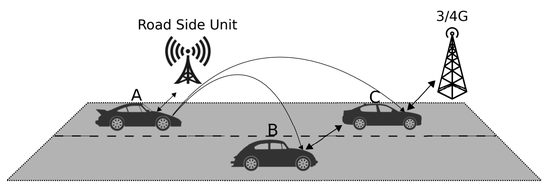
A Highly Available and Dynamically Scalable Vehicular Public-Key Infrastructure (VPKI): VPKI as a Service (VPKIaaS)

Hamid Noroozi, Mohammad Khodaei and Panos Papadimitratos
Networked Systems Security Group
KTH Royal Institute of Technology
www.ee.kth.se/nss

Background



Vehicle A:
1. Generate and sign message
2. Encapsulate message
3. Broadcast $\{Msg\}_{(P^i_v)}, \{P^i_v\}_{(PCA)}$

Vehicles B & C:
1. Validate the pseudonym, $\{P^i_v\}_{(PCA)}$
2. Verify the signature
3. Validate message content
4. Accept/reject the message
5. Re-broadcast

Figure 1: Secure and privacy-protecting V2V and/or V2I (V2X) communication [3]

VPKI Overview

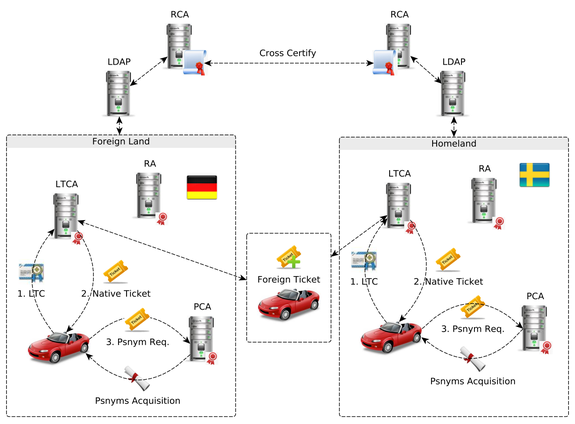


Figure 2: A Vehicular Public-Key Infrastructure (VPKI) Architecture [4]

- Vehicle registration with its home Long Term Certification Authority (LTCA), obtaining an X.509 certificate
- Anonymous ticket acquisition from the LTCA
- Anonymous certificate(s)/pseudonym(s) acquisition from any Pseudonym Certification Authority (PCA)
- Resolution process initiation by Resolution Authority (RA) (conditional anonymity)

Challenges and Objectives

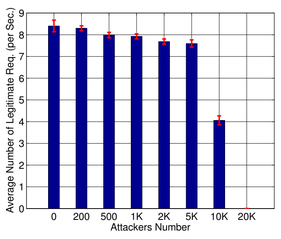


Figure 3: LTCA servers under a DDoS attack [4]

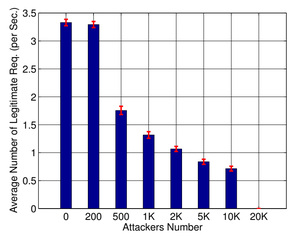


Figure 4: PCA servers under a DDoS attack [4]

- High Availability
  - Self-healing
  - SLA improvement
- Dynamic Scalability
  - Consistent performance on higher load
  - Partial resilience against DDoS
  - Resource efficiency on dynamic load

Cloud-Native Approach

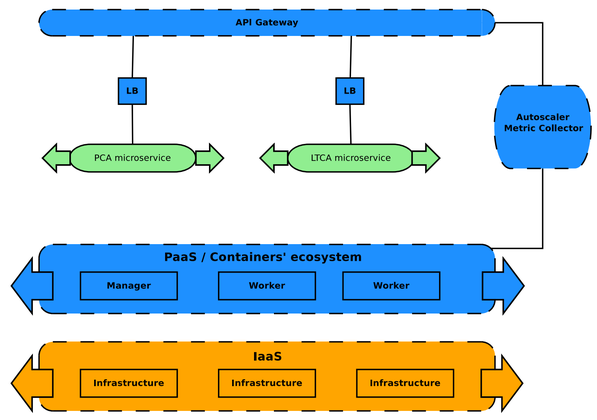


Figure 5: Cloud Native Approach

- Architect VPKI in Microservices [5]
- Plan to scale in/out services
  - Handle race and deadlock conditions
- Automate scaling in/out
  - Define load and health in metrics
  - Publish metrics

