QR CODE-BASED IDENTIFICATION WITH MOBILE DEVICES

Giovanni Schmid[1] and Francesco Rossi[2]
[1]*High Performance Computing and Networking Institute (ICAR), Naples, Italy*
[2]*University of Naples Parthenope, Naples, Italy*

Keywords: Identification (entity authentication), Zero knowledge protocols, QR codes.

Abstract: Mobile devices are becoming ubiquitous, getting rise to a pervasive network through which people can share information and get also very complex services. A key factor for the security of both consumers and providers in this emerging business scenario is the ability for a user or a service to reliably and efficiently authenticate itself. In this paper, we consider a unidirectional visual channel of interaction between the user and the service. Identification indeed takes place by using a QR Code symbol which is displayed or scanned by the mobile device of the user in the proximity of an access point for the service. We consider protocols for strong authentication which, if correctly implemented, does not reveal any useful information both to the verifier and to any unauthorized observer (zero-knowledge protocols). Our experimental results show the feasibility of our approach for a wide range of mass-market devices and applications, including physical access to restricted or pay-per-use areas (military or parking zones, etc.), logical access to resources or services (e.g., ATMs, computer systems and Internet services), and privacy-aware voting and testing centers.

1 INTRODUCTION

Thanks to computing and networking technologies, and because of the incentives of availability, lower costs and easier use, a variety of services requiring interactions in the physical world with their users are being digitalized, and somewhat virtualized and augmented.

On the other side, as camera-equipped and display-equipped mobile devices (in particular, smartphones and tablet PCs) approach ubiquity, they become excellent platforms for the deployement to million of users of the above services in a user-friendly but secure way. Indeed, because of the general proliferation of services at our disposal as mobile users, these could act as consolidation platforms that avoid the inconveniences and, in some cases, the security threats deriving by using many different special-purpose devices and tokens (e.g. identity cards, one-time password generators, credit cards, membership cards, etc.). Of course, one could object that the centralization of different security services is a threat in itself. However, we believe that simple designs - combined with software security principles and prudent engineering practices - can result in multi-purpose, service-oriented mobile platforms whose use would be by far more secure and privacy-compliant than our actual habits in managing an increasing number of different tools.

In the present work, we propose to use the camera or the display of a mobile device as a unilateral visual channel to achieve strong entity and message authentication thanks to Quick Response (QR) coding (ISO/IEC, 2006b). Although today's devices increasingly feature convenient, wireless communication interfaces (e.g. Bluetooth, WiMax and 802.11), the visual channel offers the advantage of being immune to Man-in-the-Middle (MitM) (Menezes et al., 1997) attacks in suitable usage contexts, when one or both the parties in communication were not previously authenticated. Indeed, conversely than the wireless channel, the visual one can provide for *demonstrative identification* (Balfanz et al., 2002) of the communicating devices. Moreover, by using the visual channel we can deploy cryptographic protocols whose security does not rely on those of the underlying wireless transmission protocol and its related software subsystem, which historically suffer of many vulnerabilities (for a report on Bluetooth security, see for example (Bialoglowy, 2010a; Bialoglowy, 2010b)).

In the following, we will focus on a client-server architecture and two kind of application scenarios:

1. The enrollment of a user to a service and its subsequent secure accesses to it through personal iden-

Schmid G. and Rossi F..
QR CODE-BASED IDENTIFICATION WITH MOBILE DEVICES.
DOI: 10.5220/0003807200790086
In *Proceedings of the 2nd International Conference on Pervasive Embedded Computing and Communication Systems* (PECCS-2012), pages 79-86
ISBN: 978-989-8565-00-6

tification, eventually in pay-per-use mode;

2. A trusted service giving corroborate evidence to users of having got a correct interaction and output by it thanks to a service's receipt, but in such a way that the receipt does not transfer the above evidence to unauthorized parties.

Examples fitting in scenario 1 include physical access to restricted or pay-per-use areas, such as parking zones or military areas, or logical access to resources or services like ATMs, Internet services and computer systems. In this case the user acts as a prover, that is she claims an identity to be proved, whilst the service is the verifier. Auditing and non-repudiation mechanisms should be put in place at the server-side, expecially in case of pay-per-use or critical services.

