On the Potential of PUF for Pseudonym Generation in Vehicular Networks

Jonathan Petit*, Christoph Bösch*, Michael Feiri*, Frank Kargl*†
*Distributed and Embedded Security Group
University of Twente, The Netherlands
{j.petit, c.boesch, m.feiri, f.kargl}@utwente.nl
†Institute of Distributed Systems
University of Ulm, Germany
frank.kargl@uni-ulm.de

***Abstract*—Most proposals for security of vehicular networks foresee the generation of a comparatively large number of changing pseudonyms to prevent vehicles from being identified or tracked. Most proposals rely on communication with backend pseudonym providers to refill a vehicle's pseudonym pool which creates a number of problems, one being secure storage and handling of a large amount of private key material. In this paper we investigate the usage of Physical Unclonable Functions (PUFs) and Public PUFs (PPUFs) instead of Hardware Security Modules for this purpose. We describe a possible solution that uses PUF and Fuzzy Extractors to provide the necessary stability.**

***Index Terms*—Pseudonym, PUF, PPUF, Vehicular Networks.**

I. Introduction

The world of Intelligent Transportation System is getting closer to the deployment and requires security and privacy mechanisms to ensure acceptance by users [1]. Vehicle-to-Vehicle (V2V) communications enable vehicles to exchange information about road conditions, such as accident or traffic jam. As safety-related information has a potential impact on traffic safety, this information must be secured [2]. A mandatory mechanism is the authentication of the sender. Indeed, receivers must be able to verify that the sender is an authorized vehicle. Unfortunately, authentication mechanisms break user privacy as every receiver learns the identity of the sender. Therefore, a short-term credential *–pseudonym–* should be implemented in order to prevent authentication to ease vehicle tracking. In the current solution [2], pseudonyms are first issued by a Certification Authority (CA)–also named *Pseudonym Provider* (PP), so in the rest of the paper, we use the terms CA and PP interchangeably. Then, pseudonyms are used to sign messages until a pseudonym change is triggered (on a fixed schedule, a random schedule, or triggered by Road-Side Unit). When a vehicle is running out of pseudonym, it requests a new set of pseudonyms to the PP. A solution to simplify credentials management, and to reduce the burden on the PP, is to allow vehicles to self-generate their own pseudonyms and the corresponding key pairs. This eliminates the need of pre-loading, storing, refilling, as well as obtaining pseudonyms through infrastructure connectivity. Hence, the communication overhead of pseudonym schemes would be reduced, as vehicles do not have to contact the PP for pseudonym refill for example. Self-generation of pseudonyms also improves the system usability, as privacy is not compromised if the local supply of pseudonyms is exhausted and it is not necessary to over-provision a vehicle with pseudonyms [3]. To enable self-generation, vehicles need a secret key. In this paper, we investigate how Physical Unclonable Functions (PUFs) can be used in the generation of secret keys, and especially in the context of pseudonym generation.

Key Extraction from PUFs. As introduced by Pappu et al. [4], [5], a PUF is a primitive that maps challenges C_i to responses R_i, which depend on the physical properties of the device in which the PUF is contained or embedded. Physical Unclonable Functions have essentially two parts: i) a physical part and ii) an operational part. The physical part is a physical system that is very difficult to clone. The operational part corresponds to the function. In order to turn the physical system into a *function* a set of challenges C_i (stimuli) has to be available to which the system responds with a set of sufficiently different responses R_i. Examples of PUFs include optical PUFs [4], [5], silicon PUFs [6], coating PUFs [7], Intrinsic-PUFs [8], and LC-PUFs [9]. Regardless of their particular instantiation, their unclonability, tamper evidence and tamper resistance properties have made PUFs very useful tools in Intellectual Property (IP) protection, device authentication and secure key storage applications. A common characteristic of PUF-based protocols [8], [10], [11] is the derivation of a key(s) from the PUF, which is used to encrypt (a piece of) IP and authenticate its origin. In this work we focus on the key extraction for pseudonym and signature computation. In [10], the authors observe that by using public-key cryptography in combination with a PUF on FPGA, the corresponding private-key does not need to ever leave the FPGA, even during the enrollment stage, thus increasing the security of the overall system.

The Need for a Fuzzy Extractor. Notice that PUF responses are noisy by nature. This means, that two calls to a single PUF with the same challenge C_i will output two different but closely related responses R_i, R_i'. The measure of closeness can be defined via a distance function, e.g., the Hamming distance. This distance function should be small for responses from the same device and very large for PUF responses from

different devices. Since the plain PUF responses are noisy, they cannot be used as a key. This means that the data encrypted under response R_i cannot be decrypted with response R_i'. In order to derive reliable and uniform strings from (imperfect) sources of randomness, such as a PUF, the concept of a fuzzy extractor [12] or helper data algorithm [13] was introduced. Thus, we obtain a secure master key from the fuzzy extractor. This master secret key can be the source for a key derivation scheme [14] to derive public/private key pair(s) which can then be used as a pseudonym(s).

