Contactless Smart Card as a Cache for Geocaching

Karel Burda
burda@feec.vutbr.cz
Brno University of Technology, Brno, Czech Republic

Summary
In this paper, the possibility of using a contactless smart card as a cache for geocaching is analyzed. Geocaching is an outdoor game in which players search for hidden boxes, or caches based on geographical coordinates. The problems with this game are the possibility of players cheating and the need to maintain the caches. And then there is the problem of the ignorant public accidentally discovering a cache and considering it an explosive device. This paper proposes a concept for a possible solution to the above problems by replacing the boxes with conventional contactless smart cards. Also, this concept makes geocaching more attractive by using various games. This paper proposes a system architecture as well as the cryptographic protocol required for secure communication between the player's smartphone and the card.

Key words:
Contactless smart card, geocaching, cryptography, protocol.

1. Introduction

Geocaching [1] is an outdoor game in which players search for hidden boxes, called caches, based on geographical coordinates. The caches are installed by volunteers at interesting locations, and the information needed to find them is then published on specialized servers (most commonly [2]). Sometimes, instead of the exact coordinates of the cache (the so-called traditional cache [3]), a puzzle is posted (the so-called mystery cache [4]), where the player must first obtain some information to find the coordinates - for example, from history, culture etc.

Players use portable receivers of global navigation satellite systems, such as the GPS (Global Positioning System) or GLONASS (ГЛОбальная НАвигационная Спутниковая Система), to find the cache. Currently, these are usually conventional smartphones with a navigation receiver and a corresponding application (see Fig. 1).

Caches are usually in the form of a box (see Fig. 2), which is hidden in a suitable way (e.g. in the forest in a hollow stump, in a space behind a loose brick, etc.). If the player finds the box, he writes in a diary, which is stored in the box. Alternatively, he can exchange an item of his for another item found in the box. Removing the box, writing and exchanging items should be done in a situation where the player will not be seen by others. This protects the cache from looting and damage. Finally, the player still logs in to the game server and has their find recorded there. The server records and summarizes the results of each player.

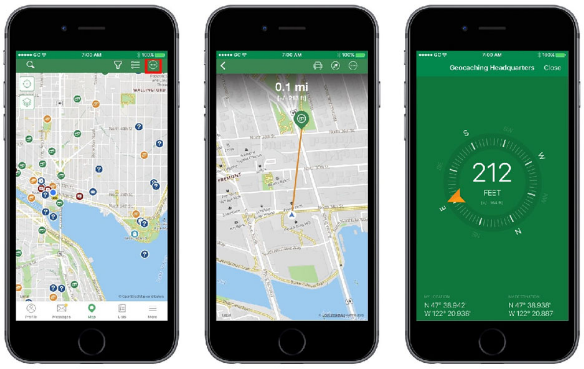

Fig. 1: Screenshots of one of the geocaching applications [2]

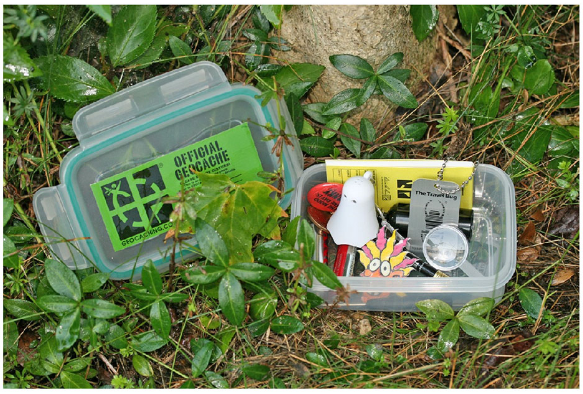

Fig. 2: Example of a cache and its contents [5]

The magic of geocaching lies in the combination of hiking and exploring. Each cache is dedicated to a specific topic, so the player can learn a lot of interesting facts while reading the description of the cache or while solving the puzzle. Along with physical movement, players can also expand their knowledge. Another interesting feature for many players is the possibility of exchanging items such as small toys, pendants, etc. The finder can take one of the items from the cache, but must place an item of comparable value in its place. This option is particularly attractive for children.