Containers Orchestration

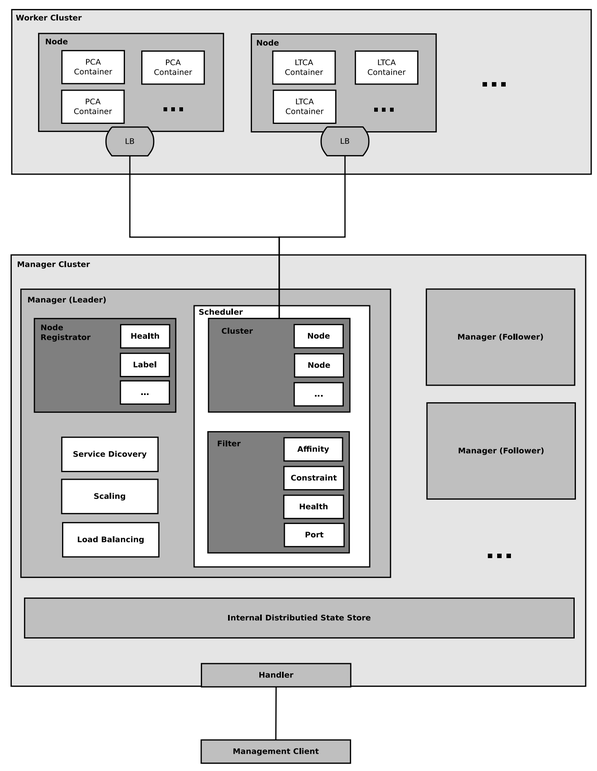


Figure 6: Containers orchestration scheme

- Platform independent (On premises or Public/Private Cloud)
- Deliverable trusted images
- TLS communication
- Quorum management
- Raft Consensus algorithm [6]
- Auto-scaler

Containerization steps

- System definition in **Topology and Orchestration Specification for Cloud Applications (TOSCA)** [7]
- Service orchestration
  - Service registry
  - Load balancing
- State Sharing (Using Raft [6])
- Publishing **Key Performance Indicator (KPI)**

Performance evaluation

- Load test
  - Increase load steadily
- Stress test
  - Intense load try to break
  - Chaos monkey test [8]
  - Negative test
- Benchmark test
  - Resource planning
  - Large-scale Vehicular Communication (VC) deployment

Future work

- Disaster Recovery as as Service
  - Geo-Replication
  - Recover after failure with data loss
- Compromised/Malicious internal VPKI entities

References

[1] M. Khodaei, H. Jin, and P. Papadimitratos, **"SECMACE: Scalable and Robust Identity and Credential Management Infrastructure in Vehicular Communication Systems,"** IEEE Transactions on Intelligent Transportation Systems, in revision (submitted 2016).

[2] M. Khodaei and P. Papadimitratos, **"Evaluating On-demand Pseudonym Acquisition Policies in Vehicular Communication Systems,"** in Proceedings of the First International Workshop on Internet of Vehicles and Vehicles of Internet, Paderborn, Germany, pp. 7–12, July 2016.

[3] M. Khodaei and P. Papadimitratos, **"The Key to Intelligent Transportation: Identity and Credential Management in Vehicular Communication Systems,"** IEEE VT Magazine, vol. 10, no. 4, pp. 63–69, Dec. 2015.

[4] M. Khodaei, H. Jin, and P. Papadimitratos, **"Towards Deploying a Scalable & Robust Vehicular Identity and Credential Management Infrastructure,"** IEEE VNC, Paderborn, Germany, Dec. 2014.

[5] M. Fowler, J.Lweis, **"Microservices, a definition of this new architectural term,"**, `https://martinfowler.com/articles/microservices.html`, March, 2014.

[6] D. Ongaro, J. K. Ousterhout, **"In Search of an Understandable Consensus Algorithm,"** USENIX Annual Technical Conference, pp. 305–319, 2014.

[7] Advancing Open Standards for the Information Society (OASIS), **"TOSCA"**, `https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca`, 2017.

[8] Netflix, **Chaos Monkey resiliency tool**, `https://github.com/Netflix/chaosmonkey`, 2017.