The application of PUF for in-vehicle security, and more specifically secure key storage and component identification for insurance application, was analyzed in [15]. We differentiate from this work by focusing on PUFs use in pseudonym scheme. Moreover, we complete their work by analyzing the potential of PPUF [16] in vehicular networks.

Organization: The rest of the paper is organized as follows. Section II gives a short introduction of different pseudonym schemes. Section III gives an overview of Physical Unclonable Function and Fuzzy Extractor. Then, we investigate how PUF and PPUF can be used for pseudonym generation in respectively Section IV and Section V. Section VI concludes this paper.

II. Pseudonym Schemes

Looking at the means of achieving pseudonymity, the schemes differ in what cryptographic mechanisms they employ. Four major categories can be distinguished for pseudonymity in vehicular networks. Schemes based on *asymmetric cryptography* aim for PKI-oriented privacy solutions. Pseudonyms are typically represented by public key certificates without identifying information. To facilitate verification by receiving vehicles, pseudonym certificates must be sent along with messages. Schemes based on *identity-based cryptography* extend this idea but remove the need of explicit public key certificates by deriving public keys from identifiers. This reduces communication overhead for pseudonym use but introduces new challenges for pseudonym issuance. Pseudonym schemes based on *group signatures* introduce one public key for a group of vehicles. Group-based schemes reduce the need for pseudonym changes but pose new challenges for pseudonym resolution and revocation. Schemes based on *symmetric cryptography* are attractive because of their computational efficiency, but must be cast into protocols that can enable reliable authentication. Due to the different challenges posed by each cryptographic paradigm, many solutions combine different mechanisms to achieve more effective schemes. In Sections II-A to II-D, we discuss the proposals of each category and emphasize the need of a secure key to generate pseudonym.

A. Asymmetric Cryptography Schemes

The first propositions to ensure privacy in vehicular networks were based on asymmetric cryptography [17], [18]. Afterwards, this approach has been followed by major initiatives such as the SeVeCom project [19], the IEEE 1609.2v2 standard [20], and the Car-to-Car Communication Consortium [21]. Indeed, pseudonymous communication can be achieved with traditional public key cryptography schemes (PKI) by equipping vehicles with a set of public key certificates and corresponding key pairs. The public key certificates are used as unlinkable pseudonyms and, therefore, contain no identifying information. Vehicles sign messages with the secret key of the currently active pseudonym and attach the signature, as well as the corresponding pseudonym certificate, to the message. Receivers can verify a message signature based on the pseudonym certificate, but are unable to determine the sender's vehicle identity. One could conclude that the central challenge of asymmetric cryptography scheme is to have a secure key.

B. Identity-based Cryptography Schemes

Identity-based cryptography (IBC) [22] is related to asymmetric cryptography with the significant difference that a node's identifier functions as that node's public key. A corresponding privacy key is derived from the identifier to sign messages. To verify the signature, knowledge of the sender's identifier is sufficient. An explicit public key or additional certificate are not required. However to prevent that any node with knowledge of another node's identifier can derive a corresponding private key, only a centralized trusted authority with full knowledge of system parameters is able to extract private keys and assign them to nodes. Thus, a node's authenticity is implicitly guaranteed rather than explicitly stated with a certificate, because only authorized nodes would receive a private key corresponding to a specific identifier.

Compared to conventional PKI, IBC avoids the use of certificates for public key verification and the exchange of public keys and associated certificates, while providing similar authentication characteristics. The resulting communication and storage efficiency make IBC attractive for authentication in vehicular communications. A drawback is the requirement that a trusted authority must extract private keys from vehicle identifiers rather than having vehicles generate their own key pairs.

C. Group Signature Schemes

The downside of using a changing set of anonymous keys as pseudonyms is the necessity for generation, delivery, storage, and verification of numerous certificates for all pseudonym public keys (or private keys in case of IBC). To mitigate this overhead, Calandriello et al. [23] presented a first approach that uses group signatures to enable vehicle OBUs to generate and certify their own pseudonyms without interacting with the CA. Basically, they used group signatures to support issuance of traditional public key certificates. The group manager (GM) is a new entity that sets group parameters, changes group public keys, and may revoke anonymity if supported by the scheme. In contrast to PP or CA, the GM role can be filled by a vehicle and not necessarily a trusted third party. In any case the GM has a key role in the key (pseudonym) generation.

