 69 + 12\log_2\left(\frac{v}{440}\right) \tag{3}$$

where v is the value from the X_f : v pairs in D_{us} . We substitute v in place of the frequency in the standard MTS equation. Since 440 Hz is a widely used standard concert A (musical note), equation (3) uses the term log_2 (v/440) to compute the number of octaves above or below the concert A. This term is multiplied by 12 to compute the semitones above the concert A. MIDI represents the concert A with integer 69 which is added for a MIDI compliant pitch number. After computing duration and pitch of the notes, the notes are arranged similar to their occurrences over the duration of a digraph to obtain the vectors \vec{p}_{us} , \vec{st}_{us} and \vec{et}_{us} . As Flight4 translates to the longest duration, all other notes overlap with it at different points. \vec{et}_{us} is computed as \vec{st}_{us} + T(v), notes corresponding to $Keyhold_{K_i}$, $Flight3_{K_iK_{i+1}}$, $Flight4_{K_iK_{i+1}}$, have the same values in \vec{st}_{us} (same start time). $Flight1_{K_iK_{i+1}}$ and $Flight2_{K_iK_{i+1}}$ have the same values in \vec{st}_{us} equal to \vec{et}_{us} values of $Keyhold_{K_i}$. $Keyhold_{K_{i+1}}$ has its \vec{st}_{us} value equal to the \vec{et}_{us} value from $Flight1_{K_iK_{i+1}}$.

Figure 2 illustrates the mapping, start time and end time of music notes generated using digraphs from D_{us} . These notes (from all digraphs in alphabetical order) when played in a sequence produce a musical tune. Figure 3 shows a collection of piano roll plots which were generated for different samples of test-phrase using our procedure. The highlighted sections represent a played note. We can observe that plots 3a and 3b appear to be similar (similar sounding music) to each other. Both were generated from different samples by the same user. Figures 3c and 3d show the same for a different user. This example is representative of observations on our entire dataset.

IV. ANALYSIS ON MUSIC FROM KEYSTROKES

We performed inter-user and intra-user distance analysis and user-music verification using random forests, naive bayes and SVM. But as standard classifiers do not differentiate between musical notes and other forms of data, we also preform verification experiments with three human-classifiers detailed below.

A. Inter-user and intra-user analysis

In each music file M_{us} , vectors $t\vec{n}_{us} = \vec{1}$, $c\vec{n}_{us} = \vec{1}$ and $\vec{v}_{us} = 75$ are constant. Therefore we perform the inter-user and intra-user analysis using only the \vec{p}_{us} , \vec{st}_{us} and \vec{et}_{us} vectors. As order of the notes in all music files are same, vectors \vec{st}_{us} and \vec{et}_{us} can be simplified to a single vector $\vec{d}_{us} = \vec{et}_{us} - \vec{st}_{us}$. \vec{p}_{us} denotes the pitch of the notes and \vec{d}_{us} denotes their duration. We chose Canberra distance as it is most suitable



Fig. 3: Examples of the piano roll plots that are obtained after mapping the keystroke features to the music features. We illustrate the piano roll plots of two test-phrase samples from two random users from our data-set, Figures 3(a) and 3(b) are from samples of user A and Figures 3(c) and 3(d) are from user B.



(b) Distances using note-duration vectors \vec{d}_{us} .

Fig. 4: Plot of density functions for inter-user and intra-user Canberra distances of the note-pitch vectors (4a) and noteduration vectors (4b) between all music files.

when dealing with vectors. The Canberra distance between two vectors \vec{a} and \vec{b} is given by:

$$c(\vec{a}, \vec{b}) = \sum_{i=1}^{n} \frac{|a_i - b_i|}{a_i + b_i}$$
(4)

Figure 4 shows the density functions of the inter-user and intra-user distances from all music files. Figure 4a is plotted

with distances using \vec{p}_{us} while figure 4b is using \vec{d}_{us} from all the music files respectively. We observe that the density curves for intra-user distance falls majorly towards the left, implying lesser intra-user differences in music, for both cases. In contrast, the density curves for inter-user distances are fall towards the right with higher distance values. We can also observe the overlapping regions between the intra-user and inter-user density curves is small. Small intra-user distances, large inter-user distances and small overlap among these curves are all desirable qualities for user verification. These qualities imply that music files of a user are fairly separable from other users.

