Critical Analysis of New Protocols on Lightweight Authentication

Ying QIU
Communication and Network Department
Institute for Infocomm Research
Singapore 138632
qiuying@i2r.a-star.edu.sg

Jiqiang LU
Communication and Network Department
Institute for Infocomm Research
Singapore 138632
jlu@i2r.a-star.edu.sg

***Abstract*—The paper critically analyzes three recently proposed lightweight authentication protocols from three categories (i.e. CAN, PUF and RFID), and points out a few delicate issues in the three protocols, particularly: 1) LeiA lightweight authentication protocol for CAN might be vulnerable to key-recovery-attacks; 2) The lockdown technique to prevent machine learning on PUFs for lightweight authentication cannot prevent from flooding attacks; and 3) SLAP succinct and lightweight authentication protocol for low-cost RFID system, but lacks scalability and retrieval capability for mass tags.**

Keywords—authentication, lightweight, security, machine learning, CAN, PUF, RFID.

I. INTRODUCTION

According to a forecast from Statista, the number of installed Internet of Things (IoT) devices will be more than 31 billion in 2020, and grow up to 75 billion in 2025 [30]. On the other hand, Symantec reported that the number of attacks relevant to IoT devices increased more than 600% in 2017 [31]. Meanwhile, the most IoT devices are low capability devices or sensors. The situation is also very serious in the vehicle industry [4], [9], [11]. Hence, the security requirement is very keen to protect the massive communications among these low capability devices/sensors and the control centers.

In this paper, we critically analyze three lightweight authentication protocols that are proposed recently from different categories, i.e. CAN, PUF and RFID.

In Section 2, we investigate the protocol "LeiA: A Lightweight Authentication Protocol for CAN", which was proposed by Radu et al. in 2016 [1] for Controller Area Network (CAN), [6], [10]. CAN is a communication standard for vehicle. LeiA protocol allows ECUs to authenticate with each other and protects the system from many attacks. We observe that a single key can be used many times after sending a series of fake *AUTH_FAIL* messages, and as a consequence the protocol is vulnerable to key-recovery-attacks. We also give some suggestions to improve the protocol.

In Section 3, we investigate the protocol "A Lockdown Technique to Prevent Machine Learning on PUFs for Lightweight Authentication", which was proposed by Yu et al. in 2016 [2] on Physically Unclonable Functions (PUFs). Yu et al.'s protocol provides a hardware solution for a server to authenticate a device through machine learning approaches. We find that the protocol is not able to prevent from flooding attacks.

In Section 4, we study the protocol "SLAP: Succinct and Lightweight Authentication Protocol for low-cost RFID system", which was proposed by H. Lou, et al in 2018 [3] for Radio Frequency Identification (RFID). The SLAP protocol is only composed of bitwise operations such as XOR, left rotation and conversion, which are easy to implement on a passive tag. Its main challenge is the scalability for mass tags.

Section 5 gives some concluding remarks.

II. ANALYSIS OF LEIA LIGHTWEIGHT AUTHENTICATION PROTOCOL FOR CAN

A vehicle could be fully controlled maliciously if a single component in the vehicle was compromised, for example, Miller and Valasek hacked the Jeep Cherokee through the head unit [4], [8].

In 2016, Radu et al. [1] proposed LeiA protocol to protect the communication in CAN. LeiA is claimed to be the first AUTOSAR [5] compliant lightweight authentication protocol for CAN. In the protocol, an Electronic Control Units (ECUs) can be authenticated by one another, and thus many attacks can be defended. Radu et al. proved that the protocol can provide the security under the unforgeability assumption of the MAC scheme under chosen plaintext /message attacks [7]. The claimed contributions of the protocol include providing higher security levels (comparing with the protocols proposed previously) without additional hardware components or substantial implementation costs.

The protocol is suitable for the scenarios of automotive application with the constraints of stringent time and low bandwidth. The implementation can be flexible with the balance between the security and the overhead of the bandwidth/computation.

A. Procedure of the LeiA Protocol

In LeiA protocol, the authentication process consists of 4 steps: initialization, session key generation, authenticated messages translation, and resynchronization.

a) Initialization:
In the initial step, an initial function – *setup(h)* was run at both the prover and the verifier sides, then a shared secret s and public parameters ns are produced, where h is called the security parameters and (s, ns) is called the security tuple.

