Offline Access to a Vehicle via PKI-Based Authentication

Arm, Jakub[1[0000-0001-8290-8255], *], Fiedler, Petr[2[0000-0001-8558-5164]], and Bastan, Ondrej[1[0000-0002-3790-7533]]

[1] FEEC, Brno University of Technology
[2] CEITEC, Brno University of Technology
Jakub.Arm@vut.cz

Abstract. Using modern methods to control vehicle access is becoming more common. At present, various approaches and ideas are emerging on how to ensure the access in use cases reflecting car rental services, car sharing, and fleet management, where the process of assigning car access to individual users is dynamic and yet must be secure. In this paper, we show that this challenge can be resolved by a combination of the PKI technology and an access management system. We implemented a vehicle key validation process into an embedded platform (ESP32) and measured the real-time parameters of this process to evaluate the user experience. Utilizing the SHA256-RSA cipher suite with the key length of 3072 bits, we measured the validation time of 46.6 milliseconds. The results indicate that the user experience is not worsened by the entry delays arising from the limited computing power of embedded platforms, even when using key lengths that meet the 2020 NIST recommendations for systems to be deployed until 2030 and beyond.

Keywords: PKI, X.509, certificate, access management, key management.

1 Introduction

Controlled vehicle access is ensured via various technical solutions. In the consumer domain, wireless technologies based on RF (Radio Frequency) and NFC (Near Field Communication) have prevailed. Currently, cars allow access (unlocking the door and starting the engine) only in the presence of the correct code transmitted from the user's unique mechanical key with an embedded transponder; thus, the user must have such a key. Although this widely used approach is considered secure enough, threats such as replay-attack and relay-attack still need to be addressed. Moreover, the solution exploiting a physical key is not convenient for scenarios like granting access to the company fleet vehicles or car sharing, i.e., scenarios where the physical key needs to be passed from one person to another.

These scenarios require a rather quick transfer of the access, ideally without the need to manipulate with the physical key. In general, solutions based on traditional car keys with an integrated RF transponder in the car sharing sector tend to cause many problems. Multiple challenges need to be solved to ensure a reliable and secure operation of key control management, including that presented [1]:

- Combating unauthorized vehicle usage;
- resolving lost and misplaced car keys;
- keeping track of the keys when the user is an outside contractor;
- reducing the risk if a third party is involved.

1.1 Standard car access methods

A rental/car share (RCS) and other management systems can employ barcodes, QR (Quick Response) codes (or NFC/RFID), GPS, and mobile apps coupled with a wireless network to enable customers to bypass the reservation desk. The backend servers and mobile applications communicate with a lock module integrated in the vehicle to maintain the valid codes (keys) and control the access. Presently, such an approach is patented by [2].

Another patent [3] describes the use of Smart card systems in connection with transportation services; the smart card acts as a user ID that not only controls the access to the vehicle but also manages the driver-specific settings.

The above patents describe a car sharing and fleet management system ensuring support for symmetric cipher technologies (mostly provided by the AES-CBC cipher algorithm), which is prone to various attacks, namely, the replay attack, cipher algorithm attack, brute force, password guessing, and cipher key compromising; such situations occur especially if insufficient key lengths are used (below 128 bits). Current recommendations for future-proof security solutions (2030 and beyond) advise using symmetric encryption algorithms with a key size of at least 128 bits [17]; however, to allow truly long-term protection against mathematical attacks and quantum computing, a key size of 256 bits is sometimes recommended [18].

1.2 Using PKI in the automotive domain

The intention to use PKI (Public Key Infrastructure) in the automotive domain is not new. In this context, for example, let us point to reference [4], which already in 2006 outlined the possibilities of incorporating more advanced security methods, including the application of PKI, in car sharing, fleet management, and various other scenarios. However, as regards car sharing or fleet management, no detailed solution involving an architecture diagram and/or a detailed description of the functionality and performance has been published to date. The PKI-based technology can be advantageously employed in the car access scenario to facilitate access rights validation including offline validation through validating the digital certificate. In general, the most typical scenarios for the use of PKI in the automotive field are:

- V2X communication;
- car or car component authentication;
- secured car access.