B. Verification experiments with standard classifiers

Even though We use the note-pitch vector ($\vec{p_{us}}$) and the note-duration vector ($\vec{d_{us}}$) as feature vectors for the music files. For the verification experiments, we use three different classifiers; Random Forests, Naive Bayes and SVM. Due to our dataset consisting of 30 music files for each user, we run the experiments with two different configurations; two-fold and three-fold cross validation. For each session 30 imposter samples are sampled randomly from users other than the genuine user.

Random forest classifier with five trees, maximum depth was restricted to five and number of child nodes was restricted to two. GINI impurity was used for the split criterion. In SVM classifier we use a RBF kernel, penalty parameter = 1 and gamma = 0.01. The Gaussian Naive Bayes (GNB) classifier implements the Gaussian Naive Bayes algorithm as shown by the following equations:

$$\hat{y} = \underset{y}{\operatorname{argmax}} P(y) \prod_{i=1}^{n} P(x_i|y)$$
(5)

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{6}$$

TABLE I: The average FAR, FRR and Accuracy; for the three standard classifiers with two-fold and three-fold cross validation experiments (on the left) and for human classifiers (on the right) on user verification.

	Standard Classifiers				Human Classifiers			
Metrics Average		Random Forest	Naive Bayes	SVM		HC1	HC2	НС3
FAR	2-Fold	0.06 ± 0.05	0.12 ± 0.07	0.14 ± 0.08	Individual	0.18 ± 0.11	0.20 ± 0.10	0.17 ± 0.11
FRR		0.09 ± 0.07	0.10 ± 0.09	0.14 ± 0.10		0.18 ± 0.10	0.19 ± 0.11	0.17 ± 0.12
Accuracy		0.92 ± 0.0	0.89 ± 0.06	0.89 ± 0.07		0.81 ± 0.09	0.83 ± 0.10	0.81 ± 0.12
FAR	3-Fold	0.06 ± 0.07	0.05 ± 0.06	0.04 ± 0.07	Overall		0.18 ± 0.11	
FRR		0.03 ± 0.05	0.03 ± 0.05	0.04 ± 0.08			0.18 ± 0.11	
Accuracy		0.96 ± 0.04	0.94 ± 0.05	0.96 ± 0.05			0.81 ± 0.10	



(a) False rejection rates or Type-1 error.



(b) False acceptance rates or Type-2 error.



(c) User-wise accuracies for verification.

Fig. 5: Results from the Human-Classifier (HC) based verification experiments.

C. Verification experiments with human classifiers

where, y is the class, \hat{y} is the predicted class, $x_1, x_2, ..., x_n$ are the features, σ_y and μ_y are the standard deviation and mean estimated using maximum likelihood. Table I presents the results of our verification experiments. We observe that the Naive Bayes and SVM classifiers perform slightly better with more number of training instances. However, all three classifiers performed similarly with high accuracy between 89% to 96%. The False Rejection Rates (FRR) and False Acceptance Rates (FAR) were low ($\leq 10\%$) in all cases except for two-fold experiments with Naive Bayes and SVM.

As these classifiers do not differentiate between music and other numerical data, these results do not highlight the merits of converting a user's keystroke data into music. Therefore, a human classifier based experiments were carried out, detailed below.