D. Symmetric Cryptography Schemes

Symmetric cryptography is less flexible than asymmetric cryptography when it comes to the realization of authentication capabilities but is highly efficient in terms of computational overhead. In symmetric schemes a (hashed) Message Authentication Code ((H)MAC) is used for message authentication. The signer hashes the message together with a secret key. Any verifier must know the same secret key to verify the MAC by performing the same operation on the message. As a consequence, any node with knowledge of the secret key can generate valid MACs, thus a node's anonymity set would extend to all nodes using the same secret key. However, sender accountability is not provided as non-repudiation cannot be achieved.

For inter-vehicle communication utilization of symmetric authentication schemes offers the benefits of short generation and verification time as well as less security overhead [24]. At the same time, the need for deployment and maintenance of certification infrastructure and associated costs, as need for asymmetric schemes, could be replaced by potentially simpler key distribution. In a naïve scheme, each OBU could have the same secret key preinstalled, or even a set of shared secret keys [25]. Due to the potential benefits, symmetric schemes have been considered for VANET authentication. However, reliable authentication requires that exposure of single secret keys should not compromise authentication of all OBUs. This requirement, paired with the desire for accountability, makes actual symmetric authentication schemes more complex.

Based on the description of the four schemes, we conclude that the generation of a secure key is the crucial phase of a secure pseudonym scheme. In the next sections we investigate the potential of PUFs and PPUFs for key extraction in vehicular networks.

III. Physical Unclonable Functions and Fuzzy Extractor

Figure 1 shows that with help of a PUF response, secret key information can be generated. Each run of the PUF will cause slight changes in the digitized output, resulting in a noisy key. This means that the PUF only produces an approximation of the response that is expected. Because we want to derive cryptographic keys from the PUF responses, it is necessary to make the PUFs output identical each time a challenge is reused. A Fuzzy Extractor can be used to create a unique key.

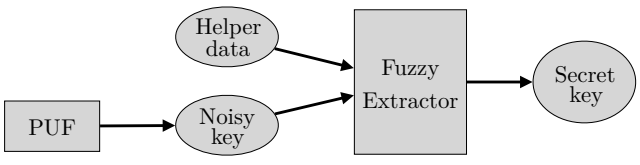


Fig. 1. Structure of key extraction

A. Physical Unclonable Function

A Physical Unclonable Function (PUF) as introduced in [4], [5] is a primitive that is bound to a physical system and extracts key information by mapping a set of challenges C_i to a set of responses R_i. This challenge-response behaviour is highly dependent on the physical properties of the device in which the PUF is contained or embedded. PUFs consist of two parts:

i) a physical part, which is an intractably complex physical system that is very difficult to clone. It inherits its unclonability from uncontrollable process variations during manufacturing. For PUFs on an IC these process variations are typically deep-sub-micron variations such as doping variations in transistors.
ii) an operational part, which corresponds to the function.

In order to turn the physical system into a *function* a set of challenges C_i (stimuli) has to be available to which the system responds with a set of sufficiently different responses R_i. The function can only be evaluated using the physical system and is unique for each physical instance because of process variations. Moreover, it is unpredictable even for an attacker with physical access.

Silicon PUFs [6] (SPUFs), for example, generate their responses based on the hidden timing and delay information of integrated circuits. The variations in the manufacturing process cause significant delay differences among different ICs, even with identical layout masks. SPUFs are very sensitive to environmental changes such as temperature and voltage. Thus they are unsuitable to be used in a vehicle.

In the rest of this Section we summarize some known PUF constructions that can be used in the context of pseudonym generation in vehicles. These include: Coating PUFs, SRAM PUFs, Bistable Ring PUFs and Public PUFs.

1) Coating PUF: Tuyls et al. [7] introduced the concept of a Coating PUF (CPUF) where an IC is covered with a protective coating. This coating contains random particles with different dielectric constants at random positions. Figure 2 shows that below this coating the IC has an array of sensors to measure the local capacitance of the coating. Due to the randomness in the dielectricity of the coating particles, also the capacitance will be random for every sensor. The authors [7] show, that each sensor can extract up to three key bits. Instead of using the challenge-response behavior of the PUF, the coating can be used to store keys inside the IC rather than in memory. Note that CPUFs need an additional manufacturing step, but are very cheap to produce.

2) SRAM PUF: To overcome the disadvantage of additional manufacturing steps the concept of *Intrinsic* PUF (IPUF) was introduced. Guajardo et al. [8] present the first construction of an IPUF based on the start-up values of the SRAM memory already present on the device. Thus, the device does not need any custom circuits or manufacturing steps to turn it into a PUF. Note that SRAM memory is widely available in almost every computing device. A SRAM cell is constructed by cross-coupling two logic inverters. Such a circuit has two logically stable states. Due to manufacturing variations there will be a

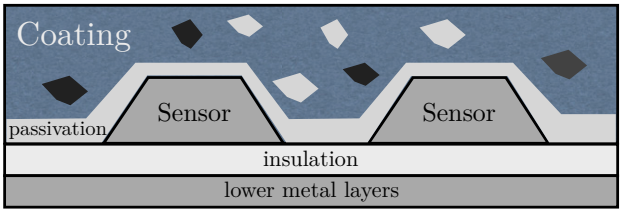


Fig. 2. Schematic cross-section of a Coating PUF IC

mismatch between the two inverters, which will determine the value of the power-up state of the SRAM cell. Each SRAM cell is (heavily) biased towards zero or one. Different SRAM cells will behave randomly and independently from each other. We consider a range of memory locations within a SRAM memory block as a challenge and the start-up values at these locations as the response. SRAM PUFs are relatively stable to temperature variations and robust over time.

