MIFARE DESFire in Practice: A Systematization of Attacks and Deployment Weaknesses

Tomáš Přeučil[1], David Oswald[2] and Martin Novotný[1]

[1]Czech Technical University in Prague: Faculty of Information Technology, Thákurova 9, 160 00 Praha 6, Czech Republic
[2]Durham University, Stockton Road, Durham, DH1 3LE, United Kingdom

Abstract

MIFARE DESFire cards are widely deployed in access control, transportation, and payment systems and are generally considered secure when configured appropriately. However, in real-world deployments, exploitable weaknesses still exist. In this Systematization of Knowledge (SoK), we consolidate and structure the existing body of work on DESFire security, covering cryptographic weaknesses, implementation flaws, protocol misuse, relay attacks, and deployment misconfigurations. We classify known attack vectors against DESFire systems and examine how problems in different parts of a deployment can combine in unexpected ways. Taking published attacks and real-world incidents into consideration, we show that secure cryptographic primitives alone do not guarantee secure deployments. Based on this analysis, we outline common mitigation strategies and present practical considerations for system designers and administrators. Our goal is to provide a structured overview of DESFire security and to encourage more robust deployment practices.

Keywords

MIFARE DESFire, RFID security, access control systems, systematization of knowledge, relay attacks, key management

1. Introduction

Contactless smartcards underpin access control, transportation, and payment infrastructures worldwide. Security failures in such systems have repeatedly led to large-scale breaches [1], yet vulnerable designs remain deployed [2, 3, 4, 5]. MIFARE DESFire (from the EV1 version) has long been regarded as the "secure" family within the MIFARE product line, despite the insecurity of its predecessors [1, 6].

Prior research has investigated selected aspects of DESFire security, including protocol design decisions, cryptographic properties, implementation weaknesses, and side-channel leakage [7, 8, 6, 9]. However, these results are fragmented across different attack classes and threat models. As a result, operators of such systems, as well as researchers, lack a unified perspective on how individual weaknesses interact and how deployment choices influence security in real-world deployments.

In this work, we present a structured analysis of DESFire security across attack vectors, deployment configurations, and mitigation strategies. We aim to consolidate prior research and clarify the practical security by examining different weaknesses appearing in real-world systems.

Our contribution.

1. **Structured overview**: We organize known DESFire attack vectors into a framework that groups cryptographic weaknesses, implementation flaws, protocol misuse, relay-based attacks, and deployment misconfigurations.
2. **Practical perspective**: We discuss how these attack classes relate to each other and how weaknesses in real world (using documented attacks and studies) at different layers (card, reader, backend, and configuration) can jointly affect system security.
3. **Mitigation framework**: We summarize countermeasures and migration strategies discussed in prior work and analyze their applicability as well as limitations in real deployments.

TDI 2026: 4th International Workshop on Trends in Digital Identity, April 20–21, 2026, Verona, Italy
tomas.preucil@fit.cvut.cz (T. Přeučil); david.f.oswald@durham.ac.uk (D. Oswald); martin.novotny@fit.cvut.cz (M. Novotný)
0000-0001-7856-5149 (T. Přeučil); 0000-0001-8524-5282 (D. Oswald); 0000-0001-6446-7257 (M. Novotný)

Positioning. We do not aim to present a new cryptographic break of DESFire. Rather, this work aims to systematize existing knowledge into a unified framework. Our analysis shows patterns that are not immediately visible when attacks are considered in isolation. The resulting overview, summarized in Figure 1, provides a view of the DESFire attack surface.

Organization. Section 2 introduces technical background. Section 3 presents the structured overview of known attack vectors (except for relay attacks). Section 4 discusses relay-based attacks. Section 5 analyzes countermeasures and deployment strategies. Section 6 concludes.

2. Preliminaries

This section introduces the technical background required for the analysis in later sections.

2.1. RFID Cards and the MIFARE Product Family

First, we clarify terminology: In this paper, we use the terms "card" and "tag" interchangeably. Contactless RFID cards commonly operate on two main carrier frequencies: 125 kHz and 13.56 MHz. As 125 kHz cards are increasingly phased out for access control, 13.56 MHz cards (including contactless payment cards) become more widespread. Vendors producing 13.56 MHz products include NXP (MIFARE) [10], Legic [11], and Thales (formerly Gemplus) [12]; this paper focuses on NXP's MIFARE lineup.

Each card exposes a Unique Identifier (UID). [1] Many MIFARE products support cryptographic operations; the primitives available depend on the product generation and configuration. MIFARE Classic variants are widely regarded as insecure [1]. MIFARE Plus and DESFire (EV1 and newer) are generally considered secure when configured appropriately.

2.2. Security Considerations

DESFire EV0 was shown to be vulnerable to side-channel attacks in 2011 [6]. The MIFARE Ultralight family targets low-cost use cases and ranges from no cryptography to AES-based variants depending on the model [13].