Manuscript received July 5, 2021
Manuscript revised July 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.7.24

In addition to the above positives, traditional geocaching faces some challenges. In particular, there is the possibility of players cheating, the problem of cache maintenance, and the public's attitude towards caches. The possibility of cheating consists in the player registering a find on the game server without having found the cache. This takes advantage of the fact that few cache owners compare the data on the game server with the data in the cache diary. Another problem is the maintenance of the cache, where the owner has to replace the described diary and refill pencils in the cache. In relation to the public, the problem of robbing and destroying caches and mistaking them for explosive devices is a major issue. Although the caches are marked with various geocaching logos (see Figure 3), it happens that the ignorant public, upon accidentally finding a cache, can call the police and bomb squad [6] thinking that it is a bomb. Therefore, this paper proposes an original concept for a possible solution to the above problems, which is based on the principle that the box of the cache is replaced by a contactless smart card.

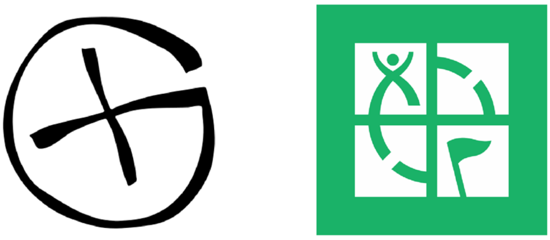

Fig. 3: Frequently used geocaching logos

2. Proposal for a solution

In the proposed solution, the cache does not take the form of a box, but takes the form of a commonly used contactless smart card (see Figure 4). Such a card has a contactless interface according to the ISO/IEC 14443 standard [7] and is equipped with a microcomputer chip. Any active device that is equipped with a more general NFC ("Near Field Communication") type interface [8] can then communicate with the card's microcomputer via this interface. The proposed solution takes advantage of the fact that many modern smartphones are equipped with NFC interfaces.

The NFC interface allows data communication between the card's microcomputer and the player's smartphone over a distance of a few centimeters and also provides power to the microcomputer from the smartphone. The relatively large computing and memory capacity of contemporary cards and smartphones offers the possibility to run various games in addition to cryptographically secure communication. One type of game may be the wandering virtual items around the world and another type may be the collecting virtual items.

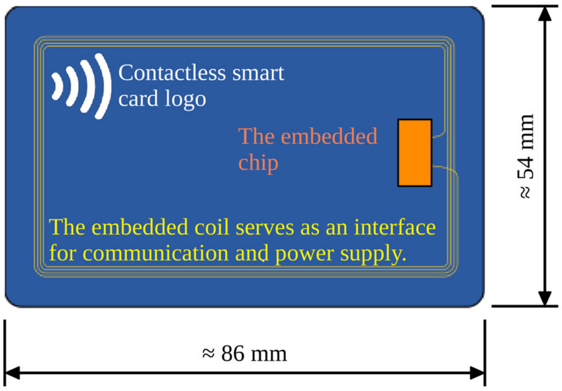
Contactless smart card logo
The embedded chip
The embedded coil serves as an interface for communication and power supply.
≈ 54 mm
≈ 86 mm

Fig. 4: Contactless smart card

In the case of wandering virtual items (see Figure 5), each player P_A has the option to register his wandering item with the GS game server. Typically, this is an image whose unique identifier I_T is sent from the game server to the player's smartphone. When the first cache C_Z is found, this identifier is automatically passed from the player's smartphone to the found cache. The cache C_Z then automatically passes this identifier to its next finder P_B. He then passes it to his next found cache C_Y, etc. The described procedure is repeated with the next finders (P_C, P_D, etc.) and with the next caches (C_X, C_W, etc.). The wandering item will thus virtually travel via the players' smartphones to various caches around the world, which the owner of the item and other players will be able to track via the game server.

