Vehicular PKI Scalability-Consistency Trade-Offs in Large Scale Distributed Scenarios

Pierpaolo Cincilla[1], Omar Hicham[1,2], and Benoit Charles[3]

[1]IRT SystemX, 8 avenue de la Vauve, 91120, Palaiseau, e-mail: pierpaolo.cincilla@irt-systemx.fr
[2]University of Pierre and Marie Curie, 4 Place Jussieu, 75005 Paris, e-mail: omar.hicham@etu.upmc.fr
[3]IDnomic, 175 Rue Jean Jacques Rousseau, 92130 Issy-les-Moulineaux, e-mail: benoit.charles@idnomic.com

***Abstract*—Intelligent Transport Systems (ITS) are becoming a mature technology: standardisation progress and pre-deployment projects are opening the way to smart mobility.**

Vehicle–to–Vehicle (V2V) and Vehicle–to–Infrastructure (V2I) communications are paramount to cooperative awareness and safety applications, and must be protected. In recent years, a lot of research has been focusing on vehicular communication security and privacy. Current ITS communication security functionalities include message authentication, which has an impact on the privacy of vehicles and drivers. At the European level, the European Telecommunications Standards Institute (ETSI) has defined a message authentication mechanism based on a Public Key Infrastructure (PKI). In order to serve a huge amount of ITS Stations (ITS–Ss), the Vehicular Public Key Infrastructure (VPKI) has high scalability requirements. To the best of our knowledge, so far no one has investigated the VPKI scalability in large-scale deployment.

In this paper, we present the first extensive measurement campaign of a fully functional ETSI-compliant PKI. In our measurement campaign we assess PKI performance and scalability, replicating the system on hundreds of machines. In particular, we evaluate different replication strategies in terms of performance and consistency implications.

***Index Terms*—Public Key Infrastructure, Security Credential Management System, Security, Privacy, C-ITS, Scalability, Cloud**

I. Introduction

Intelligent Transport Systems (ITS) have been receiving growing attention in recent years. Vehicle manufacturers and IT companies are working on related technologies (e.g. positioning systems, object-detection, etc.) and use cases (e.g. valet parking, platooning, etc.) and several prototypes are now out on the streets.

As the technology matures, standards organisations and pre-deployment tests are paving the way for the future autonomous transport systems. Both the ETSI TC ITS WG5 working group in the European Union (EU) [1] and the IEEE 1609.2 working group in United States (US) [2] deliver a set of standards for ITS. At the same time, several pre-deployment projects such as SCORE@F, PRESERVE, CORRIDOR and SCOOP@F (see [3], [4] for a survey), are testing the underlying technologies in real scenarios.

Communications between vehicles and infrastructures are fundamental to improve traffic management, road safety, mobility and comfort services. As the vehicles (and other ITS–Ss) get connected to the network, the security risks augment [3]. Attacks against Vehicular Ad hoc Networks (VANETs) are a threat not only to user privacy but also to user safety [5]–[8]: V2V and V2I communications (also called Vehicle–to–X (V2X) communications) must be protected.

In order to protect V2X communications, ETSI has defined a PKI-based message authentication mechanism as the basis to establish trust between ITS–Ss [4]. The authentication process has to guarantee both the ITS–S identity and its privacy. Message signatures can be used to track vehicles, raising privacy concerns. As ITS–Ss will periodically broadcast messages to their neighbours (e.g. beacon messages for enhanced cooperative awareness), the content of exchanged messages has to be treated as personal data, and vehicle privacy must be guaranteed.

The privacy protection mechanism design is not trivial because it should consider several ITS-specific constraints: safety applications require that ITS–Ss should be observable, and safety messages are not encrypted to be widely accessible. Moreover, the on-board units have limited computational power, and for safety messages, authentication is critical.

The message authentication schema defined by ETSI (see ETSI Technical Specifications (TS) 103 097 [9], ETSI TS 102 940 [1] and ETSI TS 102 941 [10]) meets the two objectives: ensure message security (authenticity and integrity) while preserving user and vehicle privacy. In order to provide privacy without harming the operational behaviour of applications, the ETSI authentication system is based on the use of pseudonym certificates that are not easily linkable between them. Pseudonym certificates allow ITS–Ss to communicate without disclosing their identity, and pseudonym change makes it hard to track them.

Vehicles and infrastructures will be more and more connected, making VPKI scalability a must. As the development and testing of VPKI prototypes go on, and with commercial deployment around the corner, addressing the VPKI scalability becomes necessary. So far, there are few ETSI-compliant VPKI implementations (see [11], [12]) and, to the best of our knowledge, none of them has addressed the scalability issue in a large-scale deployment. In this paper, we explore the scalability-consistency trade-offs of an ETSI-compliant VPKI in a large-scale distributed scenario with hundreds of replicas in geographically distant data centers.

