Secure Device Pairing Using Audio

William R. Claycomb and Dongwan Shin
Secure Computing Laboratory
New Mexico Tech
Socorro, NM, USA
Email: {billc,doshin}@nmt.edu

***Abstract*—Secure device pairing between mobile devices is a challenging task. The lack of a trusted authority and low computational power make it difficult for mobile devices to establish secure communication channels in ubiquitous computing environments. Solutions have been proposed using location-limited channels to transmit secure pairing information that can be verified as originating from the intended device, enabling users to establish secure channels over insecure mediums. Of particular interest is using audio as a location-limited channel, due to the widespread deployment of audio capabilities on mobile devices. We describe a solution for secure device pairing using audio, called UbiSound, which only requires a single audio transmission to authenticate both devices. We describe our communication protocol, implementation details and results, and discuss how our solution is resistant to a number of attacks. Additionally, we emphasize how our solution is usable for visually impaired users.**

I. Introduction

The challenge of securely pairing two devices over an unsecured communication channel is a difficult task in computing. The task is made considerably more difficult if neither device has any basis by which to trust the identity of the other, such as a certificate signed by a mutually trusted authority. Several approaches have been suggested to address this problem, many of which employ additional device capabilities to transmit authentic data between devices, enabling the identity of the devices to be verified.

Initial approaches relied on physical touch to transmit secure keying information. More recently, other channels such as visual tags, audio codes, ultrasound, or laser communication have been proposed. Early works required both devices to transmit key establishment information via the secondary channel. More recent works have proposed methods for establishing device identity using only one transmission via the secondary channel.

One particular method of transmitting identification data between devices, which has not been extensively explored, is to use the audio capabilities of both devices. Early works have suggested using audio features on mobile devices to play human-recognizable words or sounds to verify receipt of key establishment information. More recently, the audio channel has been proposed as a method of transmitting key establishment information itself, though with human-verification via audio sound still required to secure the communication channel.

In this paper, we explore a method of key establishment in mobile devices using the audio channel to transmit key establishment information, which we call *UbiSound*. By encoding and transmitting key verification information from one device to another, we show how to establish device authentication, enabling the creation of a secure channel over an unsecured communication medium. We further demonstrate how both devices can be authenticated using a single audio message, without requiring a human-assisted second audio transmission or string comparison for key verification. We show how our solution dramatically improves the time required for device pairing, and show how our approach is secure against complicated attacks such as man-in-the-middle. Furthermore, by using audio communication, rather than visual channels, we create a solution which is usable by visually impaired operators.

The remainder of this paper is organized as follows. Section II provides background material on secure device pairing, followed by Section III, which details our approach, including implementation details and performance analysis. Section IV provides a discussion of potential attacks, resistance, and how the solution is usable for the visually impaired. Section V concludes the paper with suggestions for future work.

II. Background and Related Works

Secure pairing between two devices, without any previous, or *a priori*, knowledge by which to verify identity, such as public keys generated by a trusted authority, was first generalized in [1]. This problem was further specialized to the field of ubiquitous computing in [2], and since then, many approaches have been proposed to address this problem. The main challenge in each approach is to authenticate the other device, that is, to ensure that secure communication is established with the intended device only, and does not include an attacker, or "man-in-the-middle."

The first proposed solution was to use physical touch between the two devices to securely transmit key establishment information [2]. However, this is impractical and cumbersome for mobile computing, so various other solutions emerged. These approaches use the various additional capabilities of mobile devices to establish a location-limited channel (LLC) between them [3]. Such a channel may not be secret, that is, it may be observable to an adversary, but it is authentic, that is, the data received by one device can be assured not to have been altered. This also ensures that the *origin* of the data is authentic; the receiving user can guarantee the data originated from the sending device. Such a transmission is sometimes referred to as *demonstrative identification* [3]. With

978-1-4244-4170-9/09/$25.00 ©2009 IEEE

the ability to transmit authentic key establishment information, two devices can use common key establishment techniques, such as Diffie-Hellman [4], to establish a shared secret key.

Various location-limited channels have been proposed for use in such a scenario. Among the first solutions was to use device displays and digital cameras [5]. This approach was also explored by [6]–[8], who proposed using two-dimensional barcodes to transmit key establishment information between two devices. [9] and [10] expanded the capabilities and performance of this approach by introducing the use of color in the barcodes. Other LLCs have been suggested, including infra-red transmission [3], ultra-sound [11], lasers [12], [13], and biometric patterns such as facial recognition [14] or grip patterns [15].