The secret parameter $s = \{K_{id_0}, \ldots, K_{id_{n-1}}\}$ is the set of long term symmetric keys K_{id} of n sensors, which are generated by the key generation algorithm $KG(1^h)$ for each identity id_i,. The public parameters $ns = \{(c_{id_0}, e_{id_0}), \ldots, (c_{id_{n-1}}, e_{id_{n-1}})\}$ are formed from the counter c_{id_i}

The work was supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore.

and the epoch e_{idi} of each identity id_i. The initial value is 0 for both the counter and epoch of each identity id_i. Then the session key K^e_{idi} for each identity id_i is generated by a session key generation function.

b) Session key generation:
A keyed hash function is used to compute the session key on both Sender *S* and Receiver *R* as below:

$$e_{idi} = e_{idi+1} \quad (1)$$
$$K^e_{idi} = MAC(K_{idi}, e_{idi}) \quad (2)$$
$$c_{idi} = 0 \quad (3)$$

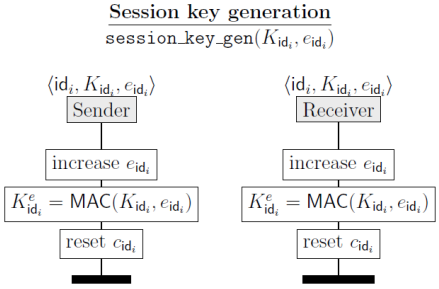

Fig 1. Session key generation between sender *S* and receiver R for message with identifier id_i.

c) Sending the Authenticated Messages:
Due to page limitation, we skip the description of the authentication step, since it does not affect the following analysis. Anyway, we refer the reader to the original paper [1] for detail.

d) Resynchronization:
If the authenticated verification is unsuccessful, an *AUTH_FALL* signal is sent from Receiver *R* to Sender *S*. After getting the *AUTH_FALL* message, the Sender *S* broadcasts a message, including its current e_{idi} *and* c_{idi} with their MAC_{id} encoded by its current session key K^e_{idi} :

$$MAC_{id} = MAC(K^e_{idi}, c_{idi}, e_{idi}) \quad (4)$$

If the received values of e_{idi} *and* c_{idi} are larger than the stored ones, Receiver *R* then updates the e_{idi} *and* c_{idi}. Otherwise it means the system is attacked by the replay attacks.

B. *Critical Analysis of the LeiA protocol*

The critical issue of the protocol is the synchronization for the epochs (e_{idi}) and the counters (c_{idi}).

Fig. 2 is snapped from the original paper in order to descript our analysis easily. Note that there is a typo in Fig.2, that is, the direction for the first message *AUTH_FAIL* should be from *Receiver* to *Sender*.

In the protocol, the epochs (e_{idi}) and the counters (c_{idi}) are defined as:

- The epoch e_{idi} is 56 bits and incremented at every vehicle start-up or when the counter c_{idi} overflows and participates in the generation of the session key;
- The counter c_{idi} is 16 bits and included in the Data Frame with the *MAC* to provide freshness.

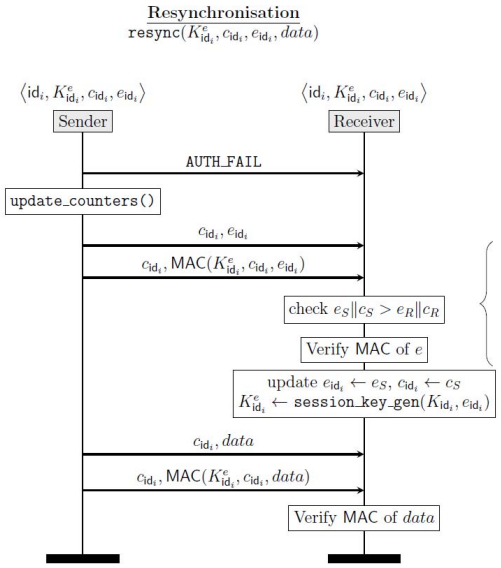

Fig 2. Message authentication failure and resynchronization procedure, between sender *S* and receiver *R* for message with identifier id_i.