Even with modern cards, deployments can be vulnerable due to configuration and system-level design choices. Many operational failures occur at the application layer (for example, in access-control management software), where incorrect parameterization or shared site keys reduce security. Common management suites include Salto [14], Gallagher [15], and others. Vulnerabilities can also arise at other layers (hardware interfaces, firmware storage of keys, etc.).

2.3. DESFire architecture

MIFARE DESFire is an ISO14443-A [16] compliant card family. DESFire cards come with 2, 4, or 8 kB of memory. Unlike other MIFARE products, the memory of DESFire is not organized into sectors or pages. Instead, it adopts a structure of applications and files, where each application is identified by an Application ID (AID). The card-level application (AID 000000) contains global configuration and the PICC master key; however, some of these can be overridden by other applications in their own contexts.

Any application (except the card application, AID 000000) can contain multiple files and a set of application-specific keys. Each file can be configured to use different ciphers (DES/3DES/AES) and different communication modes (plaintext, encrypted, CMAC-protected).

Authentication between reader and card is based on a challenge/response protocol using the selected application key. Successful authentication establishes a session key that protects subsequent communication according to the configured mode.

[1] Some cards use non-unique identifiers (NUID) when the available UID space is insufficient.

2.4. Key diversification

NXP specifies a key diversification mechanism based on AES-CMAC that derives application keys from a site master (site) key, the UID, AID and a key number (specific to the particular application) [17]. If an attacker learns the site (master) key, they can derive all per-card/per-application keys; conversely, deriving the site key from a single card key is not feasible without further weaknesses. Since NXP's algorithm uses the AID as one of the input variables, it effectively allows the card to be used at multiple sites (each site has its own AIDs and its own site key). However, the sites must agree on an AID numbering scheme as well as one master key for the master application.

In practice, DESFire deployments typically involve multiple layers: the card, reader, management software, as well as operator's diligence during configuration. That means that security depends not only on the strength of the underlying cryptography but also on key management practices, application configuration, and system integration decisions.

3. Systematization of DESFire attack vectors

In this section, we provide an overview of known attacks on DESFire cards (excluding relay attacks, which are discussed separately in Section 4). The surveyed publications were identified based on domain knowledge of the authors and established literature in the field, and were complemented by targeted searches in publicly available academic databases (e.g., Google Scholar) to ensure coverage of relevant prior work.

Based on the identified literature, we classify the attacks into three categories:

- attacks that extract keys from the reader,
- attacks exploiting system misconfiguration, and
- attacks targeting vulnerabilities in specific card implementations.

Figure 1 shows an overview of these attacks.

3.1. Key extraction from the reader

In this subsection, we describe attacks that obtain the keys required to access data on a card by extracting them from the respective reader. It should be noted, however, that the work by Hobson falls both in this section and Section 3.2.

Laurie looked at an access system that uses DESFire EV1 and whose readers are based on Atmel SAM7XC [24] in 2013 [18]. The MCU has a crypto co-processor as well as read-protected ROM (and RAM under certain cases). However, Laurie found out that the fuses, which block the readouts, can be reset. While this wipes the flash, the RAM stays intact and can then be read out. This allowed Laurie to get the master key for the system. He was then able to obtain all remaining keys using the diversification process. This allowed him to access the entire content of any card belonging to the system. He states that the master key *may* be generated on a per-system basis or that it *could* even be manufacturer-specific. When he reached out to Atmel, they confirmed that there was no way to protect the RAM and ignored his request for clarification on how to handle keys securely. Laurie notes that the keys should never pass through the RAM, which could be, for example, achieved by storing them in the flash using the *const* qualifier.

Hobson presented several attacks against an access system developed by a New Zealand-based company, which uses DESFire cards in 2023 [19]. He was able to find a hardcoded master key in the firmware of a reader, which allowed him to perform his attacks, including the cardholder ID incrementation. He also found that the respective readers have a weak PRNG, which only generates around 8000 seeds for the DESFire authentication. Therefore, he was able to get both random numbers used in the DESFire authentication process (see Section 2.3), which means he was able to forge the entire

