Secure Physical Access Control with Strong Cryptographic Protection

Jan Hajny, Petr Dzurenda and Lukas Malina
Department of Telecommunications, Brno University of Technology, Technicka 12, Brno, Czech Republic

Keywords: Authentication, Identification, Security, Proofs Of Knowledge, Physical Access Control, Cryptography.

Abstract: This paper is focused on the area of physical access control systems (PACs), particularly on the systems for building access control. We show how the application of modern cryptographic protocols, namely the cryptographic proofs of knowledge, can improve the security and privacy protection in practical access control systems. We propose a novel scheme SPAC (Secure Physical Access Control) based on modern cryptographic primitives. By employing the proofs of knowledge, the authentication process gets more secure and privacy friendly in comparison to existing schemes without negative influence on the implementation complexity or system performance. In this paper, we describe the weaknesses of existing schemes, show the full cryptographic specification of the novel SPAC scheme including its security proofs and provide benchmarks on off-the-shelf devices used in real commercial systems. Furthermore we show, that the transition from an old insecure system to strong authentication can be easy and cost-effective.

1 INTRODUCTION

We use physical access control systems (PACs) many times a day. We open parking lot gates, office complex doors or operate elevators by attaching our chipcards or RFID (Radio Frequency Identification) tags to electronic readers. Using PACs, the identification and authentication process is easy and fast. The chipcard just transmits the identifier stored in its memory to the reader. In more advanced systems, a cryptographic authentication protocol is additionally implemented to avoid the eavesdropping of identifiers by attackers.

Although the PAC systems are often used in very security-sensitive areas, such as power plants, large industrial sites, banks or critical communication infrastructures, their security is often very low. There are many known attacks exploiting improper cryptographic protocol design or wrong implementation of PAC systems.

In this paper, we describe the general PAC architecture (Section 2), show the weaknesses of popular existing systems (Section 3), identify the features necessary for next-generation PACs (Section 3.1) and propose a novel cryptographic scheme based on modern cryptographic primitives that remove most of the weaknesses and provides additional features for privacy protection (Section 4). To prove the practical readiness for implementation, we show the performance benchmarks of primitives used in our scheme on devices used in real, commercial PAC systems (Section 6). Finally, we show that the new scheme based on strong cryptography is easily implementable into existing applications (Section 6.1).

2 GENERAL ARCHITECTURE OF PHYSICAL ACCESS CONTROL SYSTEMS

The general architecture of physical access control systems is described in this section. Regardless the manufacturer, the existing commercial systems usually respect the architecture shown in Fig. 1.

Although some modifications might occur in concrete implementations, we will assume this architecture because it sufficiently reflects the most of existing systems and perfectly reflects the implementation we are aiming at with our scheme. We provide the list of key PAC devices and describe their roles in the system here.

- **User Device:** a smartcard or a smartphone used by a user for authentication. The User Device transmits the identifier or executes the authentication protocol with a reader via RFID interface.

Hajny J., Dzurenda P. and Malina L..
Secure Physical Access Control with Strong Cryptographic Protection.
DOI: 10.5220/0005524202200227
In *Proceedings of the 12th International Conference on Security and Cryptography* (SECRYPT-2015), pages 220-227
ISBN: 978-989-758-117-5

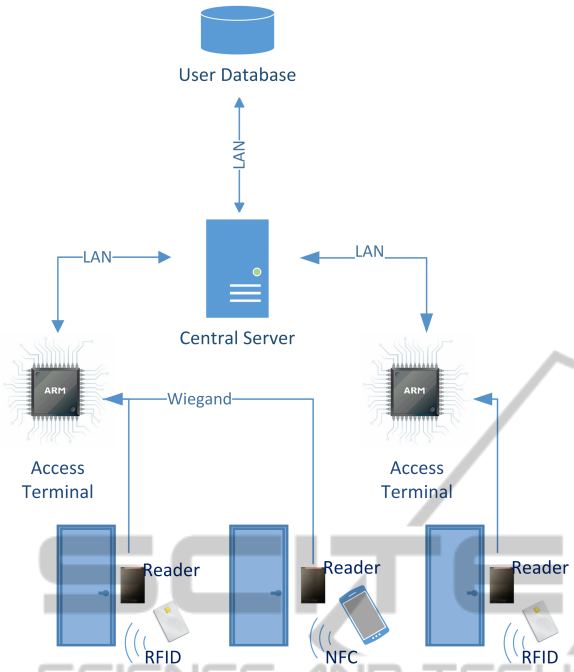

Figure 1: General architecture of physical access control systems.