The rest of the paper is organized as follows: Section II

978-1-5090-5197-7/16/$31.00 ©2016 IEEE

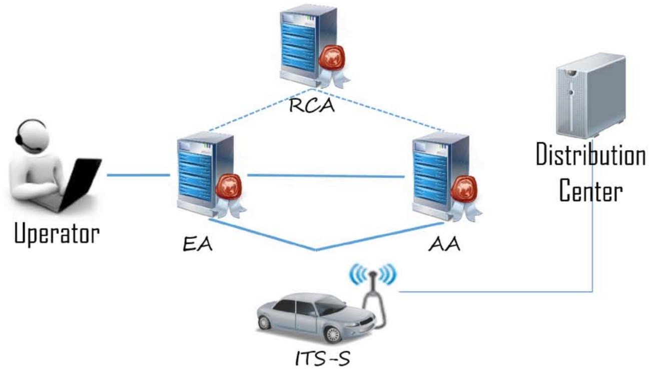
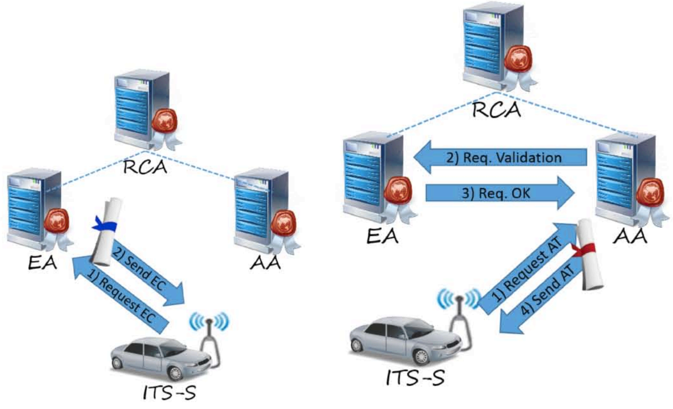


Figure 1: PKI architecture.



(a) The process to get an EC involves two steps: 1) the ITS requests the EC to the EA, 2) the EA sends the certificate (if the ITS has the permissions).

(b) The process to get an AT involves four steps: 1) the ITS requests an AT, 2) the AA asks the EA to check and validate the request, 3) the EA tells the AA if the ITS has the required permissions, 4) the AA creates and sends the certificate.

Figure 2: Authorization Ticket (AT) and Enrollment Certificate (EC) request protocol.

presents the PKI architecture and replication strategy, Section III illustrates the results of the measurement campaign. The experimental evaluation outcomes and future works are discussed in Section IV. Section V presents related works and Section VI concludes the paper.

II. PKI Architecture

A. *ETSI Architecture*

In order to provide pseudonymity to ITS–Ss and make it hard to link pseudonyms, ETSI specifications (see I) define two kinds of certificates: the Enrollment Certificate (EC), a long-term validity certificate that identifies the ITS–S to the certification authority, and the Authorization Ticket (AT), a pseudonym short-term validity certificate dedicated to the communication between ITS–Ss. Fig. 1 shows the entities involved: the Root Certificate Authority (RCA), the Enrollment Authority (EA), the Operator, the Authorization Authority (AA) and the Distribution Center (DC).

The Root Certificate Authority (RCA) is the start point of the certificate trust chain, it signs the certificates of other authorities (Authorization Authority (AA) and Enrollment Authority (EA)) and produces and maintains the Certificate Revocation List (CRL), the list of revoked authorities, and the Trust-service Status List (TSL), the list of trusted authorities with their access points. In an operational context, this entity should be managed by an actor who can guarantee a high and stable confidence level and who is federative enough, like a state or a group of states. The EA is the authority which delivers Enrollment Certificates (ECs) and validates Authorization Tickets (ATs) requests. Each EA manages its own ITS–S fleet, and different EAs can be administrated by different stakeholders (e.g. car manufacturers, road operators, public authorities, etc). The Operator is an interface to manage ITS–S authorizations and is synchronized with the EA. ITS–S authorizations refer to the specific permissions of an ITS–S, for example special permissions for ambulances or police vehicles. The AA is a trusted third-party that provides ATs to ITS–Ss. The AA does not know the ITS–S identity and relies on the EA to check whether the ITS–S is authorized or not to have the AT. The AT request contains the identity of the EA where the ITS–S is registered. Anyone can operate an AA, providing that the pseudonymity and the unlinkability concerns are respected, i.e. it does not collude with the EA. The Distribution Center (DC) is a simple warehouse for products of the RCA, like CRL, the list of revoked authorities, and TSL, the list of trusted authorities with their access points.

This architecture is meant to provide privacy to ITS–Ss and avoid tracking: the EA knows the ITS–S identity but does not know the pseudonym certificates (ATs) it uses, while the AA knows the ITS–S pseudonym certificate but does not know its identity. An ITS–S registers itself to the EA and obtains an Enrollment Certificate. The EC is used to request pseudonym identities (ATs) to the AA: when an ITS–S requests an AT, it sends in the request message its identity encrypted with the EC and the EA identifier. The AA receives the pseudonym request, reads the EA identifier and checks the EA access point in the TSL. It then asks the EA to validate the AT request. The EA checks the ITS–S EC and validates (or not) the requests. If the request is validated, the AA generates and sends the AT to the ITS–S. Fig 2 shows the AT and EC request steps.

B. *Technical ITS-Security (ISE) Architecture*

In the ISE project[1] we developed one of the first ETSI-compliant PKI implementations. Our PKI is actually being tested by car manufacturers and equipment suppliers in the frameworks of ISE and SCOOP@F[2] projects.

