Assessing the NFC Unlock Mechanism of the Tartu Smart Bike Share System

Abasi-amefon O. Affia

Institute of Computer Science,
University of Tartu, Tartu, Estonia,
amefon.affia@ut.ee

Supervised by Danielle Melissa Morgan

December, 2019

Abstract. This report, prepared for the course *Research Seminar in Cryptography (MTAT.07.022)*, assesses the unlock mechanism of the Tartu smart bike share system, a sustainable transportation initiative. It focuses on the NFC unlock mechanism, implemented for this smart bike-sharing system. In this paper, we employ a black-box approach to analyse the Tartu bus card interaction with the smart bike for the unlock procedure, understand its implementation, show possible security vulnerabilities within its implementation, and show how these vulnerabilities can be exploited. We provide security suggestions for the use of the Tartu bus card with the Tartu smart bike share system. Although other unlock alternatives can be used with this system, we focus on the official bus card for public transportation in Tartu – the Tartu bus card.

1 Introduction

Sustainable transportation through smart bike-sharing concepts are rapidly being introduced in European cities for daily mobility [5], and the city of Tartu is no exception. The development of a bike-sharing system has been one of the mobility priorities of Tartu, Estonia. The Tartu smart bike-sharing program[1] – considered a part of the public transport system – provides a considerable alternative to cars and brings about a decrease in environmental problems, parking issues, and traffic intensity issues. These smart bikes, as seen in Figure 1, typically include smart sensors (i.e. global positioning system (GPS) receiver, RFID sensors) and network capabilities (i.e. 4G) for necessary system component communication to make decisions. A vital part of this system requiring communication to achieve its function is the unlock mechanism of the Tartu smart bike share. This unlock mechanism carries out the necessary authentication and authorisation communication to grant the user access to the smart bikes. Tartu smart bike provides two forms of this unlock mechanism – unlock mechanism using Tartu smart bike mobile app and unlock mechanism using NFC smart card.

[1] https://ratas.tartu.ee/

Security in a bike share unlock mechanism is important. An attacker could attack the bike-sharing service to manipulate access operations. Authentication is essential so that the bike is unlocked only by the person who paid for the service. With authentication, the communicating parties can trust that they are speaking to the legitimate counter-party, rather than a malicious third party, masquerading as the legitimate party. Thus, unlocking the smart bikes require proper user authentication – to correctly identify an individual user before granting access to the smart bikes.

With the introduction of the Tartu smart bike program as a form of public transportation, a provision was made to use the already existing form of validation for public transportation to access/unlock the smart bikes. The Tartu bus card, widely used since 1st September 2015[2] is implemented especially for user validation on the Tartu local public transportation (i.e., shuttle bus). As such, the Tartu smart bike share system incorporates this form of user validation before granting access to its smart bikes.

The Tartu public transportation system also allows for the use of Tallinn bus cards for validation. As such, a personalised Tallinn bus card with an active ticket can be supplied to unlock the Tartu smart bike successfully. However, we will focus on the Tartu bus card for further analysis. We analyse the Tartu bus card and its interaction with the smart bikes during the unlock procedure to highlight possible vulnerabilities that could lead to possible attacks or attack scenarios and provide possible solutions.

Fig. 1: Tartu smart bike connected to the docking station

[2] `https://www.tartu.ee/en/tartu-bus-card`

2 Unlock mechanisms for Tartu Smart Bike Share System

When looking into information online about the Tartu smart bike share system, there was little to no technical information explicitly provided about the system or how its unlock mechanism works. Thus, we use a black-box approach to analyse this system by observing the mechanism in use and reviewing similar implementation use-cases in literature [2]. As previously mentioned, there are two ways to unlock the Tartu smart bike – using Tartu smart bike mobile app unlock mechanism, or Tartu smart bike NFC unlock mechanism.