The expected range is upto 5 centimeters.

- **Reader:** a simple device receiving the identifier or authentication data. In trivial implementations, the reader just re-sends data received from RFID interface to Access Terminal via a one-way Wiegand three-wire interface. In more complex implementations, the Reader is able to verify the cryptographic data received and re-send user's identifier only in case he provided correct proofs. In this case, the Reader is often equipped by SAM (Secure Access Module). Using SAM, the verification of cryptographic data might be faster and more secure due to the use of special cryptographic co-processors. The SAM might be realized by a programmable smartcard such as JavaCard (Oracle, 2015) or MultOS card (MultOS, 2015).
- **Access Terminal:** a device connected to Readers by one-way Wiegand interface and to Central Servers by two-way interface (possibly LAN). The Access Terminal maintains a list of identifiers of authorized users. This list gets updated from Central Servers. After receiving an identifier from the Reader, the Terminal checks its list and decides about the authorization.
- **Central Server:** the Central Server is the central point of administration. Using Central Server, it is possible to add, remove, block and manage all users. The information necessary for authentication (user identifiers, their keys) are distributed to other devices from here.
- **User Database:** all user information is stored in a central database directly connected to the Central Server.

The weakest point of the above described architecture from the security perspective is the communication between the User Device and the Reader. That is given by its wireless nature and the fact that it is not protected physically (like the other interfaces might be, for example by running wires inside walls or private areas of the building). Therefore, most cryptographic protocols are aiming on securing the communication between users' smartcards and SAMs inside Readers. We provide a short overview of existing solutions in the next section.

3 STATE OF THE ART IN PACS

We analyze the existing solutions used for physical access control in this section. We've chosen the most spread technologies. Although precise data regarding the market shares are not available, the technologies and manufacturers included represent the majority of all systems deployed. We included NXP's Mifare and DESfire; HID's Prox and iClass; and Legic Prime and Advant.

Mifare Classic

NXP's Mifare Classic introduced in 1994 is the most popular technology used in physical access control systems. Although very old and insecure, the technology is still used in many applications, even those security sensitive. The authentication protocol is based on a unique 4B card identifier UID. In some implementations, the card just reveals UID to the Reader without any authentication protocol. In that case, UID can be easily eavesdropped and used by an attacker for impersonation. In other implementations, an authentication protocol depicted in Fig. 2 is used. The protocol is considered insecure due to many existing practical attacks (Courtois et al., 2008; Courtois, 2009; Garcia et al., 2009) on the encryption algorithm CRYPTO1.

Mifare DESfire

The insufficient security of the CRYPTO1 algorithm used in the Mifare Classic made NXP improve the cryptographic protection and release Mifare DESFire. The old encryption algorithm was replaced by 3DES algorithm. The authentication protocol is depicted

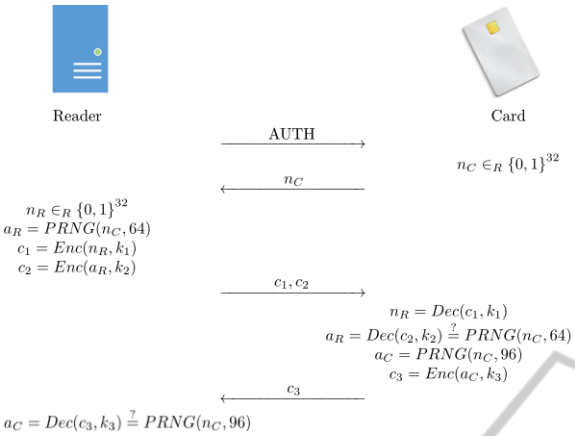

Figure 2: Mifare authentication protocol.