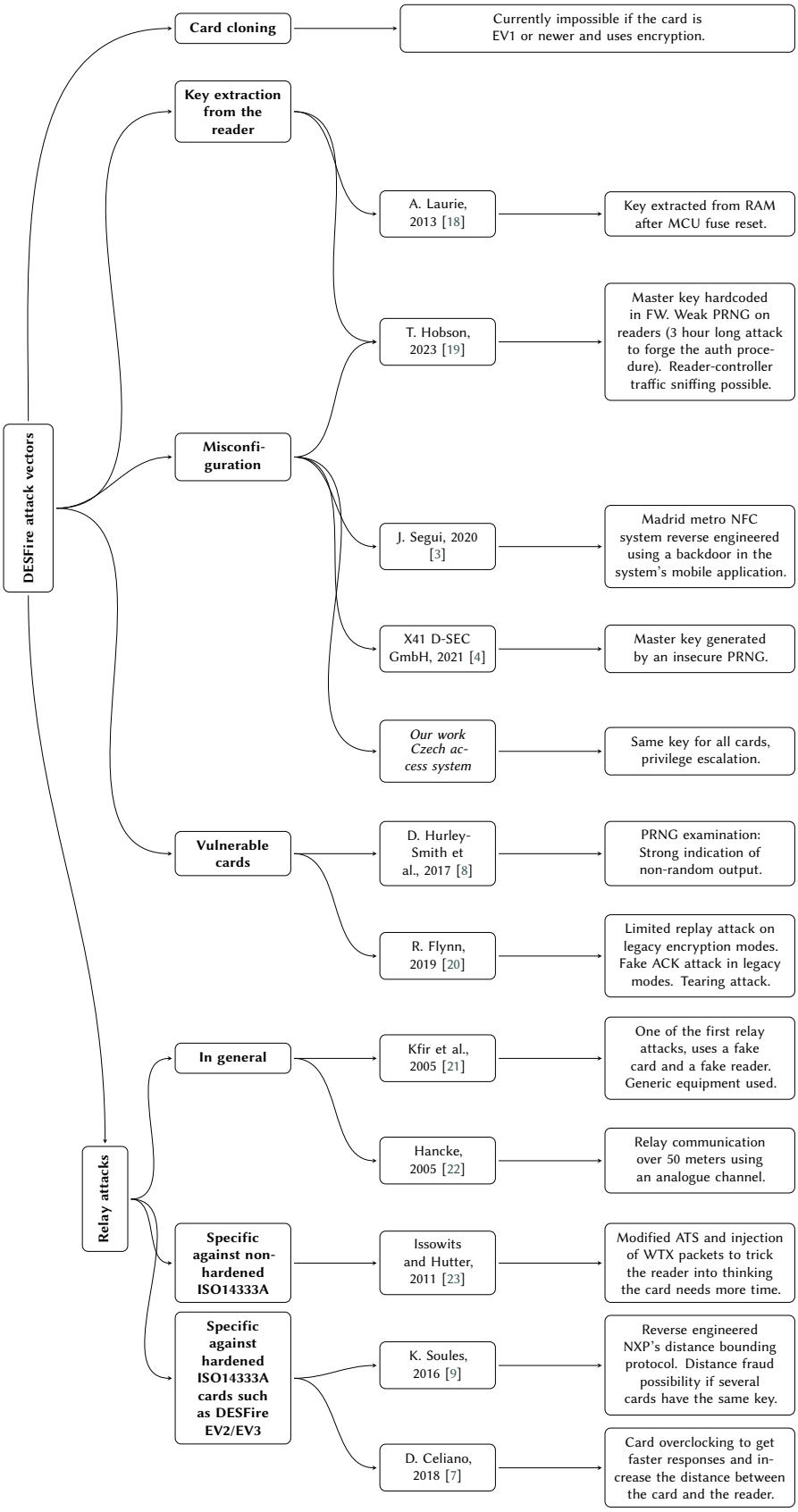

Figure 1: Systematisation of DESFire attack vectors (EV1 and newer).

DESFire authentication procedure in about three hours. At the time of his conference talk, the manufacturer claimed to be working on a fix. Besides this attack, Hobson was also able to sniff and decrypt the communication between the reader and the system controller, as well as gain code execution on the readers.

3.2. Misconfiguration

In this subsection, we present attacks which exploit the misconfiguration of an access system.

Segui was able to reverse engineer a good part of the Madrid Metro NFC system in 2020 [3]. The system uses DESFire EV1 and seems to allow communication between the server and the card via a mobile application. Segui managed to find the endpoints the application communicates with and found out that the endpoints always return the command that is to be sent to the card. This allowed him to see the entire authentication process and get the session keys. He also developed an Android application to debug this behaviour. This platform would then let the attacker keep the session with the card open (keeping the field on). Then, they could alter the data in the files accessed by the endpoints as they wished. Segui leaves out the information on whether the endpoints are used to recharge money onto the card. If this is the case, the attacker might be able to add money to the card as they wish.

Researchers from X41 D-SEC GmbH found a critical error in a system suite by Telenot called Complex compasX in 2021 [4]. This suite uses DESFire EV1 and EV2 cards and allows remote management. The researchers found out that the AES keys used in the suite are generated by the *rand()* function, which is a cryptographically insecure pseudorandom number generator function. It also uses the current Unix timestamp as a seed. Due to these oversights, the researchers were able to brute-force the AES keys. They proved the impact on a real-life example and were able to have a fake card ready in roughly three minutes. They also suggested an attack against the remote management functionality, an attack which can be done fast and offline if the initial handshake is captured, or online without the initial capture and within days.