The current role of the PKI technology in the automotive environment rests mainly in aiding message security (message authentication) within the V2X communication scenario, where communication between cars or cars and RSUs (Road-Side Units) must be digitally signed to prevent misuse of the system by an attacker or another malicious entity [5].

A generalized scenario for the application of PKI in motoring relies on authenticating the car to the outside world (third parties), in terms of authentication exploiting a unique Vehicle ID assigned to each vehicle by the vehicle manufacturer. Such a functionality is beneficial in the following use cases [6]:

- identifying the vehicle throughout its lifecycle (in repair shops, for example);
- upgrading the software or firmware in the control units of the car;
- ensure secure communication in telematic applications, including vehicle tracking and fleet management;
- securely manage or replace car keys if these are, for example, lost or broken.

Additionally, vehicle authentication is a prerequisite for V2X certificate enrollment. Another scenario employing PKI in the automotive domain is the security of networked devices (ECUs - Electronic Control Units) inside the car [7]. The goal is to prevent potential external and internal attackers from modifying the devices or spoofing the firmware during the firmware upgrade. These tasks require reliable and trusted key management at the level of the in-vehicle communication networks.

1.3 PKI problems and limitations in car access management

The implementation of the PKI technology in vehicle access management also introduces some technical challenges and limitations. The major issues include insufficient computing resources in automotive equipment. Further, we have to consider the challenging automotive qualification process as regards long-term component reliability, and, importantly the fact that the PKI concept itself brings some specific challenges. The distributed authentication can be exploited advantageously in standard situations; however, problems may occur in non-standard situations, e.g., if the key was issued correctly but returned unexpectedly before expiring (e.g., late cancelation of a vehicle reservation). A solution utilizing the standard revocation process is inadequate in some of the scenarios.

The revocation process can be performed via various methods or instruments: the CRL - Certificate Revocation List, CRT - Certificate Revocation Tree, OCSP - Online Certificate Status Protocol, Novomodo, or short-lived certificates. Each of these options has its advantages and drawbacks, such as computational complexity, connectivity dependence, and available memory [8]. Although the OCSP technique appears to be the best choice for embedded devices, it needs permanent Internet connection to remain functional. Regrettably, such connectivity may not be feasible, and offline car access key validation has to be relied on. This lack of connectivity is typical in the underground garage environment or during a DoS (Denial of Service) attack. For the scenario (use case car sharing and fleet management) concerning access to a vehicle, it is, therefore, most advantageous to use the short-lived certificate method, possibly in combination with the OCSP.

By evaluating the computational complexity of the security algorithms providing PKI operations (pairing, request creation, certificate signing, authentication), it has been found [9] that the longest operation rests in the validation of a certificate against a CA certificate or even a certificate chain. In [9], the author states that the best verification times on the Raspberry PI platform are 2.542 seconds in the RSA algorithm with the key length of 1024 bits and 8.905 seconds in the ECC algorithm with the key length of 1024 bits. Verification exceeding 1 second is hardly acceptable for the average user; it is accepted that 1 second corresponds approximately to the limit of the user's flow of thought [19] and thus also the limit of the broadly acceptable time delay while interacting with a vehicle. In bad weather conditions, above all, any noticeable delay is highly unwelcome.

Even when using computing platforms with limited computing power compared to standard platforms, we have to reduce the time required for these operations, or, more concretely, the key validation task.

1.4 Alternative approaches to car access

Importantly, a set of other, more sophisticated, approaches towards key management security for cars are available to resolve the problems above; these options involve, above all, principles on a blockchain basis. According to our survey, however, such approaches are only theoretical or have remained in the prototype phase at best.

The instruments and tools include, for example, the P3KI-based system, or Decentralized Offline Authorization for IoT (Internet of Things) in general. This system is based on the Web-of-Trust concept, using rescinding instead of explicit centralized certificate revocation. In this setup, the nodes ”change their minds” about who they trust and to what degree they do so [20].