2.1 Tartu smart bike mobile app unlock mechanism

The Tartu smart bike mobile app[3] provides a simple module called "Unlock Bike" where the user inserts a five-digit bike number to unlock the bike as seen in Figure 2. This unlock mechanism requires an internet connection and a valid user subscription. To use the mobile app, the user has to authenticate using his email address and password.

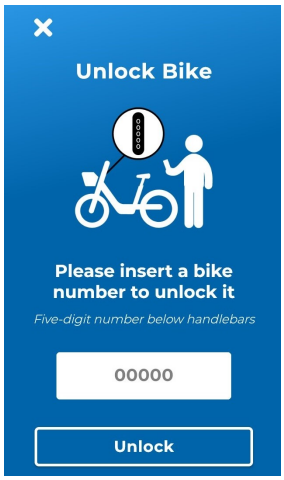


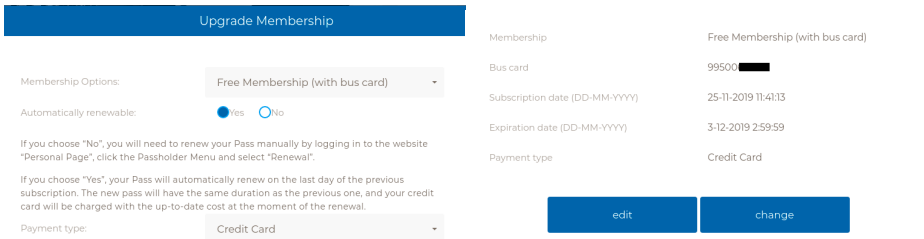
Fig. 2: Tartu smart bike mobile app unlock mechanism

2.2 Tartu smart bike NFC unlock mechanism

This type of unlock mechanism strongly supports the integration of the Tartu smart bike share system as a form of public transportation. The NFC unlock mechanism enables a contact-less interaction between a Tartu bus card (or Tallinn bus card) and the smart bike to unlock the smart bike. As we focus on the Tartu bus card interaction with the smart bike, we will not explore other NFC cards supported by the NFC unlock mechanism.

[3] `https://apps.apple.com/us/app/tartu-smart-bike/id1458482220`

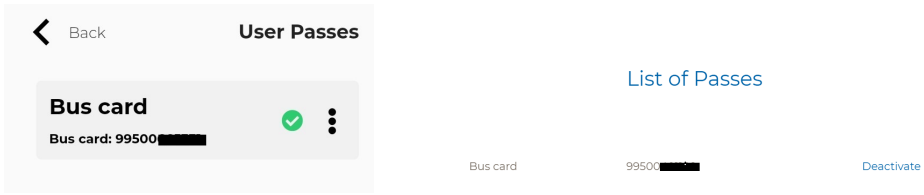
Linking bus card information. The Tartu bus card is linked to the smart bike user account using the 11-digit Tartu bus card number that is printed on the back of the card (see Figure 6b). The bus season ticket must be activated to use a bus card for the smart bike share system. This way, the user gets a bike share membership. We purchased a seasonal 10-day ticket[4] and activated this during the first validation on a public transportation bus ride. After bus validation, we can now link the card to the Tartu smart bike users profile[5]. To link the card, we obtained a user account and upgraded membership to use bus card, as seen in Figure 3a. Figure 3b shows a successfully linked bus card.



(a) Linking card to user profile (b) Linked bus card information

Fig. 3: Tartu smart bike user profile: Add bus card to user profile

Once we completed linking, we could also view a list of linked bus cards on both the mobile app and website (see Figure 4).



(a) Linked bus card on mobile app (b) Linked bus card on website

Fig. 4: List of passes

Bike unlock procedure. The smart bike is equipped with contact-less card reader to read the information needed to identify the user from the Tartu bus card. The reader is located in the centre of the bike handlebar as shown in Figure 5. When a user positions the bus card in proximity of the reader's antenna, the

[4] `https://tartu.pilet.ee/buy`
[5] `https://ratas.tartu.ee/users/my-account/`

high-speed RF communication interface allows the transmission of the data with the bus card.