We evaluated an access system designed by a Czech company in 2024. This system is distributed with DESFire EV2 or EV3, while still supporting DESFire EV1. In the examined version, other types of cards are not supported, which increases security. The system generates two keys for each site. One master key (used for the master application on the card) and one application key (used for the application/file which stores the credentials). However, these keys are not diversified on a per-card basis. The readers only store the application key, and this key does not allow the write operation. However, we found out that the application key does allow the change of permissions, which means the attacker can disable the write protection (if they had the application key). We also found out that the system does not check UIDs and that the credential files only differ in a few bytes between different cards/users. If an attacker were to gain access to the application key, e.g. through a side-channel attack, this would significantly simplify a brute-force attack.

3.3. Vulnerable cards

In this subsection, we present attacks against DESFire cards themselves.

Hurley-Smith and Hernandez-Castro examined the quality of the PRNG of DESFire EV1 in 2017 [8]. They used NIST and Diehard randomness tests, on most of which the cards confirmed good randomness. However, on the Diehard $32x32$ binary rank test, the cards performed poorly. They suggest that this is *"unsurprising, but not strongly indicative of non-random output"*.. When they tried the chi-square test, it indicated a distinctly non-uniform distribution. However, their pattern-seeking test did not fail. They note that these tests present only some characteristics of the RNG and that *"an attack exploiting the biases found has not yet been identified"*. They reported their findings to NXP.

Flynn found three platform-independent vulnerabilities affecting DESFire EV1 in 2019 [20]. The basis for his attack is a man-in-the-middle relay attack using an SCL3711 reader (as a fake card) and a Proxmark (as a fake reader). This platform allowed him to modify the data passed between the legitimate reader and the legitimate card. He did have to change the respective ATS byte as explained in Section 4.2.

His first two attacks affect the legacy modes of DESFire EV1 (plaintext, MAC'd DES and enciphered DES), but not the modern AES / 2TDEA / 3TDEA modes. The first attack makes use of the fact that in legacy mode, the IV is zeroed at the beginning of each new command (while, in modern modes, the IV is chained—see Section 2.3). This lets him repeat encrypted commands (such as crediting the card) during the same session, since this means that the transmitted data stays identical. However, this is only possible within the same session (therefore, the reader field needs to be kept on), as every session lead to a new session key.

The second attack makes use of the fact that in legacy modes, the "OK" response to a write command is unencrypted. Therefore, when passing through a barrier that would decrement the balance on the card, the attacker can simply fake the OK response and never let the data be written onto the card.

The last attack relies on a weakness of the anti-tearing mechanism in the card, and is applicable to the modern AES modes as well. The card can be configured to use so-called backup files. These files hold a copy of the respective file and only get updated on a *commit* command. If the card is powered down without the reader issuing such a command, the content of the backup file does not change, and the original file gets overwritten with the content of the backup file. This attack assumes a transit system which uses the "tap-in/tap-out" method without any actual barriers, where the tickets are checked randomly by a controller. In this scenario, the available balance (credit used to ride the public transport) is stored on the card, and once the rider enters a barrier, an amount is deducted from the card's balance. If a possible attacker walks through a barrier and taps in, they can keep the card powered on and not send the *commit* command. In the case they would be checked, they can send the *commit* command, be charged but not fined. If they are not checked, they power the card off without sending the *commit* command. In this case, they ride for free (the deduction of the balance "never happened"). This attack was reported to NXP, to which they responded by implementing transaction timers in DESFire EV3, which was released in 2020.

4. Relay attacks

This section summarizes known relay attacks against RFID cards. We first cover general attacks, then narrow the focus to ISO14443A, and conclude with those targeting DESFire EV2. DESFire EV0 and EV1 are grouped with ISO14443A cards since only EV2 and newer support protocols are supposed to provide strong countermeasures against such attacks [9].

First, we present a definition of a relay attack by Tu and Piramuthu: they define a relay attack as an attack that "*signifies that communication messages are unexpectedly passed between two parties that include an RFID tag (T) and a reader (R). It is unexpected since the two parties are not in close physical proximity of each other as is required for such communication to take place*" [25]. Relay attacks on automotive keyless systems are out of scope of this paper.

4.1. Relay attacks in general

One of the first published relay attacks by Kfir and Wool introduced the idea and terminology: a fake card (*ghost*) communicates with a fake reader (*leech*) that relays data to the genuine card [21]. The leech–ghost distance is arbitrary, and the communication is unaltered. The authors also propose several ways to increase the range between the reader and the ghost, as well as between the leech and the card.