A. Using Audio as an LLC

Audio transmission has also been proposed as an LLC [16]–[19], in several distinct applications. The first application, proposed in [16], requires both devices to generate and play a multimedia stream based on key establishment information. The streams are generated in such a way as to audibly harmonize with each other. Users of the devices listen for this harmony to ensure key establishment information has been transmitted.

The second approach involves transmitting key establishment information via an unsecured channel, then verifying its authenticity using audio playback [17]. The received key establishment information is transformed into human-understandable words and phrases, which are simultaneously played over the receiving device's speakers and displayed on the sending device's screen. The user confirms that the two sequences match, thus verifying the key establishment information.

A third approach combines the second approach with the audio transmission of key establishment information [19]. Using this approach, key establishment information is transformed into a high-speed audio transmission which is played by the sending device and recorded by the receiving device. The information is then transformed into human-understandable words for verification. Of particular interest in this approach is that the LLC is the only channel needed to establish secure communication, whereas the first approach requires both an LLC and the unsecured wireless channel to complete key establishment.

More recent solutions employ the use of *Short Authenticated Strings* [20]–[22] to secure device pairing [18]. This approach involves mapping a short string into an English sentence that is syntactically correct and understandable to humans. Users compare strings generated by both devices, either audibly or visually, to authenticate key establishment information between devices.

B. Key Establishment Protocols

First proposed in [3], the basic key establishment protocol when using LLCs is shown in Table I, using notation established in [23]. The goal is to transmit and verify a public key of each device, thus enabling a key establishment protocol such as Diffie-Hellman [4]. Using this protocol, each device first sends key verification information via the LLC. Then, each device transmits public key information over the insecure communication channel. Upon receipt of this information, each device can independently calculate the verification data of the public key information received, and compare it to the verification information received using the LLC.

TABLE I
BASIC PROTOCOL FOR EXCHANGING KEYING INFORMATION VIA A LOCATION LIMITED CHANNEL

#	Ch	Alice	Bob
1	LL	$-Addr_{Alice}, HK_{Alice} \rightarrow$	
2	LL	$\leftarrow Addr_{Bob}, HK_{Bob}-$	
3	RF	← Key exchange protocol →	

Unfortunately, the approach described above requires similar LLC capabilities on both devices. That is, if using digital cameras and color displays to transmit LLC data, each device must have these capabilities. A refinement to this protocol is proposed in [24]. Using this approach, it is only necessary for one device to transmit LLC data to verify key establishment information. This is possible because public key information is transmitted first to the sending device via the unsecured channel. The sending device combines verification data for this public key information with verification data for its own public key information, into a single LLC message. This enables independent verification of both keys by the receiving device.

A similar method of exchanging key establishment information using only one-way transmission over an LLC is proposed in [23]. This approach generates key establishment information that includes a randomly generated value. This random value is actually the only data transmitted using the LLC. However, additional user involvement is required, during which users must acknowledge receipt of keying information prior to LLC transmission, and must verify the information was received correctly after LLC transmission. This protocol, described in [23], is shown in Table II.

III. Approach

We propose using audio transmission as an LLC to securely transmit verification of key establishment information between two mobile devices, enabling secure device pairing. Furthermore, we propose a solution which eliminates the audio or string-comparison based human-verification components of previous approaches. Therefore, we do not employ the use of harmonizing sounds, human-understandable melodies, or short authenticated strings. We also improve upon previous approaches by introducing a simplified version of the device paring protocol proposed by [23]. Our entire solution, named UbiSound, is generalized in Figure 1

A. Key Establishment Protocol

We simplify the key establishment protocol described in [23] (shown in Table II), with a new protocol, shown in Table III. The simplified protocol begins with both Alice and

TABLE II
ASYMMETRIC PAIRING USING A LOCATION LIMITED CHANNEL [23] (RF: RADIO FREQUENCY CHANEL, LL: LOCATION LIMITED CHANNEL, PB: MANUAL USER INTERACTION (PUSH-BUTTON))

#	Ch	Alice		Bob
1				Directed to begin key establishment
2		Chooses random a		Chooses random b
3	RF		$-g^a \rightarrow$	
4	RF		$\leftarrow g^b-$	
5				Chooses random R_b
6				Chooses random K_b
7				$H_2 = H(I_{Bob}\|g^b\|g^a\|R_b\|K_b)$
8	RF		$\leftarrow H_2-$	
9	PB		$-ack \rightarrow$	
10	LL		$\leftarrow R_b-$	
11	RF		$\leftarrow K_b-$	
12		$H_2' = H(I_{Bob}\|g^b\|g^a\|R_b\|K_b)$		
13		Verifies $H_2 = H_2'$		
14	PB		$-outcome \rightarrow$	