Applying Smart Contracts (based on the blockchain theory) to secure car sharing systems allows these schemes to operate without a trusted intermediary. The use of smart contracts deployed on the Ethereum blockchain ensures that full-fledged car sharing functionalities along with various countermeasures to tackle malicious behavior are provided [10]. While this approach is theoretical and has not been employed yet, it embodies one of the possible ways the car sharing solution in the future.

SePCAR is a privacy-enhancing protocol to facilitate car access provision based on the public ledger. It delivers generation, update, revocation, and distribution mechanisms for access tokens to shared cars, together with procedures to solve disputes and to deal with law enforcement requests in, for instance, car accidents. The proof-of-concept implementation shows that SePCAR takes 1.55 seconds to provide car access [11]. We find this concept promising and appreciate the mathematical proof; however, incorporation into a real system necessitates a significantly higher amount of work, mostly in the form of standardization.

Through an extension and continuation of SePCAR, the HERMES tool [12] was designed and evaluated. HERMES securely outsources vehicle access token generation operations to a set of untrusted servers by concealing the secret keys of vehicles and the transaction details from the servers, including the vehicle booking details, access token information, and user and vehicle identities. The tool is built on the multipart computation protocols HtMAC-MiMC and CBC-MAC-AES. The authors suggest that the generation of a token is 42 times faster compared to the previous SePCAR and access token operations performed on a prototype vehicle on-board unit, which take approximately 62.1 millisecond. We assume this is a perfectly acceptable value because, from the user's perspective, delays close to 100 milliseconds are considered negligible and perceived as instantaneous.

In access control, standard ICT techniques already exist, including but not limited to Radius (described in RFC 3579) and Diameter (described in RFC 3588). However, these are intended for advanced access control in intangible objects (service, software). The techniques also define a specific approach to securing the communication, i.e., physical permission or denial of the communication, the actual way of negotiating with a participant, and permanent connection to the server. We, therefore, consider such solutions unsuitable for the vehicle access scenarios.

1.5 The security of the PKI system

PKI is based on X.509 certificates, which exploit one of the following asymmetric cipher algorithms:

- RSA (utilizing the factoring of large numbers, and the RSA problem);
- DSA, El Gamal (based on non-effective solving of the discrete logarithm);
- ECC (elliptic curves).

From the perspective of security, without side-channel attacks, PKI exploits the complexity of the applied cipher algorithm.

In source [13], the authors use a testbed evaluating the performance and energy consumption of Transport Layer Security (TLS) 1.2 ECC (Elliptic Curve) Cryptography and RSA (Rivest-Shamir-Adleman) cipher suites (that comply with the TLS 1.3 standard requirements, whose introduction is currently at the preparatory stage). The results show that ECC outperforms RSA in both energy consumption and data throughput for all of the tested security levels. Moreover, the importance of selecting a proper ECC curve is highlighted.

Certificates according to X.509 version 3 (RFC 3280 and its descendant, the RFC 5280 [21]) standard, are mostly utilized for the authentication and securing of communication. The standard defines the structure of the certificate and processes concerning certificate formation and validation. As the cipher suite, the SHA256-RSA or the SHA256-ECDSA is most often used followed by Base64 encoding. In addition to implementation and architecture issues, this technology also suffers from security flaws relating to the encryption mechanism itself; among the most feared attacks on the cipher algorithm is the creation of a hash collision, where two valid certificates are involved, one of which is forged and potentially dangerous. In addition, if an attacker has a valid CA certificate with the CA attribute, he or she can generate additional certificates based on this spoofed certificate, which will be valid during the authentication process [14].

An analysis of the new standard for secured communication [15] shows that some encryption algorithms (e.g., AES-CBC or MD-5) are no longer supported. Regrettably, to date, the TLS 1.3 standard has not yet been fully implemented and deployed in the embedded world. Therefore, technologies according to the TLS 1.2 standard, which is already known to contain security issues, are still being employed in new devices. The TLS1.3 standard also brings support for the embedded world, i.e., devices with less computing power, and IoT devices, i.e., devices with limited connectivity and higher security requirements. The authors also claim that some vulnerabilities remain present despite the state-of-the-art status of the TLS 1.3.