The PKI is implemented in Java. For economic and practical reasons, the key store which contains keys to decrypt requests and sign certificates is a file protected by a password (BKS

[1] http://www.irt-systemx.fr/en/project/ise/

[2] http://www.scoop.developpement-durable.gouv.fr/en/

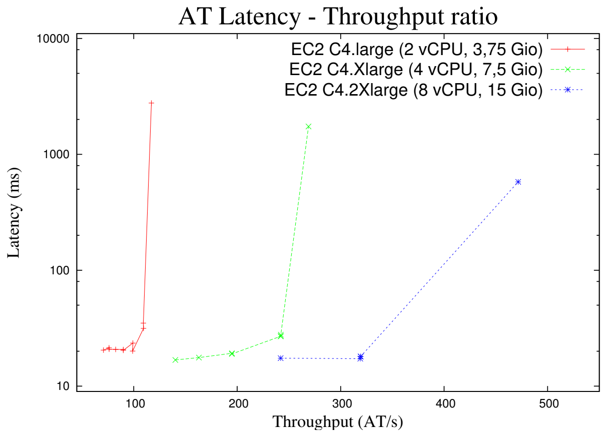


Figure 3: AT delivery performance on centralized settings.

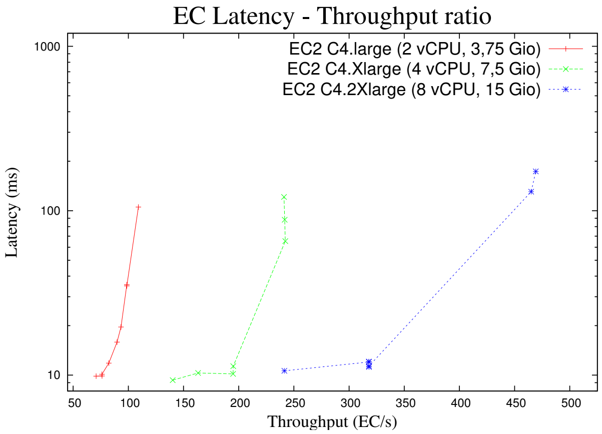


Figure 4: EC delivery performance on centralized settings.

format, provided by BouncyCastle). We do not use a hardware module.

In an operational context, the traffic load for certificate requests can be very high[3]. In order to handle such a traffic load, the system is designed to be scalable and to serve requests in parallel at multiple PKI replicas.

Since we use replicas to achieve scalability, we have to avoid synchronization bottlenecks, in particular for the EAs: if an ITS–S status is updated, or if an ITS–S has requested an EC, every EA and Operator replica must be updated. On the contrary, the AA has no synchronization needs and AAs replicas can be run in parallel without any coordination: every transaction should be logged only locally for audit requirements with no need for propagation to other replicas. We refer to these local, non-replicated, logs as *offline logs*.

To propagate updates to all replicas and keep them consistent, we use the database state machine approach [13]: each replica (EAs and Operators) manages its own copy of data storage in a separated No-SQL database (Redis) and all the replicas receive and apply all the updates in the same order. The replicas are connected in a reliable group communication channel (JGroups [14], [15]) that guarantees the message delivery order and the group membership. All the EAs and Operators are connected in the same group communication channel to synchronize update operations. For example, when an EA receives an EC request, it creates the EC and broadcasts a message containing the EC in the group communication channel. The channel delivers the message to all the members atomically (i.e. to all or none of the members) and in total order (i.e. it delivers all the messages in the same order at all members), we call those two properties of the channel *Atomic Broadcast (ABCAST)*. When a replica receives the message, it applies the update and acknowledges the sender. Once the sender has received all the acknowledgements from all the replicas, it continues the operations and answers the request. We call this *synchronous update propagation*. The combination of ABCAST and synchronous update propagation makes replica consistency *strong*: the storage is not exposed to any replication anomaly [16].

In order to boost EC delivery performance we have to reduce the synchronization time. Reducing synchronization time implies weakening the storage consistency properties. In our implementation we support four consistency levels produced by the combinations of two synchronization strategies: i) *ABCAST* vs *First In First Out (FIFO)* message delivery and ii) *synchronous* vs *asynchronous* update propagation.

The first way to reduce synchronization time is to change the group communication channel properties to use FIFO message order instead of ABCAST. The difference is that ABCAST imposes a total order on the messages, while FIFO imposes a partial order: ABCAST guarantees that all messages are received in the same order at all replicas, while FIFO guarantees that messages sent from one replica are received in the submission order by other replicas, without any guarantee on the interleaving order of messages sent from different replicas. The fact that the *inter-replica* causal order is not maintained can be very confusing for the application: for example, a replica r_1 broadcasts a message m_1, a replica r_2 receives the message m_1 and broadcasts in response a message m_2 somehow related to m_1. Now, a third replica r_3 receives first the reply message m_2, and then the message m_1. This can be confusing for r_3 because it receives the message m_2 *caused* by the message m_1 before receiving m_1. The FIFO message order is much less expensive than the ABCAST order because it does not require global coordination and has the useful property of maintaining the *per-replica* causal order, which is enough for our needs.

The second level is about the update propagation: synchronous or asynchronous. With synchronous update propagation, when a replica broadcasts a message, it waits for the update to be propagated to all other replicas before returning the operation. This guarantees that if an application connects to a replica, makes a change and then connects to another replica,

[3] According to the International Council On Clean Transportation, the number of passenger cars on EU roads in 2015 was 251 million and will reach 258 million by 2030

Instance name	vCPU	Mem (GB)	Processors
c4.large	2	3.75	Intel Xeon E5-2666 v3
c4.xlarge	4	7.5	Intel Xeon E5-2666 v3
c4.2xlarge	8	15	Intel Xeon E5-2666 v3

Table I: Amazon EC2 Instance Types

it sees its own update. With asynchronous update propagation, when a replica broadcasts a message, it does not wait for the update to be propagated to all other replicas before returning the operation. This means that if an application connects to a replica, makes an update and then connects to another replica, it may not see its own update. As for ABCAST or FIFO message order, this can be very confusing if not properly handled at the application level.