TABLE III
UBISOUND KEY ESTABLISHMENT PROTOCOL (RF: RADIO FREQUENCY CHANEL, LL: LOCATION LIMITED CHANNEL, PB: MANUAL USER INTERACTION (PUSH-BUTTON))

#	Ch	Alice		Bob
1		Chooses g^a		
2	RF		$-g^a \rightarrow$	
3				Chooses g^b
4				Chooses random R_b
5				$H_b = H(g^a\|g^b\|R_b)$
6	RF		$\leftarrow H_b-$	
7	LL		$\leftarrow R_b-$	
8	RF		$\leftarrow g^b-$	
9		$H_b' = H(g^a\|g^b\|R_b)$		
10		Verifies $H_b = H_b'$		
11	PB		$-verify \rightarrow$	

Bob selecting new public keys, g^a and g^b. Alice sends her public key, g^a, to Bob using the unsecured wireless channel. Bob then chooses a random number, R_b, of sufficient size to prevent guessing by Marvin, the adversary. Next, Bob calculates a hash value, H_b, as the concatenation of g^a, g^b, and R_b, and sends H_b to Alice using the unsecured channel.

The next step involves the LLC, which is an audio channel in our case. R_b is encoded and transmitted over this channel, followed by Bob's public key, g^b, which is sent over the unsecured wireless channel[1]. After receiving g^b, Alice has all the information she needs to calculate $H_b' = H(g^a|g^b|R_b)$, and verify that $H_b = H_b'$. Because Alice can verify that R_b came from Bob, using the demonstrative identification [3] of the LLC, she can verify that H_b was also generated by Bob. Assuming that an adversary has not compromised Bob's device, this confirms to Alice that the information she received from Bob is authentic, verifying his device, his public key g^b, and allowing key establishment to commence.

How does Bob establish that he is communicating with Alice, though? This question is addressed in [23], and the answer is reasoned as follows. Bob does not receive any communication from Alice via an LLC, which may lead to the conclusion that Bob cannot demonstratively identify Alice's device. While this would be true for completely automated devices, we have the advantage of user interaction to complete the protocol. Once Alice verifies she is communicating with Bob, she has verified that H_b is correct. Because H_b contains g^a, Alice's verification to Bob also confirms to Bob that he has used the correct values in calculating H_b, and those values can be trusted to establish a secure channel. Even if Bob were a kiosk device, he could receive confirmation from Alice via a push-button device, which only Alice would be able to press. [23] points out that it would take an extremely sophisticated attacker to develop a button-pushing device that could not be detected by the kiosk user. Of course, this also limits Bob to executing the key establishment protocol with only one device at a time.

B. Device Pairing Details

The first step towards device pairing is generating new public/private key pairs. This is to prevent replay attacks, uniquely identify each instance of secure communication (in case multiple secure channels are being established, as in a group scenario), and because no trusted authority is used to verify public key information, which would imply a static public/private key pair for each device. If too computationally expensive at run-time, several potential key pairs could be pre-generated and stored on each device. We found that using even older mobile devices[2], new RSA key pairs (1024 bit keys) can be generated in less than 1.5 seconds on average (see Table IV).

Following the protocol described above, Alice and Bob exchange several pieces of information, including the LLC transmission of R_b. Using audio as an LLC presents the challenge of encoding a random number as a series of audio tones, which can be recorded and decoded by the listening device. Standard codecs exist for this purpose; however, since our application is fairly simple, we determined that performance

[1] We discuss why this delayed disclosure of g^b is important when discussing the man-in-the-middle attack in § IV-A.