As per scenario 2, noticeable examples are those of voting and testing centers. Conversely than in the previous cases, the roles of prover and verifier are now played by the service and the user, respectively. Moreover, user's privacy concerns take now precedence over service's tracking issues. Suppose for example an automated analysis facility for a given genetic syndrome: a user would be sure to have correctly performed the test and to have got the right answer, and she would perhaps share that result with some physicians and other people. However, she would be probably displeased if such analysis results were made public or could be used against her as proof of disease by a life insurance company.

For both application scenarios, we assume a unidirectional visual channel built by displaying a single QR Code symbol by the prover and by acquiring such symbol by the verifier through a QR Code scanner or a camera.

The symbol contains the session transcript of a zero-knowledge identification protocol, that is a strong identification protocol which has the property of not revealing any useful information both to the verifier and to any unauthorized observer. We consider two of such protocols, one suitable for the requirements of scenario 1 and the other fitting those of scenario 2.

We implemented a prototypal system and run experimental tests which show the feasibility and practicality of our approach.

The paper is organized as follows. Section 2 discusses related work. Sections 3 and 4 introduce to the entity authentication problem and zero-knowledge protocols, respectively. We focus in particular on two protocols, suitable for the application scenarios discussed above. Section 5 serves as a brief introduction to QR Code technology, whilst in Section 6 we outline the sequence of operations involved in QR-Identity, a prototypal QR Code-based identification system implemented as a proof-of-concept of our approach. Some experimental results with QR-Identity are reported in Section 7. They show the good performance of the system, and the fact that, assuming a correct scanning positioning, it is is immune both to false positives and false negatives in various illumination conditions. Finally, Section 8 draws up conclusions and future work.

2 RELATED WORK

A system that uses 2D barcodes and camera phones as a visual channel was proposed in (McCune et al., 2009). The authors used that channel for human-verifiable authentication, providing the user assurance that her device is communicating with another device visually recognized by her. This is a special case of demonstrative identification (Balfanz et al., 2002), which is used to rule out MitM attacks and allowing the bootstrap of authentic public keys. The approach used in (McCune et al., 2009) can be profitably combined with ours to allow users in application scenario 2 to acquire "at runtime" the service's authentic public-key thanks to the camera of their mobile devices.

The visual channel available using barcode technology may have insufficient bandwidth for an efficient and robust implementation of cryptographic techniques, also for some 2D barcode approaches. In (McCune et al., 2009) the workaround was the use of multiple symbols to encode hash function digests of public-keys, at the dual cost of a degradation in security and a major complexity (and errors) in the scanning process. However, this and other older approaches intended to overcome the bandwidth limitation of a visual channel, as that proposed in (Laur and Nyberg, 2006), appear awkward and all in all useless after the standardization of 2D barcodes which are capable of encoding thousands of alphanumeric characters in just one code symbol.

Although we chose QR Codes (ISO/IEC, 2006b) for our implementation, some other standardized 2D barcode technologies exist which can accomplish the requirements of the application scenarios considered in Section 1, e.g. *Data Matrix* (ISO/IEC, 2006a) and *Aztec Code* (ISO/IEC, 2008). A comparison between these technologies was outside the scope of our present work.

3 ENTITY AUTHENTICATION

Entity authentication (in the following also referred as *identification*) is perhaps the most basic security service. It is a functionality required by any access control framework both in physical (e.g. gates and guard stations) and logical security domains (as in computer login, network access control, access to data, etc.); moreover, it is a pre-requisite for authenticated keys exchange in network protocols, which in turn is required in secure communications.

The general setting for an entity authentication protocol involves a *prover* or *claimant* A and a *verifier* B. B is presented with the supposed identity of A, and the goal of the protocol is to give B corroborate evidence that the identity of the claimant is indeed A. The most basic objective of an identification protocol just states its effectiveness:

- *Completeness:* in the case of honest parties A and B, B - at least with overwhelming probability - will complete the protocol having accepted A's identity.

Protocols trying to satisfy only the completeness property are called *weak* identification protocols. Examples are fixed password schemes; they are subjected to *replay* attacks (Menezes et al., 1997), exhaustive password search, password-guessing and *dictionary* attacks (Menezes et al., 1997). Therefore, we will focus only on *strong* identification protocols, that is protocols having the following two security objectives:

- *No-impersonation:* the probability is negligible that any party C different from A, carring out the protocol and playing the role of A, can cause B to complete the protocol accepting A's identity.
- *No-transferability:* the probability is negligible[1] that B can reuse a previous identification by A to successfully impersonate A in an identification session with a third party C.

