Establishing Security Best Practices in Access Control

Publication version, v.1.0
Current version available at:
www.srlabs.de/pub/acs

Andreas Rohr — andreas.rohr@rwe.com
Karsten Nohl — nohl@srlabs.de
Henryk Plötz — henryk@srlabs.de

Study supported by

RWE
The energy to lead

SECURITY RESEARCH LABS

Contents

1 Introduction

This study proposes a minimum standard for an access control system built from state-of-the-art components. The focus of the study as detailed in Figure 1 is on a) securely storing information in tokens and readers; b) securing the communication between the tokens and readers.

The solution blueprint proposed in this document is guided by three principles:

- **Openness** of algorithms and interfaces
- **Upgradeability** of security functions and keys
- **Accessibility** of access control data for fraud detection

The proposed solution requires best practice design in three dimensions: First, strong encryption is used with unique cryptographic keys for each card, which only recent contactless chips provide. Second, strong key storage in form of SAM chips is used to protect the critical master keys on the door controller (alternatively, public key cryptography could be used). Lastly, all keys can be updated in response to incidents.

A note on terminology: This study distinguishes between several concepts that are not always clearly separated in the literature on access control system:

A **reader** is a device to read and write RFID cards. Depending on the system, a reader has a medium amount of firmware complexity and may be able to make access decisions on its own, which leads to security problems if the reader is positioned outside of the protected area. Our proposed system does not include any readers.

A **transceiver** is a simple device, sometimes referred to as "active antenna" that communicates with cards with a minimal amount of firmware complexity and no security functionality. The transceiver converts data between radio channel and wired communication channel.

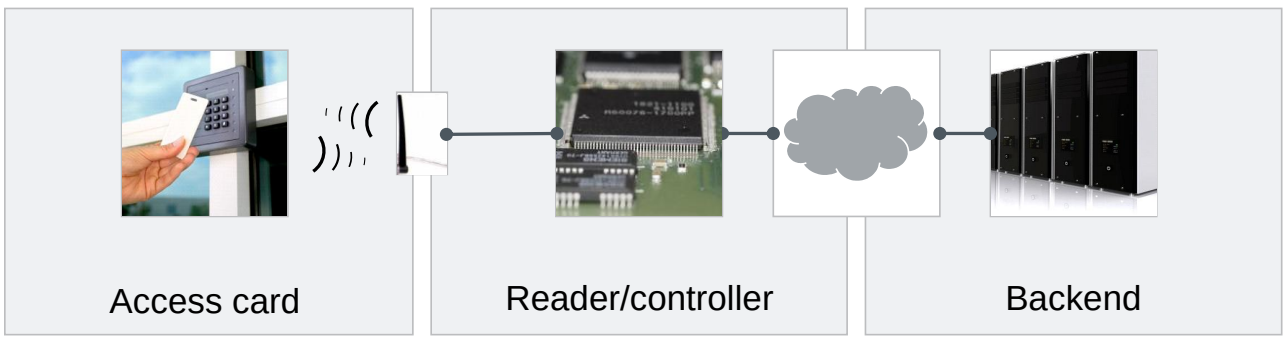


Figure 1. This study focuses on the end-to-end use of secure protocols and key management in an access control system.

A **controller**, sometimes also called “door controller”, is placed inside the protected area and is connected to one or more readers or transceivers and one or more doors. It hosts all the security functionality and makes access decisions, potentially in cooperation with a backend system.

The **backend** system is a centralized place that hosts all data regarding access permissions and may be consulted by online controllers/readers.

The next chapter provides details about current attacks and technical capabilities of RFID cards. Chapters 3 and 4 outline a best practice foundation for a modern access control system and key management system. Chapter 5 provides additional specifications to facilitate a smooth implementation of the best practice ideas. Finally, chapter 6 outlines a typical migration path from existing industry standards to the best practice standard.

2 Technical Background

Contactless access control based on RFID (Radio Frequency Identification) has replaced earlier technologies such as magnetic swipe cards in almost all security-critical applications. Two generations of RFID access cards exist: an earlier generation of cards, including Legic Prime and NXP Mifare Classic which only use basic proprietary security mechanisms, and a modern generation that leverages advances in CMOS and smart card technology to implement state-of-the-art cryptography within the resource limitations of contactless cards. The earlier generation of RFID cards is omnipresent in access control but easily clonable, while the newer cards with more appropriate security enter the market only slowly. This study outlines the requirements for a modern access control scheme and consequently requires the use of strongly encrypting cards.

Before outlining the proposed design, the next three sections discuss attacks on currently used technologies to motivate the need for more secure systems.

2.1 Attacks on Legic Prime

Legic Prime has no innate cryptographic security functions for card or reader authentication [1]. The card relies entirely on the reader to respect write or read restrictions, but the restrictions are not enforced cryptographically or even checked by the card. In most Legic Prime installations the reader relies on the card not being manipulated, e.g., on other readers respecting the write restrictions as well. Clearly, a malicious person with knowledge of the card-reader protocol can ignore all restrictions, thereby circumventing what Legic Prime provides as security.

Since there is no challenge-response protocol, cards are inherently clonable: The memory of a card can be read by a malicious reader and spoofed by a card emulator. The Master Token System Control (MTSC), which is supposed to limit and delegate card creators' abilities, can be circumvented and arbitrary master tokens can be created from scratch, using a malicious reader/writer device.

Legic Prime installations for physical access control typically store a badge number or employee number on the card which is used by the reader, controller or back-end system for access decisions. These installations rely on the promised property of the MTSC that only personalization terminals, which have been authorized by the proper master token, can create and write to a segment with the correct “genetic code”, wherefore a digital signature would be necessary. However, even using a digital signature over the badge number, and even if this

signature is tied to the card's unique identifier, the card could still be completely dumped and emulated.

Mitigations: A few mitigation measures can improve Legic Prime installations, for instance to patch the system until it can be replaced with more secure technology. Each card has a monotonic counter, called the Decremental Field (DCF), that can only be decreased and never be increased (at least on original cards). If this field were to be decreased with each door interaction, an emulated card could be detected by the system based on mismatched counter values. To the best of our knowledge, no current deployment uses this technique.

Some current systems support two-factor authentication through the additional use of PINs, or give the illusion of doing so: In some of these systems the PIN is stored (unencrypted, unprotected) on the card itself, making an attack on both factors trivial.

2.2 Attacks on Mifare Classic

Mifare Classic is the most popular access control card in the world. It uses a proprietary 48-bit stream cipher called Crypto-1 for mutual authentication between card and reader [2]. Data blocks are secured by individual keys with configurable write/read access rights. However, the cipher was demonstrated to be much weaker than expected and the key can be derived from one sniffed authentication procedure with only a few seconds of processing power on a normal laptop computer [3]. On top of the weak cipher, implementation errors exist in the regular Mifare Classic cards which make it possible to discover a key by actively interrogating the card with a specially programmed reader device within seconds to minutes [4]. Once the necessary key has been discovered, a card emulator such as the Proxmark III device [5] can be used to spoof a card to circumvent physical access control readers.