[2] HP iPAQ rx3715 (introduced in 2004, 400 MHz Samsung S3C2440 processor and 64 MB of RAM) [25] and a Dell Axim X50v (introduced in 2004, 624 MHz Intel PXA270 processor and 64 MB of RAM) [26]

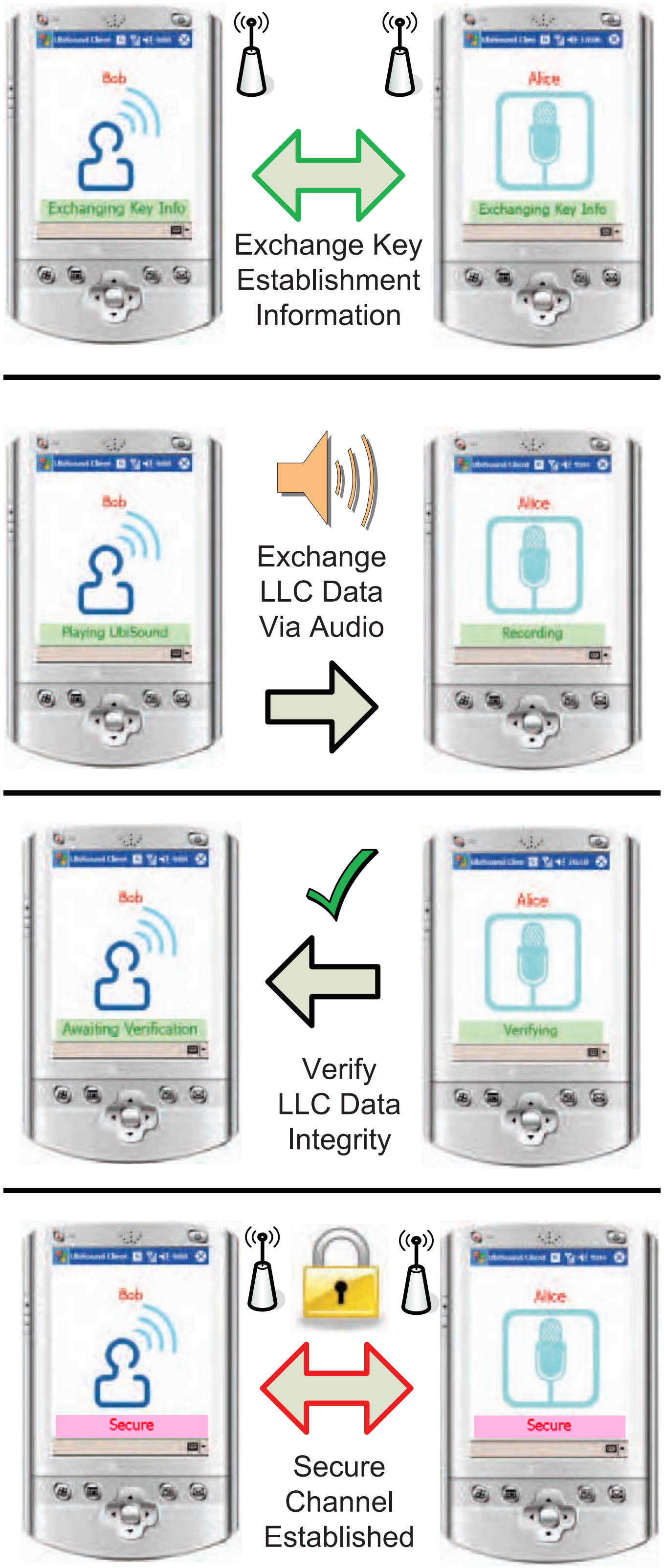

Fig. 1. UbiSound Secure Device Pairing

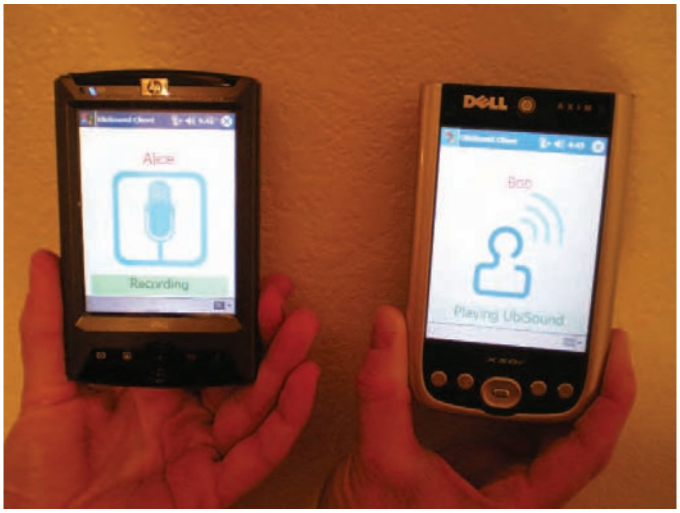

Fig. 2. UbiSound Implementation on Mobile Devices

could be fine-tuned and individualized for this specific purpose by creating our own codec.