In most recent publications, three types of relay attacks are discussed: Mafia fraud, Terrorist fraud, and Distance fraud. Although the origin of this terminology is difficult to pinpoint, all sources referenced in this work use it consistently. Note that in Figure 1, we categorize relay attacks according to

the subsections (In general, Specific against non-hardened ISO14333A and Specific against hardened ISO14333A cards such as DESFire EV2/EV3); however, all Mafia, Terrorist, and Distance fraud attacks can appear in any of these categories.

A *Mafia fraud* relay attack means that neither the verifier nor the prover is aware of the attack, while the attacker presents a fake card to the honest reader (verifier) and relays the communication over a channel to a fake reader, which communicates with the honest tag (prover). In other words, this is a man-in-the-middle attack. *Terrorist fraud* relay attack works similarly to the Mafia attack, but the tag is dishonest. In a *Distance fraud* attack, the (typically dishonest) tag is located outside the usual reading range of an honest reader and convinces the reader that it is closer than it actually is.

There are many different channels over which a relay attack can be performed. One example is a "simple" RF channel, such as in the work of G. Hancke from 2005 [22]. In the case of this Mafia attack, the author was able to relay the communication over 50 meters using an analogue channel. Another option for conducting Mafia attacks is to use a smartphone with Bluetooth or Wi-Fi [26, 27] or even relay the communication over the internet [28]. On the opposite side of the spectrum, one can also use a wired relay, which works over a (coaxial) cable, which is always going to be the fastest relay possible [29].

4.2. Relay attacks specifically against ISO14443A cards (including DESFire EV0 and EV1)

All the relay attacks presented above take advantage of the fact that most RFID cards do not deploy any countermeasures against relay attacks. Many cards and readers even feature mechanisms that allow for "slower" communication. For example, when an ISO14443A card sends an *Answer to Select* (ATS) frame, it includes one byte which specifies the *Frame Wait Time* (FWT), which defines the maximum delay the card may introduce before starting its response [16]. If the attacker performs a Mafia or Terrorist relay attack, they are effectively performing a man-in-the-middle attack. This allows them to modify the FWT value in the ATS frame so that the reader accepts longer response times, giving the attacker additional time for relaying messages and thus enabling attacks over greater distances.

The ISO 14443A requires an FWT of 86–91 µs during activation. Issovits and Hutter (2011) [23] showed that this still allows relay attacks by using a tag emulator with a copied UID (performing the protocol initialization directly on the emulator) and a modified FWT, establishing a Bluetooth relay link.

This is not, however, the only exploitable part of ISO14443A. Issovits and Hutter also showed that one could exploit the Waiting Time Extension (WTX) command [23]. This command is used by the card when it needs more time to process the data. It simply grants the card another FTW time window. The attacker only needs to send out these commands as needed. Issovits and Hutter were also able to utilize the not-acknowledge packet (NAK), which is sent when the FWT expires. Since a NAK symbolizes a retry to re-establish a connection, all the attacker needs to do is drop the NAK packets when they cross the relay channel.

4.3. Distance bounding protocols as a countermeasure

The universally accepted countermeasure against the presented attacks are distance bounding protocols. There are many different proposed distance bounding protocols. Some of the more "exotic" ones rely on sensing ambient conditions, such as temperature [30], use GPS [31] or sense the presence of other signals [32]. Some even propose the use of a Faraday cage [21] or quantum bits [33].

One of the earliest and most influential distance-bounding protocols was introduced by Brands and Chaum [34]. Their protocol relies on measuring the round-trip time (RTT) of rapid challenge/response exchanges, where the verifier sends random bits to the prover, and the prover responds by XORing these bits with a secret key. The primary goal of this construction was to mitigate Mafia fraud attacks. Building on this foundation, Hancke and Kuhn [35] as well as Tu and Piramuthu [36] proposed refinements that preserved the core principle while improving practical robustness. A further extension,

the Swiss-Knife protocol by Kim et al. [37], advanced the design by introducing not only enhanced resistance against Mafia fraud but also protection against Terrorist fraud, in addition to offering fault tolerance.

DESFire EV2 and newer, as well as MIFARE Plus EV1, use a distance bounding protocol called Proximity check, which is similar to those discussed above. This protocol was never publicly disclosed. However, it was reverse-engineered by K. Soules in his Master's thesis [9]. He found out that the protocol works as follows. Both the prover and the verifier generate 8 random bytes. These 8 bytes are then exchanged in a rapid phase during which the verifier measures the RTT. The verifier can send anywhere between one and all remaining bytes, while the prover always responds with the same number of bytes it received. After the end of the rapid exchange, operations are performed on the bytes to ensure that the prover knows the preshared secret key and that no tampering happens. The verifier then decides whether the proximity check was successful. It is also important to mention that Soules found a timing side channel leak, which allowed him to determine whether the proximity check was successful. This should not be possible as it leaks information to the attacker.