Ideally, in strong identification protocols no-impersonation and no-transferability properties should remain true also if:

- a (polinomially) large number of previous protocol executions between A and B has been run, and eventually observed by C;
- C has participated in previous protocol execution with either A or B, and;
- multiple instances of the protocol, possibly initiated by C, may be run simultaneously.

[1]The meanings of the adjectives overwhelming and negligible depend on the application, but generally imply that the probabilities of failure are not of practical importance.

Strong identification protocols achieve their objectives through a challenge-response interaction between the verifier and the prover: B generates a time-variant *challenge* for A, and A demonstrates the knowledge of a secret, associated by construction with her identity, by performing a suitable cryptographic operation on the challenge. The output of the above cryptographic operation is the *response*, and it depends on both A's secret and B's challenge.

Challenge-response identification protocols can be based both on symmetric-key techniques and asymmetric-key ones. In the first case, the claimant and the verifier share a secret key, and large systems require a trusted on-line server to effectively manage keys for all the couples of communicating parties. In our architecture, the role of trusted on-line server could be well implemented at the server side of the application. However, that would result in a server database containing the secrets of all the users enrolled with the service, a condition which exposes to massive identity thefts and, for some application scenarios, to the "big-brother" issue.

Public-key based protocols are immune to these drawbacks, since no secret must be shared by the claimant and the verifier. There are two most typical ways in which public-key based identification protocols can be obtained, which consist in using asymmetric techniques for realizing the challenge-response interaction:

- A decrypts a challenge encrypted by B under its public key, or alternatively;
- A digitally signs a challenge that B will verify.

The asymmetric system used for such mechanisms should not be susceptible to *chosen-ciphertext attacks* (Menezes et al., 1997), otherwise C may attempt to extract information by impersonating B and chosing strategic rather than random challenges. Although this concern may be addressed by incorporating a random number (a so called *confounder*) into the data over which the response is computed, a more radical approach consists in turning to *zero-knowledge* protocols. These have indeed the property of not even reveal *any* partial information which makes C's task any easier whatsoever, no matter how many protocol executions C could have been monitored.

4 ZERO-KNOWLEDGE PROTOCOLS

Zero-knowledge protocols were introduced in (Goldwasser et al., 1987) (see also (Goldwasser et al., 1989)), as special cases of *interactive proof systems*, where tipically the prover A and the verifier B repeat a three-moves commit-challenge-response protocol until B rejects a response from A, or otherwise the probability of cheating by A lowers under a previous established security threshold (in which case B accepts A's claim). The set of commit-challenge-response moves related to a protocol session is called a protocol *transcript*, and if the transcript represents a (probabilistic) proof of knowledge of a secret (e.g. a private key) by A, then the interactive system is said an interactive proof system *of knowledge*.

Technically speaking, the zero-knowledge property for an interactive proof system of knowledge, as reformulated by (Feige et al., 1988), can be stated as follows:

- *Zero-knowledge:* the system admits a *simulator*, that is a polynomial time algorithm which can produce, upon input of the assertion to be proven but without interaction with the real prover A (i.e. without knowledge of A's secret), transcripts that are probabilistically indistinguishable from those resulting from interaction with A.

Thus, the zero-knowledge property implies the no-disclosure properties previously stated for ideal strong authentication systems, but in a stronger sense, since *any* number of protocol runs does not increases the chances of subsequent impersonations or identity transfers even by computationally unbounded adversaries[2].

Compared to protocols based on asymmetric encryptions or signatures, zero-knowledge protocols offer no degradation of security with usage, being in particular immune to chosen-text attacks. Moreover, in some cases they are between one and two orders of magnitude most efficient - in terms of modular multiplications performed by the prover - than an RSA private-key operation (Menezes et al., 1997). This is particularly appealing for our application scenario 1 (see Section 1), being the prover constrained by the computational and power autonomy limits of mass-market mobile devices.

For our prototypal implementation (see Section 6) we have selected Schnorr identification protocol (Schnorr, 1990; Schnorr, 1991) and Jakobsson signature protocol (Jakobsson et al., 1996). Both are zero-knowledge protocols based on the *discrete logarithm problem* (Menezes et al., 1997), and - as such - they can take great advantage in term of memory requirements (key sizes), computations (number of modular operations) and communication bandwidth (transcript sizes) if reformulated in the arithmetic framework of *elliptic curve cryptography* (Hankerson et al., 2004).