The epoch e_{idi} is not able to synchronize due to many reasons, for example, the vehicle engine often needs more than one times to be initiated. Then, the e_{idi} at Sender side might be added by one, but the e_{idi} at Receiver side might not be updated because it did not receive the message from Sender. The communication packages among the ECUs can also be easily lost, as some ECUs are not connected to CAN directly, but through V850 in vehicles [12]. As a consequence, the session keys generated on Sender and Receiver would be different.

So an attack can be launched by following steps:

1) Since e_{idi} is often non-synchronised at both sides, the resynchronised process will be called frequently.
2) Then the generated session key K^e_{idi} is also non-synchronised frequently on the Sender and Receiver. A failed symmetric key will be needed to retrieve back the pre-stored key. Even the root key K_{idi} would be used in the process of new synchronised negotiation.
3) The c_{idi} and e_{idi} are sent in plaintext at second message in Fig. 2. According to the definitions of epoch e_{idi} and counter c_{idi}, it is easy to produce a special tuple of c_{idi} and e_{idi} by sending the *AUTH_FAIL* message continuously, because both of them are simply added by one each time and the count pool is not big enough. Thus, an attacker can tempt the victim Receiver to send forge *AUTH_FAIL* messages to Sender in order to obtain a specific tuple of c_{idi} and e_{idi}.
4) It means, from algorithm (2), a single key (K^e_{idi}) would be generated and used for a lot of times on a clear or even a specific tuple of c_{idi} and e_{idi}. Thus, it is vulnerable to such key-recovery-attacks as presented by Handschuh et al. [13] and Fouque et al. [14], i.e.:
   - From algorithm (4) that is transferred at the third message in Fig. 2, the K^e_{idi} can be recovered since

this key is used many times and the tuple of c_{idi} and e_{idi} is known or specified by the attacker.

- With the recovered K^e_{idi} and the known e_{idi}, the device's root key K_{idi} could also be recovered from algorithm (2).

C. Comments and Suggestions

The weakness of the protocol is not able to guarantee the synchronization between Sender and Receiver. An efficient synchronization is affected by many factors, such as the traffic lost, multi-attempts for the engine initiation.

Below are a few possible ways that we concern to improve the protocol:

- The first *AUTH_FAIL* message from Receiver to Sender may include a time-stamp in order to inform Sender the current status of the parameters in Receiver.
- Use a time-stamp (i.e. sequence number) instead of sending the epoch e_{idi} directly in the second message $\{c_{idi}, e_{idi}\}$ in Fig. 2.
- The increasing interval of e_{idi} could be variable.

III. Analysis of a Lockdown Technique to Prevent Machine Learning on PUFs

Usually, secure communication among low-cost resource-constrained devices and resource-rich server involves a lot of issues, such as:

1) The backbone networks are formed by secret keys.
2) The secret keys must be stored securely.
3) Non-volatile memory (NVM) is vulnerable to physical attacks.
4) Difficult to derive an exponential number of challenge/response pairs (CRPs) from a linearly-sized circuit, without a cryptographic algorithm and a secret or private key [15].
5) Probably-Approximately-Correct (PAC) theory[16].

And the following methods are adopted to solve these problems:

1) The PUFs offer physical security by not requiring static secret keys on the device to solve the issue of securely storing the keys [17~19].
2) Neither an error-correction code nor a cryptographic algorithm is employed in the scheme to avoid the issue of non-volatile memory in the low capability devices [20~24].
3) A new lockdown protocol that an adaptive chosen-challenge adversary with machine learning capabilities cannot obtain new CRPs from the traffic between devices and servers to solve the issue of deriving an exponential number of CRPs from a linearly-sized circuit without a cryptographic algorithm and a secret or private key [20~24].
4) Ensure an adversary not able to derive a PUF model with a limited amount of machine learning training data.
5) Use the threshold-based comparison to avoid the requirement of no error correction [25].

In 2016, Yu et al. [2] presented a lightweight PUF-based authentication scheme to secure the communications between a device with low capabilities and a server with powerful capabilities. Their solution is based on a server-managed challenge/response pair (CRP) lockdown protocol.