Fig. 5: Smart bike handlebar with built-in NFC card reader (in the center)

The NFC unlock procedure can thus be seen as a data exchange scenario initiated by tapping the Tartu bus card – a contact-less card, on the bike card reader – located in the centre of the handlebar to release the remotely controlled locks of the bike. The lock mechanism of the smart bike share system is a dock-based solution where bikes are locked to the docks – special bike racks located at specified stations (or secondary locks attached to the smart bike) – when the bike is returned. The unlock mechanism releases the smart bike following verification of user information stored in the Tartu smart bike share central database.

3 Tartu bus card

The Tartu bus card shown in Figure 6 is a MIFARE Ultralight C Near-Field Communication (NFC) card used by the Tartu public transportation system to validate rides on Tartu public transport. Near-Field Communication (NFC) cards are a branch of High-Frequency (HF) radio Frequency Identification (RFID)[6] technology that both operate at the 13.56 MHz frequency, and are designed to carry out secure data exchange [1]. The utilisation of Near Field Communication Technology (NFC) is rapidly widespread [8] and regarded as an effective, secure and convenient solution for user identification and access control.

The MIFARE Ultralight C smart card is widespread in the transport field because of its low price, easy implementation, and provided security. In principle, the MIFARE Ultralight C card is a 1536-bit EEPROM memory card with extra functionalities of read, write, increment and decrements, capable of transferring data with a rate of 106 Kbit/s. The MIFARE Ultralight C smart card also contains a 7-byte serial number or unique identifier (UID) with anti-cloning support.

[6] RFID technology enables identification from a distance using radio waves.

(a) Front. (b) Back.

Fig. 6: Sample Tartu bus card

3.1 Authentication protocol

The 14443 standard does not take into account security. Thus to ensure secured data exchanges, the MIFARE Ultralight C [7] adds the MIFARE authentication protocol, integrating the 2-key 3-DES encryption unit to the 14443 standard.

An authentication process is used by the MIFARE Ultralight C tag to verify that both entities – the reader and tag, hold the same secret and can be seen as a reliable partner for onward communication.

The secure MIFARE Ultralight C authentication protocol algorithm generally works, as shown in Table 1 adapted from [7]. The NFC reader is always the entity that starts an authentication procedure by sending the AUTHENTICATE command "AFh" to the smart card. The card generates an 8 byte random number $RndB$ encrypted with the key and denoted by $ek(RndB)$. This is then transmitted to the reader. The reader itself generates an 8 byte random number $RndA$ which is concatenated with $RndB'$ and encrypted with the key $ek(RndA||RndB')$. $RndB'$ is generated by rotating the original $RndB$ left by 8 bits. This token "AFh"|| 16 bytes $ek(RndA||RndB')$ is sent to the card. The card decrypts the received token to retrieve $RndA + RndB'$ and now verify the sent $RndB'$ by comparing it with the $RndB'$ obtained by rotating the original $RndB$ left by 8 bits internally. A successful verification proves to the card that both the card and the reader possess the same secret key. As the card also received the random number $RndA$, generated by the reader, it can perform a rotate left operation by 8 bits on $RndA$ to gain $RndA'$, which is encrypted again, resulting in $ek(RndA')$. This token "$00h$"|| 8 bytes $ek(RndA')$ is sent to the reader. If the verification fails, the card stops the authentication procedure and returns an error message. The reader decrypts the received $ek(RndA')$ and thus gains $RndA'$ for comparison with the reader-internally rotated $RndA'$. If the comparison fails, the reader exits the procedure and may halt the card. Finally, the card sets the state to authenticate. In the AUTHENTICATED state, READ and WRITE commands may now be performed to memory areas, that are readable or write-able [7,3].