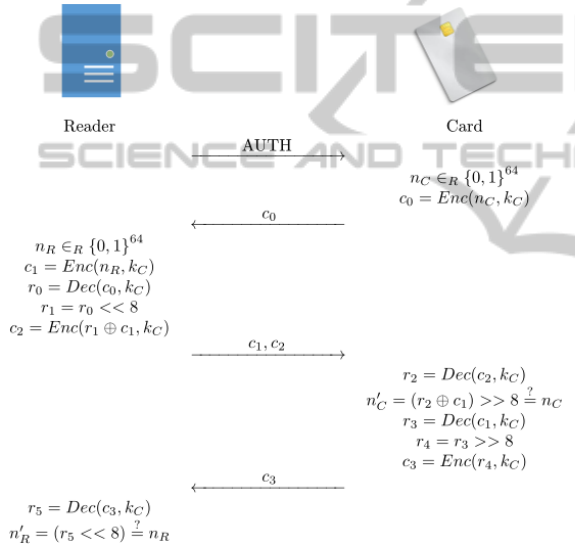

Figure 3: DESfire authentication protocol.

in Fig. 3. The authentication protocol was further improved in Mifare DESFire EV1 which supports the AES encryption algorithm (NIST, 2001). The protocol itself remained without major changes. However, even Mifare DESFire was successfully attacked, although the attacks (Oswald and Paar, 2011; Markantonakis, 2012) were aimed on implementation, not cryptographic weaknesses.

HID Prox and HID iClass

The HID Prox technology contains no cryptographic protection. HID iClass employs an authentication protocol based on the 3DES algorithm. The protocol is depicted in Fig. 4. There are attacks on this protocol available (Meriac, 2010).

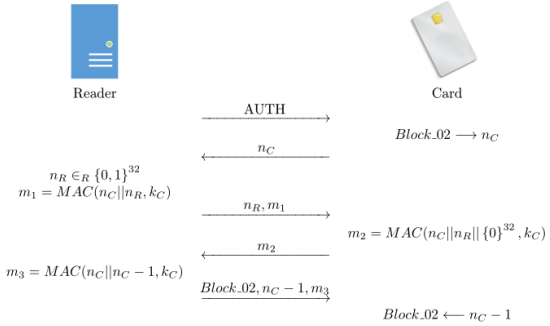

Figure 4: HID iClass authentication protocol.

Legic Prime and Legic Advant

Legic Prime has weak proprietary cryptographic protection (SRLabs, 2015). Legic Advant is protected by symmetric block algorithms (DES (NIST, 1999), 3DES, AES).

Based on the analysis of existing technologies above, we conclude that the majority of existing physical access control systems used in practice is based either on missing/insufficient cryptographic protection (Mifare Classic, HID Prox, Legic Prime) or obsolete algorithms (DES, 3DES). The rest of analyzed systems is based on AES, originally a block cipher designed for data encryption. None of the analyzed solutions is using modern authentication protocols based on the provable security concept allowing formal proofs, such as zero-knowledge protocols, Σ-protocols (Cramer, 1997) or proofs of knowledge. Privacy-enhancing features, such as the unlinkability of verification sessions, are not considered. All schemes implemented are symmetric, thus employing shared keys. That makes non-repudiation difficult in the systems. Furthermore, the symmetric keys must be present at both User Devices and Readers. That makes the risk of their extraction higher since Readers can be captured and analyzed by attackers.

3.1 Our Contribution

We specify a novel cryptographic scheme addressing the above identified weaknesses in the next sections. In particular, the SPAC scheme is designed to provide the following features.

- **Provable Security:** our SPAC scheme is based on cryptographic primitives with provable security, particularly on the interactive Schnorr-based proof of knowledge of discrete logarithm (Schnorr, 1991) in the RSA group (Rivest et al., 1978). We provide the full security analysis in Section 5.