Compared with the prior approaches that an adversary can access the "open interface" to obtain new CRPs, Yu et al.'s lockdown scheme constrains the availability CRP material against obtained by an adaptive chosen-challenge machine-learning-equipped adversary. Hence an adversary cannot get new CRPs.

A. Procedure of the Lockdown Protocol

Yu et al.'s lockdown scheme involves two protocols. The target of the first protocol is to prevent an adversary from obtaining new CRPs without the server's permission, and the second protocol is to prevent using repeated measurements by mutual authentication. Here we focus on the latter protocol as it is more complicated and the result can also be adapted to the first one.

We highlight the major steps of the second protocol below:

1) The server generates an identity (*ID*) for each device with the device's serial number.
2) Using the Strong PUF (*SPUF*) [26] to setup a pair of challenge and response. Each device has its own authentication verification model $SPUF_i$.
3) When the server needs to verify a device, the device sends to the server its *ID* and a random number c_D, which can prevent the attack of repeated measurements using the same challenge.
4) After receiving the c_D, the server builds a challenge c with c_D and its new nonce c_S.
5) The server use *SPUF* to generate a response pair r_1 and r_2. Then the server sends the message of $c_S \,||\, r_1$ to the device.
6) Based on the c_S, the device also generates the r_1' and r_2' by the *SPUF*. After comparing the r_1' and r_1 by the function of *FHD* (Fractional Hamming Distance), sends the r_2' to sever.
7) Server also compares the r_2' and r_2 with the *FHD*.
8) If all verifications are successful, the authentication is true, else fail.

B. Critical Analysis of the Lockdown protocol

The advantages of the protocol are as follows:

1) The security of the devices relies upon the strong physically unclonable functions (*SPUF*s) instead of the static secret keys on the device. It avoids the risk of secret key interception.
2) No cryptographic algorithm is employed on the devices, (except PRNG and TRNG). It can guarantee the performance in the devices.
3) The new CRPs cannot be obtained by an adaptive chosen-challenge adversary with machine learning capabilities. It prevents the attacks from the machine learning adversary.

4) The *SPUF* model is difficult to be derived by an adversary who had captured a limited amount of machine learning training data.

5) A threshold-based comparison is deployed in order to avoid the requirement of no error correction.

The disadvantages of the protocol are also observed:

First, the security of the protocol is fully based on *SPUF*, and it is impossible to upgrade adaptively during the lifetime of the hardware components. The adversary is able to obtain all c_Ds from the traffic package $id||c_D$, since the *id* can be easily deduced from a few authentications. The c_S for the current authentication can be easily guessed from the c_S for the last authentication, as its value is simply increased with 1 by each event. Therefore, it is not difficult for an adversary to achieve the device secrets from the list of sub-challenges <*c*>. Further, the r_1 and r_2 (r_2') can be captured from the last two transferred messages in Fig 3, respectively. Hence, the *SPUF* is high risk to be simulated by the attackers who obtained series of the input and output pairs of the *SPUF*. Actually, an attack is more possible because the authentication checks are based on the condition that Fractional Hamming Distance does not exceed an authentication threshold □ instead of the exact equivalence of *r* and *r'*.

Second, referring to Fig. 3, there is no mechanism in the device side to check the validity of the input from the server before executing the expensive computations of *PRNG(c)* and *SPUF(<c>)*. Therefore, it can be easily attacked by flooding (DoS) attacks. Nowadays, flooding attacks are a serious type of attacks in reality.

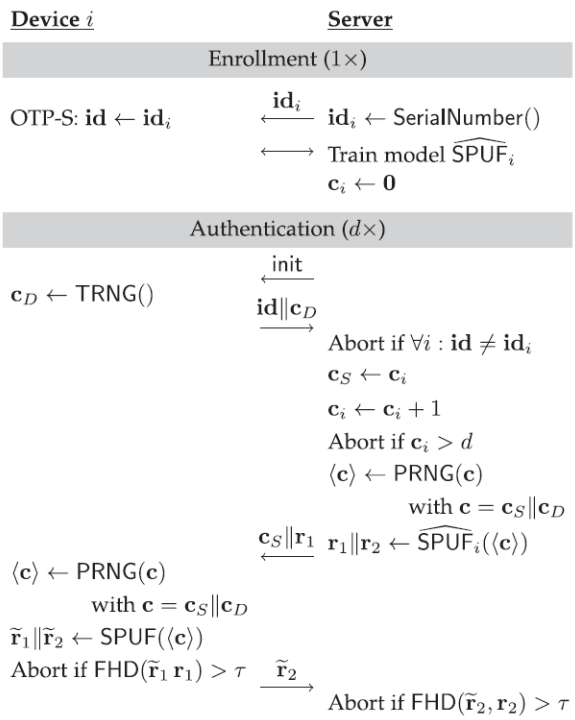

Fig 3. The 2nd Lockdown protocol prevents using repeated measurements with mutual authentication.