Table 1: MIFARE Ultralight C Authentication example (adapted from [7])

	PCD	Data exchanged	Smart card
1	start	$->1Ah$	
2		$<-AF577293FD2F34CA51$	$generateRndB$ $IV = 0000000000000000$ $ek(RndB) = 577293FD2F34CA51$
3	generate $ek(RndA\|\|RndB')$	$->AF0A638559FC7737F9F15D7862EBBE967A$	
4		$<-003B884FA07C137CE1$	$RndA' = AF3B256C75ED40A8$ $IV = F15D7862EBBE967A$ $ek(RndA') = 3B884FA07C137CE1$
5	decrypt and verify $ek(RndA')$		
6			Card sets the state to AUTHENTICATE

3.2 Enabled security features

While there are no publicly known practical attacks against the MIFARE Ultralight C cards secure authentication performed using 3-DES [7], from the prior research we know that the provided security features have not been fully utilized on the Tartu bus card (see Section 3.6.2 in [6]). We describe these observations below.

The contents of the bus cards were first scanned with the NXP TagInfo mobile application[7] to discover preliminary information such as the integrated circuit (IC) information (IC manufacturer, IC type, NFC tag type), NFC Data Exchange Format (NDEF) information, the access conditions, memory size, and the authentication information.

We see the contents of the Tartu Bus card, including its default keys, card number, card UID, OTP bytes, and signature data. By further analysing authentication type configurations, we observed that the OTP area bytes in page 3 (`0xE1101200`), show the card as being in the initialised state while all lock bytes on page 40 are set to `0x00`. Also, page 41 shows its counter value to be 0.

The above observations show two closely related vulnerabilities. First, the OTP bytes indicates that page values are writable, and block locking has not been enabled (i.e., where lock bytes on page 40 are set to `0x00`). As such, write operations can be performed on the card (see Figure 7). Second, the current authentication configuration indicates that memory protection has been disabled, and there are no protection mechanisms because the default, publicly-known authentication key is used. These vulnerabilities of the Tartu bus card can be exploited to create a cloned Tartu bus card to unlock the smart bike.

[7] `https://play.google.com/store/apps/details?id=at.mroland.android.apps.nfctaginfo&hl=en`

Fig. 7: Access condition for Tartu bus card

3.3 Memory structure

A memory dump of the card information was then extracted using the ACR112U USB NFC reader (see Figure 8) and a Python script. The contents of the memory dump were further analysed.

Fig. 8: ACR112U USB NFC reader

Figure 9 shows sample data extracted from the memory dump, including the card's UID, the signature data, the signature value, access conditions, and authentication key. The UID of the card as seen in page 0 and 1 is `0x0497342FC2833F80`. Details about the card signature start on page 4 with the first byte (`0x03`), indicating the presence of an NDEF and extends to page 38. Data from the second byte on page 5 (`0x70`), extending to the third byte on page 21 (`0x80`) is incorporated in signature data, thus providing integrity against modifications. The

signature data between page 5 byte `0x70` and page 21 byte `0x80` are the card type "pilet.ee:ekaart:3" (page 5 byte `0x70` until page 9 byte `0x33`), the card number 99500382283 (page 16 byte `0x39` until page 19 byte `0x31`), as shown on the back of the card in Figure 6b. The signature data starts on page 25, with the first byte `0x36` indicating the length of the field, and extends to page 39. Other interesting values include the one time programmable (OTP) area bytes on page 3 (`0xE1101200`), the card counter value on page 41, and default values on page 42 and page 43.

