Using ambient audio in secure mobile phone communication

Ngu Nguyen, Stephan Sigg, An Huynh, Yusheng Ji
National Institute of Informatics (NII)
Tokyo, Japan
Email: {nlnngu, zanton.zzz}@gmail.com, {sigg, kei}@nii.ac.jp

***Abstract*—We study the use of ambient audio as a seed to non-interactively and communication-free generate a secure cryptographic key among mobile distributed devices. We have implemented an ambient audio-based secure device pairing on android mobile phones. This paper discusses practical problems we encountered related to hardware, environment and time synchronisation, introduces approaches for feature generation and details results from our experimental case studies. In the case study, we consider the impact of inter-device distance on the quality of generated keys.**

I. Introduction

The pairing of unacquainted devices is a problem recently studied by various authors [1]–[4]. A central issue in mobile device pairing is the identification of communicating devices [5]. During a pairing process it is usually not transparent or easy to validate with which remote device a pairing is conducted. One solution to solve this problem is to incorporate contextual or sensor information of mobile devices. Straightforward examples are the *Seeing-Is-Believing* system proposed by McCune et al. [6] and *Loud and Clear*, proposed by Goodrich et al. [7]. In the former, a system that incorporates the camera of a mobile device to capture a 2D-barcode displayed on a second device is proposed to verify the devices identity. The Loud and Clear system utilises a similar scheme but spoken audio. A text message displayed on one device is read aloud by a user and recorded again via speech recognition by a second device. As a further example, Mayrhofer et al. present an authentication mechanism based on accelerometer readings of simultaneous shaking processes of devices [8]. When devices are shaken simultaneously by a single person, authentication was possible but unlikely for a third person mimicking the movement pattern remotely. Also, Mayrhofer derived in [9] that the sharing of secret keys is possible with a similar protocol. The proposed protocol repeatedly exchanges hashes of key-sub-sequences until a common secret is found. In contrast, Bichler et al. describe an approach in which noisy acceleration readings can be utilised directly to establish a secure communication channel among devices [2], [10]. They utilise a hash function that maps similar acceleration patterns to identical key sequences. These approaches are not unobtrusive since they require explicit user interaction. Mathur et al. [11] introduced ProxiMate that enables wireless devices in proximity to pair automatically and securely with each other using a common key generated from their shared ambient wireless signals. In their experiments, the keys are extracted from radio frequency signals broadcast from some USRP (Universal Software Radio Peripheral) devices. The distance which is suitable for the devices to create the common key depends on signal wavelength. For example, in 802.11 based wireless environment, the devices need to be very close to each other, about 6.25 cm. Besides, if using other signals such as FM-radio or TV, the devices need specific hardware.

We propose a synchronisation scheme for devices in proximity based on ambient audio for unobtrusive key generation. A set of devices willing to establish a common key conditioned on ambient audio take synchronised audio samples from their local microphones. Each device then computes a binary characteristic sequence for the recorded audio: An audio-fingerprint. This binary sequence is designed to fall onto a code-space of an error correcting code. In general, a fingerprint will not match any of the codewords exactly. Fingerprints generated from similar ambient audio resemble but due to noise and inaccuracy in the audio-sampling process, it is unlikely that two fingerprints are identical. Devices therefore exploit the error correction capabilities of the error correcting code utilised to map fingerprints to codewords. For fingerprints with a Hamming-distance within the error correction threshold of the error correcting code the resulting codewords are identical and then utilised as secure keys. The Hamming distance tolerated in fingerprints rises with increasing distance of devices. We presented a first discussion of this scheme in [12], [13].

There is a wide area of applications for secure audio-based device pairing. People can connect their smartphones by just put the phones near to each other without stopping their work to verify the PIN code. A file sharing service which only allows data transmission with client devices in a limited region can encrypt information with the common key generated from ambient sound. Besides, this scheme can be utilized to localize mobile devices.

Here, we discuss possible algorithmic solutions for the application on mobile devices in section II. We present our implementation on mobile devices in section III and detail practical problems we encountered in section III-A. Additionally, section III-B describes a case study conducted with smart-phone devices to study the accuracy of the approach in a realistic setting. Section IV states future work and section V draws our conclusion.

978-1-4673-0907-3/12/$31.00 ©2012 IEEE

II. Generating audio-fingerprints in mobile phones

Cano et al. [14] defined an audio-fingerprint as a succinct description generated from a piece of audio. This representation is used to identify and match audio contents [15]–[18]. Azizyan et al. [19] and Tazia et al. [20] also applied audio-fingerprints in indoor localization. Chandrasekhar et al. [21] provided an evaluation of some popular audio-fingerprinting systems for audio search on mobile devices.