Once encoded, the sending device plays the audio sequence, which is recorded by the listening device's microphone. The listening device decodes the message using the codec, to obtain R_b. Using this information, the receiving device completes the remaining steps in the device pairing protocol described in Table III.

C. Implementation Performance Analysis

Our solution was implemented and tested using common mobile devices, each capable of playing and recording audio. Two such devices, displaying the UbiSound implementation, are shown in Figure 2. Performance was measured in the following areas:

- Time to generate public key
- Time to generate and encode random value R_b
- Time to play audio message
- Time to decode message and verify contents
- Overall time to verify key establishment information

TABLE IV
TIME (SEC) TO GENERATE PUBLIC/PRIVATE KEY PAIRS

Device	Axim	iPAQ
Average (sec)	1.12	1.35

1) Generating Cryptographic Elements: As mentioned previously, generating new public/private key pairs is accomplished in less than 1.5 seconds, on average. Table IV details these results.

TABLE V
TIME (SEC) TO GENERATE AND ENCODE A RANDOM VALUE

Device	Axim	iPAQ
Average (sec)	< 1	< 1

2) Generating and Encoding a Random Value: A cryptographically strong random number generator was used to generate a 24-bit random value, R_b. Using our own codec, this was encoded into a sequence of audio tones and coupled with additional baseline tones to form an entire message. The additional baseline tones are necessary to provide decoding information on the receiving device, and for error correction. Table V shows the average time for each device to generate and encode a random value.

TABLE VI
TIME (SEC) TO PLAY AN ENCODED VALUE

Device	Axim	iPAQ
Average (sec)	2.5	2.5

3) Playing an Audio Message: The tones were generated as MIDI [27] sequences and played at normal volume. The receiving device was placed within 30 cm of the sending device for best quality and demonstrative identification. Table VI shows the average time to play encoded sounds.

TABLE VII
TIME (SEC) TO DECODE THE MESSAGE AND VERIFY THE CONTENTS

Device	Axim	iPAQ
Average (sec)	< 1	< 1

4) Decoding the Message and Verifying Contents: Once received, the message must be decoded and the contents verified by the receiving device. Note that a cryptographic function (hashing) is required in this step as well. Table VII shows the results of this test for each device.

TABLE VIII
OVERALL TIME (SEC) TO VERIFY KEY ESTABLISHMENT INFORMATION

Device	Axim	iPAQ
Average (sec)	6-7	6-7

5) Overall Time to Verify Key Establishment Information: The overall time, from initiation of the process (marked in our case by A generating a public key and sending it to B, triggering B to begin the process of key generation, random number generation, and audio message encoding), to the verification of key establishment information by A, is shown in Figure VIII. This figure includes the time necessary to establish both the insecure connection initially used to transmit key establishment information, as well as the secure channel used for subsequent communication.

TABLE IX
AVERAGE TIME TO ESTABLISHMENT SECURE COMMUNICATION

Method	Time
HAPADEP [19]	62-80 sec
Loud and Clear [17]	22-32 sec
Seeing-Is-Believing [24]	5-7 sec
UbiColor [9]	6-8 sec
UbiSound	6-7 sec

6) Comparison: A comparison of average time to establish secure communication for various methods of device pairing using LLCs is shown in Table IX.

IV. DISCUSSION

The solution proposed in this paper makes various improvements over existing schemes. Most importantly, our solution does not require a second audio transmission or string matching to verify key establishment information. Rather, the devices establishing communication calculate verification information and prompt the users to either accept or reject the communication request. This improvement results in much faster establishment of secure communication, because so little human involvement is required. Additionally, we describe a robust solution, which is capable of authenticating two devices when only one of them possesses LLC transmission capabilities (assuming the other possesses LLC receiving capabilities). This is particularly useful when interfacing with kiosk or other unattended and automated devices - only one human operator is needed to complete the communication protocol.

Another improvement our solution offers is with respect to transmission size. Previous approaches suggested transmitting entire public keys, or hashes of public keys, sometimes in addition to networking information. Depending on the protocol, networking information may be necessary (particularly when using the LLC as the sole method of transmitting key establishment information), but recent approaches do not assume this to be the case. While initial solutions required 80-240 bits of information to be transmitted [9], [19], a recent proposal notes that by using short authenticated strings, the amount of data necessary for LLC transmission can be reduced to 16-20 bits [18]. Similarly, our solution requires significantly less data (32 bits total) be transmitted using the LLC, and this could be reduced further if a smaller random number were used as R_b.

