Demo: A Practical Application of Visible Light Communication: Opportunistic Sharing of Encryption Keys

Jayanth Shenoy, Aditya Tyagi, Meha Halabe, Christine Julien
{jayanth.shenoy,adityatyagi6498,meha.halabe,c.julien}@utexas.edu
University of Texas at Austin
Department of Electrical and Computer Engineering

ABSTRACT

We present a demonstration of JIVE (Joint Integration of VLC and Encryption), a novel encryption key sharing framework utilizing the emerging wireless technology Visible Light Communication (VLC). Based on the idea of transmitting data by modulating light, we are able to (1) share a secret key within a constrained physical space and (2) leverage this shared key to communicate encrypted information among co-located mobile devices. In this demonstration, we showcase our complete implementation of JIVE: a VLC transmitter and a VLC receiver. Both endpoints are built using off-the-shelf components. The VLC link is used to distribute a randomly generated secret key that can only be "observed" by VLC receivers that are physically in the same space as the transmitter. Each receiver is connected over a serial connection to an Android device; we developed applications for Android that take the key from the VLC receiver and subsequently use the key to encrypt or decrypt application data. Our demo invites participants to create their own encrypted messages in the Android application and interact with the VLC prototype as it transmits encryption keys, thus illustrating our system's ability to bootstrap security among physically co-located devices.

ACM Reference Format:
Jayanth Shenoy, Aditya Tyagi, Meha Halabe, Christine Julien. 2019. Demo: A Practical Application of Visible Light Communication: Opportunistic Sharing of Encryption Keys. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19), October 21–25, 2019, Los Cabos, Mexico.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3300061.3343377

1 INTRODUCTION

Many mobile application scenarios demand the ability for applications to exchange information privately, relying on the use of encryption and decryption algorithms to secure the data. These processes require the parties to exchange keys *a priori*. In the current state of practice, many individuals establish these secret keys through "low tech" approaches like writing a key on a shared whiteboard. In this work, we demonstrate the use of JIVE, which uses the visible light in a space to establish a shared secret key that is confined to that space. In [3], we described the details of the JIVE framework, which supports the sharing of encryption keys via visible light communication. By transmitting data through visible light pulses, we developed a secure system capable of constraining encrypted information encoded signals (i.e., light) using the physical space (i.e., walls).

JIVE is particularly appropriate in situations where small groups of devices need to securely share data and are co-located in a space from which it is difficult for light to "leak", but other forms of communication (e.g., WiFi) are susceptible to eavesdropping. For example, consider an enclosed room in a doctor's office or hospital. This room may consist of a variety of health IoT devices and devices carried and used by nurses and doctors. As the IoT devices collect information about the patient, this data could be securely distributed to all of the mobile devices belonging to the patient and healthcare professionals over a WiFi or Bluetooth connection. However, to protect the patient's privacy, the communicated data should be encrypted while in transit. Because of the frequency and short durations of these interactions, establishing a shared secret should be efficient and transparent to the user. With JIVE, the lights in the room could be programmed so that, when the lights are power cycled and the door is closed, JIVE generates a novel secret key and begins sharing it by invisibly modulating the light. Any devices within the room can "see" the key, and devices outside of the room are unlikely to be able to see the key. When a visit with a patient completes, the lights can be power cycled

MobiCom '19, October 21–25, 2019, Los Cabos, Mexico
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6169-9/19/10.
https://doi.org/10.1145/3300061.3343377

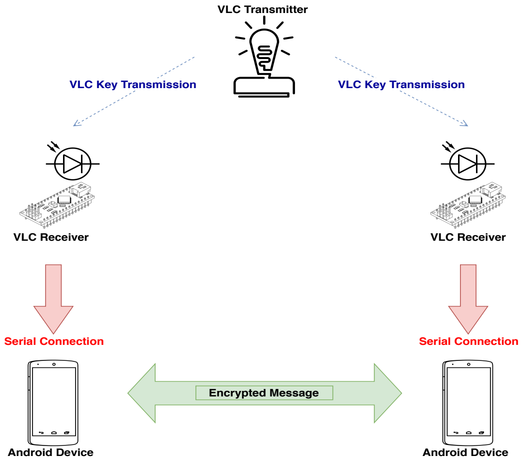


Figure 1: High-level Jive System Design. The VLC transmitter (at the top) generates and transmits a random sequence of bits that define the secret session key. VLC receivers that capture the transmission share it with their connected Android device, which can subsequently use it for encrypted communication.

and a new key generated. Jive is also applicable to the onerous multi-step setup process required to add new IoT devices to a smart home. Rather than having to "prove" co-location by carefully timing the sequence of button presses on an IoT device and a user's mobile device, co-location could be authenticated via the key shared by Jive.

