Lightweight and Space-efficient Vehicle Authentication based on Cuckoo Filter

Charalampos Kalalas and Jesus Alonso-Zarate
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona, Spain
Emails: {ckalalas, jesus.alonso}@cttc.es

***Abstract*—While the emerging vehicle-to-everything (V2X) connectivity paradigm is radically transforming the automotive sector, unprecedented security challenges arise, calling for innovative security enablers with minimum impact on the ongoing communication. In dense V2X scenarios, the 5G authentication and key agreement (5G-AKA) procedure may suffer from uncontrolled failures which result in unacceptable latency levels due to the excessive signalling overhead. In this paper, we introduce a lightweight vehicle authentication scheme, as an extension of the 5G-AKA, to adequately address a high number of authentication requests. The proposed mechanism leverages the space-efficient features of the cuckoo filter, a probabilistic data structure for approximate set membership tests, to achieve authentication of multiple vehicles at a time. Our performance analysis reveals the impact of various cuckoo filter parameter configurations on the authentication efficiency. In addition, our proposed authentication mechanism is able to outperform the standardized 5G-AKA procedure in terms of latency and protocol overhead even for high vehicle load.**

***Index Terms*—Vehicle authentication, Internet of Vehicles (IoV), 5G-AKA, Cuckoo filter**

I. Introduction

During the recent years, the automotive industry is rapidly transforming towards the realization of the Internet of Vehicles (IoV) paradigm where vehicles become increasingly aware of their surroundings and capable of communicating with each other, with road-side units, and with other road users, e.g., pedestrians, cyclists, etc. In this context, 5G (and beyond) systems are expected to play a pivotal role, offering ubiquitous vehicle-to-everything (V2X) connectivity for increased road safety, optimized driving decisions, and real-time traffic control. With the increasing level of driving automation in 5G-enabled vehicular use cases, V2X communication becomes highly vulnerable to malicious actors, opening up entirely new questions from a security and privacy perspective that have not been addressed in a similar context before [1]. In addition, the transmission of a wide range of sensitive data, e.g., vehicle's trajectory and speed, needs to be private and secured enough to avoid issues such as degraded safety, trajectory tracking, generation of false alarms for road hazards, congestion, etc.

A secure and privacy-aware network architecture is therefore required to guarantee the level of vehicle identities and protect vehicular data by ensuring the authentication of the message senders in vulnerable V2X scenarios. In this context, the 5G authentication and key agreement (5G-AKA) constitutes one of the fundamental procedures for mutual authentication between each vehicle and the network and provides keying material that can be used in subsequent security procedures [2]. Following the rationale of 4G systems, the 5G-AKA mechanism is governed by secure key exchanges assuming different roles for each of the involved network entities. However, in highly dense IoV scenarios, the excessive signalling overhead required for security context establishment in 5G-AKA may result in increased latency beyond the acceptable levels. This is especially important for *i*) mission-critical V2X use cases, e.g., road safety, where vehicle authentication should have minimum impact on the actual communication, and *ii*) roaming scenarios, where the signalling between the serving and the home network domains may introduce non-negligible latency.

Several vehicle authentication protocols have been recently proposed in the literature for mutual authentication among the involved network entities [3, and references therein]. Group-based AKA approaches, e.g., [4], allow the serving network to authenticate clusters of vehicles and reduce the message exchanges with the home network. However, the dynamic V2X network topologies result in frequent local signalling for cluster formulation and head selection which may prove harmful for the V2X data exchange when the vehicle density exceeds a certain level. On the other hand, lightweight authentication protocols, e.g., [5], may reduce the required computation cost at the expense of a higher communication overhead in terms of required size of the exchanged messages.

The potential of probabilistic data structures, such as the Bloom filter [6] and cuckoo filter [7], [8], for message authentication tasks has been only recently explored in the literature. A signature-based access protocol relying on Bloom filter is proposed in [6] aiming to integrate the authentication process with the random access channel procedure. The authors in [7] introduce a privacy-preserving vehicle authentication scheme where the cuckoo filter properties are exploited for batch message verification without the need of employing bilinear pairings. In [8], the cuckoo filter data structure is used to protect user location privacy and ensure service authentication in delay-tolerant vehicular scenarios. However, the stringent V2X performance requirements in terms of latency and protocol overhead are not explicitly taken into account in the design of the aforementioned protocols. To the best of our knowledge, the applicability of the cuckoo filter structure in the context of the 5G-AKA procedure has not been investigated so far.