III. Measurement Campaign

In the measurement campaign we test our PKI performance and scalability.

In our experiments we deploy AAs, EAs, Operators and clients. The clients are simple ITS–S mock-ups, only able to request ECs and ATs. We call the AA-EA-Operator-client quartet *elements* of a *group*. Within a group the authorities trust each other, the Operator registers the client certificate to the EA (i.e. enrolls the client as a legitimate ITS) and the client makes EC and AT requests to the EA and AA of the same group. The client measures the response latency, the success rate and throughput. In our experiments each element of the group is deployed in its own Elastic Cloud Computing (EC2) virtual machine.

In sections III-A to III-C, we present three sets of experiments with three different configurations: centralized, distributed and geographically distributed.

In the first set of experiments we measure how many EC and AT requests per second can be served by a single group.

In the second set of experiments we test the PKI performance with several groups within a data center. In order to measure the system scalability we replicate the PKI in 10, 20, 30, 40 and 50 groups and we measure how many EC and AT requests per second can be served for each setting. Note that each group contains 4 elements, so in order to deploy 10 to 50 groups we use, in our experiments, 40 to 200 machines.

In the third set of experiments we measure the PKI scalability across data centers. We deploy the groups in geographically distant sites and we analyze the impact on performance.

A. Centralized Deployment

The first set of measurements aims to show our PKI performance and the limits of a centralized deployment. A centralized deployment is the deployment of one machine for each entity of a group (AA, EA, Operator and client). We test the performance in terms of AT and EC throughput, i.e. the number of ATs and ECs delivered per second.

Fig. 3 and Fig. 4 show, respectively, the AT and EC delivery latency, depending on the throughput, on three different commodity hardware (see Table I). Note that the throughput measure masks the number of injected requests: we do not show the number of requests per second *injected* in the system (input rate) but the number of requests per second *served* by the system (throughput). Augmenting the number of requests injected, the machines start to deliver the certificates with a low latency until a saturation point. Behind that point the latency explodes without any or little benefit in terms of throughput. Fig. 3 shows that for a 2 virtual core machine with 3,75 GB of memory the saturation point occurs little after reaching 100 AT requests per second. Doubling the machine virtual cores and memory we can serve about 240 AT requests per second and more than 300 AT requests per second if we double again. Note that the bigger the machine, the less abruptly the saturation occurs.

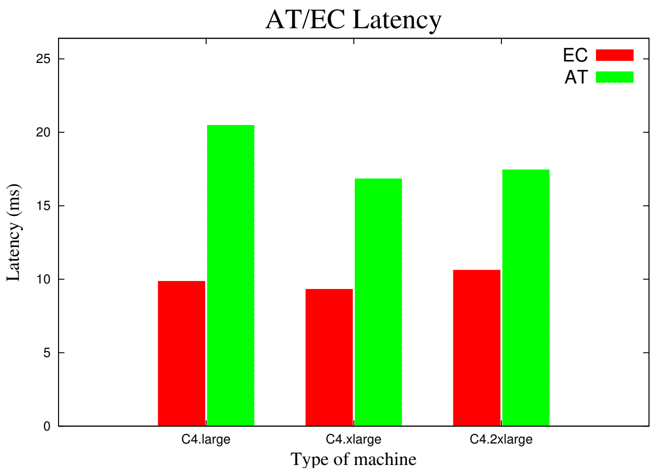


Figure 5: AT and EC latencies before machine saturation.

Fig. 5 shows the latency for AT and EC requests in the three machines listed in Table I. Unsurprisingly all the machines have similar results, intuitively this is because a bigger machine can serve a higher number of requests per second but does not serve requests faster. In average, delivering an AT takes around 20 milliseconds, while serving an EC takes around 10 milliseconds, half of the time. This is because the AA must validate the AT requests with the EA, while the EA does not need to communicate with anyone to deliver an EC (see Section II).

Those results show quite a good performance: respectively 10 and 20 milliseconds to serve an EC and an AT, but also the limits of a centralized deployment: in order to serve a huge load of requests (we can expect millions of ITS–S using billions of temporary certificates) we must replicate the PKI.

B. Distributed Deployment

The vehicular message authentication system reposes on PKI services, making its availability and scalability paramount. The PKI must be able to serve billions of certificates to millions of ITS–Ss. To achieve scalability and fault tolerance we replicate the PKI servers in hundreds of Amazon Web Services (AWS) EC2 instances.

Thanks to replication our system is able to support server failures without stopping or perturbing the system. We change

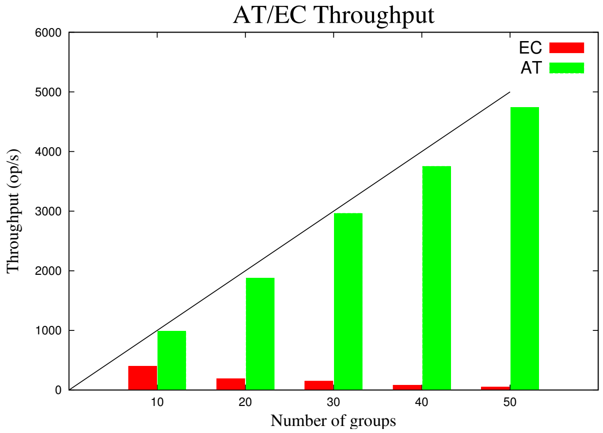


Figure 6: AT and EC throughput augmenting the number of replicas.