The true test and application of our work is to see if an average human can differentiate between the music generated from one user's typing sample to another. We recruited three volunteers (hereinafter referred to as human-classifier or HC) with little to no formal education in music. Each experiment session had a training phase and a testing phase. A user number was selected for each session and all the samples from that user were labeled "genuine" for the session. In the training phase the HC was made to listen to 15 music files from the genuine user. In the following testing phase a set of 30 music files, consisting of 15 genuine (not used in training) and 15 imposter (randomly selected from other users) files were played one after the other. At the end of each file HC classified it as either genuine user or as an imposter. The classification decisions were recorded and analyzed.

Figure 5 and Table I summarize the results of our verification experiments carried out with three HCs. Figures 5a and 5b show the Type-1 and Type-2 errors committed by the HCs respectively. Type-1 error is the case where a HC falsely rejected a genuine music sample (rejection of a true null hypothesis). Whereas, Type-2 error is the case where a HC falsely accepted an imposter's music sample (failure to reject a false null hypothesis). Figure 5c shows the user-wise accuracy of the HCs in the verification task. We observe that verifying a few users was challenging for all three HCs, especially users 6,12 and 25, reflected in the high FAR and FRR for these users. For all other users, the FAR and FRR values are low, within range of 10% to 15% in most cases. In a few cases the FAR and FRR values are 0, indicating a perfect classification by the HC. The accuracies shown in the figure 5c reflect similarly with most being in the range of 75% to 85% while being low on users 6, 12 and 25. Overall, all three HCs could easily verify the music files of users with high accuracies for most users.

V. CONCLUSION

Our work shows that information from keystroke dynamics can be translated to other representations that maintain or enhance the human interpretability of the data. A theoretical system to convert keystroke features to the aural sense has been proposed. Information on KD is conveyed through auditory music files. We show that user-specific music files exhibit high inter-user distance and low intra-user distance which is a desired quality in a feature vector to be used for authentication. Verification experiments with standard classifiers (Random Forests, Naive Bayes and SVM) show that these music files can be verified with high accuracies despite treating music as any other numerical data. Experiments with different devices, types of text and music fluency of the human classifiers are part of our future research direction.

Results from human-classifiers reveal that a user's keystroke behavior can be converted to music which is humanly verifiable. The musical melody created from each user's keystroke data using our approach has a clearly distinguishable tune unique to a user. The approach and findings of this work can be used in a variety of ways such as; a new mode of second-factor user authentication, a complementary form of data presentation for audible user specific keystroke signatures. The concept of mapping biometrics features to music can be reused with some modifications to suit other forms of biometrics such as gait, touch, swipe and fingerprints to name a few.

REFERENCES

- [1] K. S. Balagani, V. V. Phoha, A. Ray, and S. Phoha, "On the discriminability of keystroke feature vectors used in fixed text keystroke authentication," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 1070 – 1080, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865511000511
- [2] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation computer systems*, vol. 16, no. 4, pp. 351–359, 2000.
- [3] A. Serwadda, Z. Wang, P. Koch, S. Govindarajan, R. Pokala, A. Goodkind, D. Brizan, A. Rosenberg, V. V. Phoha, and K. Balagani, "Scanbased evaluation of continuous keystroke authentication systems," *IT Professional*, vol. 15, no. 4, pp. 20–23, July 2013.
- [4] T. Feng, Z. Liu, K. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen, "Continuous mobile authentication using touchscreen gestures," in 2012 IEEE Conference on Technologies for Homeland Security (HST), Nov 2012, pp. 451–456.
- [5] S. Mondal and P. Bours, "Swipe gesture based continuous authentication for mobile devices," in 2015 International Conference on Biometrics (ICB), May 2015, pp. 458–465.
- [6] A. Serwadda, V. V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems, BTAS 2013, Arlington, VA, USA, September 29 - October 2, 2013, 2013, pp. 1–8. [Online]. Available:* https://doi.org/10.1109/BTAS.2013.6712758