Contribution: Motivated by these literature gaps, our contribution in this paper is twofold:

- We propose a novel vehicle authentication mechanism

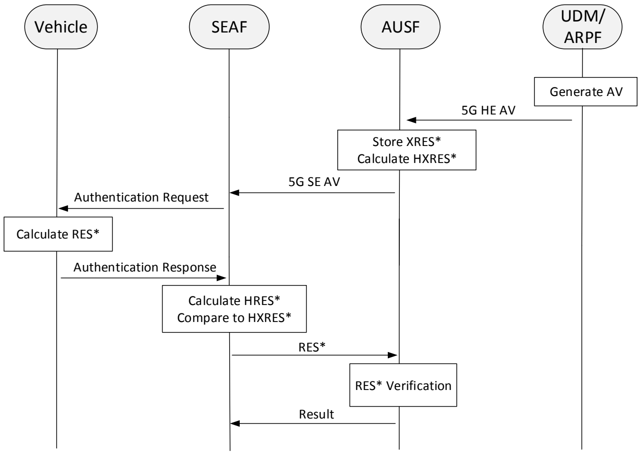

Fig. 1. Signalling flow in the 5G-AKA procedure [2].

aiming to extend the 5G-AKA procedure and address highly dense V2X connectivity scenarios. Our proposed scheme inherits the space-efficient advantages of a cuckoo filter implementation and allows for the authentication of multiple vehicles at a time with controllable false positive rates.

- We conduct an in-depth performance analysis of our vehicle authentication scheme to study the impact of different filter configurations for varying vehicle load. A properly designed cuckoo filter can significantly improve the authentication efficiency and outperforms the standardized 5G-AKA scheme in terms of end-to-end latency and protocol overhead even for high vehicle load. In addition, the introduced space cost remains close to the information-theoretic lower bound even for stringent false positive rate requirements.

Organization: The remainder of the paper is organized as follows. Section II describes the basic steps of the 5G-AKA procedure and the specific roles/operations of each involved entity for vehicle authentication. In Section III, our proposed vehicle authentication mechanism is presented and a thorough performance analysis aims to shed light on the impact of various cuckoo filter configurations on the achieved performance. The authentication efficiency of our proposed scheme is evaluated in Section IV and a performance comparison against the standardized 5G-AKA procedure is also conducted. Finally, Section V concludes the paper.

II. The 5G-AKA Procedure

The purpose of the primary 5G-AKA procedure is to enable mutual authentication between the vehicle and the network and to provide keying material that can be used in subsequent security procedures [2]. As shown in the system architecture of Fig. 1, the key 5G functional elements involved in the signalling between the vehicle and the network for mutual authentication are the following:

1) Security anchor function (SEAF): The SEAF is a security anchor which may be co-located with the access and mobility management function (AMF) in the serving network (or visited network, in the case of roaming scenarios) domain. It performs authentication at the serving network level and one of its roles is to generate a unified anchor key that can be used by the vehicle and the serving network to protect the subsequent message exchange.
2) Authentication server function (AUSF): The AUSF is handling authentication requests in the home network and implicitly performs serving network authorization via interaction with the SEAF. It provides authentication functionalities through message exchange with the unified data management (UDM), e.g., it notifies UDM for successful/unsuccessful authentication of a vehicle.
3) Authentication credential repository and processing function (ARPF): The ARPF constitutes a functional element of the UDM in the home network domain. It stores long-term credentials, e.g., the vehicle's subscriber key, used to uniquely identify a subscription and mutually authenticate the vehicle and the 5G core network.