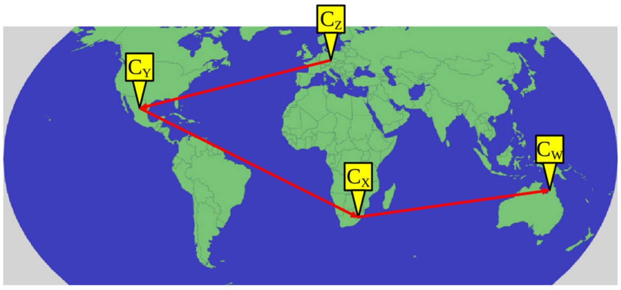

Fig. 5: Wandering virtual items

The second possible type of game is collecting virtual items. In this case, the game server operator defines different puzzles or collector's sets and the player chooses which one he wants to play in his application. With each find of any cache, the player's application then executes one step of the game. For example, if the player chooses to play a puzzle (an example is shown in Figure 6), then with each discovery of a new cache, the game application makes a new puzzle piece available to the player. If the player is collecting a collection of some virtual items (e.g. pictures of important buildings), then when the player finds a cache, the game application will give him access to an item that he does not already have in his collection.

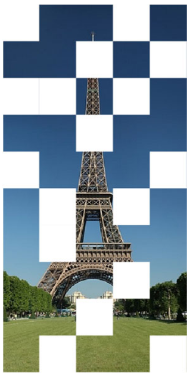

Fig. 6: Collecting virtual objects

2.1 Architecture

The proposed system (see Fig. 7) consists of a certification authority CA, a game server GS, players P, caches C and owners of these caches CO. The commonly used TLS ("Transport Layer Security" [9] and [10], respectively) protocol will be used to ensure the confidentiality and authenticity of the communication between the GS-CA, GS-CO and GS-P pairs.

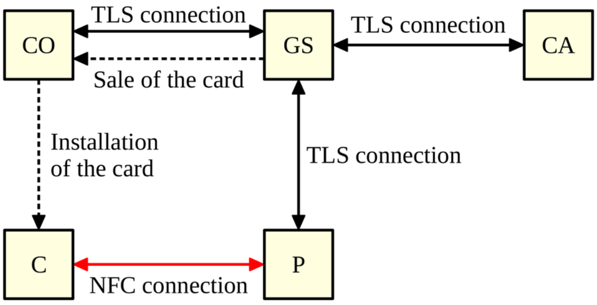

Fig. 7: System architecture

The purpose of the certification authority is to generate public key certificates for all elements of the system. A certificate $CRT(v_X)$ is a message signed by a CA for which party X has the public key v_X, where X may be GS, P, C or the CA itself. The CA uses its private key s_{CA} to sign certificates. The certificate $CRT(v_{CA})$ is available to all elements of the system, and so each element is able to use the key v_{CA} to verify the public key certificate of any other element of the system.

The game server GS mainly acts as a registrar for players and caches. It has generated a pair of private key s_{GS} and a public key v_{GS}. By signing with the private key, it authenticates itself to the other parties in the system, and its public key from the certificate $CRT(v_{GS})$ is used by the other parties to verify the identity of the GS. The game server also acts as a registration authority, so that it receives player requests for certificates, verifies the identity of the players, then sends the received requests to the CA, and finally forwards the generated certificates to the players. In addition, this server also registers and publishes the players' game results.

Each individual player P_A is represented in the system by their game application. The player downloads it from the game server and installs it on his smartphone. As part of this installation, the player also obtains the certificate $CRT(v_{CA})$, so that the application can then use the key v_{CA} to authenticate the certificates of other parties. The application will then generate a private key s_A and a public key v_A for its player. The player then registers on the GS game server with his unique nickname A and obtains the certificate $CRT(v_A)$ of his public key from the CA via GS.

The last elements of the system are the cache and its owner. The future owner CO of the cache first connects to the GS game server using a web browser. Here he registers and orders a card Z for his cache C_Z. The GS server specialists will generate the private key s_Z and the public key v_Z for the card Z, and also provide the certificate $CRT(v_Z)$. Furthermore, these specialists store the identifier Z, the private key s_Z and the certificates $CRT(v_Z)$ and $CRT(v_{CA})$ in the card. The card is then sent by regular mail to its CO owner, who installs it in the field as his cache C_Z. The owner then notifies the game server that his cache has been activated.