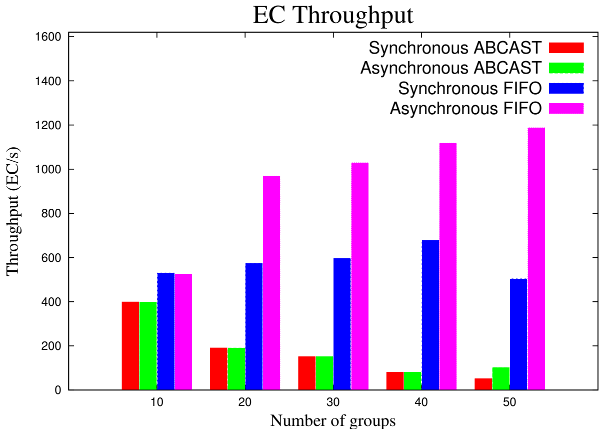


Figure 7: EC throughput with FIFO/ABCAST and synchronous/asynchronous API.

the number of replicas at run-time to exclude failed replicas or integrate new replicas transparently.

When we distribute the PKI we replicate the storage system (see Section II). The replicated storage system allows to boost read operations (it allows to parallelize reads at several replicas) but it makes the writes less efficient (and more complex) because it should propagate the updates to each replica. This has an impact on the AT and EC creation, because when we create an AT we need to read from the distributed storage but we do not need to persist any data in it, while when we generate an EC we need to persist and propagate the generated certificate in the distributed storage. This is why by replicating our EA and AA servers we expect to scale up the throughput of pseudonym identities (AT) creation at the price of lowering long term certificate (EC) throughput.

This intuition is confirmed by the experiments: the AT delivery throughput scales well, while the EC delivery throughput degrades while we augment the number of replicas.

Fig. 6 Shows the AT and EC delivery throughput in distributed settings. The AT delivery rate is very close to the optimal, linear scalability: by augmenting the number of groups from 10 to 50 (and consequently the number of replicas from 40 to 200) we augment the throughput from 1k ATs per second to 5k ATs per second. The optimal scalability corresponds to the segment bisector In Fig. 6.

Fig. 6 shows that, contrary to AT, the more we replicate the worse the EC delivery throughput is. This is due to the replicas synchronization.

From this set of experiments we conclude that we can optimize for the most common case i.e. AT requests maintaining a strong consistency in storage operations. The counterpart is then a performance degradation of the EC requests.

The EC delivery performance degradation can be acceptable if we consider that EC requests are much rarer than AT requests: the Enrollment Certificate is meant to last long time, while the Authorization Tickets are meant to be changed frequently[4]. However, from the point of view of an Operator (e.g. an ITS–Ss manufacturer) that enrolls large amounts of ITS–Ss the EC delivery performance matters. In order to enhance EC delivery performance we must relax the storage consistency properties.

We explore the trade-off between EC delivery performance and storage consistency in next Section.

1) Weakening the consistency: Fig. 7 shows the EC throughput augmenting the number of replicas with ABCAST/FIFO message delivery and synchronous/asynchronous update propagation.

With 10 groups (40 replicas), the system delivers around 400 EC per second with ABCAST and around 500 EC per second with FIFO. Interestingly, at this stage the synchronous or asynchronous update propagation strategy does not really matter. Doubling the number of replicas we observe that experiences with ABCAST message propagation order performs worse regardless the update propagation schema. The throughput improves when we relax the message delivery order from ABCAST to FIFO. With FIFO message propagation order and synchronous update propagation, the performance does not decrease augmenting the number of replicas and remains substantially stable. We observe a real gain when we use FIFO order in conjunction with asynchronous update propagation. In this configuration, passing from 10 to 20 groups (40 and 80 replicas respectively) causes the performance to double passing from around 500 EC per second to around 1000 EC per second.

Globally, we observe that the ABCAST message propagation order does not scale at all, and the more replicas we use, the worse the throughput is. Relaxing the channel delivery order property we obtain better performance and we can observe the effects of the synchronous and asynchronous update propagation strategy: with synchronous update propagation and ABCAST message propagation order we have

[4] As an example we can consider a validity period of years for ECs and hours or even minutes for ATs.

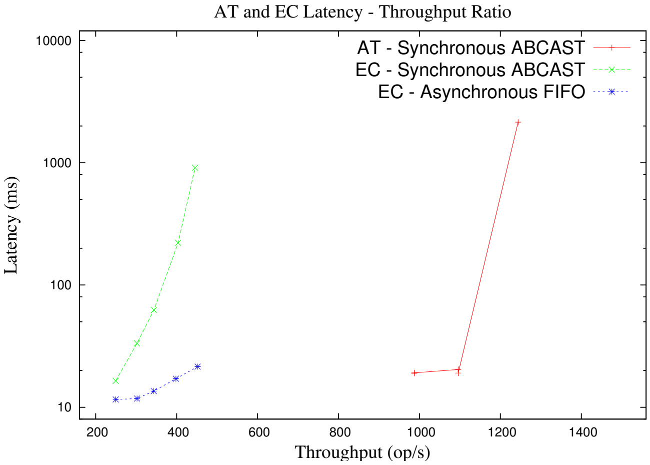


Figure 8: AT and EC delivery performances with the authorities deployed in the same data center.