Schnorr protocol is suitable in application scenarios like the first one described in Section 1, whilst Jakobbson protocol fits the requirements demanded by the voting center scenario described therein.

Jakobsson signature protocol allows for *designated verifiers* at the cost of some more computational efforts both at server-side and client-side, and a bigger transcript. The verifier B is designated in the sense that he (and he only) can simulate correct protocol transcripts. This means that a third party C cannot be convinced by B of a successful A's identification previously occured with him. Thus, Jakobsson protocol is suitable when a user B whishes to have a receipt R corroborating his correct interaction with a service A and its output, avoiding that any other party C can get the same evidence from R.

For both protocols, we are interested in their non-iteractive versions, since in any case we assumed a unidirectional visual channel used only by the prover. That can be easily obtained from protocol's original versions thanks to the *Fiat-Shamir heuristic* (Fiat and Shamir, 1987), which consists in replacing the random challenge of the verifier with a pseudo-random hash digest of the concatenation of the prover's identifier (or the message being signed by her) with the commitment. Actually, the Fiat-Shamir heuristic transforms an interactive proof of knowledge system into a non-interactive digital signature scheme. However, since the adoption of the visual channel implies the *presence* of the prover A during a protocol session, this turns out again into an identification session, provided that previous transcripts are recorded by the verifier to avoid replay attacks.

5 QR CODES

QR (Quick Response) Code is a two dimensional Barcode developed by Denso Wave Corporation and now established as an ISO (ISO/IEC18004) standard (ISO/IEC, 2006b). QR Codes data encoding capacity depends on their *version*, each version being characterized by having a different number of *modules*, that is the black and white dots that make up a QR Code. A QR Code is omni-directional readable through position detection patterns located at three of its four corners. Moreover, it has error correction capability: data can be restored even if the QR Code is partially dirty or damaged. Information is coded in vertical and horizontal directions, so the last version (version 40) can encode up to 4,296 alphanumeric characters, several hundred times more data than traditional bar-

[2]This assertion refers specifically to *perfect* zero-knowledge. In *computational* zero-knowledge the assertion remains true only for polinomially bounded adversaries.

codes.

This high capacity in data encoding and the error correction capability are very important for our approach, making QR Codes very suitable for the application scenarios depicted in Section 1. As shown in Section 6.4, it is possible to encode an identification protocol transcript in just a single QR Code of suitable version, avoiding the display of more than one image that would result in a more complex, slow and error-prone protocol. Various running tests with a previous prototype (see Section 7) show indeed that the adoption of a coding scheme producing multiple images for a single session (e.g. coding the protocol transcript by means of naive binary matrices) could result in high transmission times and high false negatives in authentication, because of severe limitations imposed by mass-market displays and acquisition devices.

Figure 1 depicts a QR code resulting from an identification session with the Schnorr zero-knowledge protocol. The code has version 13, error correction level H (High)[3], and contains an item description 55 characters long for a total of 231 characters protocol transcript. In this case, the minimum QR symbol size is of 57.75 mm for a scanning distance of 150 mm.

Figure 1: A QR code resulting from an identification session and having a minimum scanning distance of 150 mm.

6 A QR CODE-BASED IDENTIFICATION SYSTEM

As a proof-of-concept of our approach, we developed *QR-Identity*, a prototypal client-server application implementing both Schnorr and Jakobsson non interactive protocols, and which uses QR Codes symbols to convey protocol transcripts through the visual channel provided by the display of a mobile phone at the client side and a webcam at the server side. The application allows the owner of the mobile phone to perform identification or signature sessions w.r.t. an access control point implemented through a webcam-equipped laptop[4].

6.1 Application Workflow

Restricting for brevity our description to scenario 1 of Section 1, the goal of QR-Identity is to allow the owner (and only the owner) of a mobile phone to get a service thanks to a visual identification session realized via a QR Code symbol. To obtain the service (see Section 1 for a list of possible services), the user has to use its mobile device to generate and display a single QR Code symbol in proximity of the service's access point, getting back the service feedback in case of a successfull identification, and nothing otherwise. Since we make use of asymmetric cryptography, the identification process assumes that the service has got an authentic user's public key. This is achieved by means of a one-time, suitable *enrollment phase* of the user with the given service. Scopes of the enrollment phase are both user's registration in a service's database and the creation of a tamperproof copy of the client component of QR-Identity for such a user.