Existing Mifare Classic installations usually either only check the card’s unique identifier -- thereby making a spoofing attack trivial -- or store a badge or employee number in the card's data section, which is susceptible to cloning attacks due to the weak cryptography. After news of the weakness of the cipher surfaced, some vendors added an HMAC or digital RSA signature over the badge number and card UID. This check-sum prevents card clones but does not prevent emulation through devices such as a Proxmark III [5].

Mitigations: Mifare Classic allows a similar attack detection protocol as Legic Prime: A value block can be set to decrement only and used as a monotonic, decreasing counter. If a cryptographic RSA signature of this counter along with the UID is stored on the card and updated on every read, card cloning becomes harder and fraud detection easier. To the best of our knowledge, this countermeasure is not currently used,

often due to the inability of the reading devices to compute RSA signatures.

2.3 Generic Attacks

Besides attacks that exploit specific RFID designs, the technology carries some intrinsic risks that cannot fully be mitigated. Among those risks are relaying attacks and chip hacking.

For relaying, an attacker builds a communication bridge between a valid card and a reader. The card owner will not know that communication happens and does not need to be in vicinity of the reader. The attack is possible even against RFID tags with strong encryption since hardly any of the RFID systems in use today can verify the distance to an RFID tag. Most RFID protocols even provide generous time-out windows which make the attack easier.

All RFID cards in use today can be hacked using intrusive attacks by which secret key material is extracted from the cards. These chip hacking attacks are mitigated through the combination of two strategies:

a) Increase attack cost: Hacking a modern EAL 4+ card typically costs in excess of EUR 50,000. [6].

b) Decrease attack incentives: The attack value is lowered by using unique keys per card through fraud detection, and through second factor authentication for sensitive areas. Once the attack incentives are below the attack costs, the attack is supposed to be mitigated.

3 Access Control Minimum Requirements

The access control market has gone through several evolutionary steps – from metal keys, to magnetic stripe cards, to radio tokens, and to simple proprietary computerized tokens. Currently access control tokens, such as Mifare Classic and Legic Prime, are vulnerable to the attacks shown in Figure 2. The hardening process outlined in the figure should lead to a modern system with attack costs no less than EUR 100,000. These hardening steps provide the base for state-of-the-art security as outlined in the remainder of this study.

Designing access control systems with state-of-the-art microchips is long overdue. This chapter outlines the minimum requirements for de-

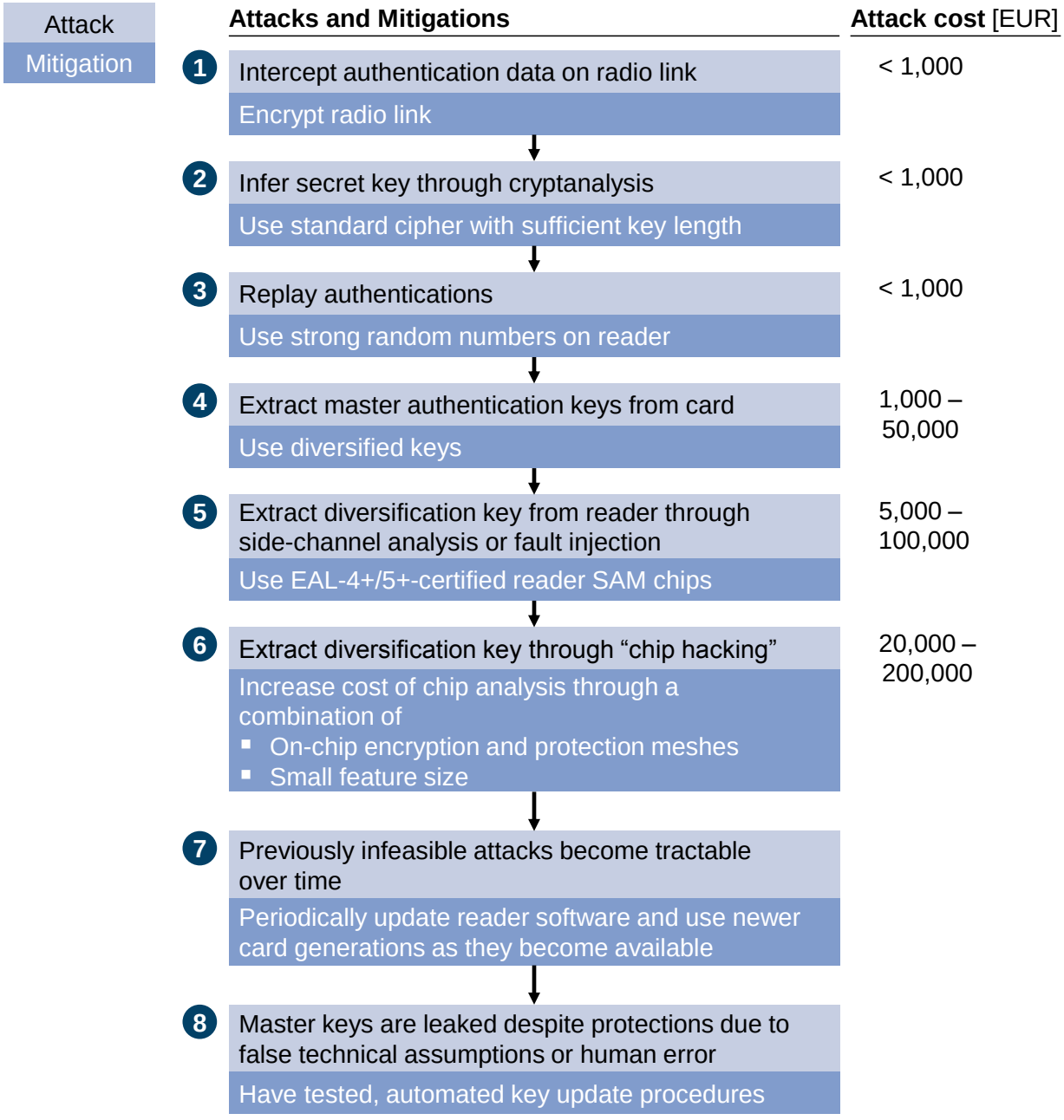


Figure 2. Adapting best practice technology follows the RFID hardening process.

ploying a modern system resistant to the attacks outlined in the figure. The information in this chapter is provided in the form typically found in a Request for Proposal (RfP). No specific brands or technologies are required, but instead the system capabilities on a functional level are laid out in such a way that different technologies and brands can be matched as long as the same security standard is maintained. Institutional readers are encouraged to use parts of this document in their RfPs.

3.1 Air Interface

As motivated in Figure 2, card communication should be mutually authenticated using a standard cipher, keys unique to a card, and strong random numbers. The cards should furthermore be EAL 4+ or 5+ certified to resist physical attacks to a certain extend [7].

Data communication between reader and card (or an attacker) is inherently untrustworthy and must be authenticated. This requirement not only applies to the air interface between card and reader, but to all data that originates outside of the protected area (see next section on reader installation). The cryptographic protections that apply to the air interface also apply here.

All cryptographic operations must be executed with published and well-researched algorithms [8]. The use of proprietary or non-published algorithms would contradict the stated goal of openness, and is also likely to jeopardize the security properties (since proprietary encryption algorithms have been shown to likely be weak). Similar considerations apply to the key derivation procedures that generate card specific keys: A published algorithm based on a secure cipher should be used.