Audio fingerprints are generated based on a set of features extracted from audio samples. Haitsma and Kalker [15] claimed that the frequency domain contains the most important features of audio. Therefore, in their proposed fingerprint extraction, they segmented the audio sequence into overlapping frames and applied a Fourier transform on every frame. They selected 33 non-overlapping frequency bands whose width is logarithmic scale to form a 32-bit sub-fingerprint value for each frame. These sub-fingerprints are made of bits that represent the energy change of the audio signal.

A commercial audio search algorithm based on spectrogram peaks was introduced in Wang's paper [16]. In this algorithm, the audio segment was converted to a spectrogram and local amplitude peaks are chosen to form a sparse set of time-frequency points. Peaks are combined with a number of their neighbours to create fingerprint hashes. Ogle and Ellis [22] adapted this approach and identify recurrent sound events.

Waveprint, a novel audio fingerprint extraction method based on computer vision techniques, was introduced in [18]. The authors generated spectral images of the audio input. For each of these images, top wavelets are extracted. Then, they reduced the wavelets to binary representation. Finally, the Min-Hash procedure produced the final sub-fingerprints. Locality-Sensitive Hashing technique was used in the matching process.

Previous research typically applies audio-fingerprints to search most similar samples in an audio database. Our approach, inspired by [15], generates an ambient audio fingerprint in a noisy environment without (!) sharing information among devices. In our research, each audio-fingerprint-bit expresses the energy-difference on frequency bands.

An ambient audio chunk which is considered as an array of signal intensity value is split into non-overlapping equal-length frames. Then we perform a Discrete Fast Fourier Transform (FFT) on these frames. After that, each frame is divided into a set of frequency bands. All of the frequency bands have the same width.

A matrix E is created as follows. Its size is *the number of frames* $\times$ *the number of frequency bands per frame*. Each value of the matrix is the total energy of a frequency band in the corresponding frame.

From the matrix E, we generate the binary fingerprint f whose bits contain the information about the energy change of frequency bands on two successive frames.

$$f(i,j) = \begin{cases} 1 & \text{, if } \begin{array}{l} (E[i,j] - E[i,j+1]) \\ -(E[i-1,j] - E[i-1,j+1]) > 0 \end{array} \\ 0 & \text{, otherwise.} \end{cases}$$

In the above formula, if the energy values of two consecutive frames increase, the corresponding element in the binary sequence is assigned with the value of 1; otherwise, it has the value of 0.

Another possible audio fingerprint extraction scheme which can be considered is Wang's algorithm [16]. We are currently adapting it in our problem. Each spectrogram is divided into 32 frequency bands, and then we select the n amplitude peaks in each band, not only one. Besides, the peaks in the upper half of the spectrogram are removed. The noise cancellation mechanism of some devices (such as Nexus One) makes the peaks in high frequency bands meaningless in the matching process. There are some patterns of the amplitude peaks in the spectrograms, as seen in Figure 1 where $n = 5$. We also choose the amplitude peaks as the local maxima in each time band. The result can be combined with the ones in frequency bands to form the audio fingerprint for an audio sample.

III. Experimental results

We utilise Android-based mobile phones in this research: One Nexus One[1] and three Nexus S[2] devices. The Android OS of the Nexus One device is CyanogenMode-7.1.0-N1 "cooked" version 2.3.7[3]. The Nexus S devices have the official Android OS version 2.3.6. The Nexus One smartphone has a secondary microphone dedicated for dynamic noise suppression, while the Nexus S devices only have "software noise cancellation".

To assure that all mobile devices can record the most similar audio data, their clocks must be synchronized. We use the Navy Clock II application, which is freely available in Android Market[4], for time synchronization. This software provides an Application Programming Interface (API) for acquiring the atomic time from a remote NTP server. Therefore, we integrate it into our application, which is used in the case study.

On each Android device, the ambient audio data are recorded in 6375 milliseconds at the sampling rate of 44100 Hz. The recorded data can be then saved in raw format for later analysis. Each file contains a list of audio signal intensity values which are represented as 16-bit integers.