4.4. Defeating the countermeasures on NXP RFID cards such as DESFire EV2 and newer

NXP's Proximity Check protocol has several inherent weaknesses that can be exploited. Soules presents one such vulnerability and corresponding attack in his work [9]. The issue arises when multiple cards share the same keys for the distance-bounding protocol, as an attacker can place just one card near the reader, while another (key-sharing) card can be located elsewhere.

The core issue is that the Proximity Check runs only once. It is carried out during the anticollision phase, before any data access takes place. As a result, an attacker can route this initial verification to a card in proximity of the reader, while redirecting subsequent communication to a card located remotely. This attack can be partly mitigated if the system rigorously verifies the UID. However, the vulnerability persists if the attacker can switch some requests to an emulator (e.g., ChameleonMini [38]) and others to the card in proximity of the reader.

Another technique to bypass the Proximity Check is based on overclocking the target card. This approach was demonstrated by D. Celiano in his thesis [7], where he showed that DESFire EV2 and MIFARE Plus EV1 cards can operate at increased carrier frequencies of 16.3 MHz and 16.5 MHz, respectively. At frequencies higher than these thresholds, the cards no longer respond. However, in comparison with the nominal operating frequency of 13.56 MHz, Celiano was able to get about 20% speedup. By exploiting this reduced response time, he calculated that an attacker could successfully relay the communication over distances of at least 40 km. His work further outlines the specific hardware components required to mount such an attack.

The last relay attack technique we want to discuss is called *Early send/late commit* (sometimes also called *early detect/late commit*). These attacks were discussed, for example, by Clulow et al. [39] or Hancke et al. [40]. The early send (or early detect) attack works by sampling each bit sooner than intended. For example, the prover on the receiving end would usually sample the respective bits halfway or by the end of the bit duration period. If the attacker samples the bits earlier, it gives them more time to react and relay the data. Late commit works in a similar way—by sending the respective bits late but soon enough for the receiving verifier to detect them correctly. If signal processing is used and/or if the verifier integrates the value of the bit over time, the attacker can simply transmit more energy in such a way that the resulting average energy matches the expected bit. However, we are not aware of any successful attacks against DESFire using these methods.

5. Best practices and suggested countermeasures

We presented several attacks targeting access and transport systems based on the MIFARE DESFire platform. This section outlines recommended countermeasures and design practices to mitigate these vulnerabilities.

The first requirement is to *use AES encryption.* Systems should avoid DESFire EV0 cards and ensure that *all applications* on each card operate in AES mode, since legacy modes are susceptible to known attacks. The system should also generate a secure, site-specific master key, which serves as the basis for key diversification. Each card must have its own unique set of diversified keys, and no keys should ever be reused across cards or applications.

If an attacker gains access to the master key, the entire system is compromised. Therefore, protecting this key is critical. The master key should be stored only within the system controller, not within readers. The most secure approach is to use a dedicated secure element or encryption co-processor, such as the MIFARE SAM [41].

If protection against relay attacks is required, the system should use DESFire EV2 or EV3 cards, enable the *Proximity Check* feature, and apply diversified keys. For systems employing anti-tearing, DESFire EV3 should be used, with the transaction timer configured appropriately. Two additional DESFire features can further improve security when used *in addition* to the abovementioned recommendations:

- Random UID mode: In this mode, the card generates a new 4-byte UID during each initialization sequence. The real UID is accessible only after authentication. Random UID helps prevent user tracking, but since the reader must authenticate to retrieve the real UID, it requires a non-diversified key, introducing a potential key management challenge.
- Originality check: Available on DESFire EV2 and EV3 (and some other NXP cards), this feature verifies that a card is a genuine NXP product. While this adds another layer of assurance, it does not prevent cloning if an attacker can replicate the entire card image (for example, onto a ChameleonMini [38]), since the originality signature can also be copied.

From an operational perspective, facilities should employ monitoring and alerting mechanisms to detect repeated unauthorized access attempts. Alerts should be sent to a designated administrator, who can promptly investigate the source. Readers could also implement a progressive lockout mechanism, disabling themselves for increasing time periods after consecutive failed attempts. To summarize, a secure DESFire-based access system should:

1. Generate a strong AES master key for each site and store it securely (e.g., in a MIFARE SAM).
2. Use the master key for per-card key diversification.
3. Deploy only DESFire EV2 or EV3 cards and configure all applications to use AES mode.
4. Enable the Proximity Check feature and secure it with a diversified key.
5. Configure the transaction timer (if anti-tearing is used).
6. Perform an originality check on every card initialization.
7. Consider using random UID mode for enhanced privacy.
8. Implement intrusion alerts and reader lockout behavior.

6. Summary

This work presented an analysis of security issues in real-world access control systems based on MIFARE DESFire cards. Although DESFire offers modern cryptographic primitives, the overall security of deployed systems strongly depends on configuration choices, key management practices, and operational procedures.