This demonstration aims to exhibit how Jive can be applied to the exchange of private information in the context of a real application. More specifically, we will use a single VLC transmitter that constantly "broadcasts" an encryption key within an enclosed environment we designed. We will have multiple VLC receivers connected via USB to Android devices. These receivers will perform one of three roles:

(1) **Sender:** The sender will obtain the shared secret key via VLC, encrypt a message with the received key, and broadcast the encrypted message to any other nearby Android devices via WiFi.
(2) **(Data) Receiver:** The data receiver will also obtain the key via VLC, listen for encrypted broadcast messages from other Android devices, and use the received key to decrypt the message.
(3) **Eavesdropper:** The eavesdropper will also receive the encrypted data via WiFi but will be unable to decrypt it because the eavesdropper will be outside of the space in which the VLC link is visible.

As part of the demonstration, we have built a miniature "room" to emulate the walls surrounding the VLC link.

2 THE DESIGN OF JIVE

Fig. 1 shows the overall setup of a Jive system. This section will provide a brief overview of the entire system. All of the components of the Jive implementation are publicly available[1].

Jive is designed to generate a new, random key every time it is powered on. This key is then continuously transmitted until the transmitter is powered off. Each VLC receiver communicates the data received on the VLC link to an Android application using a direct serial connection[1].

Hardware. We designed both the VLC transmitter and the VLC receiver based on the widely available Teensy 3.5 microcontroller. For the VLC transmitter, Jive uses a simple VLC circuit, where the microcontroller is connected to a DC LED light at a digital output. The LED is powered by 12 volt battery . The VLC receiver circuit consists of a photodiode receiving the light; the microcontroller gathers data from it through its analog input pin.

VLC Software. The software for the system relies on carefully timed periodic interrupts. The VLC link uses Manchester encoding to embed data in the light pulses, mitigating transmission flicker effects by ensuring that the light switches between high and low output states at every bit in the message. The receiver runs in low power mode until its periodic interrupt is triggered at every half symbol period. During each interrupt, the receiver's photodiode samples light data; each sample is processed by the Teensy's 10-bit ADC. The receiver determines whether the received ADC value should be high or low based on a pre-specified analog threshold. Once a sufficient number of bits are successfully received, the interrupt decodes and rebuilds each frame, assembling the received frames into the larger message. Jive segments the 128-bit generated key into 8 bit frames for transmission. Before sending a frame of data, the transmitter sends a sequence of six high "half-bits." In Manchester encoding, this sequence of six high half-bits will never be construed to be any part of a message since bits are encoded through rising and falling edges. Each message is also prefaced with a one (i.e., a rising bit) at the beginning and a zero (i.e., a falling bit) at the end to ensure that the message begins on low and ends on low for the preamble algorithm to function properly. For error correction, Jive requires the same key to be received three consecutive times before sharing it with the Android application.

Android software. We created Android applications [2] to provide support for symmetric encryption. Basic screenshots are shown in Fig 3. Once the key has been received by the application via the connection to the VLC receiver, the application creates a `SecretKeySpec` that is used to encrypt a message to send or decrypt a message received using the

[1] https://github.com/UT-MPC/JIVE_VLC

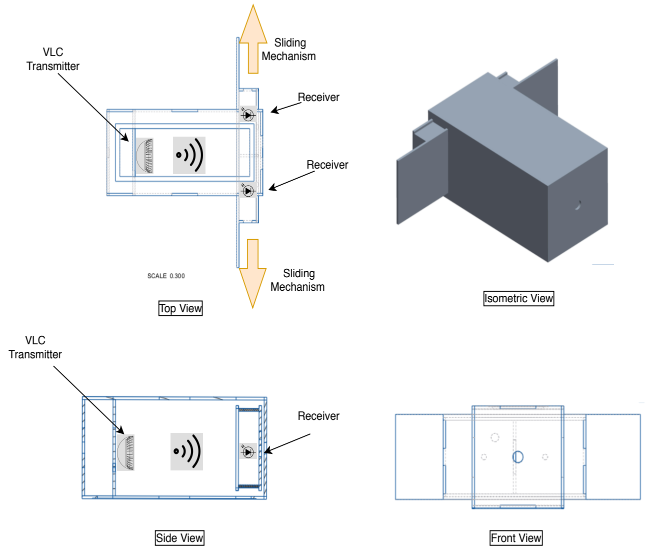


Figure 2: Orthographic projection of designed JIVE system enclosure.