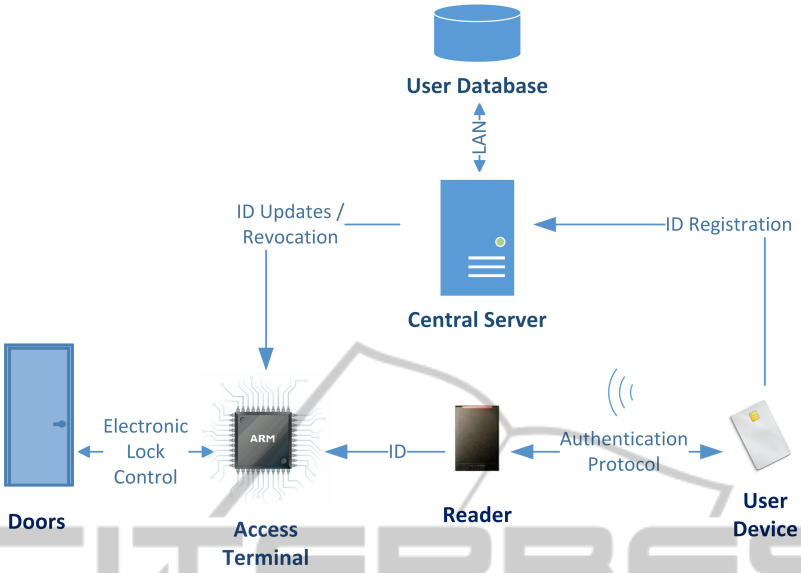

Figure 5: SPAC architecture.

- **Privacy Protection Features:** the SPAC scheme provides features for the protection of users' privacy and digital identity, namely user ID hiding and authentication sessions unlinkability.
- **Non-repudiation:** the SPAC scheme is based on asymmetric cryptography. The user is proving his identity using a private cryptographic key known only by him, not by anyone else. Therefore, users cannot repudiate their past transactions like in symmetric schemes.
- **Local Key Storage:** unlike existing schemes, the user authentication key is stored in user's device only, not in the verifier's device. That makes the key protection much easier.
- **Computational and Communication Efficiency:** the SPAC scheme is computationally efficient enough to run on low-resource devices as smartcards and SAMs. The communication is realized by a simple 3-way protocol. We prove the computational and communication efficiency by describing the implementation results in section 6.

4 SPAC CRYPTOGRAPHIC SPECIFICATION

4.1 Notation

For various proofs of knowledge or representation, we use the efficient CS notation introduced by Camenisch and Stadler (Camenisch and Stadler, 1997). The protocol for proving the knowledge of discrete logarithm of c with respect to g is denoted as $PK\{\alpha : c = g^{\alpha}\}$. The symbol ":" means "such that", "|" means "divides", "$|x|$" is the bitlength of x and "$x \in_R \{0,1\}^l$" is a randomly chosen bitstring of maximum length l. G is a set of the respective group generators. All operations are in $\mathbb{Z}_n^*$, where $n = rs$ and r, s are random large safe primes over 512 bits long. $\mathcal{H}$ denotes a hash function.

4.2 Scheme Overview

The SPAC scheme complies with the general concept of PAC systems described in section 2. The SPAC scheme is depicted in Fig. 5. The scheme is composed of following protocols.

1. `Setup`: all cryptographic parameters are generated and distributed to participating devices here.
2. `Registration`: User Device generates its private key, computes his public identifier and registers this identifier at the Central Server here.
3. `Authentication`: User Device proves the ownership of his private key corresponding to a registered identifier to the Reader, Reader forwards the identifier to the Terminal that decides about granting/rejecting the access.
4. `Revocation`: in case some user needs to be revoked, the whitelist of valid identifiers is updated at Terminals by the Central Server.

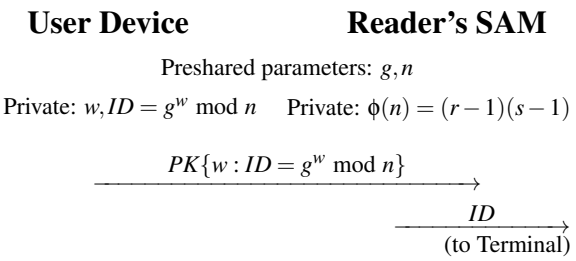

Figure 6: `Authentication` protocol in CS notation.