We organized the known DESFire attack vectors into several classes, showing how weaknesses across multiple layers, including cryptographic configuration, protocol use, backend software, and physical infrastructure. We showed that many practical risks stem from the interaction between these layers and from deployment-specific decisions.

Our analysis of documented attack classes, together with representative case studies, illustrates the gap that can exist between the theoretical security properties of DESFire and the effective security achieved in practice. This gap is often driven not by fundamental cryptographic breaks, but by misconfiguration, insufficient isolation of trust domains, and inadequate key management.

Lastly, we presented a framework for systems administrators which shows the needed configuration techniques for configuring a secure access system.

In conclusion, DESFire-based systems can provide strong security, but this heavily depends on how they are deployed and maintained. The main risks identified in this work arise not from fundamentally broken cryptography, but from weaknesses in configuration, key management, hardware protection, and system integration. A system-wide perspective is therefore essential when evaluating or deploying DESFire-based access control solutions.

7. Acknowledgments

This work was supported by the Czech Technical University (CTU) grant No. SGS23/208/OHK3/3T/18. This work was supported by the grant ACDRC (CyberSecurityHub) Embedded AI Processor for Automotive Applications (ID: 75311).

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5 (ChatGPT) and Grammarly in order to: Grammar and spelling check, Improve writing style, transfer Figure 1 from *DrawIO* into *tikz*, as well as for Abstract and title drafting. After using these services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] G. de Koning Gans, J.-H. Hoepman, F. D. Garcia, A Practical Attack on the MIFARE Classic., in: CARDIS, volume 8, Springer, 2008, pp. 267–282.

[2] M. Daley, Gallagher research, GitHub repository, https://github.com/megabug/gallagher-research, 2020.

[3] J. A. Segui, A research on how metro de madrid nfc cards works., https://github.com/jalbertseg/Mifare-Desfire, 2025. Accessed: 2025-08-31.

[4] X41 D-SEC GmbH, Advisory X41-2021-003: Telenot complex - Insecure AES Key Generation, 2022. URL: https://x41-dsec.de/lab/advisories/x41-2021-003-telenot-complex-insecure-keygen/, accessed: 2025-08-31.

[5] M. Daley, Access Control on Sesame Street — Gallagher/Cardax access control system research, Presentation at KawaiCon 2019, https://www.youtube.com/watch?v=brhXqyidiKo, 2019.

[6] D. Oswald, C. Paar, Breaking Mifare DESFire MF3ICD40: Power analysis and templates in the real world, in: Cryptographic Hardware and Embedded Systems–CHES 2011: 13th International Workshop, Nara, Japan, September 28–October 1, 2011. Proceedings 13, Springer, 2011, pp. 207–222.

[7] D. Celiano, Overclocking proximity checks in contactless smartcards, Master's thesis, University of Cambridge, 2018.

[8] D. Hurley-Smith, J. Hernandez-Castro, Bias in the Mifare DESFire EV1 TRNG, in: G. P. Hancke, K. Markantonakis (Eds.), Radio Frequency Identification and IoT Security, Springer International Publishing, Cham, 2017, pp. 123–133.

[9] K. Soules, Analysing the vulnerabilities of commercially available rfid tags, Master's thesis, University of Kent: School of Computing, 2016.

[10] MIFARE, NXP, 2023. URL: https://www.mifare.net/en/, [cit. 2023-01-24].

[11] Multi-purpose RFID ICs, LEGIC Identsystems Ltd, 2022. URL: https://www.legic.com/products/smartcards/legic-smartcard-icss, [cit. 2022-12-27].

[12] Digital Identity and Security, Thales, 2023. URL: https://www.thalesgroup.com/en/markets/digital-identity-and-security, [cit. 2023-03-07].