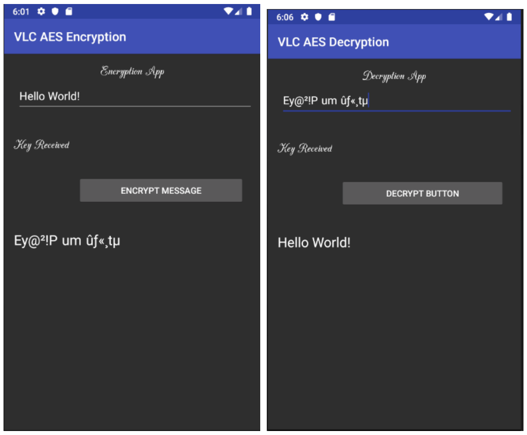
symmetric Advanced Encryption Standard (AES) algorithm. After the message has been processed, the ciphered or plain text is shown in the application for the user to view.

3 THE DEMONSTRATION

Our demonstration setup consists of Android devices with wireless communication capabilities, multiple VLC receiver devices connected to the Android devices via USB, and a single VLC transmitter. The Android phones will be connected to WiFi. When the transmitter is powered on, it will generate a random 128-bit key and project the key within the enclosed space (see Fig. 2) with VLC.

Multiple receivers can be attached to the box, as shown in Fig. 2. With a small slot in the box, we can isolate the light to either one or both receivers. To mimic the eavesdropping scenario and exhibit how our system adds a physical layer of security, we will attempt to decrypt the ciphered message with the receiver isolated from the light. This shows that without the key received via VLC, the message cannot be decrypted and is only accessible to the devices that can physically see the light.

We establish several default parameters for our VLC link. The transmitter and receivers will be approximately .304 meters apart. The length of our encryption keys will be 128 bits and the VLC message frames will be 8 bits. Though in a real deployment of JIVE, the signal in the light is imperceptible, for the purposes of the demo, we will set the data rate to a value such that the transmitter light pulses are perceivable to the human eye. If the system's symbol period is at least 800 μs, the modulation of the signal is detectable by a human [3].



Hello World!

Ey@²!P um ûƒ«¸tµ

Ey@²!P um ûƒ«¸tµ

Hello World!

Figure 3: User interface of Android application.

Encryption: We begin the demonstration by powering on the transmitter, which causes JIVE to generate a random key. With our box in Fig. 2, we can isolate the receivers so that one does not receive the key through the VLC link. We then ask the user to start the encryption app on the Android phone and connect it to the receiver that sees the light through a USB cable. Once the Android application has received the complete encryption key, it informs the user. We then ask the user to enter a message they would like to encrypt and then press the "Encrypt Message" button, triggering the AES encryption and printing out the ciphered text on screen. Finally, we ask the user to send the ciphered text to the other phone over WiFi, depicting the communication of encrypted data over a possibly unsecured medium.

Decryption: To show how our system adds a physical layer of security, we ask the user to use another phone for decrypting the encrypted text. However, we do not remove the barrier from the box, keeping the connected VLC receiver still in the dark. As the user attempts to decrypt the message, they will notice that the message cannot be decrypted, displaying that the secure key cannot be received by a device that is not physically in the room with the light. Once we adjust the box so that both receivers see the light, we ask the user again to decrypt the ciphered input by tapping the "Decrypt Button." This time the message will be successfully decrypted and the plain text can be viewed on the screen, reaffirming that only devices with access to the light are able to view the encrypted data.

REFERENCES

[1] Felipe Hernandez. 2014. USBSerial. https://github.com/felHR85/UsbSerial.

[2] Tau Ceti Co operative Ltd. 2013. Legion of the Bouncy Castle, Inc. Retrieved May 23rd, 2019 from https://www.bouncycastle.org/

[3] J. Shenoy, A. Tyagi, M. Halabe, and C. Julien. May, 2019. JIVE: Joint Integration of VLC and Encryption. Under Review. (May, 2019).