page	B1 B2 B3 B4	ASCII		page	B1 B2 B3 B4	ASCII
0	04 97 34 2F	..4/		1	C2 83 3F 80	..?.
2	FE 48 00 00	.H..		3	E1 10 12 00	....
4	03 8B 94 11	....		5	35 70 69 6C	5pil
6	65 74 2E 65	et.e		7	65 3A 65 6B	e:ek
8	61 61 72 74	aart		9	3A 33 66 0F	:3f.
10	5F 26 06 31	_&.1		11	35 30 38 33	5083
12	31 59 04 20	1Y..		13	20 20 20 6E	...n
14	22 5A 13 33	"Z.3		15	30 38 36 34	0864
16	39 30 30 39	9009		17	39 35 30 30	9500
18	33 38 32 32	3822		19	38 33 53 07	83S.
20	04 97 34 C2	..4.		21	83 3F 80 54	.?.T
22	02 00 01 51	...Q		23	03 3C 53 69	.<Si
24	67 01 04 00	g...		25	37 30 35 02	705.
26	19 00 82 76	...v		27	C7 18 46 5D	..F]
28	5C B4 CC 9C	\...		29	4F 84 BB 97	O...
30	A8 FC BB CE	....		31	AB 6D E1 7F	.m..
32	CB 16 02 18	....		33	1D A6 92 C7	....
34	70 BB D4 CB	p...		35	53 3D 82 D7	S=..
36	4F 81 F5 06	O...		37	09 C6 BC 56	...V
38	CD 9F 96 05	....		39	46 41 49 4C	FAIL
40	00 00 00 00	....		41	00 00 00 00	....
42	30 00 00 00	0...		43	00 00 00 00	....
44	Key 1 page 0			45	Key 1 page 1	
46	Key 2 page 0			47	Key 2 page 1	

Fig. 9: Memory dump from a Tartu bus card

4 Implementing Tartu bus card cloning attack

We introduced a test MIFARE Ultralight C Tartu bus card to implement the cloning attack. This test card holds no active seasonal subscription, neither is it linked to a user on the Tartu smart bike share app. Following the hypothesis that the Tartu bike share system checks only the card number to unlock the Tartu smart bike, we decided to modify only the memory value storing the bus card number to match the bus card number of the victim's card.

We first create a memory dump of the test card information to view the memory information. A Python script was used to perform write operations on the card memory fields (page 16 byte `0x39` until page 19 byte `0x31`). In this script, we use the `pyscard` Python smart card library to communicate with the MIFARE Ultralight C card (Tartu bus card). The script first verifies the card

type, initiates a connection when the card is placed in the NFC card reader, then executes the specified write command on the card. The card number is specified in the Python script in hex format for its corresponding memory pages.

Using the ACR112U NFC reader and a Python write script, we were able to successfully overwrite the card number of the test card with the card number of a linked and validated bus card.

We attempted to unlock the bike by tapping the test card against the bike card reader. This attack was successful (see Figure 10), granting us access to the smart bike and thus confirming the hypotheses mentioned above.

Fig. 10: Successful unlock of bike with Test card

Ride receipts were sent to the validated cardholder once the bike was docked, showing that the Tartu smart bike share system registered this ride as a legitimate event. Thus, we conclude that with the knowledge of the victim's Tartu bus card's number, it is possible to create a fake bus card that will allow the attacker to impersonate the victim.

We also observed that cloning the whole memory block of the victim's card did not result in a successful unlocking of the bike. This means that the NFC reader of the bike verifies that the UID under the signature matches the uncloneable UID of the card. This of course does not provide security, because NFC reader does not check the signature and hence is not able to verify the integrity of the UID contained in the signed data block.

4.1 Automated exploitation

The attack scenario described above assumes the ability of the attacker to somehow learn the card number of the victim's Tartu bus card (e.g., by visually inspecting victim's bus card). In this section, we highlight automated exploitation scenario where for a successful attack the attacker does not have to obtain the card number of particular victim's Tartu bus card.

Following the same principles of the described attack, an attacker can perform an automated exploitation attack. This is a brute force type attack that exploits the vulnerability of the Tartu bus card number being serial numbers and where an attacker can check validity information online using the official bus ticket site[8] without authentication. Work has already been done by Martin Paljak[9] to automatically generate bus card numbers and check the type of pass on the card, thereby increasing the attack likelihood. Once a list of valid tickets is collated, appropriate tools can be used to run through these card numbers until a valid user card number is reached, and the bike is unlocked.

While in these experiments we used an USB NFC reader and a computer program to write data onto the Tartu bus card, we note that since majority of modern smart phones have built-in NFC readers, the automated attack could be made user-friendly by implementing it as a mobile application.