`Setup` Protocol
The public parameters of $\mathbb{Z}_n^*$ group (modulus n, generator g) and general parameters (k - user key size, l - challenge size, m - protocol error rate) are generated and distributed by the Central Server to all participating devices. Private parameters (the factorization of n) are distributed to SAM modules of Readers. This phase can be realized by existing means of cryptography or by simply pre-sharing all values in software, thus we don't cover it in more details.

`Registration` Protocol
User Device generates a random private key $w \in_R \{0,1\}^k$ and computes the identifier $ID = g^w \bmod n$. By means of traditional cryptography or personal interaction, the ID is registered at Central Server and distributed to Terminals.

`Authentication` Protocol
The authentication protocol is the key part of the scheme. It runs between the User Device and the SAM module of the Reader. The Reader sends user's ID to the Terminal after the protocol is successfully finished. If the ID is valid (present on Terminal's whitelist), the access is granted by, e.g., unlocking an electronic door lock.

The authentication protocol is realized by a standard interactive proof of knowledge of discrete logarithm protocol (Camenisch and Stadler, 1997) based on Schnorr protocol (Schnorr, 1991). The authentication protocol is depicted in CS notation in Fig. 6. Using this protocol, the User Device provides a cryptographic proof that he knows a private key w to its unique identifier ID. If the proof is constructed correctly, the Reader's SAM learns ID and is convinced that the User Device knows corresponding w. However, w is never disclosed.

The proof of knowledge can be practically implemented as a Σ-protocol (Cramer, 1997) depicted in full notation in Fig. 7. An interactive version using a random challenge e generated by the Reader, instead of a non-interactive version based on Fiat-

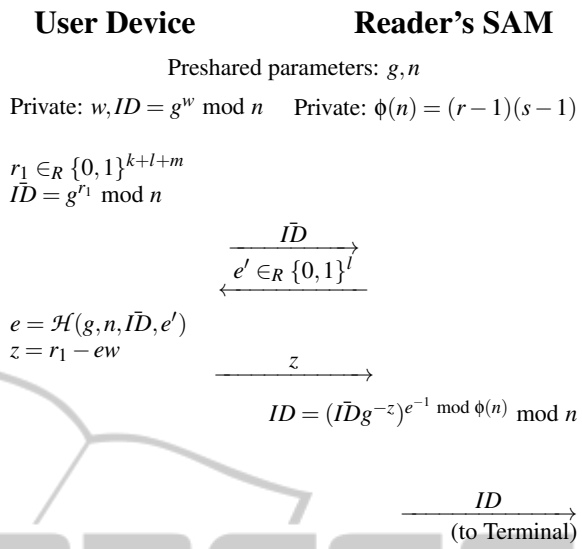

Figure 7: `Authentication` protocol in full notation.

Shamir heuristic (Fiat and Shamir, 1987), is chosen due to Reader's hardware limitations (no clocks, limited memory). First, the User Device chooses a randomization value r_1 and computes corresponding commitment $I\bar{D}$. $I\bar{D}$ is sent to the SAM. The SAM module responds with a randomly generated challenge e'. The User Device closes communication by sending the response $z = r_1 - ew$. In the response, the private key is perfectly hidden by r_1 and e. From the knowledge of $\phi(n)$ and the answer z, the SAM module is able to compute ID and send it to the terminal that decides about its validity.

5 SPAC SECURITY ANALYSIS

The authentication protocol used in our scheme is based on standard cryptographic constructions, therefore the security proof follows the standard approach for proofs of knowledge systems (Rosen, 2006). We prove completeness, soundness and semi-honest verifier zero-knowledge of the protocol.

Completeness
Completeness property proves that honest, valid users who know private key w are almost[1] always accepted by the protocol. We prove the completeness from the authentication verification equation:

[1]Except low probability given by the protocol error rate $P_{Err} = 2^{-m}$

$$ID \equiv (I\bar{D}g^{-z})^{e^{-1} \bmod \phi(n)} \equiv (g^r g^{-(r-ew)})^{e^{-1} \bmod \phi(n)} \equiv$$
$$\equiv (g^{r+ew-r})^{e^{-1} \bmod \phi(n)} \equiv g^{ewe^{-1} \bmod \phi(n)}$$
$$\equiv ID \pmod{n} \tag{1}$$

Soundness
Soundness property proves that dishonest, invalid user who does not know the private key w, is accepted by the protocol with negligible probability. We prove soundness by proving that a user who does not know w is able to correctly answer at most 1 challenge e.