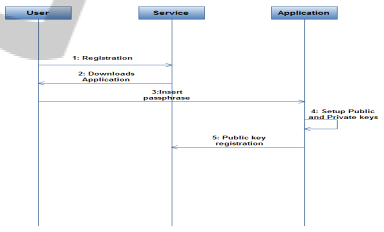

Figure 2: Sequence diagram of the enrollment phase for QR-Identity.

We suppose that the user registers itself to the service through the service website and a standard HTTPS connection. The registration process (see Figure 2) requires the input of information that properly identifies users according to service's requirements, plus the choice of a *passphrase P* as authentication token to access QR-Identity client-side. After registration is successfully completed, the user can download and install a trusted copy of the client component of QR-Identity, released with the identifier *A* of the user w.r.t. the service and preconfigured to be accessed

[3]This is the highest correction level, with a maximum of 30% restorable codewords.

[4]We must stress here that, in the cases for which we considered the Jakobsson protocol (i.e. the application scenario 2 of Section 1), the QR Code symbol should actually be displayed at the server side and captured through the user's camera-equipped mobile phone. We considered this aspect uninfluential for our proof-of-concept

only with the passphrase P defined during the registration process. The authenticity and tamperproof properties of a such client component are guaranteed by a suitable digital signature generated by the service and added to the code.

To access the application, the user has to input P, which is also used to encrypt on disk her private key thanks to a symmetric cipher (e.g. AES (FIPS, 2001)). At its first run, the client of QR-Identity generates a couple $(a, \tilde{a})$ of private-public keys for A and, by connecting through the login coordinates (A, P), it registers $\tilde{a}$ as A's authentic public key in the service's database.

After the out-of-band, one-time enrollment phase described above, the user can access the service a potentially unlimited number of times through the identification process described in Figure 3. The sequence diagram depicted therein reflects the software modules composing the application both at the client-side and at the server-side. As stated in Section 1, the client-side acts as prover in case of application scenario 1, and as verifier in case of scenario 2. For the server-side, roles are obviously inverted.

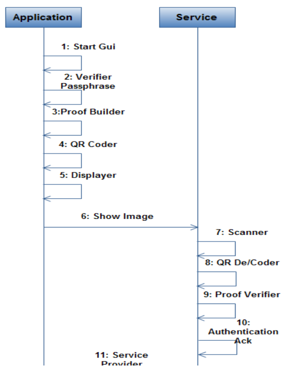

Figure 3: Sequence diagram of the identification phase for QR-Identity.

6.2 Computational Costs and Bandwidth

The computational costs for our application are mainly related to the modular arithmetic computations (exponentiations in the first place and, secondary, multiplications) which are required by prover and verifier to create and verify a protocol transcript, respectively, since the transmission of transcripts along the visual channel occurs with almost no delay.

A Schnorr protocol identification session requires one modular exponentiation and one modular multiplication by the prover, whereas the verifier has to perform two modular exponentiations and one modular multiplication.

A signature verification session with Jakobsson protocol requires instead four modular exponentiations and two modular multiplications by the prover, whilst the verifier must perform six modular exponentiations and three modular multiplications.

Bandwidth occupancy of the visual channel is particularly relevant w.r.t. the robustness of the application, since too big transcripts result in QR code symbols that are difficult to scan.

Assuming an identifier A 64 bits long and for the system parameters the default sizes of $|p| = 1024$ and $|q| = 160$ bits, a Schnorr protocol session results in a transcript of 1264 bits, including separators. Thus, Schnorr identification transcripts can be conveniently coded by using a QR Code version 12 symbol with the highest error correction level H (ISO/IEC, 2006b). The adoption of the highest error correction level is motivated by the fact that, depending on the operating environment (illumination, presence of shadows, light reflections, occlusions, etc.), the visual channel is generally affected by noise.

With the same sizes for p, q, and assuming 256 bits[5] for the receipt R, Jakobsson transcript takes instead 2824 bits including separators, requiring a version 20 symbol for the H error correction level.

Both transcripts, public-key sizes, and computations can be substantially reduced by considering versions of the above protocols based on elliptic curve cryptography, as we did in QR-Identity (see Section 6.4). This is especially advisable for applications like those described in scenario 2 of Section 1. In this case, in particular, the transcript sizes for Schnorr and Jakobsson scale down to 584 and 1416 bits, respectively, allowing ligther symbols (i.e. versions 8 and 13) to encode sessions.