3) Bistable Ring PUF: A Bistable Ring PUF (BR-PUF) as introduced by Chen et al. [26] is another example of an intrinsic PUF. The basic idea is similar to the SRAM PUF which uses two cross-coupled inverters per SRAM cell. The BR-PUF makes use of a ring of an even number of inverters. This ring has two possible stable states. When powered up, the ring falls into one of its two possible states, thus producing a 1-bit output. This single ring can be turned into a BR-PUF by duplicating the inverters and adding multiplexers and demultiplexers between the stages. This allows an exponential number of Challenge-Response-Pairs (CRPs). The BR-PUF is a temperature sensitive PUF, but the authors claim, that with additional hardware and protocol measures the problem can be addressed. BR-PUFs are relatively reliable against aging.

4) Public PUF: A Public PUF (PPUF) as introduced by Beckmann and Potkonjak [16] is a PUF that can be reverse engineered and expressed as a function. The function characterizes the PUF and thus it is possible to emulate the PUFs behavior for a given input. The emulation needs to be slower than the real-time behavior of the actual PUF. The PPUF characteristic function plays the role of the public key. A holder of the PPUF characteristics can choose a secret key sk and some public parameters P_{pub} at random. With these parameters as the input for the PPUF, the user calculates a public key $pk = \mathsf{PPUF}(sk, P_{\mathsf{pub}})$ and sends the all public information to the party holding the real PPUF. The holder of the real, fast PPUF can then, with help of the public parameters, generate the shared secret key $sk = \mathsf{PPUF}(P_{\mathsf{pub}}, pk)$.

A similar concept to PPUF called "SIMulation Possible, but Laborious" (SIMPL) is proposed by Rührmair [27]. SIMPL systems are disordered, unclonable physical systems with a complex input-output behavior. Similar to PPUFs, a SIMPL system has a publicly known numeric description, which allows everyone to simulate the systems output slowly. Only the system holder can determine the output in a fast way with help of the physical measurements.

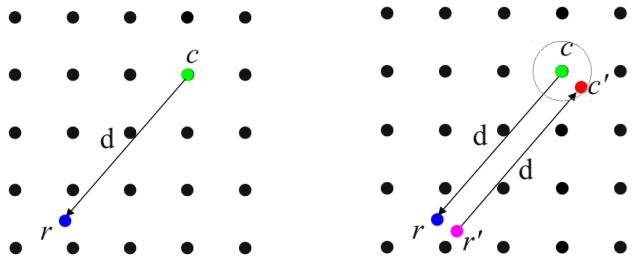
B. Fuzzy Extractor

The responses of a PUF can not be used as a key (as in e.g. [7]) in a cryptographic primitive for two reasons. First, PUF responses are obtained through measurements which are typically noisy. This leads to a problem since cryptographic functions are very sensitive to noise on their inputs. Even a single bit difference in the response cannot be tolerated. Second, PUF responses are not uniformly distributed. Hence, even if there was no noise, the response would not form a cryptographically secure key. In order to deal with both issues a Fuzzy Extractor or Helper Data algorithm has to be used. For the precise definition of a Fuzzy Extractor and Helper Data algorithm we refer to [12], [13]. A Fuzzy Extractor deals with both issues by implementing first an *information reconciliation phase* and second, by applying a *privacy amplification* or randomness extraction primitive. In order to implement these two primitives, helper data W are generated during the *enrollment phase*. During this phase, carried out in a trusted environment, a probabilistic procedure called Gen is run. Later, during the *key reconstruction* or authentication phase, the key is reconstructed based on a noisy measurement r' and the helper data W. During this phase, a procedure called Rep is performed. We present one of the constructions for such procedures previously described in [12], [28]. Other constructions as well as constructions for other metrics can be found in [12].