We prove soundness by following the proof described in (Camenisch and Shoup, 2003).

Theorem 1. *Under the assumptions that factoring of n is hard and $\log_g ID$ is unknown, given a modulus n, along with elements g, ID, it is hard to compute integers a, b such that*

$$1 \equiv g^a ID^b \bmod n \text{ and } (a \neq 0 \text{ or } b \neq 0). \tag{2}$$

Proof. Suppose there is an algorithm $\mathcal{A}$ that inputs n, g, ID and outputs a, b valid in (2). Then we can use $\mathcal{A}$ to either factor n or compute $\log_g ID$, both violating assumptions. The output (a, b) satisfies $1 \equiv g^a ID^b \equiv g^a g^{\alpha b} \equiv g^{a+\alpha b} \bmod n$, therefore $a + \alpha b \equiv 0 \bmod ord(g)$. We have two cases:

Case 1. Let us consider $a + \alpha b = 0$. Then discrete logarithm $\log_g ID$ can be efficiently computed as $\log_g ID = \alpha = -\frac{a}{b}$. Case 1 violates the discrete logarithm assumption.

Case 2. Let us consider $a + \alpha b \neq 0$. $\mathcal{A}$ can be used to factor n by choosing α in random, inputting (n, g, g^α) and getting the output (a, b). Using $(a + \alpha b)$, which is a non-zero multiple of $\phi(n)$, the adversary can factor n. To efficiently compute a proper factor of n, the adversary can use the technique originally developed for RSA (Boneh, 1999). Case 2 violates the factorization assumption. □

Using the Theorem 1, we can prove soundness like in (Camenisch and Shoup, 2003), thus by constructing the knowledge extractor and assuming that the factorization of n is hard. The extractor uses the standard rewinding technique, thus inputs two different valid answers z, z' on two different challenges e, e' with the fixed first step $I\bar{D}$. The verification equation must hold for both answers:

$$ID \equiv (I\bar{D}g^{-z})^{e^{-1} \bmod \phi(n)}$$
$$ID \equiv (I\bar{D}g^{-z'})^{e'^{-1} \bmod \phi(n)}$$

therefore

$$ID^e \equiv (I\bar{D}g^{-z})$$
$$ID^{e'} \equiv (I\bar{D}g^{-z'})$$

Now we divide the equations and get:

$$ID^{e-e'} \equiv g^{-z+z'}$$

That equation can be transformed into:

$$ID \equiv g^{(-z+z')/(e-e')}$$

From the Theorem 1, the User must have used $\log_g ID$, since the factorization of n is unknown. Based on the Case 1 of Theorem 1, $(e - e')$ divides $(-z + z')$, therefore the extractor can extract $w = \frac{-z+z'}{e-e'}$. We reached the contradiction to the original assumption.

Honest Verifier Zero-Knowledge
Zero-knowledge property guarantees that no information about private w leaks from the protocol. That is proven by the existence of a simulator M that is able to simulate all the protocol values indistinguishably from real protocol values without the knowledge of w. Therefore, if all protocol values can be simulated without the knowledge of w, they do not release any information about w. The protocol simulator is constructed in the standard way (Damgård, 2000) and works in following steps:

1. M randomly generates an answer z' from the interval $< 2^{klm} - 2^{kl}, 2^{klm} >$,
2. M randomly generates a challenge $e'' \in_R \{0,1\}^l$,
3. M computes the first message of the protocol $I\bar{D}' = ID^{e''} g^{z'} \bmod n$.

The simulated values $I\bar{D}', e'', z'$ are then computationally indistinguishable from the real run of the protocol.

