Poster: VLC-based Authenticated Key Exchange

Ying-Shen Chen[†], Chung-Yi Lin[†], Hsu-Chun Hsiao[†§], Yueh-Hsun Lin[‡], Hsin-Mu Tsai[†]
[†]National Taiwan University [§]CITI, Academia Sinica [‡]Samsung Research America

I. Introduction

Cryptographic keys are essential for secure communication. To establish pairwise secret keys in a group of devices, the devices can first exchange their authentic public keys and then use asymmetric key agreement (e.g., Diffie-Hellman) to compute shared secrets among themselves. Authenticated public-key exchange in a group (even a small one) is nevertheless challenging [1], as a man-in-the-middle or group-in-the-middle adversary can impersonate others or segregate the group.

State-of-the-art authenticated public-key exchange protocols often heavily rely on human users acting as Out-of-Band (OOB) channels to defend against strong in-band attacks. Users may be asked to type passcodes on each device, take a picture of a barcode displayed on each device's screen, or confirm whether every device displays the same information. However, human OOBs are slow and human error complicates the design and validation of security protocols.

This work explores Visual Light Communication (VLC) as an alternative OOB channel for reducing user intervention in authenticated key-exchange protocols. VLC is an emerging wireless communication technology with the potential to enhance security and user experience. VLC can provide high inherent security down to the physical layer because of its line-of-sight propagation and ease of isolation. Moreover, VLC is designed to transmit data while maintaining illumination, thus allowing data exchange in the background without annoying users. By contrast, other OOBs (e.g., humans, NFC, or LED blinking patterns) are either slow, designed for pairwise interactions, or are intrusive to the user experience.

Although leveraging VLC for security applications has been mentioned in the literature [2], there still lacks a systematic exploration of its benefits and challenges with respect to security, especially when it comes to actual deployment. Hence, in this work, we design a proof-of-concept VLC-based key exchange protocol, analyze its security and performance, and highlight practical considerations based on our experience with the ongoing implementation. In addition, we plan to formalize the problem and threat model under a more realistic setting where, for example, the visual light channel is not ideal.

II. Background

Visible Light Communications (VLC) is a type of wireless communications that utilizes visible optical signal to carry digital information wirelessly. Such a system typically alters the output intensity of an LED light source over time to represent the information being transmitted. An optical sensor is used at the receiving end to convert the optical signal to an electrical signal for the demodulation of transmitted information. A special type of VLC, Camera Communications (CamCom), makes use of a commodity camera as the main receiving component instead of a special-purpose optical sensor [3], and the transmitted optical signal can be directly extracted from the pixels occupied by the transmitter (e.g., a light fixture) in camera-captured images without any hardware modifications.

CamCom has a number of advantages. First, modern mobile devices with built-in cameras can easily become CamCom receivers after installing an app. Second, because the optical transmission is highly directional and cannot penetrate visual obstacles, the technology is **less susceptible to eavesdropping, jamming, and falsified transmission**. Finally, it is easier for the user to **visually verify whether a transmission comes from a legitimate transmitter** by looking at the appearance of the pixels where the received information is extracted and determining whether the transmission is coming from that transmitting object. The main downside of CamCom is that, due to the camera's limited frame rate of about 30 frames per second (fps), its throughput is only approximately 10 byte/s with a standoff distance of a few meters [3].

III. Protocol Design and Preliminary Analysis

Two concrete examples are considered for ease of illustration: 1) Conference participants siting in the same room would like to securely exchange their public keys and contact information by simply putting their phones on the table with the cameras facing the light source on the ceiling. 2) In home or corporate environments, an administrator would like to ensure that newly-installed devices can automatically discover authentic information about other devices nearby.

We propose VAKE, a proof-of-concept key exchange protocol that leverages VLC for improved security and user experience. VAKE enables an efficient pairwise key exchange for a small group of N devices ($N \geq 2$). VAKE allows each device to securely exchange its public key with all other $N-1$ devices; therefore this device itself could establish pairwise secrets with any device in the same group using Diffie-Hellman (DH) key agreement [4].

VAKE begins with a group of m devices as participants and a trust server S. Figure 1 provides the full protocol description. Unlike conventional protocols, VAKE leverages two types of VLC as OOB channels (unicast and broadcast) to offer secrecy and integrity between S and all participants. The VLC communication is assumed to be secure but has a low bit rate (10byte/s). Thus, WiFi is used as an in-band channel for primary data communication between the participants and S.

VAKE consists of three phases. The first phase prepares system parameters before an exchange, such as generating a

DH public key on each participant and generating a session key on S. Most important is to set the number of participants in the exchange group to all devices and S, in order to prevent potential Sybil attacks by injecting or duplicating public keys during the exchange. In the second phase, server S distributes session key k_i, WiFi SSID, and corresponding identifier d_i to each device via the directional OOB channel, to ensure the authenticity and integrity of gathered DH public keys from all participants.

Once S gathers all legitimate public keys from every participant, it redistributes the list of gathered public keys through the in-band channel, and at the same time broadcasts the data commitment of the list of public keys via its visual OOB channel. At the end of VAKE, each device should save all other participants' DH public keys, allowing each device to compute pairwise secrets by Diffie-Hellman or even establish group secrets with more than two devices from the exchange group using group Diffie-Hellman protocols [5].