The basic steps of the mutual authentication signalling flow along with the roles/operations of the involved 5G functional elements can be summarized as follows:

1) After successfully completing the random access channel procedure, each vehicle transmits a registration request message to initiate the authentication procedure. Upon the reception of this message, the SEAF invokes the authentication service by sending an authentication request message to the AUSF, including the serving network name. In turn, based on the serving network name, the AUSF notifies the UDM in the home network. In particular, the ARPF executes cryptographic algorithms based on the long-term security credentials and generates the 5G home environment authentication vector (5G HE AV).
2) The UDM returns the 5G HE AV to the AUSF which, in turn, extracts the expected response (XRES*), stores its value, and generates the 5G serving environment authentication vector (5G SE AV). The hash expected response (HXRES*) is one of the key fields included in the 5G SE AV and it is later used for verification. The AUSF then sends the 5G SE AV to the SEAF which extracts the random challenge (RAND) and the authentication token (AUTN). The RAND and AUTN values constitute part of the authentication request message subsequently sent by the SEAF to the vehicle.
3) Upon the reception of the authentication request message, the vehicle verifies the freshness of the AUTN and authenticates the network by locally computing the output of a network authentication function. The vehicle also derives the RES* value which is then sent as part of the authentication response message to the SEAF. In this step, the network-towards-vehicle authentication has been completed.

4) In turn, the SEAF calculates the hash response (HRES*) and compares it with the HXRES* received from AUSF in Step 2 for authentication. If they coincide, the SEAF shall consider the authentication successful from the serving network point of view. The SEAF then sends an authenticate-request message to the AUSF containing the RES* received from the vehicle. A comparison between the RES* and the stored XRES* (i.e., part of the 5G HE AV received from the UDM) is then performed at the AUSF. If the RES* and XRES* are equal, the AUSF shall consider the authentication successful from the home network point of view and will send an authentication event notification message to UDM declaring "success".

In the next section, we present in detail our proposed authentication mechanism that extends the 5G-AKA scheme allowing for the authentication of multiple vehicles at a time.

III. Proposed Vehicle Authentication

We consider a scenario where a high number of vehicles transmit in a near-simultaneous manner registration request messages to initiate their authentication procedure. The proposed authentication mechanism aims to minimize the required signalling for the security context establishment between the home/serving network and the vehicles. In this regard, we leverage the space-efficient features of a cuckoo filter data structure at the expense of introducing false positives in a controlled rate. We provide the implementation details in the following.

A. Cuckoo filter basics

A cuckoo filter offers a compact probabilistic way to represent an item by storing its fingerprint, i.e., a bit string derived from the item using a hash function, in a hash table. The fingerprint length can be determined according to the target false positive rate $\hat{p}_{\text{fp}}$. Compared to similar data structures, such as the Bloom filter [9], the cuckoo filter not only supports dynamic item insertion and deletion, but also achieves higher lookup performance in terms of time and space efficiency. A cuckoo hash table consists of an array of buckets and each inserted item has two candidate buckets determined by the hash functions h_1 and h_2. In particular, the indices of the candidate buckets for an item X are computed based on a partial-key cuckoo hashing [10], as $i_1 = h_1(X)$ and $i_2 = i_1 \oplus h_2(f(X))$, where $f(X)$ denotes the fingerprint of item X.

As illustrated in the item insertion example in Fig. 2, the item X can be placed in either buckets 2 or 6 of the hash table with a total number of 7 buckets. If either of the buckets is empty, the algorithm inserts X to that free bucket and the insertion completes. If neither bucket is free, as in this example, the algorithm selects one of the candidate buckets (e.g., bucket 6) for X, removes the existing item (in this case A), and re-inserts it to its own alternate location. This dynamic insertion process may require removing an additional item and, thus, it may repeat until a vacant bucket is found or a maximum number of displacements is reached. A set membership query for item X simply searches both buckets of the hash table for the fingerprint of X, and returns true if an identical fingerprint is found.