6 IMPLEMENTATION

Many existing cryptographic schemes remain only theoretical because their computational complexity is too high for practical implementations on restricted devices like smart cards and card readers equipped by only very limited hardware. Therefore, we provide detailed analysis and benchmarks of the primitives used in our scheme on real, off-the-shelf hardware commonly used in PAC systems. We focus on operations inside Reader's SAM module because that is the weakest point of the systems. Other devices are more computationally powerful (Central Server - standard server, Terminal - ARM-based system, User Device - NFC-enabled smartphone or smartcard with cryptographic co-processor).

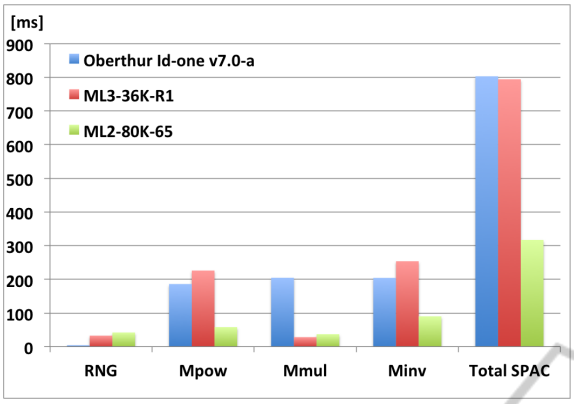

Figure 8: Benchmarks of SAM modules.

From the cryptographic specification of the authentication protocol in Section 4, the primitive operations are easily identifiable. For the authentication protocol implementation, the SAM module must be able to efficiently compute:

- 1 random number generation (RNG),
- 1 modular inversion (Minv),
- 2 modular exponentiation (Mpow),
- 1 modular multiplication (Mmul).

We measured the time necessary for the computation of the above specified operations in a modular group with 1024 b modulus and parameters $k = 160$, $l = 160$, $m = 80$. The smartcards Oberthur Id-one v7.0-a (JavaCard), ML3-36K-R1 (MultOS) and ML2-80K-65 (MultOS) were used. These cards are commonly used as SAM modules. The results of individual operations and the total verification time are shown in Fig. 8. We present the arithmetic mean of 25 measurements.

The total time needed for the user verification varies according to the module used from 300 to 800 ms. The best choice for our protocol is the MultOS ML2 card (marked green in the graph). Further speed-up is possible by using pre-computation techniques and better implementation of some operations (e.g., the modular inverse operation is currently implemented naively using modular exponentiation). However, we consider even the times measured fast enough for practical implementations.

The implementation on the User Device is easier. If smartphones are used, no resource restrictions apply due to enormous computational power of modern devices. In case smartcards are required, the MultOS ML2-80K-65 is again the best choice for implementation. All operations of the cryptographic proof require 167 ms on this device.

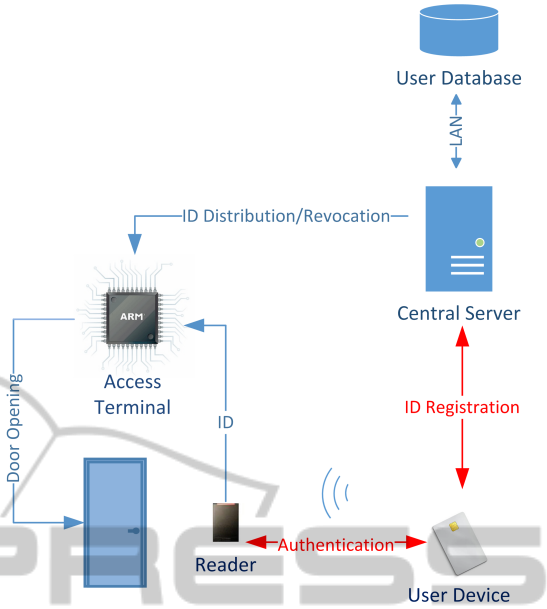

Figure 9: Compatibility of SPAC with existing PAC systems.

Based on the times measured, it is possible to compute the total time of the protocol, including computations on the user's smartcard and the SAM module. The total time excluding communication is 484 ms without optimization. This value proves the practical implementability of the scheme since the industry limit is considered 0.5 s for physical access control protocols.