VAKE: Key exchange protocol
Setup Phase
1. $U_0 \xrightarrow{UI} S$: setups the number of devices, i.e., m
2. S: generates ephemeral session key k_i and identifier d_i for device D_i where $i \in \{1, \cdots m\}$
3. $U_i \xrightarrow{UI} D_i$: setups the number of devices i.e., m
4. D_i: generates self Diffie-Hellman public key g^{n_i}

Collect Phase
5. $S \xrightarrow{VLC} D_i$: unicasts k_i, $ssid$, and d_i to D_i via a directional VLC
6. $D_i \xrightarrow{WiFi} S$: connects to S using $ssid$ and sends $g^{n_i}||d_i||\zeta_{k_i}(g^{n_i}||d_i)$ where ζ is a MAC function
7. S: gathers all public keys g^{n_i}, and verifies their authenticity with $k_i \forall i$

Distribute Phase
8. $S \xrightarrow{WiFi} *$: broadcasts the list of public keys g^{n_i}
9. $S \xrightarrow{VLC} *$: broadcasts commitment $H = h(g^{n_1}||g^{n_2}||\cdots||g^{n_m})$
10. D_i: verifies H using $m-1$ received g^{n_j} $(j \neq i)$ and g^{n_i}
11. $D_i \xrightarrow{WiFi} S$: reports the verification result $res||\zeta_{k_i}(res)$
12. $S \xrightarrow{WiFi} D_i$: gathers all verification results from D_i, $\forall i$, send back the confirmation message signed with k_i to D_i, e.g., $conf||\zeta_{k_i}(conf)$
13. D_i: computes pairwise keys $k_{ij} = g^{n_i n_j}$ for D_j, $\forall j \neq i$

Fig. 1. VAKE with a trusted server S and a group of m devices.

A. Security Verification and Comparison

We compare VAKE with SafeSlinger [1] and a simple pairwise pairing method. The table below summarizes the comparison results. VAKE requires only 6 communication exchange (Steps 5, 6, 8, 9, 11, and 12 in Figure 1), which is significantly fewer than the other methods. An immediate advantage of having fewer protocol runs is that the protocol can be formally verified using existing tools [6]. Our current verification assumes an ideal VLC that provides perfect authenticity and secrecy; modelling a non-ideal channel in the verification remains open for future work.

As for human effort, VAKE users only need to count the total number of devices in a group, whereas SafeSlinger additionally requires users to compare and select matching phrases. The pairwise pairing method is much more complicated than VAKE because users have to perform $O(n^2)$ actions for the pairwise setup.

	Prtcl Rounds	Human Efforts
Pairing	$O(n^2)$	type codes (or compare phrases) for each pair
SafeSlinger [1]	13	Counting, phrase comparison
Ours	6	Counting

IV. Implementation Challenges

We encountered several challenges while implementing VAKE using commodity mobile devices. Here, we highlight these technical issues and describe how to resolve them. The first interesting challenge is how to implement VLC-based unicast channels. For N participating devices, one solution is allowing server S to install the same number N number of light sources, facing each device directly. A more economic approach is installing only one light source, but rotating it to deliver signals to all N devices at a steady speed (like a round robin approach). The second challenge is the trade-off between the receiving data rate and distance when leveraging the VLC channel. According to our experiments, the error rate increases exponentially when the distance between the light source and the deployed mobile device increases. By applying error-correcting codes, we could mitigate this issue at the cost of the data rate and transmission latency. We are investigating additional methods to overcome these implementation challenges.

V. Conclusion and Future Work

This study aims to explore the challenges and opportunities of using Visual Light Communication for authenticated key exchange. We propose a secure, scalable and easy-to-use protocol that is suitable for establishing pairwise keys in a large group of constrained devices. Our preliminary work so far indicates several interesting observations and future directions. For example, we observe that the use of VLC can significantly simplify protocol, thus making it suitable for automated security validation. Future work includes (1) relaxing assumptions such as the trusted server and directional lights; (2) protocol implementation, and 3) understanding the realistic behaviors of VLC, especially for security applications.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology, National Taiwan University, and Intel Corporation under Grants MOST 105-2633-E-002-001 and NTU-ICRP-105R104045.

References

[1] M. Farb, Y.-H. Lin, T. H.-J. Kim, J. McCune, and A. Perrig, "Safeslinger: easy-to-use and secure public-key exchange," in *ACM MobiCom*, 2013.
[2] A. Jovicic, J. Li, and T. Richardson, "Visible light communication: opportunities, challenges and the path to market," *Communications Magazine, IEEE*, vol. 51, no. 12, pp. 26–32, 2013.
[3] H.-Y. Lee, H.-M. Lin, Y.-L. Wei, H.-I. Wu, H.-M. Tsai, and K. C.-J. Lin, "Rollinglight: Enabling line-of-sight light-to-camera communications," in *ACM MobiSys*, 2015.
[4] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.
[5] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 11, no. 8, pp. 769–780, 2000.
[6] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *Computer aided verification*. Springer, 2008, pp. 414–418.