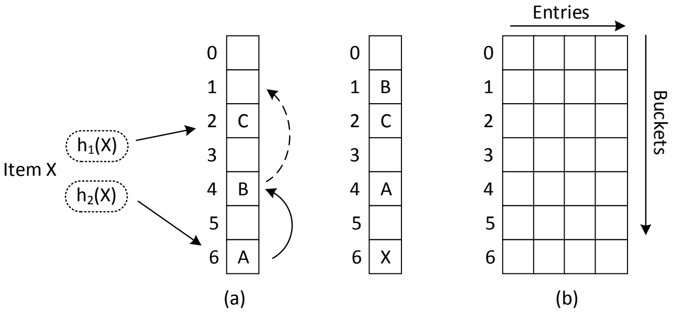

Fig. 2. (a) Item insertion in a cuckoo filter. (b) A cuckoo filter may have multiple entries per bucket [10].

B. Authentication mechanism based on cuckoo filter

Following the cuckoo filter principles described before, we hereby propose an extension of the 5G-AKA procedure tailored to minimize signalling exchanges for vehicle authentication in highly dense IoV scenarios. In particular, our approach introduces the following protocol extensions:

1) Cuckoo filter generation at AUSF: According to the Step 1 of the 5G-AKA procedure, the UDM/ARPF generates a 5G HE AV corresponding to each registration request message transmitted by every vehicle. In turn, the AUSF derives the XRES* values from the received 5G HE AVs and generates a cuckoo filter, henceforth denoted as CF(XRES*). In particular, the AUSF performs the item insertion operation, as described in Section III-A, for each XRES* and stores the fingerprints corresponding to all vehicles that requested registration to the network. Each stored fingerprint in the CF(XRES*) is thus unequivocally associated with each vehicle. The 5G SE AV is then constructed as 5G SE AV = RAND || CF(XRES*) || AUTN, where || denotes the message concatenation operation, and it is subsequently sent to the SEAF.
2) RES* verification at SEAF: Upon receiving an authentication response message by each vehicle, as indicated in Step 3 of the 5G-AKA procedure, the SEAF extracts the RES*, calculates its fingerprint, and performs the set membership query operation in the CF(XRES*). In particular, the SEAF calculates the output of the hash functions $h_1(\text{RES*})$ and $h_2(\text{RES*})$ to derive the indices of the two candidate buckets where the fingerprint of RES* may be stored. If, in neither of the two locations in CF(XRES*), the stored fingerprint of XRES* does not coincide with the fingerprint of RES*, then authentication fails. Otherwise, if the fingerprint of XRES* is identical to the fingerprint of RES* in either of the two buckets, then the authentication may be successful, as a false positive may have occurred.
3) Notification from SEAF to AUSF: After the end of the RES* verification phase, the SEAF needs to notify the

AUSF about the outcome of the authentication process. In particular, leveraging the item deletion operation of the cuckoo filter, the SEAF removes the non-matching fingerprints in the CF(XRES*), leaving stored only those fingerprints correctly authenticated during the previous phase. The CF(XRES*) is then transmitted to the AUSF as part of the authenticate-request message.

C. Analysis

The basic operations of the CF(XRES*), i.e., item insertion, query, deletion, are independent of the hash table configuration. However, the parameter configuration of the CF(XRES*) can significantly affect its authentication performance. In what follows, we analyse the CF(XRES*) performance in terms of two fundamental metrics: *i*) the XRES* insertion failure probability and *ii*) the false positive probability.

Let us consider the CF(XRES*) construction process where the AUSF has received multiple XRES* from the UDM corresponding to a number of N vehicles requesting registration to the network (out of T vehicles in total). The AUSF inserts the various XRES* in the empty filter of $m = cN$ buckets with b entries per bucket for a constant $c > 0$. Let also f denote the fingerprint size (in bits) and assume that a reference XRES* has entered in its bucket. According to the item insertion principles of the cuckoo filter, if $j-1$ XRES* have the same buckets with the reference XRES*, the following must simultaneously hold: *i*) their location indices derived from h_1 and h_2 are the same, which occurs with probability $2/m$, and *ii*) they have the same fingerprint, which occurs with probability $1/2^f$. Therefore, the conditional probability of such j XRES* sharing the same two buckets is $\left(2/m \cdot 1/2^f\right)^{j-1}$ and when $j = 2b+1$, the insertion fails. Thus, the XRES* insertion failure probability is given by

$$p_c = \left(\frac{2}{2^f \cdot m}\right)^{2b} = \left(\frac{2}{2^f \cdot cN}\right)^{2b}, \qquad (1)$$

while the expected number of colliding XRES* is derived as $\binom{N}{2b+1} \cdot p_c$, as there are $\binom{N}{2b+1}$ different combinations of $2b+1$ XRES* out of N in total.