The validation process of a certificate which complies with the X.509 standard is performed against the ancestor certificate, which itself is validated against its ancestor certificate. This hierarchic queue of certificates is called a chain of certificates [16]. Such mechanism implies that a leaf (client) certificate must be validated against its ancestor certificate (a car sharing company or a fleet management service in our case).

2 The proposed concept

The essence of the proposed approach lies in using the PKI (Public Key Infrastructure) technology based on X.509 certificates to assure the security of the in-vehicle access control. In the car sharing and fleet management scenario, the access is managed by a backend server (Figure 1). In our solution, the key empowers the user only for a short time (short-lived certificates to assure off-line certificate revocation), and the key is linked to the individual vehicle reservation. Furthermore, the reservation cancelation function based on OCSP (Online Certificate Status Protocol) is also implemented in our solution to manage changes in the status of the individual keys (reservations); the backend server uses the OCSP-based revocation; this function, obviously, requires an active connection. When the vehicle is off-line, the short-lived certificate technology is used as a backup to increase the resilience of the solution against a DoS attack at the connectivity.

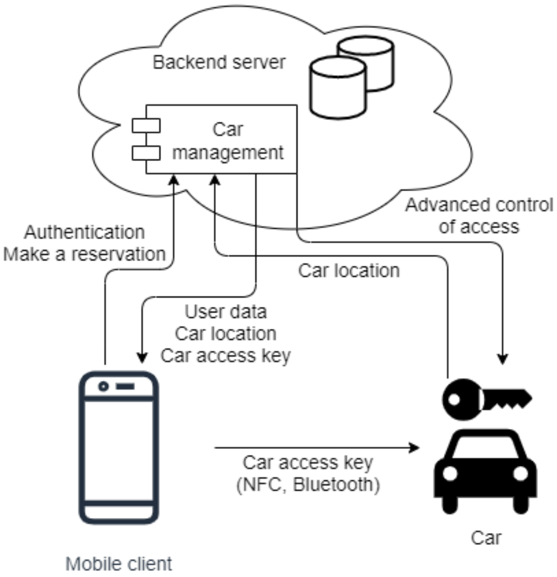

Fig.1. The block diagram of the vehicle access system (car sharing or fleet management).

2.1 Architecture

By adopting PKI for the purposes of the access control system in physical objects, we, map services normally provided by the standard components of the PKI system (RA, VA, CA) to the individual components of the car sharing system. The access device (an in-vehicle ECU controlling the door lock via a CAN communication network) is integrated into the physical object (a vehicle), thus acting as a validation authority (VA). As such, it verifies the validity of the certificates that are presented to it when the user attempts to open the vehicle. Within the certificate, an active reservation with valid parameters is embedded. A mobile user application (installed in the client's phone), i.e., the reservation service, acts as a registration authority (RA) because it ensures the creation of new keys (reservations) with certain parameters (the start and end of the reservation; company name; and, optionally, vehicle designation) by forwarding these requests to the certification authority (CA).