Returning to Fig. 7, we see that the communication between the player's application and the cache has not yet been discussed. In order not to cheat here, cryptographic techniques are incorporated into this protocol. Specifically, the player's smartphone application and the cache's microcomputer authenticate each other using digital signatures, whereby the player also receives a microcomputer-signed confirmation of finding the cache. The relevant protocol is described below.

2.2 Cryptographic protocol

The protocol for communication between the player's application and the cache card is designed based on a two-party challenge-response authentication protocol in which digital signatures are used to prove identity ([11], p. 405). This protocol is presented in graphical form in Fig. 8. After finding the cache C_Z with the card Z, the player P_A brings his smartphone closer to this cache. In doing so, a transformer coupling between the two devices ensures the transfer of electricity from the smartphone to the microcomputer in the card. By subsequently influencing the magnitude of the AC supply current in both directions, a two-way data transfer is also ensured between the two devices.

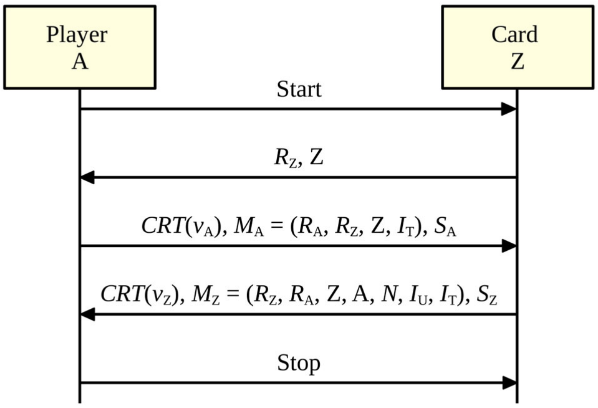
Player A
Card Z
Start
R_Z, Z
$CRT(v_A)$, $M_A = (R_A, R_Z, Z, I_T)$, S_A
$CRT(v_Z)$, $M_Z = (R_Z, R_A, Z, A, N, I_U, I_T)$, S_Z
Stop

Fig. 8: Cryptographic protocol between the player application and the cache card

Once the transmission channel between the smartphone and the card is established, the player's application sends a Start message to start the protocol. The card then sends a random number R_Z and its identifier Z to the counterparty. The player's application first checks its list against the Z identifier to see if this cache has already been found by the player. If so, the player's application terminates the protocol run with a Stop command, and otherwise the protocol continues. The above measure serves to ensure that the cache issues only a single confirmation of its find to each finder.

The smartphone then sends the player's certificate $CRT(v_A)$ and the M_A, S_A pair to the cache, where M_A is the player's message and S_A is the player's signature of that message. The cache first verifies the validity of the player's certificate using the public key v_{CA} of the CA. If everything is OK, it uses the player's public key v_A from this certificate to verify that the signature S_A is the player's signature of the message M_A. If the mentioned signature is authentic, then the cache has thus verified the identity of player A (its identifier is known from the certificate) and the authenticity of the received message $M_A = (R_A, R_Z, Z, I_T)$.

In that message, the cache verifies the correctness of both the value of the random number R_Z and its identifier Z, and also writes down the value of R_A, which is a random number generated by the player's application for the given protocol run. The numbers R_A and R_Z are unique for a given protocol run due to their randomness, and checking their values prevents attacks by exploiting data that has been transmitted in other protocol runs. The value I_T is the identifier of a wandering item that is stored in the player's smartphone. If this value is zero, it means that there is no wandering item in the player's smartphone.

The cache then generates an acknowledgement of its discovery, which is a message $M_Z = (R_Z, R_A, Z, A, N, I_U, I_T)$, for which it creates a signature S_Z using its private key. The value N is the sequence number of player A in the sequence of all players who have discovered the cache so far, and the value I_U is the identifier of the wandering item stored in the cache. If this value is zero, there is no wandering item in the cache.