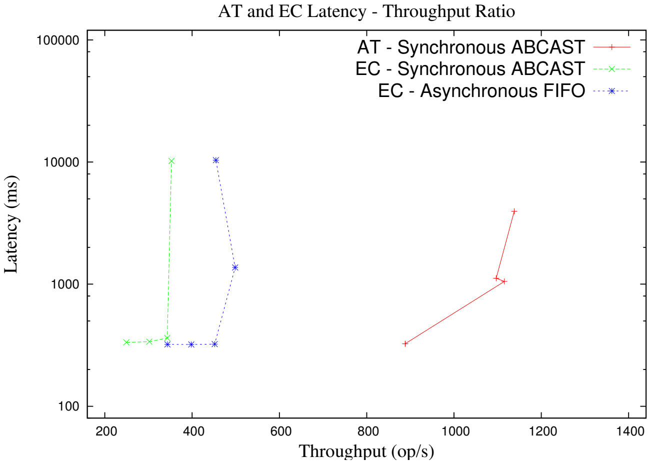


Figure 9: AT and EC delivery performances with the authorities deployed in distant data centers. The EA is deployed in Europe (Ireland) and the AA in the US (west coast)

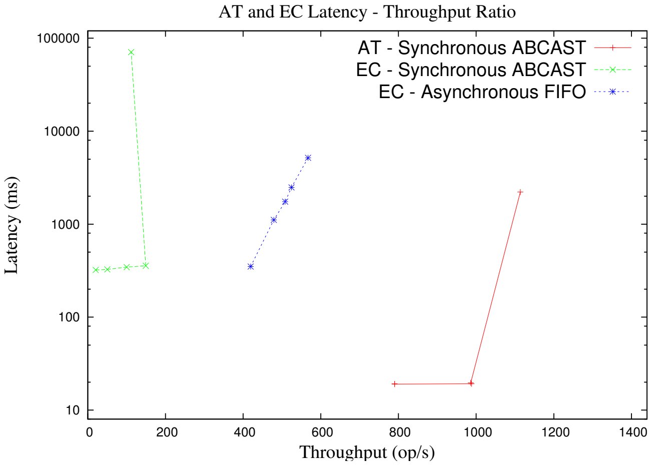


Figure 10: AT and EC delivery performances with the authorities replicated in both Europe and United States data centers.

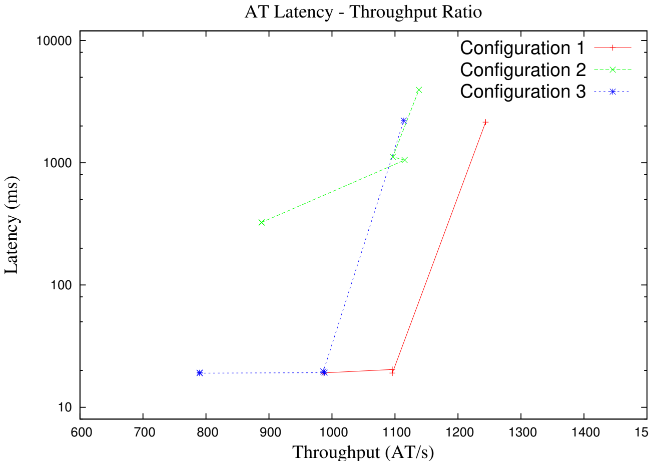


Figure 11: AT delivery performances in the three configurations.

good performance compared to FIFO message order delivery but poor scalability. Relaxing both the message order delivery properties and the update propagation schema, we obtain much better performance (close to 1200 EC per second) and the system scales. Of course the scalability is not comparable to the near-linear AT scalability because even using a weak consistency model the synchronization effort for write operations increases with the number of replicas.

C. Geographic Deployment

In this section we present a set of experiments that shows the system scalability and performance when its elements are geographically distant. The interest of those experiments is to investigate the impact of ITS–Ss mobility when they are far away from the certification authorities.

We compare three configurations: in the first configuration we deploy 10 groups with all the elements in Europe, in the second configuration we deploy 10 groups with the clients and AAs in the US west coast and the EAs and Operators in Europe (Ireland), in the third configuration we deploy 10 groups in the US west coast with the EAs and Operators replicated in Europe.

Fig. 8 shows the base case: the latency/throughput ratio with 10 groups deployed in a single data center in Ireland. When the system is not saturated ITS–S obtain EC and AT in few milliseconds. As for previous experiments, ATs latency is higher than ECs latency but they scale better. We can also observe the differences between synchronous ABCAST and asynchronous FIFO for EC.

Fig 9 shows the latency/throughput ratio when the AAs are deployed in a data center in the west coast of the United States, and the EAs in a data center in north Europe. In terms of throughput we observe a similar behaviour to the one presented in Fig. 8, but at a much higher latency due to the latency of cross-Atlantic messages sent between AAs and EAs and between ITS–S and EAs. In the mono-data center configuration of Fig. 8 the EC and AT delivery latency is between 10 and 20 milliseconds, while in the cross-data center configuration of Fig 9 the EC and AT delivery latency is one order of magnitude higher (around 300 milliseconds).

The high latency of EC and AT delivery is due to the high latency of cross-Atlantic communications between entities deployed in data centers in Europe and United States. We can avoid those high communication latencies by replicating all the members of a group (EA, AA, Operator and client) in the two data center. In this configuration clients in a geographical region contact the authority in that region avoiding high communications latencies. This strategy works well for ATs because the clients request the AT to a close AA, which validates the request communicating with an EA in the same region. So the AT request does not incur in the high latency of contacting a geographically distant replica. However, replicating the group in distant data center does not improve EC requests performance because the storage system is replicated in distant data centers and the updates are propagated to geographically distant sites.