Construction Based on Code Offset. In order to implement the procedures Gen and Rep an error correction code $\mathcal{C}$ and a set $\mathcal{H}$ of universal hash functions [29] is required. The Gen-procedure takes as input a PUF response(s) r and produces as output a key K and helper data $W = (W_1, W_2)$. This is achieved as follows (cf. Figure 3(a)). First, a code word $c \leftarrow \mathcal{C}$ is chosen at random from $\mathcal{C}$. Then, a first helper data vector equal to $d = c \oplus r$ is generated and W_1 is set to d. Furthermore, a hash function h_i is chosen at random from the set $\mathcal{H}$ and the key K is defined as $K \leftarrow h_i(r)$. The helper data W_2 is set to i. During the key reconstruction phase (cf. Figure 3(b)) the procedure Rep is run. It takes as input a noisy response r' from the same PUF and helper data W and reconstructs the key K. This is accomplished according to the following steps:

1) *Information Reconciliation* (see Figure 3(b)):
   a) Using the helper data W_1, $c' = d \oplus r'$ is computed.
   b) The decoding algorithm of $\mathcal{C}$ is used to obtain c.
   c) From c, r is reconstructed as $r = d \oplus c$.
2) *Privacy amplification*: The helper data W_2 is used to choose the correct hash function $h_i \in \mathcal{H}$ and to reconstruct the key as $K = h_i(r)$.

The security of the above constructions has been established in [12], [13], [28], [30], [31]. By security here we mean two complementary things. First, [12], [13] provide a bound on the number of bits of entropy left after the fuzzy extractor operates on the source bits of the PUF. Second, [28], [30], [31] show that given the public helper data information, negligible information is learned about the derived secret. Finally, [30], [31] show how to protect the helper data against tampering



(a) Enrollment. (b) Key reconstruction.

Fig. 3. Phases of a Fuzzy Extractor.

and modification.

The first implementation of a Fuzzy Extractor on FPGA was done by Bösch et al. [32] who also provide explicit constructions and investigate the hardware costs of Fuzzy Extractors on FPGAs. Their results show, that implementing a Fuzzy Extractor on an FPGA requires less than 450 Slices and is thus feasible in practice.

IV. PUF for Pseudonym Generation

Now that we described pseudonym schemes in Section II and PUF in Section III, we investigate how PUF could be used in the pseudonym generation phase.

In the SeVeCom project [2], the CA generates a set of pseudonyms for each vehicle. Each pseudonym contains an identifier of the CA, the lifetime of the pseudonym, the public key, and the signature of the CA, and thus, no information about the identity of the vehicle. Pseudonyms are stored and managed in the on-board pseudonym pool, with their corresponding secret keys kept in the Hardware Security Module. This ensures that each vehicle has exactly one key pair (its own pseudonym and private key) that is active during each time period. But this solution relies on a secure key storage, which is expensive. Indeed, SRAM is already present in vehicles, and therefore, the cost of a SRAM PUF is negligible. But beyond the notable cost difference, the PUF is proved to be more secure than an HSM [8], [31], [33]. Moreover, as an attacker cannot tamper the PUF without breaking it, the authenticity of the signer is always proved. The PUF can also be used for IP Protection to ensure that in-vehicle components are legitimate [15]. Thus, we propose to use a PUF in replacement of an HSM for pseudonym generation.

Most of the public key schemes require a source of randomness for the generation of the key material. To be dependent of the PUF, we can use a pseudo-random number generator or a hash function with the PUF responses as an input vector to generate randomness. This randomness is then the input for the key generation of the used public key scheme (e.g., HMQV [34]) to generate public/private key pair. Figure 4 depicts how a PUF participates in the pseudonym generation. During the vehicle's enrollment phase, the vehicle generates its master secret key (msk) with help of the PUF. From msk, the vehicle derives a master public key (mpk) and sends it to the CA. To create a pseudonym during the *everyday driving* phase, the vehicle challenges the PUF to generate a random private/public key pair (sk, pk) that forms the pseudonym. Then, the vehicle signs pk with its master secret key msk and sends it to the CA. As the CA knows mpk, it can extract pk and signs it, which creates the certificate of the pseudonym. The CA stores the link pk-mpk for pseudonym resolution phase. The certificate $cert_{sk}$ is sent to the vehicle and proves that (sk, pk) is validated by the CA. The vehicle could sign message with sk and appends $cert_{sk}$. As the certificate is linked to the private key, only the owner of sk can use $cert_{sk}$. This protects against Man-in-the-Middle attack, where an attacker tries to steal certificate for example.

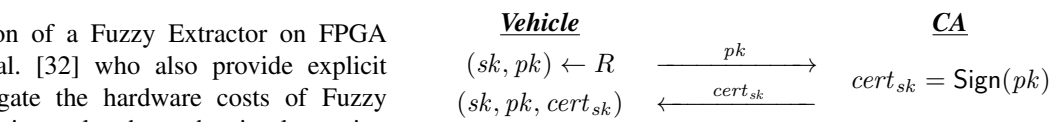


Fig. 4. Simplified protocol: PUF used for key generation and certificate distribution. The vehicles public key will be signed with msk before sending. The CA is able to verify the received signature with mpk.