A. Obstacles and pitfalls

During our experiments we observed serious differences between different device classes. We utilised the Nexus S and Nexus One devices[5]. We observed that the recording start time of the Nexus One was heavily delayed compared to the Nexus S. Furthermore, due to the Nexus One pre-processing the audio signal for noise reduction, the resulting audio sequence differed significantly. Figure 2 depicts these effects. The figure shows the spectrogram of synchronised

[1] Nexus One Technical Specifications: http://www.google.com/phone/detail/nexus-one

[2] Nexus S Technical Specifications: http://www.google.com/nexus/tech-specs.html

[3] CyanogenMode: http:/www.cyanogenmod.com

[4] Navy Clock II application: https://market.android.com/details?id=com.cognition.navyclock

[5] Nexus One with Android version 2.3.7 and Kernel version 2.3.37.6; Nexus S with Android version 2.3.6 and Kernel version 2.6.35.7

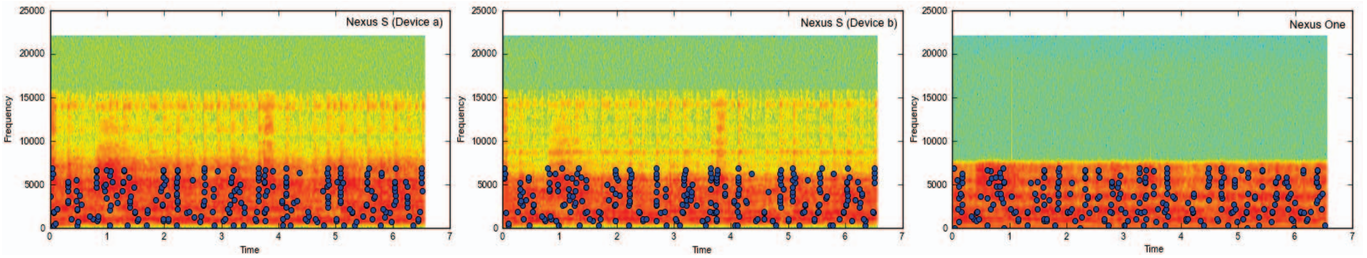

Fig. 1. Amplitude peaks on the spectrograms of synchronised audio recordings from three devices. There are some similar patterns in all three spectrograms.

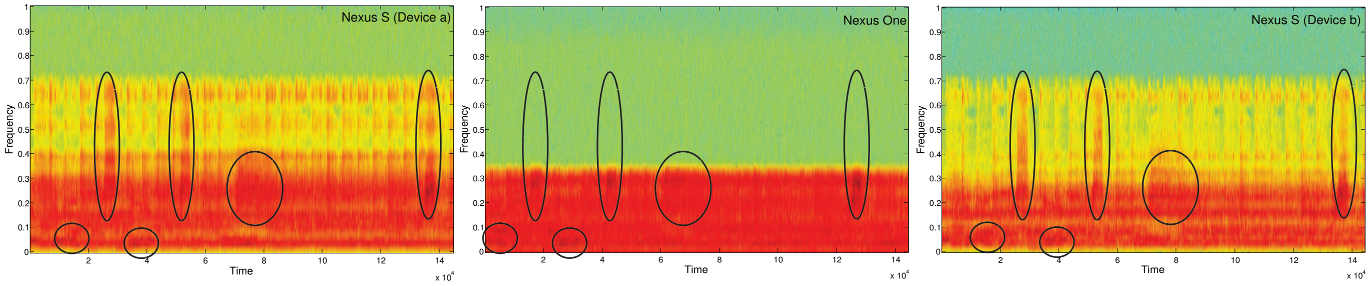

Fig. 2. Spectrogram of synchronised audio recordings from three devices. A similar distribution of energy is found in all three spectrograms. For the Nexus One, the hardware noise cancellation cuts higher parts of the frequency spectrum. The recording of the Nexus One is delayed compared to the two Nexus S devices which are tightly synchronised.

audio recordings from a Nexus One and two Nexus S devices. A pattern of significant energy peaks in the spectrum is highlighted in the figure. A similar pattern re-appears in all figures. However, for the Nexus One device (middle), the pattern occurs earlier in the recorded sequence. Hence, the start of the audio recording of this device was delayed. Additionally, we observe from the figure that the audio spectrum for the Nexus One is cut for higher frequency bands.

Both these effects are unfortunate for the described fingerprinting method. As described in [12], [13], audio-recordings of remote devices should not differ by more than few ten milliseconds in order to achieve sufficiently matching fingerprints. In our case, however, we frequently experienced a delay of about 0.5 seconds.