Fig. 3 illustrates the XRES* insertion failure probability as a function of the number of vehicles N for various cuckoo filter configurations. As expected, the number of colliding XRES* increases with increasing vehicle load due to the higher contention. However, collisions rapidly drop when a larger bucket size is considered; as the number of entries per bucket increases, the set of candidate locations for the inserted XRES* expands, resulting in a lower probability for groups of $2b+1$ XRES* colliding during the filter construction process. Alternatively, collisions can be reduced when a longer fingerprint size is used, since the probability of an exact fingerprint match becomes lower.

Due to the probabilistic nature of the cuckoo filter, false positives, i.e., non-existent XRES*, may occur during the filter construction process at the AUSF. The false positive probability p_{fp} is then defined as the probability that the SEAF returns a successful match for a non-existent XRES* when querying the $2b$ filter entries during the lookup operation. Given that in each entry the probability that a query matches a stored fingerprint and returns a false positive successful match is $1/2^f$, the total probability of a false fingerprint hit can be expressed as

$$p_{\text{fp}} = 1 - \left(1 - 1/2^f\right)^{2b} \approx 2b/2^f. \qquad (2)$$

The impact of the CF(XRES*) parameter configurations on p_{fp} is shown in Fig. 4. It can be observed that p_{fp} rapidly decreases with increasing f, as a longer fingerprint size renders the false positive match less probable. It is also worth noting that as the number of entries per bucket increases, p_{fp} also increases, since for larger size of buckets, each lookup checks more entries and thus has a higher chance to identify fingerprint matches. Larger buckets therefore require longer fingerprints to retain the same false positive rate. Setting a target false positive probability $\hat{p}_{\text{fp}}$, i.e., $2b/2^f \leq \hat{p}_{\text{fp}}$ according

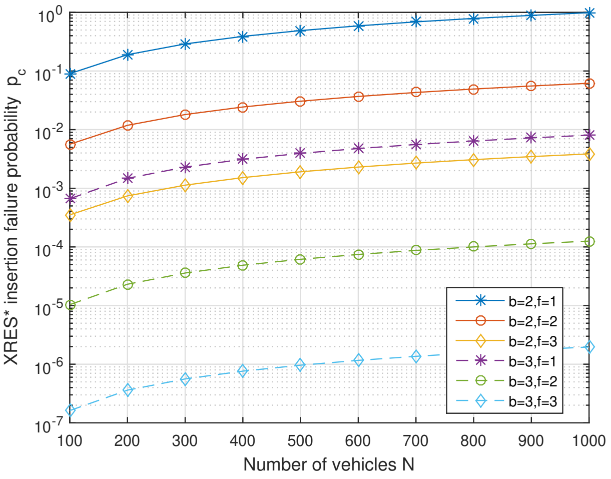

Fig. 3. XRES* insertion failure probability for various cuckoo filter configurations and increasing vehicle load.

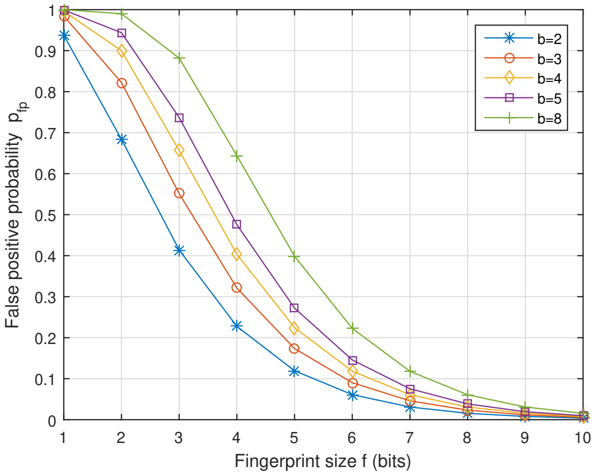

Fig. 4. False positive probability for various cuckoo filter configurations.