- [7] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor." JCP, vol. 1, no. 7, pp. 51–59, 2006.
- [8] D. Gafurov, E. Snekkenes, and P. Bours, "Gait authentication and identification using wearable accelerometer sensor," in 2007 IEEE workshop on automatic identification advanced technologies. IEEE, 2007, pp. 220–225.
- [9] C. Wan, L. Wang, and V. V. Phoha, "A survey on gait recognition," ACM Comput. Surv., vol. 51, no. 5, pp. 89:1–89:35, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?id=3230633
- [10] B. Li, H. Sun, Y. Gao, V. V. Phoha, and Z. Jin, "Enhanced free-text keystroke continuous authentication based on dynamics of wrist motion," in 2017 IEEE Workshop on Information Forensics and Security, WIFS 2017, Rennes, France, December 4-7, 2017, 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/WIFS.2017.8267642
- [11] J. Yang, Y. Li, and M. Xie, "Motionauth: Motion-based authentication for wrist worn smart devices," in 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). IEEE, 2015, pp. 550–555.
- [12] E. M. Kim, "Harmonograph: A visual representation of sound," in 2012 International Symposium on Intelligent Signal Processing and Communications Systems, Nov 2012, pp. 489–494.
- [13] S. Matta, H. Rudolph, and D. K. Kumar, "Auditory eyes: Representing visual information in sound and tactile cues," in 2005 13th European Signal Processing Conference, Sep. 2005, pp. 1–4.
- [14] P. B. L. Meijer, "An experimental system for auditory image representations," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 2, pp. 112–121, Feb 1992.
- [15] D. I. Rigas and D. Memery, "Utilising audio-visual stimuli in interactive information systems: a two domain investigation on auditory metaphors," in *Proceedings. International Conference on Information Technology: Coding and Computing*, April 2002, pp. 190–195.
- [16] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard acoustic emanations revisited," ACM Trans. Inf. Syst. Secur., vol. 13, no. 1, pp. 3:1–3:26, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1609956.1609959
- [17] J. Roth, X. Liu, A. Ross, and D. Metaxas, "Biometric authentication via keystroke sound," in 2013 International Conference on Biometrics (ICB), June 2013, pp. 1–8.
- [18] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 453–464. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660296
- [19] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, "Snooping keystrokes with mm-level audio ranging on a single phone," in *Proceedings of the 21st Annual International Conference* on Mobile Computing and Networking, ser. MobiCom '15. New York, NY, USA: ACM, 2015, pp. 142–154. [Online]. Available: http://doi.acm.org/10.1145/2789168.2790122
- [20] J. Roth, X. Liu, A. Ross, and D. Metaxas, "Investigating the discriminative power of keystroke sound," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 333–345, Feb 2015.
- [21] T. Paul, S. Vainio, and J. Roning, "Towards personalised, dna signature derived music via the short tandem repeats (str): Proceedings of the 2018 computing conference, volume 2," in *Intelligent Computing. SAI* 2018. Advances in Intelligent Systems and Computing, vol 857. Springer, Cham,, 01 2019, pp. 951–964.
- [22] A. K. Belman, L. Wang, S. S. Iyengar, P. Sniatala, R. Wright, R. Dora, J. Balwdin, Z. Jin, and V. V. Phoha, "Su-ais bb-mas (syracuse university and assured information security - behavioral biometrics multi-device and multi-activity data from same users) dataset," 2019. [Online]. Available: http://dx.doi.org/10.21227/rpaz-0h66
- [23] www.cde.ca.gov, "Visual and performing arts framework," 2004, http://www.cde.ca.gov/ci/cr/cf/documents/vpaframewrk.pdf, [Accessed : 14-March-2019].
- [24] B. C. Florea, "Midi-based controller of electrical drives," in *Proceedings* of the 2014 6th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Oct 2014, pp. 27–30.
- [25] http://kenschutte.com/midi, "Ken schutte midi matlab toolbox," 2015, https://github.com/kts/matlab-midi.
- [26] M. M. Association, "Midi tuning specification," http://www.midi.org/techspecs/midituning.php,[Accessed : 14-March-2019].