A. Attacks

a) Man-In-The-Middle: The man-in-the-middle attack, shown in Table X against the Diffie-Hellman Key Exchange, is the driving motivation behind using LLCs to secure device pairing in the first place. The attack is prevented by exchanging authenticated data between devices, that is, data which can be independently verified as belonging to the device in question. Normally this is verified using trusted authorities, but a trusted authority cannot be assumed when authenticating mobile devices. Therefore, another method of authenticating key establishment must be used. The LLC provides such a method. Even though the data transmitted using the LLC is not secret (an attacker can easily observe and capture the data), it is authentic - it is guaranteed to originate from the sending device.

In a simple example, using an LLC either prevents Marvin from delivering his own key establishment information (p, g, M), to Bob, or prevents Marvin from delivering the malicious response, M, to Alice. Additionally, due to the LLC, Bob knows he has received the correct information from Alice, (p, g, A), or Alice knows she has received the correct information back from Bob, B. This prevents Marvin from establishing a man-in-the-middle attack.

TABLE X
MAN-IN-THE-MIDDLE ATTACK ON THE DIFFIE-HELLMAN KEY EXCHANGE

Alice	Marvin	**Bob**
Chooses p, g, and a. $A = g^a \bmod p$		
	$\xrightarrow{p,g,A}$	
	Chooses m. $M = g^m \bmod p$	
	$\xrightarrow{p,g,M}$	
		Chooses b. $B = g^b \bmod p$
	$\xleftarrow{B}$	
	Sends M to Alice	
	$\xleftarrow{M}$	
$K = M^a \bmod p$	$K_{Alice} = A^m \bmod p$, $K_{Bob} = B^m \bmod p$	$K = M^b \bmod p$

Our solution operates in a very similar way. In order to launch a man-in-the-middle attack, Marvin would first to provide Bob with a malicious key, g^{m_a}, not g^a. Marvin would then alter the information returned to Alice via the unsecured channel, to be $H_m = H(g^a|g^{m_b}|R_m)$. Next, Bob sends the LLC message (R_b), which Marvin is not able to modify, due to the properties of LLC transmission. Finally, Bob sends g^b to Marvin, who would then send g^{m_b} to Alice, as he needs Alice to use g^{m_b} for the attack to succeed. However, since Marvin cannot alter the information transmitted between Bob and Alice via the LLC, the only possibility of the attack succeeding is if $R_m = R_b$. In that case, $H_m = H'_b$, where H'_b is the value Alice would calculate based on the information she received from both Bob and Marvin. However, unless Marvin correctly guesses R_b, these values will not be equal, so Alice would not prompt Bob to establish a secure channel. As long as R_b is large enough to prevent guessing by Marvin, the attack is unsuccessful. We chose the size of R_b to be 24-bits, giving Marvin a $1/(1.67 \times 10^7)$ chance of guessing correctly.

The placement of the transmission of g^b after the transmission of the LLC data (R_b) is important to preventing this attack. If g^b were transmitted prior to R_b, then for a brief moment, Marvin would know g_a, g_b, and H_b. Recall that $H_b = H(g^a|g^b|R_b)$. If R_b is only 24-bits long, then Marvin could conceivably do a brute-force attack on H_b, trying all 1.67×10^7 possibilities of R_b to obtain the correct value. Then, Marvin would know R_b prior to LLC communication, and could create $H_m = H(g^a|g^{m_b}|R_b)$ to satisfy $H_m = H'_b$.

b) Denial of Service Attack: Launching a denial of service (DoS) attack against the audio LLC would be somewhat difficult to conceal, in contrast to DoS attacks against wireless or other network traffic. Because of the nature of the LLC, any DoS attack would require the use of audible sound, and could be easily traced back to the source. However, an unintentional DoS attack could be carried against this solution if the background noise around the devices was too loud to allow audio information to be successfully transmitted.

c) Modifying LLC data: It is conceivable that an attacker could try to somehow modify the audio transmission between Bob and Alice to modify the message received (i.e. change R_b into R_m). If successful, this would enable a man-in-the-middle attack. However, to do so would require foreknowledge of R_b, and would require very fast processing power to determine how to modify the audio data in real-time. Because our codec randomizes the tones used to represent bit sequences during each transmission, and only plays baseline tones after the data payload has been transmitted, Marvin would need to entirely cancel out and replace Bob's transmission with his own. We feel this is extremely unlikely. Additionally, this attack would require a hidden speaker capable of playing sounds that could not be distinguished from Bob's sounds.

d) Disruption attack: If Marvin is not attempting to launch a man-in-the-middle attack, but rather simply disrupt communication, he could potentially succeed by carrying out the man-in-the-middle steps described previously. This would require him to be able to receive and modify all communication between Alice and Bob over the insecure channel (which is highly unlikely considering the distance between Alice and Bob is very small). If he could somehow accomplish this, he could successfully prevent secure communication from being established using our protocol. This is shown in Table XI.