IV. Analysis of SLAP Succinct and Lightweight Authentication Protocol for Low-cost RFID System

For RFID system, the security of data is always struggling against the low capability and poor performance of the RFID devices [27~29].

In 2018, Luo et al. [3] proposed an ultra-lightweight authentication protocol --- SLAP succinct and lightweight authentication protocol for low-cost RFID system. SLAP protocol is claimed as ultra-lightweight because it adopts only three kinds of bitwise operations (namely, XOR, left rotation and conversion). Luo et al. think that the protocol is more secure due to its conversion properties: irreversibility, sensibility, full confusion and low complexity as well as less computation and short size of communication messages. Owing to these features, SLAP is more preferable for a low-cost RFID system.

A. Procedure of the SLAP Protocol

In SALP protocol, only there kinds of bitwise operations (i.e. *XOR* $\oplus$, left rotation, and conversion) are adopted which are easy to implement on a passive tag (e.g. RFID tag).

A novel conversion operation is one of the major contributions of the protocol. The conversion operation has 3 phases:

a) Grouping: At the first step, two related strings *A* and *B* were slipped into several small pieces according to the Hamming weights of their partner, respectively.

b) Rearrange: At the second step, both strings A and B are re-grouped according to the segmentation patterns of the partner.
- Regrouping string *A* into *k* groups (*Ak, k*<=*n*) depends on the segmentation way with *B*.
- Regrouping string *B* into *j* groups (*Bj, j*<= *n*) depends on the segmentation way with *A*.

c) Composition: *A'* and *B'* are the new groups of strings *A* and *B*, respectively. Operate $\oplus$ on *A'* and *B'* to get the conversion final result

$$Con(A,B) = A' \oplus B'.$$

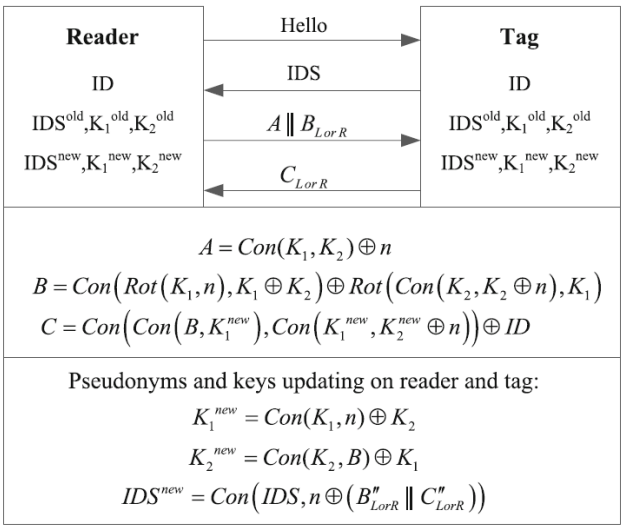

Fig 4. SLAP protocol

The authentication procedure of SLAP is below:

