Location-based Access Control Providing One-time Passwords Through 2D Barcodes

Mirco Schönfeld, Martin Werner, Florian Dorfmeister

Mobile and Distributed Systems Group
Ludwig-Maximilians-University Munich
mirco.schoenfeld@ifi.lmu.de
martin.werner@ifi.lmu.de
florian.dorfmeister@ifi.lmu.de

Abstract: Location-based Access Control often relies a location proof, which ensures that a specific user is at a specific location. These locations are most often inferred from measurements. As a consequence, such systems are never trustworthy. An attacker can simply fake sensor measurements or even sensor data in absence of a trusted measurement module. Moreover, the measurement data is typically stable over time at a fixed location and can thus be replayed at later times. With this paper, we propose a system, which can provide functionality for a location proof, which does not rely on measurements and does not suffer from replay attacks. Therefore, a self-contained system is generating signed one-time passwords and communicates them via 2D barcodes for authentication of camera-enabled devices being in a specific location.

1 Introduction

An attractive and fast-growing business area in mobile context is tying certain services to specific geographic locations. Such services often provide information to which access is granted if and only if the user is physically present at a specified location. Many of these location-based services rely heavily on the correctness of location information. The verification of such information presents an active area of research, because there are certain scenarios in which tampering with sensor data to fake locations is feasible.

Moreover, for the use-case of location-based access control it is often much simpler to couple access control with interaction. Especially, if access control points are rare. A common technique to provide such functionality is given by Near-Field-Communication [ABPW07, AK06, GS08]. However, Near-Field-Communication is not implemented in recent Apple devices and hence does not allow for a "Bring-Your-Own-Device" philosophy.

If sensor data should not be trusted, infrastructure-based localization is needed, which usually introduces severe privacy risks. Therefore we can conclude, that only an architecture, which provides proximity or location information in line with the user's intention can be successful. The proposed system consists of a backend service the user wants to authenticate with and a small computing unit that is placed at a specific geographic location. To activate a computing unit against our backend service a public/private key pair is created inside this computing unit and the associated public key is registered in the backend service. After activation the unit should be physically bound to a fixed location.

For each user requiring access to the location-based information the computing unit constructs a one-time password. For this purpose it uses the cryptographic hash function SHA-1 to generate a digest from an internal counter and a timestamp. This message digest is then encrypted with the unit's private key. The resulting one-time token consists of the plain-text representation of both the counter and the timestamp together with the message digest and is transmitted to the user via a QR code. To gain access the user authenticates himself to our backend service using this one-time token. As only the backend service is able to verify the integrity of the token and as the counter effectively prohibits replay attacks, the system is secure. Moreover, the intention of the user gets expressed by scanning a visual code and hence, privacy problems occuring due to tracking of people without their attention and permission are absent.

In this paper we present a prototypical implementation of our architecture based on the open-source prototyping platform Arduino [Ard]. For transmitting the authentication token we employ QR-codes. We therefore equipped the Arduino with a LCD display, which is capable of displaying such two-dimensional barcodes. In contrast to using NFC for the token transmission merely all smartphones ship with the required hardware: a sufficient camera.

2 Implementation Details

As an appropriate basis of our prototypical implementation we employed the Arduino Mega2560. Arduino is an open-source prototyping platform that ships with a complete toolchain for implementing and running C code. The Mega2560 is run by an Atmel ATMega2560 microcontroller at 16MHz that is part of the Atmel AVR microcontroller-family. It is capable of calculating signature digests in a reasonable amount of time.

Figure 1: Arduino Prototype With LCD Display Attached

The Arduino-based controller was equipped with a SainSmart 3.2" TFT LCD Display module driven by an SSD1289 display controller, where a display interface library is readily available [UTF].

The open source software *qrencode* was modified to run on the Arduino and to display QR-codes via said library. Cryptographic routines are based on Colin Plumb's BigNum libary [BNL], which basically provides modular exponentiation of big integers to systems with CPUs supporting 32 bit integer data natively.

As for hashing, the well-known and widely used SHA-1 digest was used, though it is known to have some smaller weaknesses. In 2005, Wang et al. successfully attacked an SHA-1 digest with fewer than 2^{69} hash calculations [WYY05], which they were able to lower to 2^{63} calculations only a few months later [SHA]. Thereby, they reduced the theoretical bound of 2^{80} hash calculations of a brute-force-attack. However, we consider this acceptable for our prototypical implementation as the hashing algorithm could be easily replaced later on.