As one can see in Figure 4, the number of interactions between the vehicle and the CA is limited. The vehicle contacts the CA either for every pseudonym or for a set of pseudonyms.

Moreover, as the vehicle generates its own pseudonyms, using PUF reduces the load on the CA. Therefore, the scalability issue of one CA receiving requests of pseudonyms refill from thousands of vehicles is improved. Regarding the vehicle's computation overhead, the generation time of a secure key from a PUF is in the order of magnitude of 10^{-4} second [32] and is considered negligible.

But the main benefit of using a PUF for pseudonym generation is that this solution does not require a secure key storage. Indeed, intrinsic PUFs (SRAM-PUF, BR-PUF) have interesting properties for use in secret key generation and storage. Since the key is generated from intrinsic randomness introduced by inevitable manufacturing variability, no explicit key-programming step is required, which simplifies key distribution. Moreover, since this randomness is permanently fixed in the (sub-)microscopical physical details of the chip, no conventional non-volatile key memory is required. This also offers additional security against probing attacks and possibly other side-channel attacks, since the key is not permanently stored in digital format, but only appears in volatile memory when required for operation [35]. In our context of vehicular networks, the key is generated on-the-fly when the vehicle's engine is started. Furthermore, since many cryptographic security applications require a source of pure randomness, any secure processor should implement some type of random number generation algorithm on-chip. Hardware algorithms have been proposed before [36], however it is also possible to use the existing PUF circuitry to generate a random number which is acceptable for cryptographic applications [37], [38], and proved to be an efficient source of entropy for key generation [39].

V. PPUF for Pseudonym Generation

A PPUF (or the related SIMPL system) extends the PUF concept to allow public key cryptography [16], [27]. The physical PPUF in this system represents the private key, while a simulated representation of its characteristics serves as a public key. The necessary asymmetry stems, e.g., from the different time required to calculate the PPUF or the simulation or some other intrinsic asymmetry. While PPUFs are in an early stage of research, we assume existence of such a timing-based PPUF as an asymmetric cryptographic primitive and describe how it could be used for V2V security.

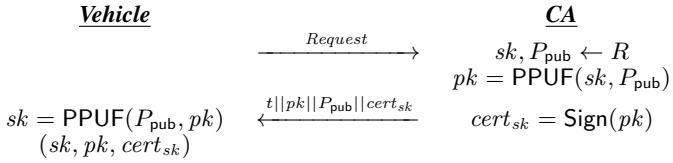


Fig. 5. Simplified protocol: PPUF used for key generation and certificate distribution. The CA will sign the message before sending to allow the authentication by the client.

Such PPUF enables two parties A and B to exchange a secret key. We assume that A is in possession of a PPUF. B has a simulated version of A's PPUF (also named *PPUF characteristic*), and uses it to simulate its output for some input. As the "simulated-PPUF" is slower than the real PPUF [16], it is used as a resource testing mechanism to prevent spoofing.

Figure 5 shows in a simplified way how a PPUF can be used for key generation and certificate distribution. Assume the PPUF characteristic of the vehicle (i.e. the "simulated-PPUF") is only known by the CA. To issue a new pseudonym, the CA selects a secret key sk at random and also chooses at random some public parameters P_{pub}. The CA simulates the PPUF with sk and the public parameters as the input. The output of the PPUF is the public key $pk = \mathsf{PPUF}(sk, P_{\mathsf{pub}})$. The public key pk is then signed with the master secret key of the CA to create the certificate $cert_{sk}$ linked to the secret key sk, and thus, to the PPUF owner. Before sending the public parameters for the new pseudonym to the vehicle, a time stamp t will be included in the tuple. This will prevent replay attacks, where an attacker resends an earlier eavesdropped message to the vehicle. Then, the tuple $(t||pk||P_{\mathsf{pub}}||cert_{sk})$ will be signed by the CA and sent to the vehicle. After receiving a message from the CA, the vehicle first checks the authenticity and integrity of the message with help of CA's signature and the time stamp. If the message is accepted, the vehicle runs its PPUF with the public parameters to generate the corresponding secret key $sk = \mathsf{PPUF}(P_{\mathsf{pub}}, pk)$ necessary for signing messages. This process is used for every pseudonym used by the vehicle. Depending on the scheme, either the CA or the vehicle can trigger a pseudonym change. The tuple $(sk, pk, cert_{sk})$ will be the new pseudonym of the vehicle. As in the case of PUF, the certificate $cert_{sk}$ proves that (sk, pk) is validated by the CA. The vehicle can now sign messages with sk and append $cert_{sk}$. As the certificate is linked to the private key, only the owner of sk can use $cert_{sk}$.

The original concept of PPUF assumes that the PPUF characteristics are public, and thus, everyone in possession of the PPUF characteristics can generate (simulate) a key pair. Even if an attacker creates a valid key pair, he cannot generate a valid certificate corresponding to the generated secret key. Also, an attacker cannot sign the tuple $(t||pk||P_{\mathsf{pub}}||cert_{sk})$ on behalf of the CA.