Fig. 10 shows the trade-off between AT and EC performances when the PKI is replicated in geographically distant sites: AT latency is order of magnitude better, as low as in the first configuration (around 10 ms), while EC performance does not improve from the previous configuration. AT requests are much more relevant than EC requests because they are much more frequent (see Section II and Section III-B) and represents much of the load. Moreover, from the ITS–S perspective, AT requests are more safety-critical than EC requests because AT are needed to sign safety messages.

The PKI geographic replication makes possible for the PKI operator (e.g. a car manufacturer) to replicate the system in distant sites without penalty on the AT delivery latency. Fig 11 compares the AT delivery performances in the three configurations and shows clearly that our architecture can be efficiently replicated in distant data centers for a dramatic decrease for the AT delivery latency.

IV. Discussion and Future Works

In this paper we present benefits of a weak consistency model for VPKI data replication. Weakening the consistency makes more scalable the system but exposes applications (e.g. the Operator) to replication anomalies (for concurrency anomalies in distributed databases see [17]–[19]). Those anomalies are due to the fact that the storage replicas are not in the same state at all times and do not have a common update history: different replicas can receive and apply updates in different order.

Replication anomalies can be very confusing for the application and should be properly handled. There are several ways to mask storage inconsistencies at application level. For example when an Operator connects to a replica and make an update, we can direct all successive connections to the same replica until the update is propagated to all the replicas. In this way we assure that an Operator will always observe its own modifications (read after write semantic).

In our measurements campaign we consider a fixed amount of replicas, in future works we plan to handle load changes by dynamically adjusting the number of replicas serving requests.

Another promising research axe is the misbehaviour detection. We plan to investigate the possibility to identify misbehaving vehicles taking as input the internal Security Credential Management System (SCMS) data.

V. Related Work

Vehicular Security and Privacy-preserving Architecture (VeSPA) [20] presents a VPKI that implements a ticket-based authentication mechanisms for pseudonyms requests similar to Kerberos [21]. The work in VeSPA is not comparable to ours because they use only one ticket to request a bulk of pseudonyms in one request, while we have one request for each pseudonym. Also the focus of their work is not the scalability of the VPKI, which is the focus of our work.

[22] improves VeSPA VPKI to a multi-service architecture decoupling the Long Term Certificate Authority (LTCA) (EA in our terminology) and the Pseudonym Certificate Authority (PCA) (AA in our terminology). The proposed architecture supports services across multiple domains (each domain can represent different country and has its set of rules). They show that the average time for a vehicle to obtain one ticket containing a single service identifier from the LTCA is 100.95 ms and acquisition of 1000 pseudonyms takes 16.46 sec including entire communication, verification and storage at the vehicle. They measure the multi-domain protocol latency at 363 ms (285.4 ms for the distant native domain and 104 ms for the foreign domain). As they state, the communication with the distant LTCA has the dominant latency. In our work we show how LTCA (EA following ETSI terminology) geo-replication reduces significantly the communication latency and the pseudonym acquisition time (see III-C).

SERvice Oriented Security Architecture for Vehicular Communications (SEROSA) [23] presents a service oriented security and privacy preserving architecture for Vehicular Communication (VC). Their architecture synthesis the current VC and Internet based standards by having the LTCA as Identity Provider (IdP) and the PCA as Service Provider (SP). In [12], Khodaei et al. enhance VeSPA protocol to improve performance and security. Differently than us, [23] and [12] take measurements from the vehicle point of view (i.e. how long it will take for a vehicle to obtain pseudonyms) while we take measurements from the infrastructure point of view (i.e. how many pseudonym the infrastructure can serve per second). In their work, they investigate the delay to obtain pseudonyms when the number of pseudonyms in a request increases. They only replicate the PCA on two servers behind a proxy, while in our work we focus much more on scalability and replicate both the AA and the EA in dozen of servers. Moreover, we show the benefits of varying the consistency level of the PKI storage system, an aspect that has not been studied in any previous work.

In [24] they show a comparison of the latency for issuing 100 pseudonyms for different VPKI: VeSPA [20] 817 ms, SEROSA [23] 650 ms, PUCA [25] 1000 ms, SR-VPKI [12] 260 ms. In our VPKI each request is for one pseudonym and we serve requests in parallel which leads to a latency of around 20 ms for each pseudonym and a throughput of more than 200 requests per second with a single machine per authority (see III-A). Moreover our VPKI is easily scalable and can serve thousands of pseudonym requests using PCA (AA in our terminology) replication while maintaining the 20 millisecond latency.

VI. Conclusion

We have implemented and tested a fully functional prototype of a vehicular credential management system compliant with ETSI specifications [1], [10]. In this work, we investigate the scalability-consistency trade-offs of a replicated PKI. At the best of our knowledge, this is the first large scale experimentation of the credential management system being standardized at the ETSI. From our measurements campaign, we conclude that our architectural design makes easily scalable the AT delivery operations. Regarding the EC delivery operations, they are more complex to scale because they involve the propagation of updates in a distributed storage system. We show how we can make scalable those operations by relaxing the storage consistency and we discuss how we can handle anomalies introduced by the weak consistency replication model.