6.3 Security Considerations

Since 2D barcodes are intended to be machine readable only, a human cannot distinguish between a valid and a tampered code. In many usage scenarios, that circumstance turns out in 2D barcodes as possible attack vectors toward automatic readers. (Kieseberg et al., 2010) gave a comprehensive analysis of QR code tampering in view of obtaining DOS or various command injection attacks. These threats however do not apply for our usage scenarios, where barcode symbols do not code cleartext information to access some service or resource, but are instead crypto-

[5]According to (ISO/IEC, 2006b), alphanumeric encoding requires about 5.5 bits/char. Thus 256 bits corresponds to about forthy five alphanumeric characters.

graphic tokens that have to be processed by a control enforcement point. Moreover, in our case symbols are displayed at video rather than printed on paper. Code tampering based attacks are thus ruled out, and the only threat to be considered is symbol cloning for identification thieft purposes, which on the other hand is nullified by the immunity of the visual channel to MitM attacks and by the adoption of strong identification protocols.

6.4 Implementation Details

Although we deployed QR-Identity on a Samsung S5660 with Android OS (client-side) and a x86-based laptop with Windows Vista OS (server-side), it can be easily ported to a wide range of hardware devices and operating systems. It was indeed developed in the Java programming language. Besides portability, Java offers a complete set of APIs both for ordinary and elliptic curve cryptography (through the proprietary java SE 6 (Sun-Oracle, 2010) or the open-source *Bouncy Castle* (Bouncy-Castle, 2011) crypto package). Moreover, the JRS-257 standard (Java-Community, 2011) has been published which adheres to ISO/IEC18004.

At the server side, we used the Bouncy Castle package both to implement public key cryptography on elliptic curves and to do hash computations. At the client side, we instead used for all crytographic computations the Spongy Castle(Tyley, 2011) library, a special porting of Bouncy Castle for the Android operating system.

Barcode management was realized both in client and server through the Zxing library (ZXing-Community, 2011), which currently is the most adopted open-source, multi-format 1D/2D Barcode image processing library.

7 EXPERIMENTAL RESULTS

The main critical aspect of identification systems based on barcodes is the fact that symbols could be not readeable because of a wrong scanning distance or other factors such as optical occlusions and background "noise" in the scanning environment. Of course the scanning range decreases as symbol version is increased, so our first test set was to check the practicality of our approach in term of admissible distances between the user's mobile phone and the camera of the service's access point.

Figure 4 shows a comparison of the scanning range for four QR Code symbols having the same size of 45 mm (which can be displayed on the majority of,

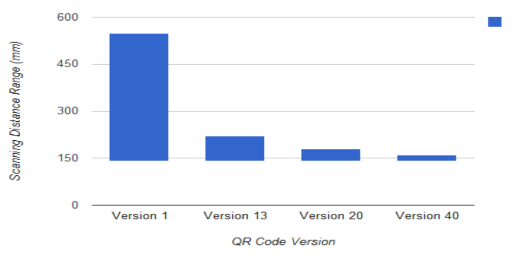

Figure 4: Scanning range of four QR code symbols having the same size (45 mm) but different versions.

if not all, the mobile phones nowadays on the market), and whose versions include those effectively used by QR-Identity (see Section 6.4). The results show a scanning range from about 150 mm to about 200 mm for all the considered symbol versions, which has to be considered viable for many usage scenarios.

The second test set simulated scanning environment contexts that could not allow symbol decoding. We did precisely the following three kinds of tests:

- *Scene Background Noising.* For a correct acquisition of a symbol, it is very important that the symbol is detached from the background. We considered three working hypothesis: uniform background, low and high noised background. The background was reproduced as a black and white pattern. In all cases symbols were correctly decoded thanks to the position detection patterns included in the QR Code structure.
- *Partial Occlusion of Position Detection Patterns.* Damaging or partially covering one or more detection patterns resulted in no acquisition of symbols. Instead, symbol decoding was possible in case of data areas partially occluded or damaged, provided that the error correction level parameter was adequately set.
- *Addition of Faked Position Detection Patterns.* The addition of false position detection patterns did not affect negatively the decoding process, since it makes use of spatial information that avoid the recognition of faked corner in the scene.

All tests were made with day brightness conditions, and without shadows or light variations.