VI. Conclusion and Future Challenges

Vehicular networks require pseudonymous communication to be accepted by users. To ensure pseudonymity, a key needs to be generated in a secure manner. Therefore, in this paper, we investigated the potential of Physical Unclonable Function (PUF) to generate pseudonym key pairs for vehicular networks. After giving an overview of pseudonym schemes and PUF, we discussed how PUF and Public PUF (PPUF) can be used in the context of pseudonym generation, proposed a protocol and analyzed the benefits and limits.

The challenge how to apply PPUF for broadcast authentication is left open and matter of our future research. Indeed, one could envision to use the (P)PUF challenge/response mechanism not only to generate pseudonyms but also to authenticate messages. This may require an interactive protocol, which is per se not suitable for broadcast communication. Moreover, if every vehicle knows the PPUF characteristics, then the PPUF owner will be always identified, which breaks its privacy and the pseudonym scheme. A possible solution could be that PPUF characteristics (P_{pub}) act as pseudonyms. Then, vehicles publish a list of PPUF characteristics and have a strategy to change it. We will investigate this in the next step of our research.

Regarding a real experimentation of PUF in Vehicular Networks, the PRESERVE Project[1] is currently developing a security ASIC that includes a Bistable Ring PUF [40] that we plan to use to implement the proposed pseudonym generation scheme in a real world system and to run benchmark.

Acknowledgement

The authors like to thank Arjan Jeckmans for fruitful discussions. The research leading to these results has also received funding from the European Union's Seventh Framework Programme project PRESERVE under grant agreement n°269994.

References

[1] F. Schaub, Z. Ma, and F. Kargl, "Privacy Requirements in Vehicular Communication Systems," in *Symposium on Secure Computing, IEEE International Conference on Privacy, Security, Risk, and Trust (PASSAT 2009)*. Vancouver: IEEE, 2009, pp. 139–145. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5283398

[2] F. Kargl, P. Papadimitratos, L. Buttyan, M. Muter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: implementation, performance, and research challenges," *Communications Magazine, IEEE*, vol. 46, no. 11, pp. 110–118, november 2008.

[1] http://www.preserve-project.eu

[3] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in vanet," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, ser. VANET '07, 2007, pp. 19–28.

[4] R. Pappu, "Physical One-Way Functions," Ph.D. dissertation, MIT, 2001.

[5] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical One-Way Functions," *Science*, vol. 297, pp. 2026–2030, 2002.

[6] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon Physical Random Functions," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2002, pp. 148–160.

[7] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-Proof Hardware from Protective Coatings," in *Cryptographic Hardware and Embedded Systems — CHES 2006*, ser. LNCS, L. Goubin and M. Matsui, Eds., vol. 4249. Springer, October 10-13, 2006, pp. 369–383.

[8] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," in *Cryptographic Hardware and Embedded Systems — CHES 2007*, ser. LNCS, P. Paillier and I. Verbauwhede, Eds., vol. 4727. Springer, September 10-13, 2007, pp. 63–80.

[9] B. Škorić, T. Bel, A. Blom, B. de Jong, H. Kretschman, and A. Nellissen, "Randomized resonators as uniquely identifiable anti-counterfeiting tags," Philips Research Laboratories, Technical Report, January 28th, 2008.

[10] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Physical Unclonable Functions and Public Key Crypto for FPGA IP Protection," in *Proceedings of the 2007 International Conference on Field Programmable Logic and Applications - FPL 2007*, ser. Amsterdam, The Netherlands, IEEE, August 27-30, 2007, pp. 189–195.

[11] E. Simpson and P. Schaumont, "Offline Hardware/Software Authentication for Reconfigurable Platforms," in *Cryptographic Hardware and Embedded Systems — CHES 2006*, ser. LNCS, L. Goubin and M. Matsui, Eds., vol. 4249. Springer, October 10-13, 2006, pp. 311–323.

[12] Y. Dodis, M. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology —- EUROCRYPT 2004*, ser. LNCS, C. Cachin and J. Camenisch, Eds., vol. 3027. Springer-Verlag, 2004, pp. 523–540.

[13] J.-P. M. G. Linnartz and P. Tuyls, "New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates," in *Audio-and Video-Based Biometrie Person Authentication — AVBPA 2003*, ser. LNCS, J. Kittler and M. S. Nixon, Eds., vol. 2688. Springer, June 9-11, 2003, pp. 393–402.

[14] D. Pavlovic and C. Meadows, "Deriving secrecy in key establishment protocols," in *Proceedings of the 11th European conference on Research in Computer Security*, ser. ESORICS'06, 2006, pp. 384–403.