Furthermore, to guarantee freshness and prevent replay attacks, both reader and card must generate and use cryptographically strong random numbers during the mutual authentication procedure.

Thus, the minimum requirements for the air interface are: mutual authentication and message authentication between card and controller with a secure cryptographic algorithm, based on strong random numbers and a securely diversified card specific key. Session encryption between card and controller is optional but desirable from a privacy perspective. The only reason we can think of not to use encryption could be a restriction resulting from export regulations. If encryption is used, similarly a secure algorithm must be used.

3.2 Reader/Controller

The system architecture is based on distributed intelligence utilizing a standard IP communications network. If the communication between

backend and the controllers fails, the individual controllers run in autonomous (offline) mode.

In order to minimize the risk of a key disclosure by extracting the keys from a SAM chip of a stolen reader, the "visible" transceiver should be separated from the actual controller, where the keys preferably live in a SAM chip. Since the transceiver only works as a transparent relay from the air interface to the controller, one can implement all functionality out of an attacker's reach.

The door controller has to be located within the protected area. The same applies to the cabling, where not-cryptographically-signed protocols are used. Where transceivers for different security zones are connected to one controller, the controller has to be placed in the highest of these zones. Any network based communication between the backend and the uplink of the controller has to be mutually authenticated and encrypted (see also Section 4.2 on authentication protocols).

The controller has to be kept patchable to fix vulnerabilities and enhance functionality in the future. The communication between transceiver and controller should work on a minimum distance of 100m, utilizing a physical protocol such as RS-232 or—preferably—an Open Supervised Device Protocol-based (OSDP) RS-485 protocol such as HADP. This protocol should be documented and standardized to have the free choice of transceivers. At best the source code is made available.

3.3 Data APIs

A central element of risk mitigation in access control is the constant monitoring of irregularities. These irregularities can be caused by an attacker who successfully circumvented technical protection measures, or by a thief who stole an access token but is behaving differently from the token's owner. Intrusion detection systems provide the capability to detect such irregularities. The basic design pattern for effective intrusion detection is the central collection of all data generated in an access attempt as shown in Figure 3.

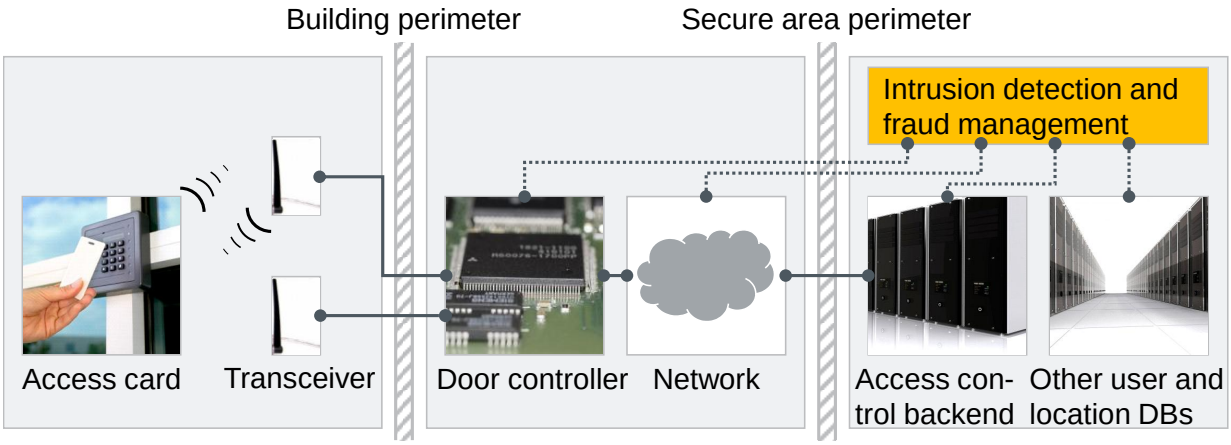


Figure 3. The fraud and intrusion management system collects all data that could indicate irregular usage, and executes real-time checks on the consistency of the data with past events.

At a minimum, the following data is made available to the intrusion monitoring system:

- A log of all access attempts, specifying at least: card ID, transceiver ID, time (time zone), result (i.e., door opens), card counter value (where applicable), and round-trip delay. This data should be provided in a "common log format"
- Fraud management returns a status code on every access attempt data item: OK, INFORM OPERATOR, DENY ENTRY
- GPS locations for all transceiver IDs; where available: building topology indicating at least which transceivers are outside
- Key generation per controller and card
- Work schedules and holiday schedules for locations (+time zones)
- User data; for example: assigned card ID(s), hiring date, type of contract (internal, temporarily internal, external)
- Card revocation list, including revocation date and reason
- Backend server statistics: server load, free disk space, access attempts processed

To protect employee's interest against the ability of a company to track their "movements" as a performance index, the data have to be parsed anonymously. This can be achieved by replacing the user reference with a unique number or another (random) identifier. In the case of an alert it should be possible to revert the underlying bidirectional mapping. Abuse of the data must be prevented by processes enforcing a four-eyes-principle and proper logging.

The network communication between the controller and the backend system should be mutually authenticated and encrypted.

3.4 Multi-Factor Authentication

To gain access to an area with a high level of sensitivity, such as a computing center or a command and control station, it should not be suffi-

cient to possess a (stolen) access card, but multiple factor authentication should be required. For such applications biometric characteristics or additional knowledge, each associated with the card owner, are widely used.

If there is no need for having the controller being capable of working offline (with respect to the backend online system), the second factor such as a PIN or a biometric attribute should be stored in the backend and not on the card. If an additional attribute is stored on the card, it should be cryptographically signed.

In some scenarios, human interaction should provide the second factor: An operator verifies the card's associated owner picture with the CCTV camera in order to manually unlock the door.

Another measure to mitigate the risk of unauthorized access with cloned or stolen cards is an access card swap in front of an area with a higher level of security. This is typically implemented in such a way that a security guard keeps all access cards to that area. In order to get one of the cards one has to provide a legitimate ID document, which is registered and tied (by listing with timestamps) to a specific access card that is handed over in exchange for the ID card. Thus, the access cards for the security area never leave the premises. This procedure might be accompanied by an enforced return policy on leave.

4 Access Control Key Management

All functions in an access control scheme should be authorized through secret keys. Keys are needed on the access cards, on most door controllers, in the backend system, and in the card processing facility. Keys more vulnerable to attacks such as those on access cards should be derived from better protected keys, for example in the card processing facility and the backend system. Strong authentication protocols and key storage protect the secret keys from attacks.

4.1 Authentication Schemes

Access cards are authenticated to readers and to the backend system using strong cryptography. All modern cards support symmetric cryptography such as 3DES or AES, while some higher-grade cards already support asymmetric cryptography such as RSA. Where asymmetric encryption is used, no valuable master keys need to be stored in the door controller, which makes the resulting design and maintenance less complex since SAM chips are not necessary.