The system basically runs in a loop. However, in the initialization phase, a public/private key pair is generated and the public key is displayed using a QR-code. This QR-code can then be scanned and transmitted to a backend, where this one-time password generator is then accepted for certain service authentications. The private key is held in the device. For high-security applications, this initialization scheme should of course be implemented using a tamper-proof module or a smart card. However, a physical shielding of the device is sufficient in most situations.

Now, the system runs in a loop with configurable delay, incrementing the internal counter at each iteration, generating a signed one-time password from this counter and a timestamp, and showing this information as a QR-code readable by almost any smartphone. By transmitting this signed one-time token to a backend, the service is able to uniquely identify any known unit and trigger an appropriate response - the backend decides whether or not to grant access. A simplified scheme is given in figure 2.

In this way, location-based authentication can be triggered on a backend without an interconnected infrastructure. Neither active nor passive localization is needed, nor an Internet connection of any device except for the device, which wants to authenticate to an online service, of course.

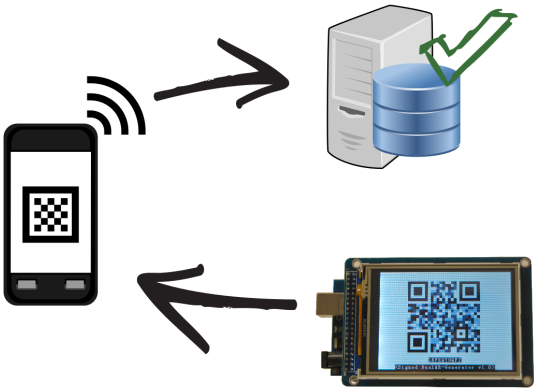

Figure 2: Simplified Authentication Scheme

3 Use-Cases

The possible application of our architecture is diversified. It could be used to grant physical access to a restricted area since the computing unit is uniquely identifiable through its public/private key pair. Therefore, the geographic location of each unit has to be provided together with the public key in the initialization phase. Additionally, each user has to be registered with the backend through a seperate public/private key pair, too. The one-time token that has been extracted from the QR-code is signed by the user's private key before being transmitted to the backend. Thereby, the backend can both identify the user and the location that the user tries to gain access to. And thereby, this scheme can be incorporated into a fine-grained, centrally manageable room access control system.

Another use-case is mobile payment. A gas pump at a gas station for example encrypts all necessary payment details into a signed QR-code. The user decodes the QR-code and transmits the signed token to a backend. The server verifys both the gas pump and the user and triggers a payment process after successful verification.

4 Conclusion

With this paper, we presented a framework for presence-based authentication with respect to Internet services, while the authentication device does not need on any network connection.

Moreover, we developed a prototyping platform for display-based communication and gave a working proof-of-concept implementation of our approach. From a hardware perspective, the framework only needs a microcontroller, which is able to calculate hashes and perform RSA encryption and a diplay to transmit a digest via QR code. This display can easily be replaced by any near-field communication technology such as NFC or Bluetooth. However, using a display leads to authentication with a clear intention of the user. It is rather unlikely to accidentally scan a matrix code as compared to performing an NFC touch.

In future work, this will be integrated into a middleware supporting NFC, Bluetooth and other wireless personal area networks for transmission of said access tokens aiming at transparent support of interaction-based authentication for online services using smartphones, including NFC-enabled ones and rather simple ones.

References

[ABPW07] Y. Anokwa, G. Borriello, T. Pering, and R. Want. A user interaction model for NFC enabled applications. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*, pages 357–361. IEEE, 2007.

[AK06] Z. Antoniou and D.N. Kalofonos. NFC-based mobile middleware for intuitive user interaction with security in smart homes. In *Proceedings of IASTED CSN'06*, 2006.

[Ard] Arduino Open Source Electronics Prototyping Platform. Online, last accessed: 17.10.2012. `http://arduino.cc/`.

[BNL] bnlib - An SDK for Big Number Arithmetic. Online, last accessed: 17.10.2012. `http://philzimmermann.com/EN/bnlib/index.html`.

[GS08] K. Griffin and C. Stone. Near Field Communication Activation and Authorization, 2008. US Patent App. 12/241,557.

[SHA] New Cryptanalytic Results Against SHA-1. Online, last accessed: 17.10.2012. `http://www.schneier.com/blog/archives/2005/08/new_cryptanalyt.html`.

[UTF] UTFT Library. Online, last accessed: 17.10.2012. `http://henningkarlsen.com/electronics/library.php?id=52`.

[WYY05] X. Wang, Y. Yin, and H. Yu. Finding collisions in the full SHA-1. In *Advances in Cryptology–CRYPTO 2005*, pages 17–36. Springer, 2005.