Also, when parts of the spectrum completely differ due to audio pre-processing, fingerprints are not sufficiently similar to establish a common secure key.

We are currently working on solutions to these problems (see section IV). In this paper, we will describe our results in favourable conditions with Nexus S devices only.

B. *Case study*

We conducted a case study with three Nexus S mobile phones to study the similarity in fingerprints for devices at various distances. In particular, we are interested in the Hamming distance in the generated fingerprints in relation to the distance among mobile devices. We have observed in [12], [13] already that the bit-errors in fingerprints are not related to the distance to an audio source or the loudness of the recorded audio. Here, we consider the deterioration in the similarity of fingerprints while the distance among recording devices is gradually increased.

As detailed in [12], [13] we can use error correcting codes (ECC) to account for the Hamming distance in fingerprints and generate an identical secure key at devices without communication among them.

With this study we aim to estimate the diameter of the security aura a device is surrounded by. Devices farther away can then not join the secure audio-based communication. Furthermore, the study will provide insight whether the approach can be utilised for some sense of indoor localisation.

For the experiment, we implemented the fingerprinting method based on energy differences as described above. Each fingerprint was created from synchronised ambient audio consisting of 18 frames of 6384 samples each and 33 frequency bands of width 497 Hz. We utilised three Nexus S mobile devices placed in an angle of $22.5°$ and $-22.5°$ to an audio source and altered the distance between devices. In several sets of experiments, we increased the distance among devices and to the audio source while keeping the angle to the audio source fixed. Figure 3 depicts our results.

The figure shows the median fraction of identical bits in fingerprints created by the two devices. Each point was created from 10-12 separate experiments with identical environmental conditions. The method achieved an accuracy of about 0.75 to 0.85 in all tests. In particular, the Hamming distance in fingerprints only slightly decreases with increasing distance. With error correcting codes, the remaining hamming distance can be corrected [12], [13]. This demonstrates that a good synchronisation is feasible with our implementation on mobile devices. Also, devices in about 2 m distance can be considered to be in the same security aura for this environment. Devices farther away can be excluded by properly configuring the error correction threshold of the ECC utilised. In future

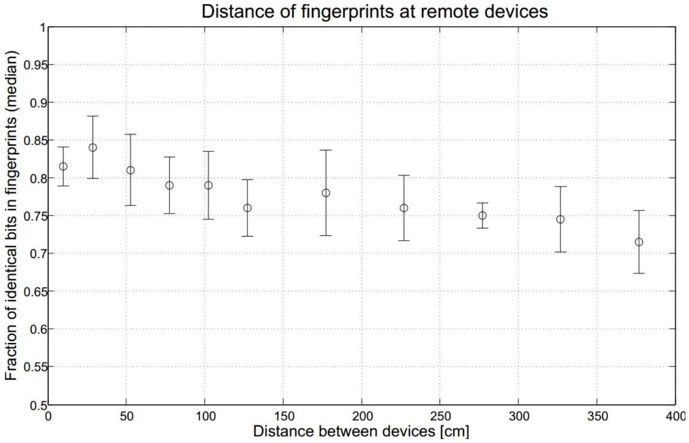

Fig. 3. Median fraction of identical bits in fingerprints created from synchronised audio recordings at two Nexus S devices

work we will consider further environments and algorithmic modifications to reduce the diameter of the security aura.

IV. Next steps

During our studies we encountered several issues which we address in our current and future studies. The most important are a more accurate time synchronisation of devices, differences due to different hardware and software realisations as well as a too low tolerance to different frequency ranges for recordings on the devices.

Regarding time synchronisation, we will consider features from ambient audio which are less time dependent. This will hopefully also positively affect the problem of different recording hardware and software instrumentations.

Furthermore, we are working on more elaborate features from audio to have the distance among fingerprints decrease more significantly with increasing distance among devices.

V. Conclusion

We presented first results from our implementation of an audio-based secure device pairing for mobile devices and discussed practical issues we face. We identified a secure sphere of about 2m outside which devices can be prevented from secure audio-based pairing. Most serious issues are currently the weak time synchronisation of mobile devices which might necessitate a new pairing scheme among devices and differences in audio recording. We discussed possible solutions we are considering in future work.

Acknowledgment

This work was supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD)

References

[1] R. Mayrhofer and H. Gellersen, “Spontaneous mobile device authentication based on sensor data,” *information security technical report*, vol. 13, no. 3, pp. 136–150, 2008.