The two resulting design options detailed in

Figure 4 are using more expensive cards and cheaper controllers with RSA encryption, or cheaper cards with more complex controllers and AES or 3DES encryption. Unlike in micro-payment, where asymmetric encryption is found in national applications [9], no comparable solutions exist for access control yet. Therefore, this study details an

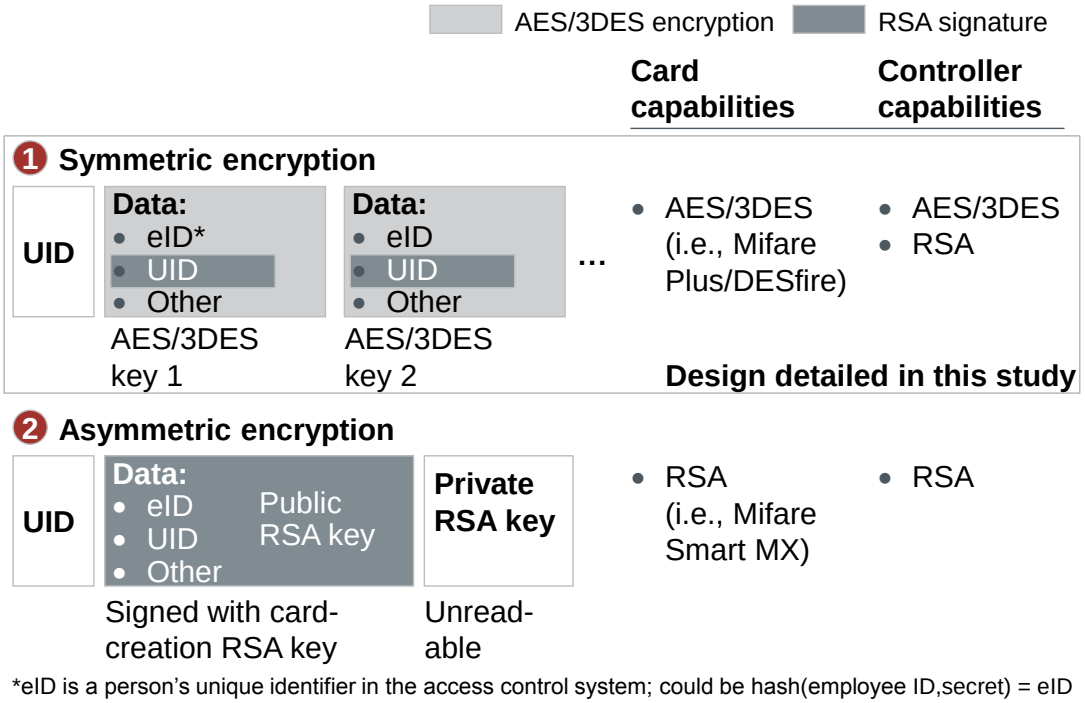


Figure 4. Two options exist for integrating access cards into a PKI-infrastructure. Depending on the card capabilities, asymmetric RSA encryption is used actively by the card, or passively by storing RSA-signed data on the card.

AES/3DES-based approach.

Even when lower-grade cards with no embedded RSA-functionality are used, a link should be established to the corporate PKI using RSA signatures stored on the card. These signatures bind the UID of a card to the person that card is assigned to (e.g., employee's identifier, eID). By doing so, access credentials cannot easily be copied to another card. Instead an emulator platform has to be used, which is more difficult and more detectable due to differing timing behavior.

4.2 Authentication Protocol

The security protocol is executed between the card, a transceiver that converts radio signals to digital communication, a door controller that authenticates the card, and a backend system that manages access rights and provides intrusion detection functionality. In some scenarios, the transceiver and the door controller can be combined into one device as long as that device is not accessible from outside the secured area. Separating the two components as recommended in this study eases implementation, upgradeability and exchangeability of components.

A secure channel is assumed to exist between door controller and backend system at all times (illustrated in Figure 5 as **step 0**). This channel can either be long-lasting and persistent (preferred), or be established transparently when needed. The channel should employ a secure and well-analyzed protocol such as TLS and must provide mutual authentication between controller and backend, message integrity/authentication and replay protection. Confidentiality is optional but desirable for privacy protection.

In **step 1** the transceiver performs an anticollision and card activation procedure for the underlying radio protocol (ISO 14443-3 or -4, type A or B).

In **step 2** the transceiver informs the controller of the presence and activation of a new card in the field, including its UID.

In **step 3** the controller, using the transceiver as a bridge and protocol converter, activates the access control application on the card and performs preparation steps as necessary.

In **step 4** controller and card engage in a mutual authentication protocol with session key derivation for message authentication. The secret for this authentication must not leave the controller security module (SAM chip). A well analyzed and secure challenge-response scheme should be used. All messages between controller and card after this step must be protected by a message authentication code (MAC).

Step 4 can also be used to set up a session key to encrypt the remainder of the session. While such encryption is preferable from a privacy standpoint, it is not strictly necessary for security purposes as long as the MAC is secure. Encrypting a session may lead to export restrictions.

There is a design option for scenarios where the door controller is online; all steps from step 4 on may terminate at the backend system instead of the controller. A hybrid mode is also possible, where only step 4 is performed by the backend, that transmits the negotiated session key to the controller, which then executes the remainder of the protocol as depicted.

In **step 5** the controller reads the signed data block from the card. The card must only allow that to happen after a/the successful authentication of the controller. If a secure channel has been established in step 4, this data block will also be encrypted. In any case the data is securely linked to the active session by the MAC.