[15] M. Asim, J. Guajardo, S. Kumar, and P. Tuyls, "Physical unclonable functions and their applications to vehicle system security," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, april 2009, pp. 1–5.

[16] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Information Hiding*, S. Katzenbeisser and A.-R. Sadeghi, Eds., 2009, pp. 206–220.

[17] L. Gollan and C. Meinel, "Digital signatures for automobiles," in *Proceedings of Systemics, Cybernetics and Informatics (SCI)'02*, 2002.

[18] M. E. Zarki, S. Mehrotra, G. Tsudik, and N. Venkatasubramanian, "Security issues in a future vehicular network," in *European Wireless*, 2002, pp. 270–274.

[19] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: implementation, performance, and research challenges," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 110–118, Nov. 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4689253

[20] IEEE, "IEEE 1609.2v2 - Standard for Wireless Access in Vehicular Environments (WAVE) - Security Services for Applications and Management Messages," 2011.

[21] C2C-CC, "Public key infrastructure memo," Car 2 Car Communication Consortium, Tech. Rep., 2010.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in cryptology CRYPTO 84*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 47–53. [Online]. Available: http://dl.acm.org/citation.cfm?id=19478.19483

[23] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," in *VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*. New York, NY, USA: ACM, 2007, pp. 19–28.

[24] J. Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks," in *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*. New York, NY, USA: ACM, 2005, pp. 79–87.

[25] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang, "Enforcing privacy using symmetric random key-set in vehicular networks," in *Eighth International Symposium on Autonomous Decentralized Systems ISADS'07*, March 2007, pp. 344–351.

[26] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Rührmair, "The bistable ring puf: A new architecture for strong physical unclonable functions," in *HOST 2011, Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 5-6 June 2011, San Diego, California, USA*. IEEE Computer Society, 2011, pp. 134–141.

[27] U. Rührmair, "Simpl systems: On a public key variant of physical unclonable functions," *IACR Cryptology ePrint Archive*, vol. 2009, p. 255, 2009.

[28] A. Juels and M. Wattenberg, "A Fuzzy Commitment Scheme," in *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 1999, pp. 28–36.

[29] L. Carter and M. N. Wegman, "Universal Classes of Hash Functions," *J. Comput. Syst. Sci.*, vol. 18, no. 2, pp. 143–154, 1979.

[30] X. Boyen, "Reusable Cryptographic Fuzzy Extractors," in *ACM Conference on Computer and Communications Security—CCS 2004*. New-York: ACM Press, 2004, pp. 82–91. [Online]. Available: http://www.cs.stanford.edu/ xb/ccs04/

[31] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith, "Secure remote authentication using biometric data," in *Advances in Cryptology — EUROCRYPT 2005*, pp. 147–163.

[32] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient Helper Data Key Extractor on FPGAs," in *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, ser. Lecture Notes in Computer Science, E. Oswald and P. Rohatgi, Eds., vol. 5154. Springer, 2008, pp. 181–197.

[33] M. van Dijk and U. Rührmair, "Physical unclonable functions in cryptographic protocols: Security proofs and impossibility results," *IACR Cryptology ePrint Archive*, p. 228.

[34] H. Krawczyk, "Hmqv: A high-performance secure diffie-hellman protocol," in *Advances in Cryptology CRYPTO 2005*, ser. Lecture Notes in Computer Science, V. Shoup, Ed. Springer Berlin / Heidelberg, vol. 3621, pp. 546–566.

[35] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Towards Hardware-Intrinsic Security*, ser. Information Security and Cryptography, A.-R. Sadeghi and D. Naccache, Eds. Springer Berlin Heidelberg, pp. 3–37.

[36] C. Petrie and J. Connelly, "A noise-based ic random number generator for applications in cryptography," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 47, no. 5, pp. 615–621, may 2000.

[37] C. W. Odonnell, G. E. Suh, and S. Devadas, "Puf-based random number generation," in *In MIT CSAIL CSG Technical Memo 481 (http://csg.csail.mit.edu/pubs/memos/Memo-481/Memo-481.pdf*, 2004, p. 2004.

[38] G. E. Suh, C. W. Odonnell, I. Sachdev, and S. Devadas, "Design and implementation of the aegis single-chip secure processor using physical random functions," in *In Proceedings of the 32 nd Annual International Symposium on Computer Architecture (MIT-CSAIL-CSG-Memo-483 is*. ACM, 2005, pp. 25–36.

[39] T. Ignatenko, G.-J. Schrijen, B. Skoric, P. Tuyls, and F. Willems, "Estimating the secrecy-rate of physical unclonable functions with the context-tree weighting method," in *Information Theory, 2006 IEEE International Symposium on*, july 2006, pp. 499–503.

[40] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Ruhrmair, "The bistable ring puf: A new architecture for strong physical unclonable functions," in *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, june 2011, pp. 134–141.