[13] MIFARE Ultralight, NXP, 2023. URL: https://www.nxp.com/products/rfid-nfc/mifare-hf/mifare-ultralight, [cit. 2023-01-24].
[14] Salto inspired access, SALTO Systems, 2022. URL: https://saltosystems.com/en/, [cit. 2022-12-27].
[15] Gallagher security, Gallagher, 2022. URL: https://security.gallagher.com/en-GB, [cit. 2022-12-27].
[16] ISO/IEC, ISO/IEC 14443-3:2011 – Identification cards – Contactless integrated circuit cards – Proximity cards – Part 3: Initialization and anticollision, 2011. URL: https://www.iso.org/standard/50942.html.
[17] Application note AN10922 Symmetric key diversification, Rev. 2.2 — 2 July 2019, NXP, 2019.
[18] A. Laurie, Atmel SAM7XC Crypto Co-Processor key recovery (with bonus Mifare DESFire hack), 2013. URL: https://adamsblog.rfidiot.org/2013/02/atmel-sam7xc-crypto-co-processor-key.html, accessed: 2025-08-31.
[19] T. Hobson, Honey, the kids tried crypto, Talk at CHCon 2023, Ngaio Marsh Theatre, Christchurch, New Zealand, 2023. URL: https://pretalx.com/chcon-2023/talk/GVCBPN/.
[20] R. Flynn, An Investigation of Possible Attacks on the MIFARE DESFire EV1 Smartcard Used in Public Transportation, PhD thesis, University of Cambridge, 2019. URL: https://www.researchgate.net/publication/344479867_An_investigation_of_possible_attacks_on_the_MIFARE_DESFire_EV1_smartcard_used_in_public_transportation. doi:10.13140/RG.2.2.19591.42401.
[21] Z. Kfir, A. Wool, Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems., IACR Cryptology ePrint Archive 2005 (2005) 52.
[22] G. P. Hancke, Practical Attacks on Proximity Identification Systems (Short Paper)., in: IEEE Symposium on Security and Privacy (SP), 2006, pp. 328–333.
[23] W. Issovits, M. Hutter, Weaknesses of the ISO/IEC 14443 protocol regarding relay attacks, in: 2011 IEEE International Conference on RFID-Technologies and Applications, IEEE, 2011, pp. 335–342.
[24] Atmel Corporation, Atmel SMART SAM7XC512/256/128 Datasheet, 2015. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-6209-32-bit-ARM7TDMI-Microcontroller-SAM7XC512-SAM7XC256-SAM7XC128_Datasheet.pdf, accessed: 2025-06-28.
[25] Y.-J. Tu, S. Piramuthu, On addressing RFID/NFC-based relay attacks: An overview, Decision Support Systems 129 (2020) 113194.
[26] R. Silberschneider, T. Korak, M. Hutter, Access without permission: A practical RFID relay attack, 2013.
[27] L. Francis, G. P. Hancke, K. Mayes, K. Markantonakis, Practical NFC Peer-to-Peer Relay Attack Using Mobile Phones., in: Workshop on RFID and IoT Security (RFIDSec), volume 2010, 2010, pp. 35–49.
[28] S. Klee, A. Roussos, M. Maass, M. Hollick, Nfcgate: Opening the Door for NFC Security Research with a Smartphone-Based Toolkit., in: Workshop on Offensive Technologies (WOOT), 2020.
[29] P.-H. Thevenon, O. Savry, S. Tedjini, On the weakness of contactless systems under relay attacks., in: International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2011, pp. 1–5.
[30] D. Ma, N. Saxena, A contextaware approach to defend against unauthorized reading and relay attacks in RFID systems, Security and Communication Networks 7 (2014) 2684–2695.
[31] D. Ma, N. Saxena, T. Xiang, Y. Zhu, Location-Aware and Safer Cards: Enhancing RFID Security and Privacy via Location Sensing., IEEE Transactions on Dependable and Secure Computing 10 (2013) 57–69.
[32] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, P. Nurmi, Comparing and fusing different sensor modalities for relay attack resistance in Zero-Interaction Authentication., in: Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), 2014, pp. 163–171.
[33] A. Abidin, E. Marin, D. Singelée, B. Preneel, Towards quantum distance bounding protocols, in: G. P. Hancke, K. Markantonakis (Eds.), Radio Frequency Identification and IoT Security, Springer International Publishing, Cham, 2017, pp. 151–162.
[34] S. Brands, D. Chaum, Distance-Bounding Protocols, Springer Berlin Heidelberg, 1994.
[35] G. P. Hancke, M. G. Kuhn, An RFID Distance Bounding Protocol., in: Security and Privacy in

Communication Networks (SecureComm), 2005, pp. 67–73.
[36] Y. Tu, RFID Distance Bounding Protocols, First International EURASIP Workshop on RFID Technology (2007) 67–68.
[37] C. H. Kim, G. Avoine, F. Koeune, F.-X. Standaert, O. Pereira, The Swiss-Knife RFID Distance Bounding Protocol., in: International Conference on Information Security and Cryptology (ICISC), 2008, pp. 98–115.
[38] Kasper & Oswald, ChameleonMini, GitHub repository, https://github.com/emsec/ChameleonMini, 2014-2024.
[39] J. Clulow, G. Hancke, M. Kuhn, T. Moore, So near and yet so far: Distance-bounding attacks in wireless networks, 2006, pp. 83–97. doi:10.1007/11964254_9.
[40] G. P. Hancke, M. G. Kuhn, Attacks on time-of-flight distance bounding channels., in: Conference on Security and Privacy in Wireless and Mobile Networks (WISEC), 2008, pp. 194–202.
[41] NXP Semiconductors, MIFARE SAM AV3 Secure Access Module, Technical Report, NXP Semiconductors, 2024. URL: https://www.nxp.com/docs/en/data-sheet/MF4SAM3X_S.pdf.