In **step 6** the controller verifies the signature of the block against its

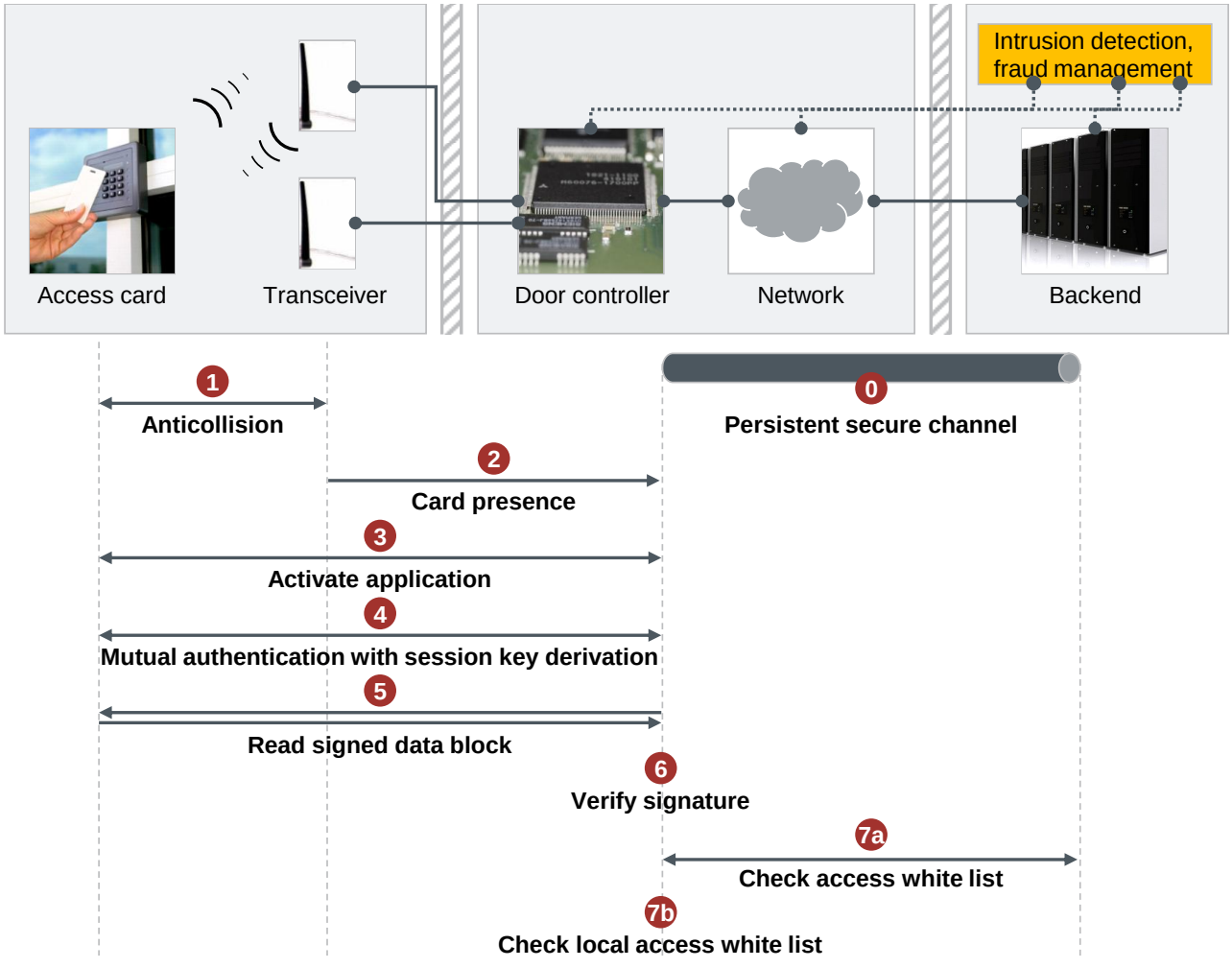


Figure 5. The card-reader protocol authenticates both sides using strong cryptography.

stored public key of the signing authority.

In **step 7a** - in case of an online controller, or if all the actions from step 4 onwards have been performed by the backend system - the entity, that is currently communicating with the card and that has verified the signed data block and its binding to the card, checks the user identifier in the signed data block against the white list in the backend system to make a final abort/accept decision on whether to open the door. The result is transmitted to the door controller so that it can open the door and end the transaction, including sending the log entry. Alternatively, if the controller is currently offline, it will perform this check against its local database, illustrated as **step 7b**.

There is a hybrid design option here: If the controller is online, but has been performing the card interactions from step 4 onwards by itself, it normally will only check against the white list in the backend. It is recommended to implement a fall-back to check against the local white list if the backend does not respond after a certain timeout (which would depend on several considerations, including the desired maximal authentication time). This behavior maintains the benefit of up-to-date white lists in the online case if the backend responds, with the advantage of not unnecessarily delaying authentication if the backend is temporarily unresponsive, and with only a small loss in security (cards that have been revoked will be continued to be accepted until the next local database update, just as with full-offline mode). The considerations for full-offline mode (e.g., white list update in configurable intervals, log storage and forwarding) still apply.

Logging: Starting at step 2, the controller creates a log entry to be sent to the intrusion detection system. The logging follows three rules:

1. Each time step 2 is reached, a log entry is generated.
2. No gratuitous log entries should be generated, i.e., don't send the entry right away when step 2 is reached, but at the latest possible moment with the most amount of data. In most cases logging will take place after step 7 (success case).
3. Log entries must not be dropped; if the controller is currently offline, it should queue all entries for submission at a later time.

4.3 Protocol Design Options

In summary, the design choices within the outlined framework are:

Encrypted channel starting at step 4

- Pro: Privacy
- Con: Needs additional cryptography on the card, may lead to export control difficulties

Direct card control by the backend system, starting at step 4

- Pro: Does not need card-related SAM in the controller
- Con: Only works in full-online mode
- Con: Several round-trips of network latency

Hybrid card control: Backend in step 4, remainder by controller

- Pro: Does not need card-related SAM in the controller
- Con: Only works in online mode

Hybrid white list check: Check white list on backend, resort to local database upon timeout

- Pro: Puts a hard and predictable limit on the time needed for authentication
- Con: Potentially delayed card revocation

The recommended balance between usability and security is: encrypted channel + hybrid white list check. Controllers in high-security areas that are guaranteed not to need the full-offline mode can use hybrid or direct card control. Even in these controllers the SAM is necessary to store the secrets that are used to establish the secure channel to the backend system (step 0), but it does not need to store key-generating keys that would allow to derive card specific keys which would allow to clone cards.

4.4 Key Storage

Secure Access Modules (SAM) chips maintain and protect secret keys. The smart card chips in the SAM resist intrusive attacks to a large degree—typically in excess of EUR 100,000 attack cost—as attested through a EAL 5+ certification [7]. SAM chips are used to store master keys in door controllers. In the card production facilities even stronger Hardware Security Modules (HSM) should be considered. The master keys never leave the SAM chips and HSMs.

Besides the non-functional capability to strongly protect keys, SAM chips und HSMs provide the following features:

Key derivation:

- Derive card specific key from master key and UID
- Common: Ability to perform card authentication and derive session keys in such a way that the card specific key never leaves the SAM chip
- Optional: Ability to encrypt/decrypt and authenticate/verify messages so that the session key never needs to leave the SAM chip

Risk Limitation:

- Enforce an upper limit (ceiling value) on the number of key derivations that can be performed to limit the usefulness of stolen SAMs
- Periodically receive encrypted and signed ceiling value updates/counter resets (obviously, these updates are not produced anymore for stolen SAM chips)

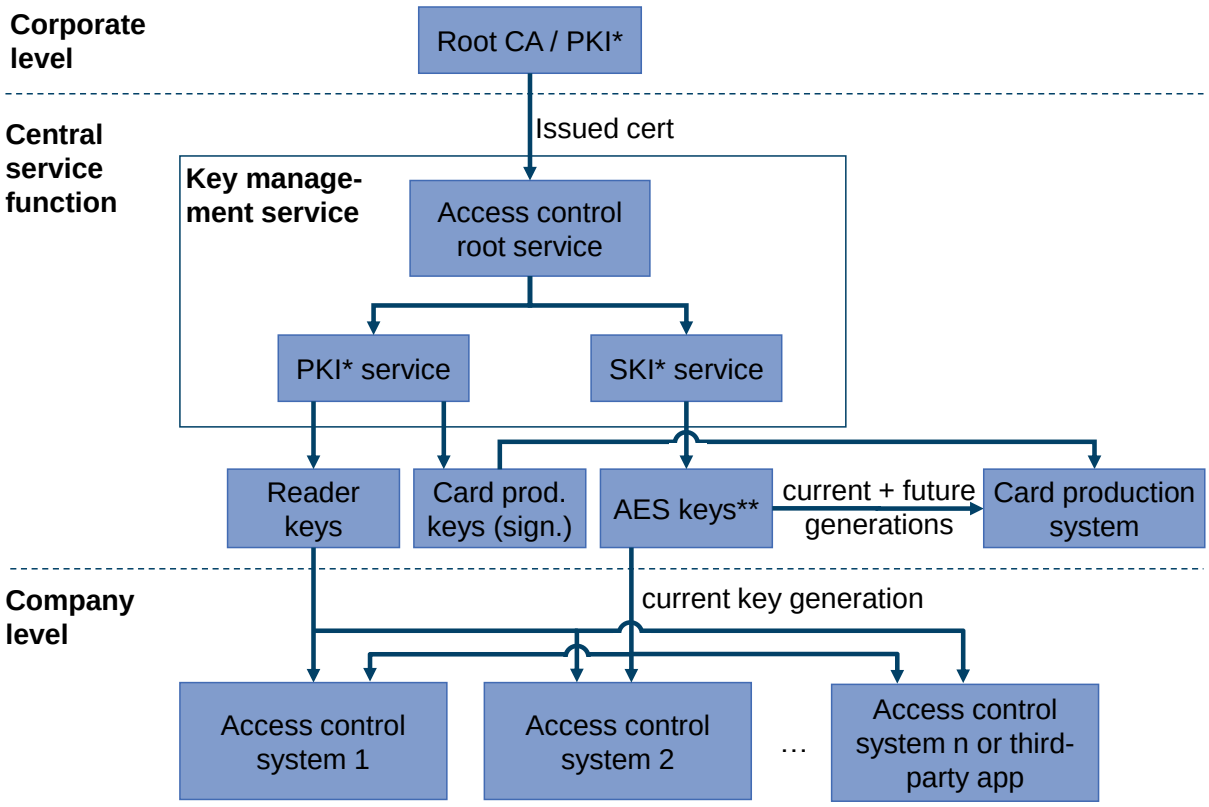
Key management:

- Store several generations of master keys, some of which might be deactivated until a certain 'magic number' is received
- Receive additional key generations, encrypted and signed for a particular SAM chip (implies the ability to check RSA signatures)

4.5 Key Management

The previous section dealt with the secure storage of master keys in SAM chips. The management and procedure to securely enroll and update master keys to the SAM illustrated in Figure 6 facilitates a secure key management based on a secure key storage.

Each controller SAM chip should hold its own RSA key pair. The public key is exported to the Access Control Root Service (derived from the company's PKI). In addition, the PKI Signing Key (certificate with the public key) is stored in the SAM chip.



* PKI: Public Key Infrastructure; SKI: Symmetric Key Infrastructure
** AES keys are only transferred in PKI-secured channels (online or offline)

Figure 6. Symmetric access keys are deployed to card production and reading devices in PKI envelopes.

Symmetric (site specific) master keys are sent to the controller SAM chip encrypted with the SAM chip's public key. It should be avoided to initialize or update the keys via the radio interface of the reader. If key updates from the building outside are disabled, performing malicious key updates becomes harder.

Master keys are signed by the PKI service to ensure the integrity and authenticity of the update package, which has to be checked by the SAM chip prior to accept the new key generation.

While a disclosed (site specific) master key can be updated for all readers with a reasonable effort, the clients (cards) are typically not as easy to update. Therefore, the card production service holds more generations of master keys than the controller SAMs to seed cards with current and future key generations. The storage of master keys in the card production environment can be done in a SAM chip at the card production server or on a PIN-secured SAM chip card per production operator.

5 Detailed specifications

The minimum requirements for an access control system span a large design space. From this space we provide specific designs for selected system components. The suggested designs have proven effective in practice.

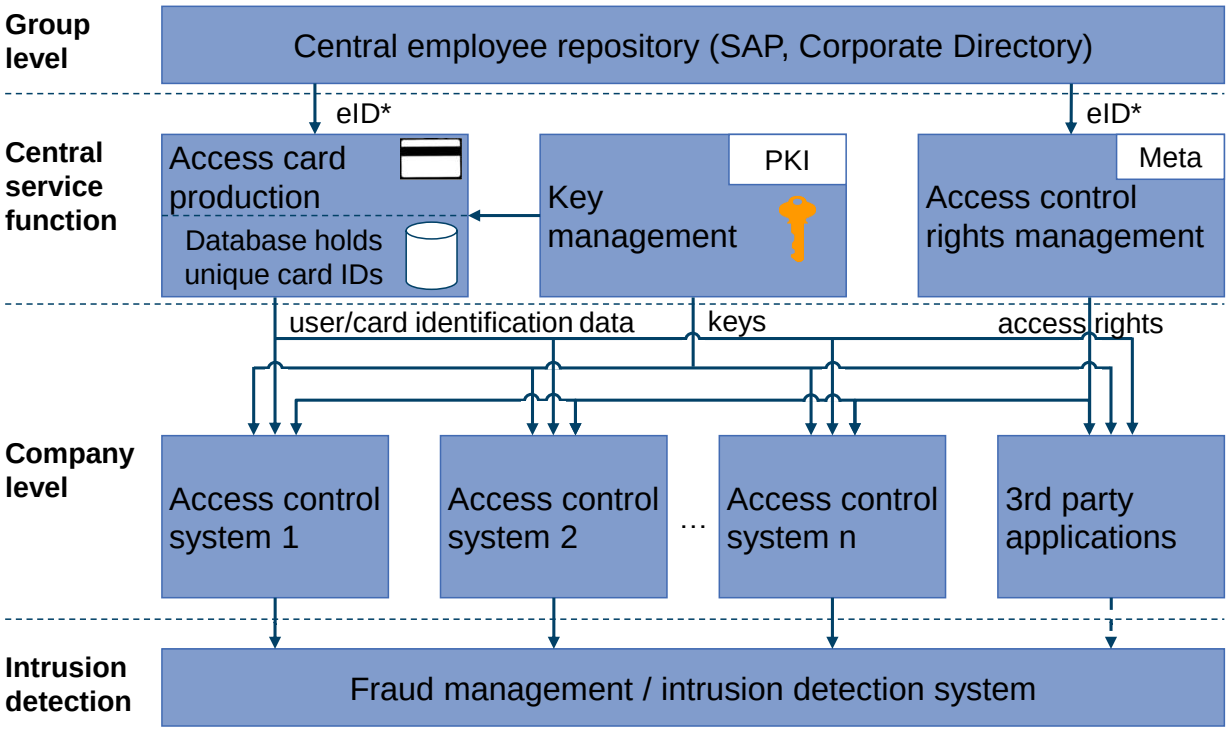
5.1 Card Layout

The data on an access card should be associated with an employee in such a way that the association can only be created by card personalization terminals. The card layout should furthermore anticipate future security upgrades that introduce new encryption or key management functionality. Card production and access rights management should be strictly separated and only the former should manage cryptographic keys as illustrated in Figure 7.

Two types of data should never be stored on the card itself:

A) Access rights of the card holder. This information needs to be managed in the backend systems and made available to the controllers from there rather than through the card.

B) Biometric information or additional knowledge (i.e., PIN). This data used as a second authentication factor needs to stay separate from the first authentication factor (i.e., the card) to provide a security gain.



*eID : unique employee ID

Figure 7. Multi-purpose employee service card as a group wide common service for access control.

Throughout the following description the term “block” should be understood in an abstract sense as an independent data storage area. On smart cards with a file system, one block would likely correspond to an EF.