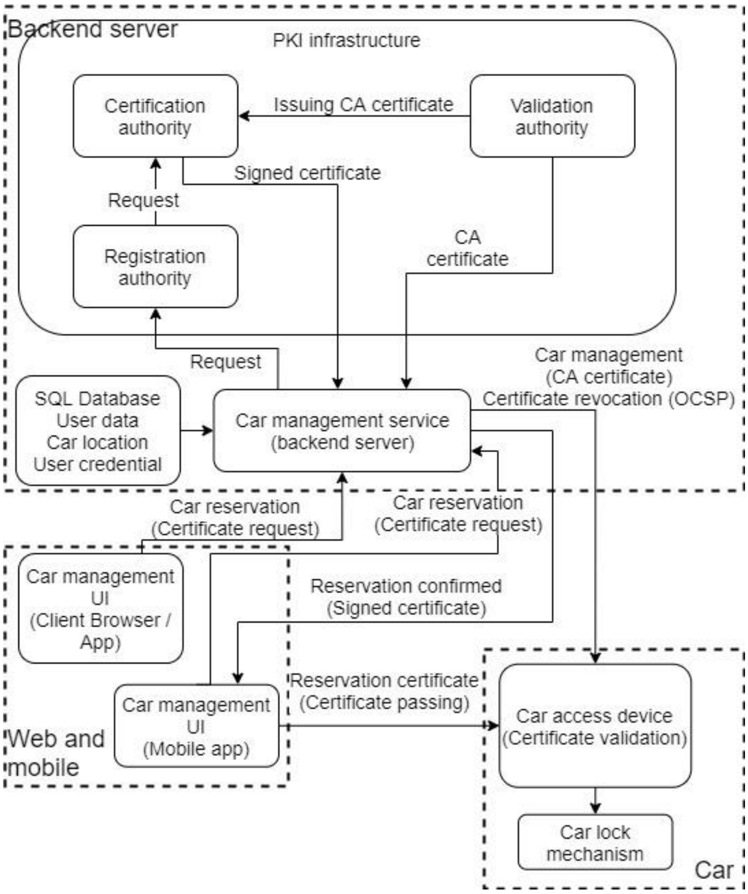
Backend server
PKI infrastructure
Certification authority
Issuing CA certificate
Validation authority
Signed certificate
Request
Registration authority
CA certificate
Request
Car management (CA certificate) Certificate revocation (OCSP)
SQL Database User data Car location User credential
Car management service (backend server)
Car reservation (Certificate request)
Car reservation (Certificate request)
Car management UI (Client Browser / App)
Reservation confirmed (Signed certificate)
Reservation certificate (Certificate passing)
Car management UI (Mobile app)
Web and mobile
Car access device (Certificate validation)
Car lock mechanism
Car

Fig. 2. The block diagram of the proposed car access system, where the PKI technology is incorporated to ensure car key creation and validation.

The advantage of this solution is that the protected physical asset (a vehicle) does not have to be actively connected to the Internet during the key validation process (reservation). This corresponds to situations where, for example, the car is in a garage or a place without Internet connectivity. On the other hand, the business model of the company providing the service must be set appropriately to prevent misuse of the possibility of making reservation just before accessing the object (vehicle) while the object itself is without active connectivity. These situations are outlined at the end of the paper.

Figure 2 describes a diagram of the proposed solution, where the PKI infrastructure ensures the dynamic access control to the vehicle. The system is designed as a server database application in conjunction with a mobile application providing an interface for the customers. The individual components of this system, therefore, take over the individual functions of the PKI technology.

This architecture thus represents a case where the entire PKI-based infrastructure runs and is provided by the operator company (car sharing or fleet management). The second, and perhaps the more frequent case will be the architecture, where the PKI infrastructure, including the root certificates, will be outsourced to a specialized company providing the PKI services. In such cases, the service providers (car sharing or fleet management) will only be intermediaries, i.e., they will maintain intermediate certificates (Figure 3).

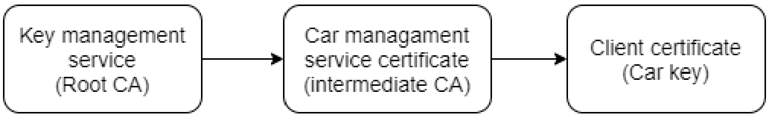
Key management service (Root CA)
Car managament service certificate (intermediate CA)
Client certificate (Car key)

Fig. 3. A certificate chain reflecting the mode in which the key management service can be autonomous, instead of being a part of the car sharing or fleet management system.