8 CONCLUSIONS AND FUTURE WORK

In this work we propose to use the camera or the display of a mobile device as a unilateral visual channel to achieve strong entity and message authentication thanks to Quick Response coding. The joint use of a visual channel and zero-knowledge protocols results in very efficient, reliable identification and signature systems suitable for various usage scenarios in which a contactless interaction between the user and the service access point is required or preferred. As a proof-of-concept of our approach, we developed *QR-Identity*, a prototypal client-server application which allows the owner of the mobile phone to perform identification or signature sessions w.r.t. an access control point implemented through a webcam-equipped laptop. The experimental results show the good performance of QR-Identity, and the fact that, assuming a correct scanning positioning, it is is immune both to false positives and false negatives in various illumination conditions.

Future work will include a comparison of QR Codes with other 2D barcodes such as Aztec codes and Matrix Data, through the realization a comprehensive test-set verifying their readability in presence of occlusions, shadows, rotations, light reflections, etc. Moreover, we are working on new zero-knowledge protocols having specific security properties, and whose computational costs allow their utilization in scenarios like those considered in this paper.

REFERENCES

Balfanz, D., Smetters, D., Stewart, P., and Wong, H. C. (2002). Talking to strangers: authentication in adhoc wireless networks. In *Symposium on Network and Distributed Systems Security (NDSS)*. Internet Consortium.

Bialoglowy, M. (2010a). Bluetooth security review, part 1. http://www.symantec.com/connect/articles/bluetooth-security-review-part-1.

Bialoglowy, M. (2010b). Bluetooth security review, part 2. http://www.symantec.com/connect/articles/bluetooth-security-review-part-2.

Bouncy-Castle (2011). Crypto apis version 1.46. http://www.bouncycastle.org/.

Feige, U., Fiat, A., and Shamir, A. (1988). Zero-knowledge proofs of identity. *Journal of Cryptology*, 1.

Fiat, A. and Shamir, A. (1987). How to prove yourself: practical solutions of identification and signature problems. In *Advances in Cryptology - CRYPTO 86, A.M Odlyzko (Ed.), LNCS 263*. Springer.

FIPS (2001). *Federal Information Processing Standards Publication 197 - AES*.

Goldwasser, S., Micali, S., and Rackoff, C. (1987). The knowledge complexity of interactive proof systems. In *Advances in Cryptology - CRYPTO.*. Springer.

Goldwasser, S., Micali, S., and Rackoff, C. (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1).

Hankerson, D., Menezes, A., and Vanstone, S. (2004). *Guide to Elliptic Curve Cryptography*. Springer.

ISO/IEC (2006a). *Information technology - Automatic identification and data capture techniques - Data Matrix bar code symbology specification*.

ISO/IEC (2006b). *Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification*.

ISO/IEC (2008). *Information technology - Automatic identification and data capture techniques – Aztec Code bar code symbology specification*.

Jakobsson, M., Sako, K., and Impagliazzo, R. (1996). Designated verifier proofs and their applications. In *EUROCRYPT 96, U. Maurer (Ed.), LNCS 1070*. Springer.

Java-Community (2011). Java community process: Contactless communication api. http://www.jcp.org/en/jsr/detail?id=257.

Kieseberg, P., Leithner, M., Mulazzani, M., Munroe, L., Schrittwieser, S., Sinha, M., and Weippl, E. R. (2010). Qr code security. In *Fourth International Workshop on Trustworthy Ubiquitous Computing (TwUC 2010)*. ACM.

Laur, S. and Nyberg, K. (2006). Efficient mutual data authentication using mutually authenticated strings. In *Cryptology and Network Security (CANS), LNCS 4301*. Springer.

McCune, J., Perrig, A., and Reiter, M. K. (2009). Seeing-is-believing: using camera phones for human-verifiable authentication. *Int. J. Security and Networks*, 4(1-2).

Menezes, A., van Oorschot, P., and Vanstone, S. (1997). *Handbook of Applied Cryptography*. CRC Press.

Schnorr, C. P. (1990). Efficient identification and signatures for smart cards. In *Advances in Cryptology - CRYPTO 89, G. Brassard (Ed.), LNCS 435*. Springer.

Schnorr, C. P. (1991). Efficient signature generation by smart cards. *Journal of Cryptology*, 4.

Sun-Oracle (2010). Java platform standard edition 6 release. http://www.oracle.com/technetwork/java/javase/overview/index-jsp-136246.html.

Tyley, R. (2011). Spongycastle crypto apis. https://github.com/rtyley/spongycastle/.

ZXing-Community (2011). Zxing - open-source, multi-format 1d/2d barcode image processing library. http://code.google.com/p/zxing/.