For example, a minimal card layout could look as follows:

```
-------------------------------------
Preamble: Anti-collision UID
-----
Block 1: < Metadata of block 2 >, ... , < Metadata of
block N >
-----
Block 2: < Data as specified in metadata >
-----
...
-----
Block N: < Data as specified in metadata >
-------------------------------------
```

Blocks 2 through N require authentication to be read. All blocks require authentication to be written to.

The first data block specifies the format and encryption keys of all remaining blocks and consists of:

```
<Metadata> :=
 <Data layout version, 8-bit>,
 <Key generation, 16-bit>,
 <Signature key generation, 8-bit>
```

Both data layout and cryptographic keys should be upgradable in order to not constrain future applications and security upgrades.

Say, the metadata in preamble and block 1 are set to:

```
-----
UID = 123
-----
Metadata block 2 =
 Data layout version = 1,
 Key generation = 5,
 Signature key generation = 2
-----
```

The contents of block 2 then could follow this suggested data layout:

```
-----
Block 2 =
 Secured with [key diversified from master key 5 with
UID 123]
 {
   hash(employee ID + shared secret),
   //   'employee   ID'   is   a   person   identifier
   // unique throughout the access system
   signature with [RSA private key generation 2] of
   {
     hash(employee ID),
     UID = 123,
     Data layout version = 1,
     Key generation = 5
   }
 }
-----
```

All door controllers know the RSA public key of the signature so they can check the cryptographic association between card and employee. The private key needed to create the signatures is only known to the card creation terminals, which consequently need to be better protected.

Key diversification: It is advisable to never use system-wide master keys to access cards, but instead to use keys derived from the master key. These master keys would only be stored in SAM chips, where they are reasonably secure from attacks, while each card is encrypted with unique keys.

In cases where the system environment does not allow for key diversification (i.e., during a migration in an infrastructure with no SAM chips), a few key generations could be reserved for non-diversified keys. Door controllers upgraded with SAM chips would consequently ignore sectors secured with non-diversified keys and instead access a different sector of the card that uses a diversified key.

Unused sectors: Blocks not yet filled with data should be secured nonetheless so they cannot be written to or rendered useless. The keys that secure empty sectors are needed only when new data blocks are rolled out and should only then be made available to the reading devices.

Block 1 protection: The data in block 1 cannot be used to modify critical data on the card (i.e., to gain illegitimate access to buildings). How-

ever, modifying the data can render the card useless. For this reason, the write key of block 1 should stay secret. In addition, this write key should also be upgraded with every new key generation. The current write key would then be that one associated with the most recent data block. In the example above, where only block 2 is filled with key generation 5, the write key for block 1 would be the one associated with key generation 5.

Counter sector: An additional extension to the suggested card layout, that provides data for fraud detection, is a counter sector on the card. That sector simply counts the number of times the card was used. In case of card clones, the counters on the two cards will divert, which can be detected algorithmically in the backend. Many RFID cards provide one-way counters that cannot be reverted, which should be used where available.

5.2 Intrusion Detection Data

Assuming that all automated access control system architectures can be "tricked" with enough effort, an additional layer of security is needed to detect such actions. Although it is desirable to have a real time intrusion prevention, we advise to implement in a first step a post-mortem approach, which will trigger alarms when detecting suspicious actions. After a reasonable long learning phase with manual checks of triggered alarms, it might be considerable to implement the detection schemes in real time to actually prevent unauthorized access.

There exist a wide range of approaches in the field of financial fraud detection that are based on rules, data mining algorithms, and statistical analysis. For access control systems such algorithms are not well researched. We propose three different modules that – when applied all together – will reduce the probability of missing unauthorized access attempts or actions respectively.

All data (logs, metadata of reader locations, etc.) must be available, relevant, adequate, and structured. A minimum set of relevant data is listed in Section 3.3.

5.3 Intrusion Detection Strategies

Intrusion detection must be based on fuzzy decisions since there is no clear line between legitimate and illegitimate behavior. Measurements for these decisions include rule checking and statistical analysis.

Module 1: Rule based detection

This module is deterministic and often allows the detection of impossible and implausible circumstances. Examples:

- "Card in two places at once": Access log entries of one card ID in different places where the necessary travel time by plane would be larger than the given time window
- "Pass back": If a card passes a security turnstile with user separation and registration of in- and out-bound direction, a user cannot enter twice or leave when he has not entered the site. This typically occurs if a card is used to initiate the passing of a security turnstile and if the card is provided to a third person with the intention to gain access a second time
- "First instance of card usage to gain access to a secured area": A card is used the very first time on an entry point of a specific secured area

Module 2: Statistical modeling

It is possible to model the usage of access control entry points with respect to time, single cards/users, and any accumulation of an access controlled area, a building, and a site.

Typically, a fit to a statistical distribution function is performed from the data in order to define a probability threshold or confidence interval. New access log entries are compared to the learned data or confidence interval respectively. If it fits, the new data point is used to adjust the model. Examples:

- "Untypical time usage of a specific card": A user has not usually opened any door at this time
- "Untypical time usage on a specific secured area": No user usually opens this area at this time
- "Untypical usage of a specific card on a specific secured area": A user has not usually opened a door to a specific secured area

Models can be built for any combination/accumulation of users, entry points, time, areas within the building, etc. The probability thresholds for a model or the length of the respective confidence interval determine the detection sensitivity.

Module 3: Combination of modules 1 and 2

To strengthen the detection specificity it is useful to combine rules defined in module 1 and 2. Example question: "Does the entering of a user at its very first time correspond to normal access time frames for the area the user has entered?" with the single rules:

"First instance of card usage to gain access to an secured area" AND "Untypical time usage on a specific secured area": If both detection rules match, it is quite unlikely that a user who has never accessed a site does this at an unusual time. We would expect an access that is accompanied by someone who explains the site.

6 Migration

The security of all popular access control cards—including Mifare Classic, Legic Prime and HID Prox—has been shown to be weak. Organizations using these cards now need to revise their risk analysis. The risk analysis typically leads to the need of upgrading the installed technology. The organization will consequently need a strategy for migrating to a new access control architecture.

6.1 Migration Target

Especially in large scale installations, access cards and door controllers cannot be migrated all at once. Two technologies must be supported by access cards for the duration of the migration. The two technologies can either be provided by a dual-antenna card, that hosts two independent chips. This option is needed for Legic Prime migrations since no other card is compatible with the lower levels of Legic Prime communication. For other card technologies, in particular Mifare Classic, the legacy card can be realized as an applet in a newer card, such as Mifare DESFire, Plus, or SmartMX.

For the time being it is difficult to implement a Legic Prime migration scenario in conjunction with the Mifare DESFire technology, since its relay/replay attack detection scheme leads to a high probability to be triggered when such a hybrid card is used in a Legic Prime environment. After a couple times of usage the Mifare DESFire chip reaches the upper limit for the attack detection counter and disables itself. To mitigate that, there exists a recommended special antenna layout to reduce the probability of triggering the false attack detection significantly. However, this problem does not apply to dual-antenna cards with Legic Prime/Mifare Plus and Legic Prime/SmartMX combinations. It is also worth noting that NXP is going to release a Mifare DESFire version soon, which will not show the behavior mentioned above.

6.2 Migration Dependencies

Access cards are often used to enable third party applications, such as cafeteria payment, proprietary RFID-based offline door cylinders not managed by the online access control management system, gasoline billing systems, and locker keying.