to Eq. (2), the minimum required fingerprint size (in bits) can be derived as

$$f_{\min} = \lceil \log_2(1/\hat{p}_{\text{fp}}) + \log_2(2b) \rceil \;\; \text{bits.} \tag{3}$$

Due to the inter-dependency of Eqs. (1)-(3) on the filter parameters b and f, we hereby define two new performance metrics aiming to integrate their impact on the authentication performance. In particular, in an effort to reflect the efficiency of the proposed authentication scheme, we define the average goodput as

$$\mathrm{E}[G] = \frac{N - \binom{N}{2b+1} p_c}{N + p_{\text{fp}}(T - N)}, \tag{4}$$

where the numerator accounts for the non-collided XRES* during insertion while the second term in the denominator corresponds to the average number of false positives in CF(XRES*) from a number of $(T - N)$ non-existent XRES*. In addition, in order to quantify the space efficiency of the proposed scheme, we introduce the space cost per stored XRES*, C, as the average number of bits required to represent each XRES*. To account for the insertion failures which may leave some filter entries unoccupied [10], we can express C as

$$C = \frac{f_{\min}}{1 - p_c}, \tag{5}$$

where $f_{\min}$ is calculated using Eq. (3) and the denominator represents the average load factor of the filter.

In the following section, a detailed performance assessment of our proposed authentication scheme is presented based on the previously derived performance metrics.

IV. Results and Discussion

Based on the conducted analysis of our authentication scheme, the aim of this section is twofold: *i*) to assess the authentication efficiency in terms of goodput performance and space cost per XRES* for various cuckoo filter parameter configurations; *ii*) to provide a performance comparison in terms of latency and protocol overhead of our proposed mechanism against the standardized 5G-AKA procedure. For the latter, a simplified event-driven simulator of the involved functional entities and exchanged signals, as depicted in Fig. 1, has been implemented. In the simulation setup, N out of T vehicles generate near-simultaneous registration requests and as $N \rightarrow T$, the network is progressively driven to overload. Thus, starting from a medium-load scenario, the authentication performance can be evaluated under different filter configurations as the system operates close to its capacity limits.

Fig. 5 depicts the average goodput performance with increasing vehicle load, for a fingerprint size of $f = 4$ bits and a varying number of entries per bucket b. It can be observed that, while b is not a binding parameter in the high vehicle load regime, the goodput performance for lower vehicle loads largely depends on b, due to the presence of false positives. In particular, as b increases, the p_{fp}, as shown in Eq. (2), also increases resulting in high performance degradation for a relatively low value of fingerprint size. However, as illustrated in Fig. 6, a greater value of f, i.e., $f = 8$ bits, is able to counteract the harmful impact of a large b in the goodput performance. On the other hand, a longer fingerprint size comes at the cost of increased space cost per XRES*, as shown in Eq. (5). The resulting trade-off between $\mathrm{E}[G]$ and C largely depends on the target false positive probability $\hat{p}_{\text{fp}}$ which is typically application-dependent.

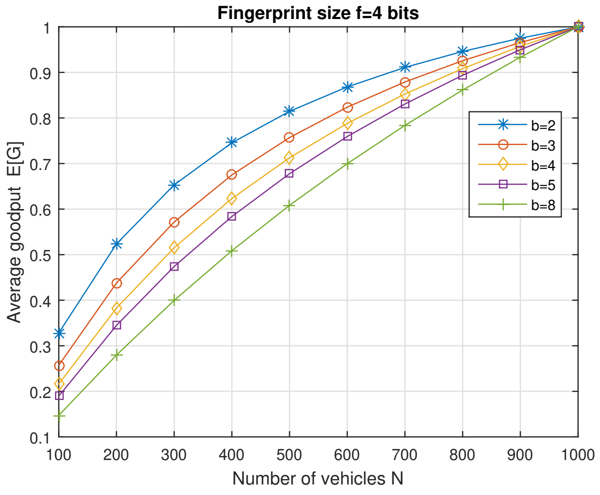

Fig. 5. Average goodput performance for various bucket sizes and increasing vehicle load ($f = 4$ bits).