B. Visually Impaired Users

Our solution provides an alternative to visually-based secure device pairing techniques. This is particularly useful to visually-impaired users. Because devices do not have to be precisely aligned, but merely placed within proximity of each other, this also makes it easier for users without fine muscle control. It could be argued that demonstrative identification requires a user to visually verify the device sending the LLC data. However, we would argue that visually identifying a device playing sound is rather difficult, even without a visual impairment. Rather, we suggest that users could *feel* the sound being played by the other device. Human touch is sensitive enough to be able to both detect that a device is playing sound, and verify that the sounds being felt correspond to the sounds being played, simply by placing a hand on the device playing the sound. This would be especially true for visually impaired users, who often have heightened senses of touch and hearing.

C. Usability Concerns

Usability remains a serious concern for any audio-based device pairing solution. As background noise increases, the reliability of data transmitted via the LLC decreases. This can be compensated for by using error correcting codecs, but at

TABLE XI
DISRUPTION ATTACK (RF: RADIO FREQUENCY CHANEL, LL: LOCATION LIMITED CHANNEL)

#	Ch	Alice		Marvin		Bob
1		Chooses g^a				
2	RF		$-g^a \rightarrow$			
3				Chooses g^{m_a}		
4	RF				$-g^{m_a} \rightarrow$	
5						Chooses random R_b
6						$H_b = H(g^a \| g^b \| R_b)$
7	RF				$\leftarrow H_b-$	
8				Chooses random R_m		
9				$H_m = H(g^a \| g^{b_m} \| R_m)$		
10	RF		$\leftarrow H_m-$			
11	LL		$\leftarrow R_b$ ————	————————	————	
12	RF				$\leftarrow g^b-$	
13	RF		$\leftarrow g^{b_m}-$			
14		$H_b' = H(g^a \| g^{b_m} \| R_b)$				
15		$H_b \neq H_b'$				
16		Does not verify key agreement with Bob				

some point, it becomes impossible to transmit successfully. Additionally, the distance between the devices plays a major role in determining the reliability of the data transfer. In implementation testing, we found that data reliability dropped dramatically as the distance between the devices increased beyond about 40 cm. Again, this can be compensated for by error correcting codecs, slowing down the playing speed, or increasing the volume of the transmission, all of which could in turn affect the usability by increasing the overall time required to establish secure communication.

V. CONCLUSION

We have demonstrated UbiSound, a protocol and process for establishing secure device pairing using audio as an LLC. Furthermore, we have demonstrated how our solution does not require additional audio messages or string comparisons of the key establishment data. We only require an acknowledgement to the other user that data was successfully and correctly received. We have demonstrated an implementation that dramatically improves on previous audio-based approaches with regard to the time necessary to establish secure communication, while maintaining the same level of security. Finally, we described how our solution is resistant to various attacks, and how it can be of use to visually impaired users.

Future work will include usability studies with actual users. Refinements in processing and codecs could also result in lower overall time to establish secure communication. We believe that audio capabilities are fairly standard on most varieties of mobile computing devices, making audio LLCs a good solution for secure device pairing. Given the increasing popularity of mobile computing and mobile computing devices, we can definitely see the potential for advanced audio-based device pairing techniques in the future.

REFERENCES

[1] R. L. Rivest and A. Shamir, "How to expose an eavesdropper," *Commun. ACM*, vol. 27, no. 4, pp. 393–394, 1984.

[2] F. Stajano and R. J. Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks," in *Proceedings of the 7th International Workshop on Security Protocols*. London, UK: Springer-Verlag, 2000, pp. 172–194.

[3] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," in *In Symposium on Network and Distributed Systems Security (NDSS 02*, 2002.

[4] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, November 1976.

[5] J. McCune, A. Perrig, and M. Reiter, "Seeing-is-believing: using camera phones for human-verifiable authentication," in *The 2005 IEEE Symposium on Security and Privacy*, May 2005.