If an organization is migrating to a new card, these dependencies have to/need to be solved transparently to the user. In other words, the user should not need to care about the card generation and whether it is working for a specific application.

Typically, the additional functions of an access card are owned by third parties, which are independently organized and feel responsible for their own system only. From their point of view, the access card tech-

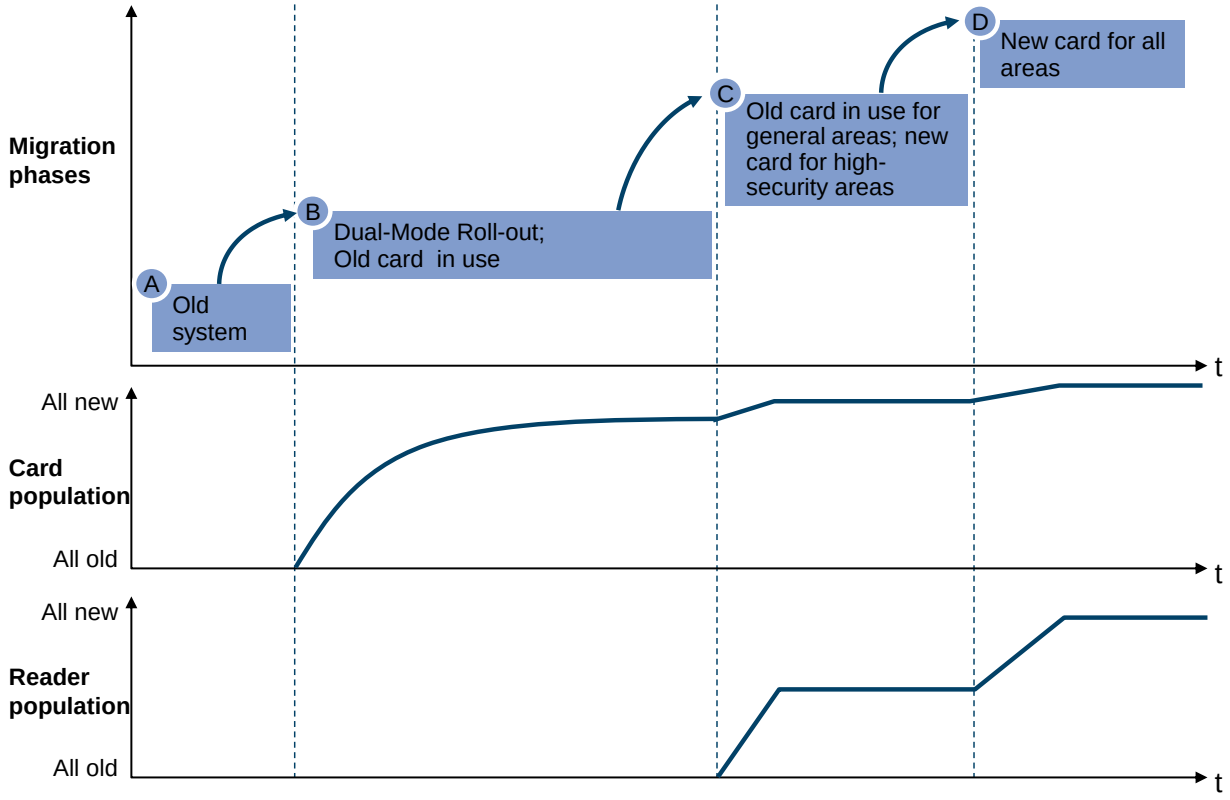


Figure 8. Cards and door controllers are upgraded to the secure target platform in four stages.

nology and data structure is considered as a steady foundation, which is never going to be changed. Thus, additional time for adjusting to the new technology must be given and a good rationale provided to get their support for a migration scenario.

If it is necessary to replace the backend system due to a vendor or integrator change, a second layer of applications can have dependencies to those backend structures. An example is the integration with an ERP system, where external companies are paid according to polled time stamps of their access card usage for service functions.

6.3 Migration Strategy

A migration is typically executed in four phases as shown in Figure 8. During the planning and preparation phase, **phase A,** decisions on the new card layout, third party dependency issues, and a roll-out prioritization according to a risk assessment of access controlled areas are made. In addition, card production has to adapt the new technology to handle the dual interface cards (encoding) and incorporate the new key management.

In **phase B** a roll-out/card replacement with the new hybrid access card for all affected users of the access controlled areas that are going to be upgraded next is performed.

According to the prioritization list of phase A, an access controlled area wise exchange takes place in **phase C**. It is obvious that once an area is equipped with the new technology, door controllers should not grant access any more using the old (insecure) technology.

Phase D finalizes the migration by replacing the remaining door controller instances and stopping the production of hybrid access cards. All future access cards will not be equipped with the old RFID chip technology. Clearly, all dependent third party systems must have been transformed at this time to work with the new technology and card layout.

7 Conclusion

State-of-the-art access control is the symbiosis of minimizing risks through modern technology and reducing impact through intrusion detection and update procedures.

Deployed access control technologies fall short of the protection level that modern RFID smart cards provide. They use cryptographically insecure encryption often in combination with predictable secret keys. A modern system must use standardized encryption and a multi-tier key management scheme that protects valuable master keys in secure hardware modules. Managing such a centralized access control scheme becomes a group responsibility rather than the responsibility of individual divisions' facility management teams. Consequently, access control key management should be overseen by the group's IT security or corporate security team and held up to the same requirements and risk management procedures as other IT assets.

Besides these organizational changes, migrating to a modern access control system requires controllers to be equipped with secure key storage chips and access cards to be replaced. Through the use of dual-interface cards, the migration is stretched over a manageable time period, starting at the most valuable locations.

Even with a strong technology base, incidents should be expected. Update procedures for reader software and keys assure keeping the time period affected by a data leak small. Intrusion monitoring provides a vehicle to detect access attempts from stolen cards.

The access control market is ripe for a change and ready to gain the same protection level that we have been experiencing in IT systems for years.

8 References

[1] Nohl, K. and Plötz, H. Legic Prime: Obscurity in Depth. *26C3*, 2009.

[2] Nohl, K., Evans, D., Starbug and Ploetz, H. Reverse-Engineering a Cryptographic RFID Tag. *USENIX Security Symposium*, 2008.

[3] Soos, M., Nohl, K. and Castelluccia, C. Extending SAT Solvers to Cryptographic Problems. *SAT,* 2009.

[4] Garcia, F.D., Rossum, P.v., Verdult, R. and Schreur, R.W. Wirelessly Pickpocketing a Mifare Classic Card. *IEEE Symposium on Security and Privacy*, 2009.

[5] Jonathan Westhues. Proxmark III hardware device. *proxmark.org*. 2007.

[6] Tarnovsky, C. Hacking the Smartcard Chip. *BlackHat DC*, 2010.

[7] Common Criteria. Application of Attack Potential to Smartcards v2.7. *Evaluation Standard*, 2009.

[8] ECRYPT II Yearly Report on Algorithms and Keysizes (2009 - 2010).

[9] Lutgen, J. The Security Infrastructure of the German Core Application in Public Transportation. *VDV White Paper*, 2007.