2.2 X.509 adaptation

The virtual reservation key is represented by a valid certificate according to the X.509v3 standard (described in RFC 3280). As mentioned earlier, this standard defines not only the process of handling certificates but also their structures. Thus, the certificate (acting as a key to the vehicle) encapsulates parameters such as the beginning and end of the reservation, or, optionally, the object designation in the case of using one root key for multiple devices (vehicles). The user to whom a valid certificate is issued (e.g., a paid reservation) produces this certificate (sent utilizing the short-range technology) at the car access device of the given object, whose access is controlled. To allow usage in the vehicle access control scenario, i.e., in the use case for car sharing and fleet management, the attributes of the certificate are filled in according to Table 1.

Table 1. The attributes modified against the X.509 standard.

Attribute	Description	Meaning in our scenario	Sample data
Version	Certificate version	Unchanged	1

Serial Number	Certificate serial number	Certificate serial number of car-sharing company PKI	2
Signature Algorithm	Cipher suite description	Unchanged	SHA256-RSA
Validity	Certificate validity	The time span which defines the exercisability of the virtual key	
Not Before			May 9 13:30:00 2021 GMT
Not After			May 10 15:00:00 2021 GMT
Subject	Subject of the certificate	Customer (car sharing) or employee (fleet management)	Customer name
Subject Public Key Info		Unchanged	
Public Key Algorithm	Public key algorithm identification	Unchanged	RSA 2048 bit
Public key		Unchanged	Data
Issuer Unique Identifier	(optional)	Not used	
Subject Unique Identifier	(optional)	Not used	
X509v3 extensions	(optional)	Not used	

2.3 Special situations

When using the PKI technology to represent and manage virtual keys for use cases like car sharing and/or fleet management, various situations may arise due to the combination of business models and security technologies that do not meet the limitations of the applied technologies.

Such situations include late cancellation of the reservation of a vehicle, which itself is off-line. In such a situation, the canceled reservation will be considered valid when validated because the car access device has no response from the OCSP server. The possible solutions are:

- Do not accept the reservation cancelation request when the vehicle is parked in a location without connectivity. This solution should be reflected in the business model and service conditions.
- Do not deliver the certificate (create certificate) until the reservation is non-cancelable.

Another situation leading to potential trouble is one where, for some reason, two or more certificates (virtual keys) are valid at the same time. This includes the scenario in which there is a group of people traveling together and thus needing to access the same vehicle. Another scenario that might require concurrent access to the vehicle includes various emergencies, including the need to access the vehicle for unplanned maintenance/service. Therefore, the basic solution described above still has to be enhanced to accommodate for such non-standard uses, e.g,. by using additional attributes that would in more detail define the rights associated with the given certificate (unlock the vehicle, switch on the vehicle, drive the vehicle, service the vehicle, etc.).

2.4 Evaluating the validation time for a certificate

We measured (Table 2) the validation time of the end certificate (virtual vehicle key) against the root certificate on an Espressif ESP32 platform, employing an Extensa 32-bit core @ 240 MHz, 530 KiB SRAM, with a cryptographic HW Accelerator to support SHA-256, AES, RSA, and RNG). We used Mbed TLS libraries with TLS 1.2 version as the internal API for the cryptographic operations (TLS 1.3 is not fully implemented at present). For all measurements, the root and client certificates were already in the ESP32 memory, and the number of levels of the certification chain equals one, i.e., we validated the client certificate against one parent certificate (issued by the car sharing service or fleet management service). During the measurements, all necessary services had already been initialized (performed in the MCU start-up phase).

The executed measurements show the results of the verification of a certificate based on RSA (1024 bits, 2048 bits, 3072 bits, and 4096 bits) and ECDSA exploiting the X9.62 curve (256 bits). Each result was calculated from 1,000 measurements, and therefore the average, maximum, and variance values are provided. For each algorithm, we separately distinguished the length of the encryption key.

Table 2. The measurement of the validation time.