1) The reader (R) sends an initiate message to the tag (T).
2) T responds R with its pseudonym IDS.
3) R uses the received IDS as an index to search the corresponding entry from the database and computes:
   a) $A = \{Con(K_1, K_2) \oplus n\}$ is used to conceal the random nonce n with a mask. (1)
   b) $B = \{Con(Rot(K_1,n), K_1 \oplus K_2) \oplus Rot(con(K_2, K_2 \oplus n), K_1)\}$ is the authentication data used to confirm the legal status of R. (2)
   c) Where: $Rot(X, Y)$ denotes cyclic left rotation of X according to Y's Hamming weight. K_1 and K_2 are the 2 secret keys shared with the backend server. (3)
   d) Then R sends the 2 new group $A||B$ to T. (4)
4) After receiving messages from R, T extracts the n first, then computes the local B' with local key pair K_1 and K_2. If $B=B'$, continue; else, fail.
5) Update the keys:
   a) $K_1^{new} = Con(K_1,n) \oplus K_2$; (5)
   b) $K_2^{new} = Con(K_2, B) \oplus K_1$; (6)
   c) $IDS^{new} = Con(IDS, n \oplus (B'||C'))$; (7)
   d) $C = Con(Con(B, K_1^{new}), Con(K_1^{new}, K_2^{new} \oplus n)) \oplus ID$ (8)
   e) *Then T sends message C to R* (9)
6) Upon receiving C from T, T compute local C' and check if $C'=C$. if yes, the authentication is successful. Then R updates the K pair (K_1 and K_2) and new IDS (IDS^{new}) (10)

B. Critical Analysis of SLAP protocol

SLAP is a neat and tiny authentication protocol. Its major contribution is the conversion operation. In the first phase *a)*, the partner's Hamming Weights are used to increase the key's confusion; and in the second phase *b)*, the rotating operation increases the complexity. In order to avoid the message leaking, the protocol uses a half party of the string B and the string C for message exchanging and authenticating, meanwhile the other half are used for computing.

Anyway, there are a few delicate issues that should be taken into consideration thoroughly.

First, we observe that when one of the two keys A or B is *0xFFFF* the conversion operation is void, although it does not affect the security as the conversion operation never works along (always with other parameters).

Second, an acknowledgment is required from Reader to Tag at the end of the protocol in order to ensure the synchronization between them, otherwise the updating procedure may be done in Tag side, but not done on Reader side.

Third, the protocol lacks scalability. The protocol is based on the preshared keys. It is always a challenge to distribute and install the key pairs into a massive amount of tags.

At last, searching and matching the entries based on the pseudo *IDS*s in Step 3 is an extreme issue for a low-capability device, since IDSs are variable each time.

V. Conclusion

In this paper, we critically analyse three newly proposed lightweight authentication protocols from the three different categories CAN, PUF and RFID.

1) "LeiA: A Lightweight Authentication Protocol for CAN" that is suitable for the scenarios of automotive application with the constraints of stringent time and low bandwidth. But it could be vulnerable by key-recovery-attacks. We have provided some suggestions to improve the protocol, accordingly.

2) "A Lockdown Technique to Prevent Machine Learning on PUFs for Lightweight Authentication". Its pros don't need any pre-shared keys, can prevent machine-learning attacks, etc. Its cons are not able to prevent from flooding attacks (DOS) and vulnerable to the attackers who know both input and output of the SPUF.

3) "SLAP: Succinct and Lightweight Authentication Protocol for low-cost RFID system" that is neat and tiny algorithm for authentication. Its challenge is the scalability for the mass tags and the retrieval of pseudo identities.

Acknowledgment

This work was supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-31) and administered by the National Cybersecurity R&D Directorate.

References

[1] A.F. Radu, F. D. Garcia, "LeiA: A Lightweight Authentication Protocol for CAN". ESORICS 2016: 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part II (pp.283-300)

[2] M.-D. Yu, M. Hiller, J. Delvaux, R. Sowell, S. Devadas, and I. Verbauwhede. "A lockdown technique to prevent machine learning on PUFs for lightweight authentication". IEEE Transactions on MultiScale Computing Systems (TMSCS), 2(3):146–159, July 2016. doi: 10.1109/TMSCS.2016.2553027.

[3] H. Luo, G. Wen, J. Su, et al. "SLAP: Succinct and Lightweight Authentication Protocol for low-cost RFID system". Wireless Netw (2018) 24: 69. https://doi.org/10.1007/s11276-016-1323-y

[4] A. Drozhzhin. "The full story of how that Jeep was hacked". Black Hat USA 2015, August 2015. https://www.kaspersky.com/blog/blackhat-jeep-cherokee-hack-explained/9493/

[5] AUTOSAR: AUTOSAR Specification 4.2, http://www.autosar.org/ specifications/release-42/

[6] CAN: Controller Area Network standard ISO 11898, the International Organization for Standardization (ISO), 1993

[7] F. Grieu, "A Chosen Messages Attack". on the ISO/IEC 9796–1 Signature Scheme.