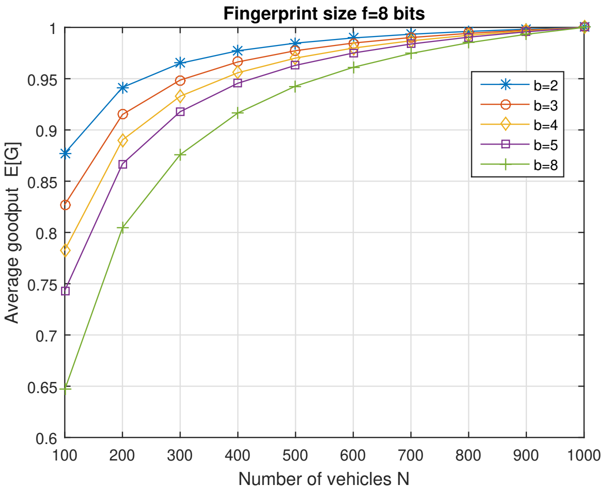

Fig. 6. Average goodput performance for various bucket sizes and increasing vehicle load ($f = 8$ bits).

A performance comparison between our proposed authentication scheme and the default 5G-AKA procedure is illustrated in Fig. 7. In order to perform a fair comparison of the two authentication procedures, we have made the following assumptions: *i*) the latency introduced due to the signal exchange between the home and the serving network has the same (deterministic) value in both schemes; and *ii*) the timeout period for the identification of an expired authentication vector

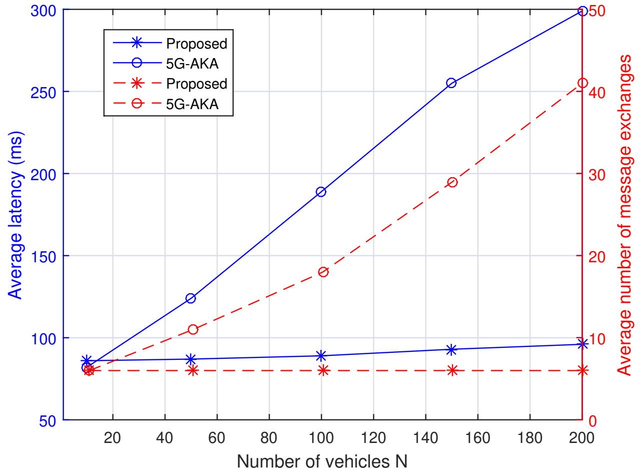

Fig. 7. Latency and protocol overhead performance of our proposed scheme and the 5G-AKA procedure for increasing vehicle load.

TABLE I
Space Efficiency Comparison

Required bits per XRES* to achieve $\hat{p}_{\text{fp}}$			
Data structure	Target false positive probability $\hat{p}_{\text{fp}}$		
	10^{-5}	10^{-4}	10^{-3}
Space-optimized Bloom filter	$\lceil 23.9179 \rceil$	$\lceil 19.1343 \rceil$	$\lceil 14.3507 \rceil$
Cuckoo filter ($b = 2, f = 4$)	$\lceil 18.6141 \rceil$	$\lceil 15.2914 \rceil$	$\lceil 11.9687 \rceil$
Entropy lower bound	$\lceil 16.6096 \rceil$	$\lceil 13.2877 \rceil$	$\lceil 9.9658 \rceil$

at each vehicle has been also set to an equal value in both schemes. A superior performance of our proposed mechanism in terms of average authentication latency per vehicle can be observed, even for high vehicle load. In addition, our scheme leverages the space-efficiency gains of the cuckoo filter implementation to keep at minimum the message exchanges required for successful authentication. On the other hand, 5G-AKA procedure suffers from uncontrolled latency and increasing number of signal interactions. This is mainly due to the high queuing delay in satisfying the authentication requests from a high number of vehicles which in turn leads to frequent expirations of the authentication tokens and re-initiation of the authentication procedure.