Algorithm	Key length [bits]	Average [ms]	Max [ms]	Variance [ms]
RSA	1024	6.15	10.62	0.04
RSA	2048	21.63	31.88	0.21
RSA	3072	46.61	64.88	0.67
RSA	4096	81.22	108.43	1.48
ECDSA	256	635.15	636.05	0.05

2.5 Limitations of the study

While the incorporation of the PKI technology into the vehicle access system (car sharing or fleet management) was found successful, several limitations arose simultaneously, due in particular to the limitations of PKI technology and associated standards. For example, to ensure the non-repudiation and traceability of the vehicle “ownership” a car key management system (the backend part) cannot be implemented in such a way that the business operator can freely create and pass car keys. This is surely an advantage from the security perspective but also a drawback because there is no full privileged master in the system, capable of solving every special situation that was unaccounted for during the business process design.

In addition, the system is vulnerable to X.509 attacks, i.e., the following issues:

- The private root could be compromised.
- Revocating the root certificate is a challenge.
- The car access rights are aggregated in a single file, implying problems with the car key management (mostly in a mobile app.).
- The root CA cannot prohibit intermediated CAs (car sharing companies) from issuing certificates (car keys) for vehicles they do not own.
- The car key management company (car sharing or fleet management) has to form a long-term certificate to create the public authority imported into every car access device. Before the root certificate expires, the new root certificate must be uploaded in every device. Such a situation can be solved by a secured OTA update.

3 Conclusion

We proposed and defined a car management system architecture, exploiting the PKI technology for the car access scenario. Using the X.509, we defined the attributes of a car access key/certificate and the process of creating, signing, and validating it as incorporated into the car access system. On the ESP32 platform, we performed multiple measurements of the certificate validation process duration. The results show that the validation of the RSA certificate with the key length of 3,072 bits takes only 46.61 milliseconds on the average and 64.88 milliseconds at the maximum. This key length meets the protection level recommendations relating to the period after 2030, as provided by NIST[17] and NSA [18]. Moreover, the results indicate that a solution requiring the validation of multiple certificates to grant access to a vehicle can provide satisfactory short entry delays.

Thus, low computing power does not constitute a problem if we utilize key lengths that provide a high level of security for both the present time and the near future.

Acknowledgment

The research was supported by Brno University of Technology. A portion of the work was carried out under the support of the core facilities of CEITEC – the Central European Institute of Technology. The investigation procedures received funding from the following projects: FV30037 Research and development of new control systems for purchasing platforms, FV40196 Research and development of the monitoring of immobile persons who are tethered to the bed in terms of risk of suffocation – decubitus, FV40247 Cooperative robotic platforms for automotive and industrial applications, TF04000074 Digital representation of Assets as a configurable AAS for OT and IT production systems, TH02030921 The sophisticated wireless network with elements of IoT for plant protection and water management, FEKT-S-20-6205 Industry 4.0 in automation and cybernetics, and 783119-1 SECREDAS Product Security for Cross-Domain Reliable Dependable Automated System (H2020-ECSEL, EU). All of these grants and institutions facilitated the efficient performance of the research and associated tasks.