6.1 Compatibility with Existing Applications

Novel systems are sometimes difficult to practically implement into existing systems. Usually, it is difficult to completely replace all devices in existing systems. The SPAC scheme proposed in this paper was created in cooperation with an industrial PAC system manufacturer who required backward compatibility. Therefore, we designed the SPAC scheme in a way that allows re-using of some existing components, namely the Central Server, User Database and the Terminals. Only the Reader and the User Device must be re-designed because a new authentication protocol is used. The components and protocols that need new implementation are marked red in Fig. 9.

7 CONCLUSIONS

The paper contains cryptographic specification of a

novel physical access control scheme called SPAC. In comparison to existing schemes, SPAC is based on modern asymmetric primitives, that provide features currently unavailable in existing systems, such as provable security, local key storage, non-repudiation and authentication session randomization. Although the scheme is based on asymmetric cryptography, it is very fast even when implemented on resource-limited devices. In the section focused on implementation, we show, that the verification part realized on a smartcard SAM module takes around 300 ms and the proving part implemented on user's smartcard takes around 170 ms. Moreover, we expect a significant speed-up in the optimized version of our implementation, which is our next step.

ACKNOWLEDGEMENTS

Research described in this paper was financed by the National Sustainability Program under grant LO1401, Technology Agency of the Czech Republic project TA04010476 "Secure Systems for Electronic Services User Verification" and by the Czech Science Foundation under grant no. 14-25298P. For the research, infrastructure of the SIX Center was used.

REFERENCES

Boneh, D. (1999). Twenty years of attacks on the rsa cryptosystem. *NOTICES OF THE AMS*, 46:203–213.

Camenisch, J. and Shoup, V. (2003). Practical verifiable encryption and decryption of discrete logarithms. In *Advances in Cryptology - CRYPTO 2003*, pages 126–144. Springer-Verlag.

Camenisch, J. and Stadler, M. (1997). Proof systems for general statements about discrete logarithms.

Courtois, N., Nohl, K., and O'Neil, S. (2008). Algebraic attacks on the crypto-1 stream cipher in mifare classic and oyster cards. *IACR Cryptology ePrint Archive*.

Courtois, N. T. (2009). The dark side of security by obscurity and cloning mifare classic rail and building passes, anywhere, anytime.

Cramer, R. (1997). *Modular Design of Secure Yet Practical Cryptographic Protocols*.

Damgård, I. (2000). Efficient concurrent zero-knowledge in the auxiliary string model. In Preneel, B., editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 418–430. Springer Berlin Heidelberg.

Fiat, A. and Shamir, A. (1987). How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO 86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Berlin / Heidelberg.

Garcia, F. D., van Rossum, P., Verdult, R., and Schreur, R. W. (2009). Wirelessly pickpocketing a mifare classic card. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 3–15. IEEE.

Markantonakis, K. (2012). Practical relay attack on contactless transactions by using nfc mobile phones. *Radio Frequency Identification System Security: RFIDsec*.

Meriac, M. (2010). Heart of darkness-exploring the uncharted backwaters of hid iclasstm security. *Heart*.

MultOS (2015). Multos webpage. "http://www.multos.com".

NIST (1999). Federal information processing standards publication (FIPS 46-3). Data Encryption Standard (DES). -.

NIST (2001). Federal information processing standards publication (FIPS 197). Advanced Encryption Standard (AES). -.

Oracle (2015). Java card webpage. "http://www.oracle.com/technetwork/java/embedded/javacard/overview/index.html".

Oswald, D. and Paar, C. (2011). Breaking mifare desfire mf3icd40: Power analysis and templates in the real world. In *Cryptographic Hardware and Embedded Systems–CHES 2011*, pages 207–222. Springer.

Rivest, R., Rivest, R., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *COMMUNICATIONS OF THE ACM*, 21:120–126.

Rosen, A. (2006). *Concurrent Zero-Knowledge With Additional Background by Oded Goldreich*. Springer.

Schnorr, C. P. (1991). Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174.

SRLabs (2015). "https://srlabs.de/analyzing-legic-prime-rfids/".