The cache then sends a triple of $CRT(v_Z)$, M_Z, S_Z to the counterparty. The player's application first verifies the validity of the received cache certificate using the CA's public key v_{CA}. If all is well, the public key v_Z from this certificate is used to verify that the signature S_Z is the signature of the cache Z for the message M_Z. If the signature is authentic, then the player's application verifies the identity of the cache Z and the authenticity of the message M_Z in this way.

The application then starts processing the message $M_Z = (R_Z, R_A, Z, A, N, I_U, I_T)$. In this message, it first verifies the correctness of the values of both random numbers R_Z and R_A, as well as the correctness of the identities of Z and A. The message is effectively a confirmation of cache Z that player A is the N-th finder of this cache, and within this find, there has been an exchange of wandering items so that the cache now contains an item with identifier I_T and the player's smartphone now contains an item with identifier I_U.

Finally, the player's application sends a Stop command to the cache, terminating the protocol run. Upon receiving this command, the cache increments the value N and stores it for the next finder. Via the Stop command, the cache can also terminate the protocol run. This situation occurs when any checks of the data received from the player's application are negative. In this case, the player's smartphone first needs to go away from the cache and then be brought closer again. This will start a new run of the protocol.

After the player's application has verified the authenticity of the message M_Z, it updates the state of the player's game in the specified way (e.g. adds one of the missing puzzle pieces). The player's application also forwards the received triple $CRT(v_Z)$, M_Z, S_Z to the game server GS. The game server checks the delivered certificate and signature. If everything is OK, it publishes player A as the N-th finder of the cache Z. In addition, if the value of I_T, resp. I_U is non-zero, it also publishes that the wandering item I_T is now stored in the cache Z, resp. the wandering item I_U is now in player A's smartphone. In the latter case, the server GS still sends the player a picture of the wandering item so that he can see what item he has discovered.

2.3 System security

The security of the designed system depends on the security of its individual elements and also on the security of the protocols used to communicate between them. Since the TLS protocol is generally considered to be secure, let us briefly discuss only the security of the protocol from the previous subsection. The security of this protocol relies mainly on the random number pair R_A and R_Z and the signature pair S_A and S_Z. The random numbers, together with their checking during the protocol, guarantee the consistency of the communication (i.e., a possible attacker cannot use data from other protocol runs for his attack). At the same time, these numbers also make it impossible for an adversary to obtain a signature for a possible spoofed message. The message signatures in the proposed protocol guarantee the identity of both parties and the authenticity of the exchanged data.

The security of the application on the player's smartphone and the security of the cache are also important for the safe operation of the proposed system. The security of the cache is mainly due to the technical protections of the smart cards and the fact that the configuration of these cards is done by the game server staff. In the case of the application, the player installs this application on his smartphone. As part of this installation, the application securely obtains the certificate of the public key v_{CA} and uses this key to perform, among other things, integrity checks on the application. This eliminates possible attacks on the game system by players. The first possible attack of a rogue player is that he modifies the application on his smartphone. However, such an application is immediately blocked upon launch because an initial check of its integrity detects unauthorized modification of the program code. The second possible attack is that the attacker creates a custom version of the application along with a fake certificate authority. However, this fake authority does not know the private key of the CA, and the cache will detect that the player's certificate $CRT(v_A)$ of the public key is fake.

3. Discussion

Now is the time to evaluate the proposed solution. For the geocaching operator, the most significant disadvantage is probably the necessity to build and operate a certification authority and to update the game server software. On the other hand, it can be stated that microcomputer caches can make geocaching more attractive. For example, players can collect pictures of different sets, they can collect pieces of different puzzles, they can exchange virtual items through caches, etc. These possibilities have the potential to increase the interest of the general public in geocaching, which could motivate existing geocaching operators to build and operate a certification authority and also to modify their game servers.