Acknowledgement

This research work has been carried out in the framework of the Technological Research Institute SystemX with the support of the AWS in Education Grant award.

References

[1] "ETSI TS 102 940 v1.1.1: Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management," *ETSI WG5 Technical Specification*, pp. 1–29, June 2012. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102900_102999/102940/01.01.01_60/ts_102940v010101p.pdf

[2] M. S. Al-kahtani, "Survey on security attacks in vehicular ad hoc networks (vanets)," in *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*, Dec 2012, pp. 1–9.

[3] A. Boudguiga, A. Kaiser, and P. Cincilla, "Cooperative-its architecture and security challenges: a survey," in *22nd ITS World Congress*, Bordeaux, France, Oct. 2015.

[4] B. Lonc and P. Cincilla, "Cooperative its security framework: Standards and implementations progress in europe," in *In Proceedings of Workshop of Smart Vehicles at IEEE WoWMoM*, July 2016.

[5] G. Guette and O. Heen, "A tpm-based architecture for improved security and anonymity in vehicular ad hoc networks," in *2009 IEEE Vehicular Networking Conference (VNC)*, Oct 2009, pp. 1–7.

[6] A. Wasef, R. Lu, X. Lin, and X. Shen, "Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]," *IEEE Wireless Communications*, vol. 17, no. 5, pp. 22–28, October 2010.

[7] M. S. Al-kahtani, "Survey on security attacks in vehicular ad hoc networks (vanets)," in *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*, Dec 2012, pp. 1–9.

[8] A. Boudguiga, A. Boulanger, P. Chiron, W. Klaudel, H. Labiod, and J. C. Seguy, "Race: Risk analysis for cooperative engines," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, July 2015, pp. 1–5.

[9] "ETSI TS 103 097 V1.1.1: Intelligent Transport Systems (ITS); Security; Security header and certificate formats ," *ETSI WG5 Technical Specification*, April 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/103000_103099/103097/01.01.01_60/ts_103097v010101p.pdf

[10] "ETSI TS 102 941 v1.1.1: Intelligent Transport Systems (ITS); Security; Trust and Privacy Management," *ETSI WG5 Technical Specification*, June 2012. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102900_102999/102941/01.01.01_60/ts_102941v010101p.pdf

[11] N. Bißmeyer, H. Stübing, E. Schoch, S. Götz, J. P. Stotz, and B. Lonc, "A generic public key infrastructure for securing Car-to-X communication," in *World Congress on Intelligent Transport Systems (ITS)*. Fraunhofer SIT, Oct. 2012, pp. 12–24.

[12] M. Khodaei, H. Jin, and P. Papadimitratos, "Towards deploying a scalable & robust vehicular identity and credential management infrastructure," in *2014 IEEE Vehicular Networking Conference (VNC)*, Dec 2014, pp. 33–40.

[13] F. Pedone, R. Guerraoui, and A. Schiper, "The database state machine approach," *J. of Dist. and Parallel Databases and Technology*, vol. 14, no. 1, pp. 71–98, 2003.

[14] A. Montresor, *The Jgroup Distributed Object Model*. Boston, MA: Springer US, 1999, pp. 389–402. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-35565-8_31

[15] A. Montresor, R. Davoli, and O. Babaoğlu, "Middleware for dependable network services in partitionable distributed systems," *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 1, pp. 73–96, Jan. 2001. [Online]. Available: http://doi.acm.org/10.1145/371455.371463

[16] H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O'Neil, and P. O'Neil, "A critique of ansi sql isolation levels," in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '95. New York, NY, USA: ACM, 1995, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/223784.223785

[17] J. Gray, P. Helland, P. O'Neil, and D. Shasha, "The dangers of replication and a solution," ACM SIGMOD. Montréal, Canada: ACM Press, Jun. 1996, pp. 173–182.

[18] D. J. Abadi, "Consistency tradeoffs in modern distributed database system design," vol. 45, no. 2, pp. 37–42, Feb. 2012.

[19] M. T. Özsu and P. Valduriez, *Principles of distributed database systems*. Springer Science & Business Media, 2011.

[20] N. Alexiou, M. Laganà, S. Gisdakis, M. Khodaei, and P. Papadimitratos, "Vespa: Vehicular security and privacy-preserving architecture," in *Proceedings of the 2Nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy*, ser. HotWiSec '13. New York, NY, USA: ACM, 2013, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/2463183.2463189

[21] B. C. Neuman and T. Ts'o, "Kerberos: an authentication service for computer networks," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33–38, Sept 1994.

[22] N. Alexiou, S. Gisdakis, M. Laganà, and P. Papadimitratos, "Towards a secure and privacy-preserving multi-service vehicular architecture," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, June 2013, pp. 1–6.

[23] S. Gisdakis, M. Laganà, T. Giannetsos, and P. Papadimitratos, "Serosa: Service oriented security architecture for vehicular communications," in *2013 IEEE Vehicular Networking Conference*, Dec 2013, pp. 111–118.

[24] M. Khodaei and P. Papadimitratos, "The key to intelligent transportation: Identity and credential management in vehicular communication systems," *IEEE Vehicular Technology Magazine*, vol. 10, no. 4, pp. 63–69, Dec 2015.

[25] D. Frster, F. Kargl, and H. Lhr, "Puca: A pseudonym scheme with user-controlled anonymity for vehicular ad-hoc networks (vanet)," in *2014 IEEE Vehicular Networking Conference (VNC)*, Dec 2014, pp. 25–32.