Finally, the space-efficiency gains of the cuckoo filter implementation are compared in Table I for various target $\hat{p}_{\text{fp}}$ with those achieved by a space-optimized Bloom filter, an alternative randomized data structure for test membership [9]. According to [10], the required number of bits per inserted XRES* for a space-optimized Bloom filer depends only on the target $\hat{p}_{\text{fp}}$, i.e., $C_{\text{Bloom}} = \lceil 1.44 \log_2(1/\hat{p}_{\text{fp}}) \rceil$ bits. Instead, based on Eq. (5), the space cost for the cuckoo filter varies according to the parameter configurations. For the chosen set of b and f values, it can be observed that the cuckoo filter achieves superior space efficiency for various stringent values of target $\hat{p}_{\text{fp}}$, as imposed by mission-critical services in V2X scenarios. It is also worth noting that the space cost remains closer to the information-theoretic lower bound of $\lceil \log_2(1/\hat{p}_{\text{fp}}) \rceil$ bits [11].

V. Conclusions

A lightweight and space-efficient vehicle authentication scheme tailored for mission-critical IoV scenarios is introduced in this paper. Our proposed mechanism leverages the advantages of a cuckoo filter implementation to achieve high authentication efficiency and space cost gains even for dense vehicular scenarios. A performance comparison with the standardized 5G-AKA procedure reveals the superiority of our approach in terms of end-to-end latency and protocol overhead. Future work aims to enhance the proposed vehicle authentication mechanism by exploiting the deployment of road-side units which are capable of providing message verification via broadcast messages in neighbouring vehicles.

Acknowledgment

This work has been partially supported by the H2020-INSPIRE-5Gplus project (under grant agreement No. 871808), by the CHIST-ERA FIREMAN project funded by the Spanish national foundation (PCI2019-103780), by the Spanish MINECO project SPOT5G (TEC2017-87456-P), and by the Generalitat de Catalunya under Grant 2017 SGR 891.

References

[1] G. Arfaoui, P. Bisson, R. Blom, R. Borgaonkar, H. Englund, E. Félix, F. Klaedtke, P. K. Nakarmi, M. Näslund, P. O'Hanlon, J. Papay, J. Suomalainen, M. Surridge, J. Wary, and A. Zahariev, "A Security Architecture for 5G Networks," *IEEE Access*, vol. 6, pp. 22466–22479, 2018.

[2] ETSI TS 133 501 v15.4.0, "5G; Security architecture and procedures for 5G System ," May 2019.

[3] P. Bagga, A. K. Das, M. Wazid, J. J. P. C. Rodrigues, and Y. Park, "Authentication Protocols in Internet of Vehicles: Taxonomy, Analysis, and Challenges," *IEEE Access*, vol. 8, pp. 54314–54344, 2020.

[4] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure Message Communication Protocol Among Vehicles in Smart City," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4359–4373, 2018.

[5] M. Wazid, A. K. Das, N. Kumar, V. Odelu, A. Goutham Reddy, K. Park, and Y. Park, "Design of Lightweight Authentication and Key Agreement Protocol for Vehicular Ad Hoc Networks," *IEEE Access*, vol. 5, pp. 14966–14980, 2017.

[6] N. K. Pratas, S. Pattathil, C. Stefanović, and P. Popovski, "Massive machine-type communication (mMTC) access with integrated authentication," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[7] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "SPACF: A Secure Privacy-Preserving Authentication Scheme for VANET With Cuckoo Filter," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10283–10295, 2017.

[8] J. Ni, X. Lin, and X. Shen, "Toward Privacy-Preserving Valet Parking in Autonomous Driving Era," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2893–2905, 2019.

[9] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM*, vol. 13, pp. 422–426, 1970.

[10] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo Filter: Practically Better Than Bloom," in *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, (New York, NY, USA), p. 75–88, Association for Computing Machinery, 2014.

[11] S. Lovett and E. Porat, "A Lower Bound for Dynamic Approximate Membership Data Structures," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 797–804, 2010.