From the point of view of the cache owner, the disadvantage of the proposed solution is the higher price of the cache. The price of conventional caches ranges from units to several tens of US dollars and the price of microcomputer caches is in the several tens of US dollars. However, this one-time cost may not be an unreasonable amount for potential owners. The electronics of the cache are encapsulated in high quality plastic, giving the owner a durable cache with a long lifespan. And unlike a traditional cache, maintenance costs are minimal, as there is no need to replace the described diaries, refill pencils, etc. Another advantage is the unobtrusiveness of the cache. Contemporary cards have dimensions on the order of centimeters and so can be disguised as ordinary, inconspicuous objects (an information sign, a fungus on a tree, etc.). So, they will not attract the attention of potential vandals or cache robbers. And there is no risk of mistaking the cache for an explosive device.

Another advantage for the cache owner is the automation of finding control. In the case of traditional geocaching, the players themselves register their finds on the game server, and these records should be checked by the cache owner against the cache diary. If the owner does not find some record in the diary, he should delete this record from the game server. However, the described procedure is laborious and is therefore often not carried out by cache owners. In the case of a microcomputer cache, checking for a cache find is simple. After the application receives a signed confirmation from the cache, it sends it to the game server either immediately via the GSM network ("Global System for Mobile Communications") of the telephone operator, or later via the first suitable Wi-Fi network. The server first verifies the signature and just then registers the player's cache find.

From the point of view of some players, the basic disadvantage of the described solution is the necessity to purchase a smartphone with an NFC interface. However, given the range of contemporary smartphones and their price, this is not a problem at present. The benefit for these players will be the diversification of geocaching through various games.

4. Conclusions

This paper proposes the replacement of traditional caches in the form of boxes with contactless smart cards. The basic elements and architecture of the system are described, as well as the methods of communication between these elements. Furthermore, this paper proposes a cryptographic protocol for communication between the player's smartphone and the cache card. The proposed solution is assessed from a security perspective and finally discussed in terms of positives and negatives for all stakeholders.

The proposed concept appears to be technically and economically feasible. Its advantage is that it both solves the contemporary problems of traditional geocaching and offers new possibilities to make geocaching more attractive.

Another advantage of this concept is that it can be implemented in parallel within contemporary geocaching in a way that owners of many traditional caches simply stick the microcomputer card on their existing cache.

References

[1] –: Geocaching. Wikimedia Foundation, San Francisco. https://en.wikipedia.org/wiki/Geocaching

[2] –: Geocaching.com. Groundspeak, Seattle. https://www.geocaching.com/play/search

[3] –: Traditional Caches. GeoWiki. https://bit.ly/3zV8DCa

[4] –: Mystery Caches. GeoWiki. https://bit.ly/3gVb7sX

[5] –: We Love Geocaching. Facebook, Palo Alto. https://bit.ly/2XmEUQP

[6] Savage, D.: Geocaching: the unintended results. BBC News. 5 July 2011. https://www.bbc.com/news/uk-england-leeds-14039229

[7] –: Standard ISO/IEC 14443. Wikimedia Foundation, San Francisco. https://en.wikipedia.org/wiki/ISO/IEC_14443

[8] Navneeta D. aj.: How NFC Tag technology in contactless payment cards is delivering value-added benefits for consumers, retailers and issuers. Infineon Technologies, Neubiberg. https://bit.ly/3kfNRFq

[9] Rescorla E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. Internet Engineering Task Force, Fremont 2018. https://tools.ietf.org/html/rfc8446

[10] Dierks T., Rescorla E.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246. Internet Engineering Task Force, Fremont 2008. https://tools.ietf.org/html/rfc5246

[11] Menezes A., Oorschot P., Vanstone S.: Handbook of Applied Cryptography. CRC Press. N. York 1996.

Karel Burda received the M.S. and Ph.D. degrees in Electrical Engineering from the Liptovsky Mikulas Military Academy in 1981 and 1988, respectively. During 1988-2004, he was a lecturer in two military academies. At present, he works at Brno University of Technology. His current research interests include the security of information systems and cryptology.