5 Possible countermeasures

Countermeasures exist to mitigate the attack discussed. This includes enabling the security features of the MIFARE Ulralight C card, using only the mobile app to unlock the bikes, checking the card signature or in addition, checking the UID of the card.

1. **Enable security features of MIFARE Ultralight C card.** The MIFARE Ultralight C card includes a strong authentication feature and a field-programmable read-only locking function per page for the first 512-bit and a read-only locking per block for the memory above 512 bit. An implementation of this is seen in the Rimi card, which is also a MIFARE Ultralight C card. Figure 11 shows the access conditions for the Rimi card. The Rimi card requires authentication to read and write the memory fields (see page 16 byte `0x39` until page 19 byte `0x31`), unlike the Tartu bus card. By enabling read-only locking and requiring authentication to access these blocks, we can completely prevent risks brought about by the analysed vulnerabilities. However, for this countermeasure to be effective, the whole Tartu bus card system and bike share system has to be upgraded to accept only the reissued Tartu bus cards with these locking and authentication features enabled. From a key management perspective, the card authentication system will have to store the symmetric secret authentication key, which must not leak not to compromise the security.
2. **Use only the mobile app.** The Tartu smart bike mobile app provides bike unlocking over the internet using an authenticated user profile, thus, avoiding the described risks of using the Tartu bus card unlock mechanism. From the user's perspective, however, the use of the NFC bus card to unlock the bike is more convenient than entering the bike number in the mobile

[8] `https://www.pilet.ee/viipe/uhiskaart/activetickets`
[9] `https://github.com/martinpaljak/yhiskaart/tree/gh-pages/py`

application. Thus, disabling the NFC unlock mechanism will degrade user experience.

3. **Checking the card signature.** If the current configuration of Tartu bus card has to be used, an additional check could be to verify the card signature. Checking the card signature prevents the attacker from trivial attack where only the Tartu bus card number is written on the cloned card, requiring the attacker to obtain over the NFC full memory dump of the victim's card. This countermeasure would complicate the current attack and prevent automated exploitation since guessing a valid card number would not be enough to clone the card.
4. **Checking that UID of the card.** In addition to checking the card signature, a check that the unique 7-byte serial number (UID) of the card matches the UID under the signature data, would make the attack considerably harder. This is because the UID of the blank Tartu bus card cannot be overwritten, requiring the attacker to use special-purpose hardware to emulate MIFARE Ultralight C card with specific UID. This countermeasure, combined with checking the card signature, would provide an equivalent security level as provided by the current use of the Tartu bus card for authenticating rides in the public transport[10].