[8] A. Greenberg, "Hackers Remotely Killa Jeep", on the Highway–withmeinit (2015), http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[9] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, et. al "Comprehensive Experimental Analyses of Automotive Attack Surfaces". In: 20th USENIX Security Sym- posium (USENIX Security 2011). San Francisco (2011)

[10] ISO: 11898-1: 2003 - Road Vehicles - Controller Area Network. International Organization for Standardization, Geneva, Switzerland (2003)

[11] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaaˆniche, Y. Laarouchi, "Survey on Security Threats and Protection Mechanisms in Embedded Automotive Networks". In: Dependable Systems and

Networks Workshop (DSN-W 2013), 2013 43rd Annual IEEE/IFIP Conference on. pp. 1–12. IEEE (2013)

[12] J. Eltze, "Double-CAN Controller as Bridge for Different CAN Networks," in Proceedings of the 4th International CAN Conference, Erlangen, Germany, 1997.

[13] H. Handschuh, B. Preneel, "Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms". In: Wagner D. (eds) Advances in Cryptology – CRYPTO 2008. Lecture Notes in Computer Science, vol 5157. Springer, Berlin, Heidelberg

[14] PA. Fouque, G. Leurent, P.Q. Nguyen, "Full Key-Recovery Attacks on HMAC/NMAC-MD4 and NMAC-MD5". In: Menezes A. (eds) Advances in Cryptology - CRYPTO 2007. Lecture Notes in Computer Science, vol 4622. Springer, Berlin, Heidelberg

[15] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in Proc. 9th ACM Conf. Comput. Commun. Security, 2002, pp. 148–160.

[16] L. Valiant, "A theory of the learnable," Commun. ACM, vol. 27, no. 11, pp. 1134–1142, 1984.

[17] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications," in Proc. 2nd IEEE Int. Conf., 2008, pp. 58–64.

[18] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Controlled physical random functions," in Proc. 18th Annu. Comput. Security Appl. Conf., 2002, pp. 149–160.

[19] B. Gassend, "Physical random functions," in Master's Thesis, Dept. Electrical Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, 2003

[20] M. Majzoobi, M. Rostami, F. Koushanfar, D. Wallach, and S. Devadas, "Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching," in Proc. IEEE Symp. Security Privacy Workshops, 2012, pp. 33–44.

[21] M. Yu, D. M'Raihi, I. Verbauwhede, and S. Devadas, "A noise bifurcation architecture for linear additive physical functions," in Proc. IEEE Int. Symp. Hardware-Oriented Security Trust, 2014, pp. 124–129.

[22] J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, "A survey on lightweight entity authentication with strong PUFs," ACM Com- put. Surveys, vol. 48, no. 2, pp. 26:1–26:42, 2015.

[23] R. Maes, "Physically unclonable functions: Constructions, proper- ties and applications," in PhD Thesis, Dept. Electr. Eng., Katho- lieke Universiteit Leuven, Leuven, Belgium, 2012.

[24] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Secure lightweight entity authentication with strong PUFs: Mission impossible?" in Proc. 16th Int. Workshop Cryptographic Hardware Embedded Syst., 2014, pp. 451–475

[25] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: A very compact and a threshold implementa- tion of AES," in Proc. 30th Annu. Int. Conf. Theory Appl. Crypto- graphic Tech., 2011, pp. 69–88.

[26] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications," in Proc. 2nd IEEE Int. Conf., 2008, pp. 58–64.

[27] J. Shen, et al. "A practical RFID grouping authentication protocol in multiple-tag arrangement with adequate security assurance". In 2016 18th international conference on advanced communication technology (ICACT). IEEE.

[28] H.-Y. Chien. "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity". IEEE Transactions on Dependable and Secure Computing, 4(4), 337–340.

[29] A. Juels, D. Molnar, and D. Wagner. "Security and privacy issues in E-passports". In SecureComm 2005. First internationa conference on security and privacy for emerging areas in communications networks, 2005. IEEE.

[30] Statista, "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025", https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[31] Symantec, "Executive Summary: 2018 Internet Security Threat Report", https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-executive-summary-en.pdf .