[2] D. Bichler, G. Stromberg, M. Huemer, and M. Loew, “Key generation based on acceleration data of shaking processes,” in *Proceedings of the 9th International Conference on Ubiquitous Computing*, J. Krumm, Ed., 2007.

[3] L. E. Holmquist, F. Mattern, B. Schiele, P. Schiele, P. Alahuhta, M. Beigl, and H. W. Gellersen, “Smart-its friends: A technique for users to easily establish connections between smart artefacts,” in *Proceedings of the 3rd International Conference on Ubiquitous Computing*, 2001.

[4] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, “Amigo: Proximity-based authentication of mobile devices,” *International Journal of Security and Networks*, 2009.

[5] D. Balfanz, D. Smetters, P. Stewart, and H. C. Wong, “Talking to strangers: Authentication in ad-hoc wireless networks,” in *Proceedings of the Symposium on Network and Distributed System Security*, 2002.

[6] J. M. McCune, A. Perrig, and M. K. Reiter, “Seeing-is-believing: Using camera phones for human-verifiable authentication,” in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, 2005.

[7] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, “Loud and clear: Human-verifiable authentication based on audio,” in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, 2006.

[8] R. Mayrhofer and H. Gellersen, “Shake well before use: Authentication based on accelerometer data,” *Pervasive Computing*, pp. 144–161, 2007.

[9] R. Mayrhofer, “The Candidate Key Protocol for Generating Secret Shared Keys from Similar Sensor Data Streams,” *Security and Privacy in Ad-hoc and Sensor Networks*, pp. 1–15, 2007.

[10] D. Bichler, G. Stromberg, and M. Huemer, “Innovative key generation approach to encrypt wireless communication in personal area networks,” in *Proceedings of the 50th International Global Communications Conference*, 2007.

[11] S. Mathur, R. D. Miller, A. Varshavsky, W. Trappe, and N. B. Mandayam, “Proximate: proximity-based secure pairing using ambient wireless signals.” in *MobiSys*, A. K. Agrawala, M. D. Corner, and D. Wetherall, Eds. ACM, 2011, pp. 211–224. [Online]. Available: http://dblp.uni-trier.de/db/conf/mobisys/mobisys2011.html#MathurMVTM11

[12] S. Sigg, “Context-based security: State of the art, open research topics and a case study,” in *Proceedings of the 5th ACM International Workshop on Context-Awareness for Self-Managing Systems (CASEMANS 2011)*, 2011.

[13] S. Sigg and Y. Ji, “Pintext: A framework for secure communication based on context,” in *Proceedings of the Eighth Annual International ICST Conference on Mobile and Ubiquitous Systems:Computing, Networking and Services (MobiQuitous 2011)*, 2011.

[14] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, “A review of audio fingerprinting,” *Journal of VLSI Signal Processing Systems*, vol. 41 Issue 3, 2005.

[15] J. Haitsma and T. Kalker, “A highly robust audio fingerprinting system,” in *3rd International Conference on Music Information Retrieval (ISMIR 2002)*, Paris, France, 2002, pp. 107–115.

[16] A. Wang, “An industrial-strength audio search algorithm,” in *Proc. 2003 ISMIR International Symposium on Music Information Retrieval*, 2003.

[17] M. Mueller, F. Kurth, and M. Clausen, “Audio matching via chroma-based statistical features,” in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005, pp. 288–295.

[18] S. Baluja and M. Covell, “Content fingerprinting using wavelets,” in *Proceedings of the Conference of Visual Media Production*, London, UK, 2006.

[19] M. Azizyan, I. Constandache, and R. R. Choudhury, “Surroundsense: mobile phone localization via ambience fingerprinting,” in *Proceedings of the 15th annual international conference on Mobile computing and networking (MobiCom ’09)*, New York, NY, USA, 2009.

[20] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, “Indoor localization without infrastructure using the acoustic background spectrum,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services (MobiSys ’11)*, New York, NY, USA, 2011.

[21] V. Chandrasekhar, M. Sharifi, and D. Ross, “Survey and evaluation of audio fingerprinting schemes for mobile audio search,” in *International Symposium on Music and Information Retrieval (ISMIR)*, Miami, Florida, October 2011.

[22] J. P. Ogle and D. P. W. Ellis, “Fingerprinting to identify repeated sound events in long-duration personal audio recordings,” in *IEEE International Conference on Acoustics, Speech and Signal Processing 2007 (ICASSP 2007)*, 2007.