[6] D. Shin and S. Im, "Visual device identification for security services in ad-hoc wireless networks," in *Proceedings of 20th International Symposium on Computer and Information Sciences (ISCIS'05)*, Istanbul, Turkey, October 2005.

[7] D. Shin, "Securing spontaneous communications inwireless pervasive computing environments," in *ISM '05: Proceedings of the Seventh IEEE International Symposium on Multimedia*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 662–667.

[8] S. Im, "Validating secure connections between wireless devices in pervasive computing using data matrix," in *Multimedia and Ubiquitous Engineering, 2008. MUE 2008. International Conference on*, 2008, pp. 186–190.

[9] W. R. Claycomb and D. Shin, "Secure real world interaction using mobile devices," in *Proceedings of the Pervasive Mobile Interaction Devices (PERMID) 2006 Workshop*, Dublin, Ireland, May 2006.

[10] ——, "Using a two dimensional colorized barcode solution for authentication in pervasive computing," in *Proceedings of the IEEE International Conference on Pervasive Services 2006*, Lyon, France, June 2006.

[11] T. Kindberg and K. Zhang, "Validating and securing spontaneous associations between wireless devices," in *Proceedings of the 6th Information Security Conference (ISC03)*, 2003.

[12] ——, "Secure spontaneous device association," in *UbiComp 2003: Ubiquitous Computing*, 2003, pp. 124–131. [Online]. Available: http://www.springerlink.com/content/m5ynud3qn3u49jyq

[13] R. Mayrhofer and M. Welch, "A human-verifiable authentication protocol using visible laser light," in *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 1143–1148.

[14] I. R. Buhan, J. M. Doumen, P. H. Hartel, and R. N. J. Veldhuis, "Secure ad-hoc pairing with biometrics: Safe," in *First International Workshop on Security for Spontaneous Interaction, Innsbruck, Austria*, Innsbruck, Austria, September 2007, pp. 450–456.

[15] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis, "Feeling is believing: A secure template exchange protocol," in *Advances in Biometrics International Conference, ICB 2007*, 2007, pp. 897–906.

[16] T. Kindberg and K. Zhang, "Securing spontaneous interactions," in

Proceedings of the 2nd UK-UbiNet Workshop, Cambridge, UK, May 2004.

[17] M. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and clear: Human-verifiable authentication based on audio," *Cryptology ePrint Archive, Report 2005/428*, 2005.

[18] M. T. Goodrich, M. Sirivianos, J. Solis, C. Soriente, G. Tsudik, and E. Uzun, "Using audio in secure device pairing," *Int. J. Secur. Netw.*, vol. 4, no. 1/2, pp. 57–68, 2009.

[19] C. Soriente, G. Tsudik, and E. Uzun, "Hapadep: Human asisted pure audio device pairing," Cryptology ePrint Archive, Report 2007/093, 2007, http://eprint.iacr.org/.

[20] S. Vaudenay, "Secure communications over insecure channels based on short authenticated strings," in *Advances in Cryptology CRYPTO 2005*, 2005, pp. 309–326. [Online]. Available: http://dx.doi.org/10.1007/11535218_19

[21] M. Cagalj, S. Capkun, and J. Hubaux, "Key agreement in Peer-to-Peer wireless networks," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 467–478, 2006.

[22] S. Laur and K. Nyberg, "Efficient mutual data authentication using manually authenticated strings," in *The 5th International Conference on Cryptology and Network Security, CANS 2006*, ser. Lecture Notes in Computer Science, D. Pointcheval, Ed., vol. 4301. Suzhou, China: Springer, December 2006, pp. 90–107, a shortened version of ePrint Report, http://eprint.iacr.org/2005/424.

[23] F.-L. Wong and F. Stajano, "Multi-channel protocols," in *13th International Workshop in Security Protocols*, April 2005.

[24] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan, "Secure device pairing based on a visual channel (short paper)," in *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 306–313.

[25] L. Hewlett-Packard Development Company, "Hp ipaq rx3715 mobile media companion," 2005. [Online]. Available: http://h18000.www1.hp.com/products/quickspecs/11960_na/11960_na.HTML

[26] I. Dell, "Dell axim x50v and x50 handhelds," 2009. [Online]. Available: http://www.dell.com/content/topics/segtopic.aspx/brand/axim_x50?c=us\&l=en

[27] J. Lazzaro and J. Wawrzynek, "RFC 4695: RTP payload format for MIDI," IETF RFC Publication, 2006.