References

1. Morse Watchmans: Optimizing Fleet Management with Key Control. Available at https://www.morsewatchmans.com/hubfs/Content%20offers/Fleet-Management-Whitepaper.pdf. Last accessed May 2021
2. Enterprise Holdings Inc.: Rental/Car-Share Vehicle Access and Management System and Method. US20140309842A1. Inventors: James E. Jefferies, Rod W. DeMay, Gurgen L. Lachinyan. US patent
3. Smart card systems in connection with transportation services. US20060157563A1. Inventor: David Marshall. US patent
4. Adelsbach A., Huber U., Sadeghi AR. (2006) Secure Software Delivery and Installation in Embedded Systems. In: Lemke K., Paar C., Wolf M. (eds) Embedded Security in Cars. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-28428-1_3
5. Thales eSecurity. Securing the connected vehicle. Available at: http://go.thalesesecurity.com/rs/480-LWA-970/images/ThalesEsecurity_Connected_Vehicle_sb.pdf. Last accessed May 2021
6. Nexus: PKI for vehicle ID. Available at: https://doc.nexusgroup.com/display/PUB/PKI+for+vehicle+ID. Last accessed May 2021
7. Alfred, Jim: Automotive security and trust management: The case for PKI. BlackBerry Technology Solutions, Certicom. Available at: https://www.certicom.com/content/dam/certicom/images/pdfs/wp-automotive%20security%20and%20trust%20management-jul2015.pdf. Last accessed May 2021
8. van Oorschot P.C. (2020) Public-Key Certificate Management and Use Cases. In: Computer Security and the Internet. Information Security and Cryptography. Springer, Cham. https://doi.org/10.1007/978-3-030-33649-3_8
9. Dahlmann, Florian: PKI For Automotive Applications. Bachelor thesis. Supervisor: Prof. Dr. Thomas Eisenbarth. Universitat zu Luebeck, 2018. Last accessed May 2021
10. Madhusudan, Akash. Applying Smart Contracts to Secure CarSharing Systems. Master thesis. Supervisor: Prof. dr. ir. Bart Preneel. KU Leuven, 2018. Last accessed May 2021
11. Symeonidis I., Aly A., Mustafa M.A., Mennink B., Dhooghe S., Preneel B. (2017) SeP-CAR: A Secure and Privacy-Enhancing Protocol for Car Access Provision. In: Foley S., Gollmann D., Snekkenes E. (eds) Computer Security – ESORICS 2017. ESORICS 2017. Lecture Notes in Computer Science, vol 10493. Springer, Cham. https://doi.org/10.1007/978-3-319-66399-9_26
12. Symeonidis, Iraklis & Rotaru, Dragos & Mustafa, Mustafa A. & Mennink, Bart & Preneel, Bart & Papadimitratos, Panos. (2021). HERMES: Scalable, Secure, and Privacy-Enhancing Vehicle Access System.
13. Suárez-Albela M, Fraga-Lamas P, Fernández-Caramés TM. A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security Energy-Efficient Fog and Mist Computing Devices. Sensors. 2018; 18(11):3868. https://doi.org/10.3390/s18113868

14. Kaminsky D., Patterson M.L., Sassaman L. (2010) PKI Layer Cake: New Collision Attacks against the Global X.509 Infrastructure. In: Sion R. (eds) Financial Cryptography and Data Security. FC 2010. Lecture Notes in Computer Science, vol 6052. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14577-3_22
15. Levillain O. (2021) Implementation Flaws in TLS Stacks: Lessons Learned and Study of TLS 1.3 Benefits. In: Garcia-Alfaro J., Leneutre J., Cuppens N., Yaich R. (eds) Risks and Security of Internet and Systems. CRiSIS 2020. Lecture Notes in Computer Science, vol 12528. Springer, Cham. https://doi.org/10.1007/978-3-030-68887-5_5
16. van Oorschot P.C. (2020) Public-Key Certificate Management and Use Cases. In: Computer Security and the Internet, pp. 217-221. Information Security and Cryptography. Springer, Cham. https://doi.org/10.1007/978-3-030-33649-3_8
17. Barker, E.: NIST Special Publication 800-57 Part 1 Revision 5: NIST Special Publication 800-57 Part 1 Revision 5, NIST, May 2020. http://dx.doi.org/10.6028/NIST.Spp.800--57pt1r
18. Commercial National Security Algorithm (CNSA) Suite Factsheet, MFS U/OO/814670-15, NSA, January 2016. https://apps.nsa.gov/iaarchive/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/commercial-national-security-algorithm-suite-factsheet.cfm. Last accessed May 2021.
19. Nielsen, J.: Usability Engineering. Academic Press, San Diego, CA, USA., 1994.
20. Jehle, G.: P3KI Explained: Decentralized Offline Authorization for IoT, version 13, October 2019. P3KI_Explained__Decentralized_Offline_Authorization_for_IoT__1.3.pdf Last accessed May 2021.
21. Cooper, D. at al.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008. Available at: https://datatracker.ietf.org/doc/html/rfc5280. Last accessed June 2021.