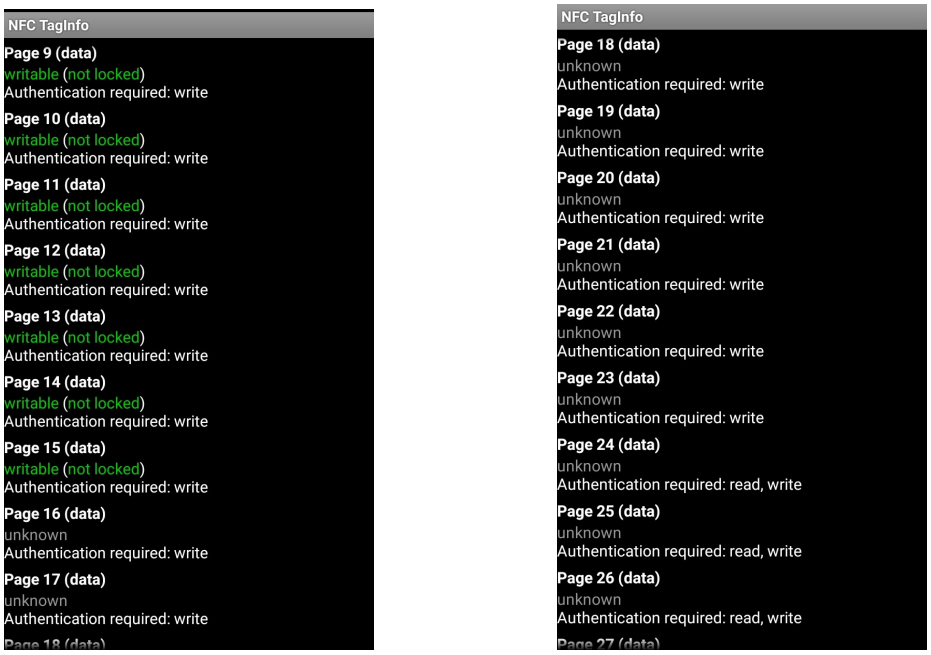
6 Future work

Another interesting element is the possibility to use other NFC cards currently used for public transportation (e.g., a personalised Tallinn bus card) to unlock the smart bike. This introduces an additional attack vector due to the nature of each of these cards. The Tallinn bus card, especially, is a MIFARE Classic 1k card [6], with the vulnerabilities discussed in this work, and several other vulnerabilities and known practical attacks [4,9]. This increases the threat landscape of the Tartu smart bike therefore the security of NFC unlock mechanism using these cards should also be assessed.

7 Responsible vulnerability disclosure

We have shared our findings with Tartu City Government on December 19, 2019 and all security holes are presently being patched.

[10] Note that UID is verified. However, since the signature is not verified, the UID check alone does not add security.



(a) (b)

Fig. 11: Access condition for Rimi MIFARE Ultralight C Card

8 Conclusion

This paper follows a black-box approach to analyse the unlock mechanism of the Tartu smart bike with a focus on the Tartu bus card and its interaction with the smart bikes during the unlock procedure. We highlight the current vulnerabilities of this implementation by examining the card linking process and the card memory information. These identified vulnerabilities led to a successful attack scenario, where with knowledge of a victim's card number, an attacker can unlock the smart bike while impersonating the victim. This attack leads to security and financial consequences on the victims and stakeholders of the Tartu smart bike share system. Countermeasure suggestions for this attack were presented, referring to existing security implementations to be emulated. Other interesting details about the card unlock mechanism were discussed, presenting some avenue for future work. The legal conclusion is that because of such weak authentication security, in the case of a dispute, the users should not be held liable unless there is valid additional evidence supporting the fact that the alleged user unlocked the bike.

References

1. Cerruela García, G., Luque Ruiz, I., Gómez-Nieto, M.Á.: State of the art, trends and future of bluetooth low energy, near field communication and visible light communication in the development of smart cities. Sensors **16**(11), 1968 (2016)
2. Deleenheer, W., Jáneš, L., Jayakumar, A.: Development of an Electric Bicycle for a Sharing System in Prague. Acta Polytechnica CTU Proceedings **12**, 24–31 (2017)
3. Ibrahim, A., Dalkılıc, G.: Review of different classes of RFID authentication protocols. Wireless Networks **25**(3), 961–974 (2019)
4. de Koning Gans, G., Hoepman, J.H., Garcia, F.D.: A practical attack on the MIFARE Classic. In: International Conference on Smart Card Research and Advanced Applications. pp. 267–282. Springer (2008)
5. Midgley, P.: The role of smart bike-sharing systems in urban mobility. Journeys **2**(1), 23–31 (2009)
6. Morgan, D.: Security of Loyalty Cards Used in Estonia. MSc thesis, Tallinn University of Technology (2017)
7. NXP-Semiconductors: MF0ICU2. MIFARE Ultralight C–Contactless ticket IC, Rev **3.3**, 36 (2019)
8. Purwanda, I.G., Adiono, T., Situmorang, S., Dawani, F., Samhany, H.A., Fuada, S.: Prototyping design of a low-cost bike sharing system for smart city application. In: 2017 International Conference on ICT For Smart Society (ICISS). pp. 1–6. IEEE (2017)
9. Tan, W.H.: Practical attacks on the MIFARE Classic. Imperial College London (2009)