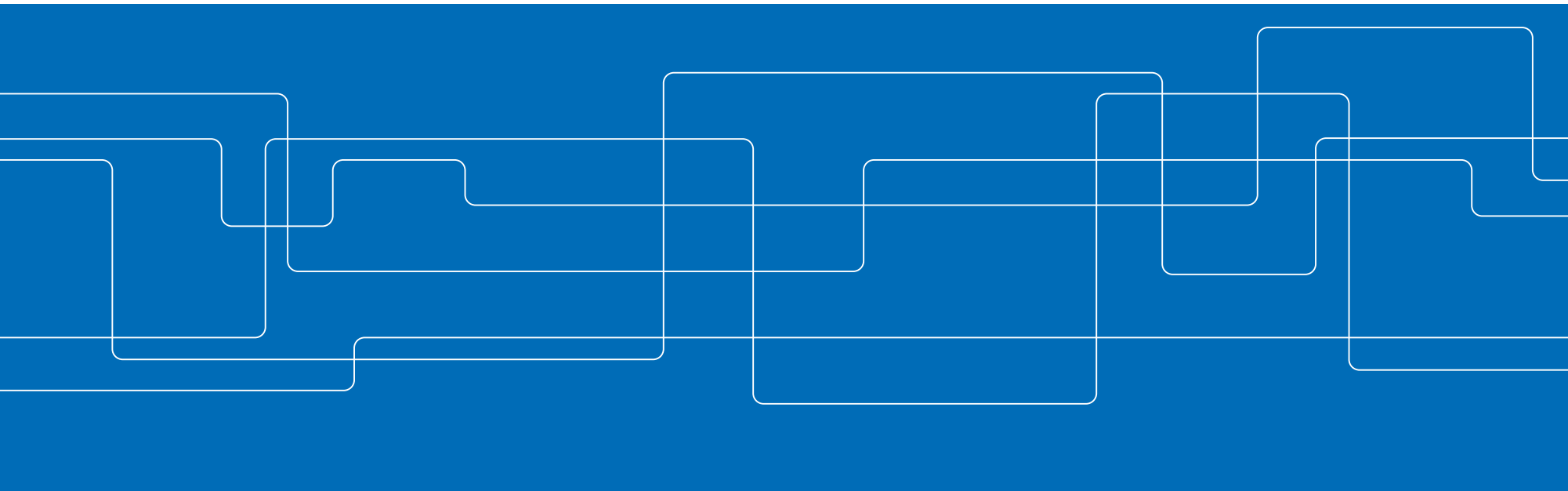
KTH ROYAL INSTITUTE
OF TECHNOLOGY

SECMACE+: Upscaling Pseudonymous Authentication for Large Mobile Systems

M. Khodaei, H. Noroozi, and P. Papadimitratos
Networked Systems Security Group

March, 2023



Vehicular Communication (VC) Systems

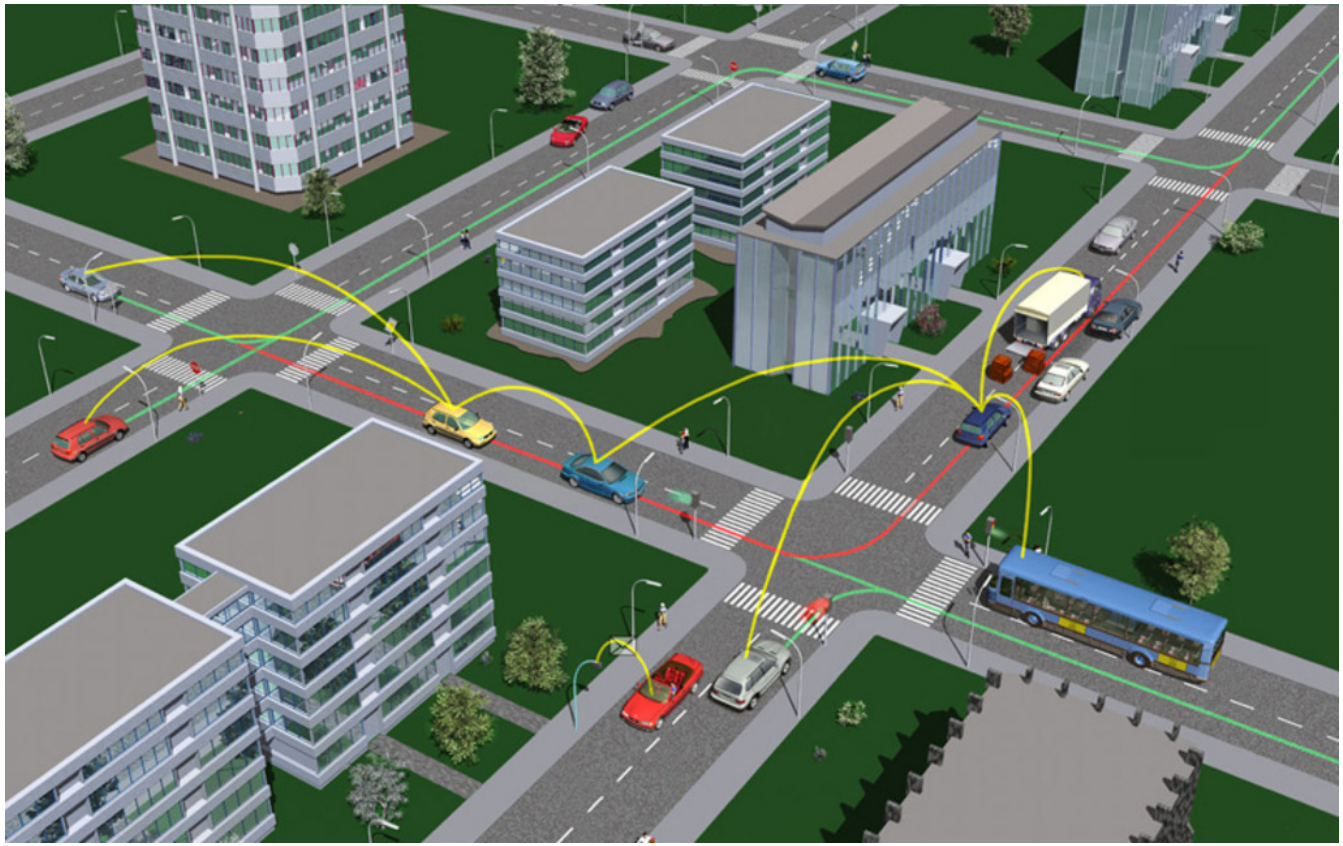
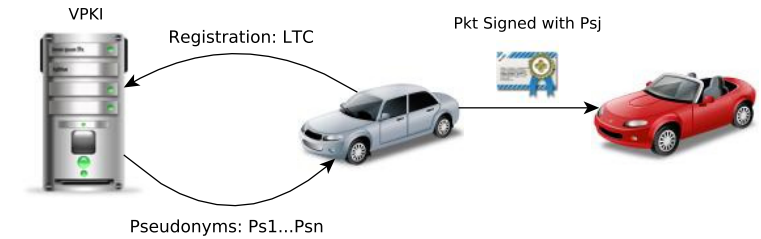


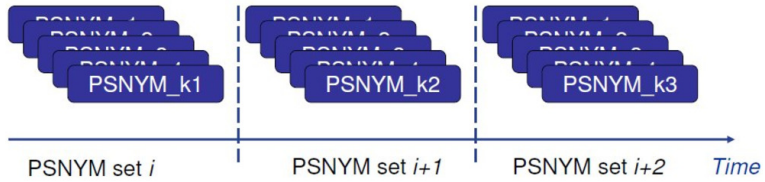
Figure: Photo Courtesy of the Car2Car Communication Consortium (C2C-CC)

Security and Privacy for VC Systems

Basic Requirements

- Message authentication & integrity
- Message non-repudiation
- Authorization & access control
- Entity authentication
- Accountability
- Anonymity (conditional)
- Unlinkability (long-term)

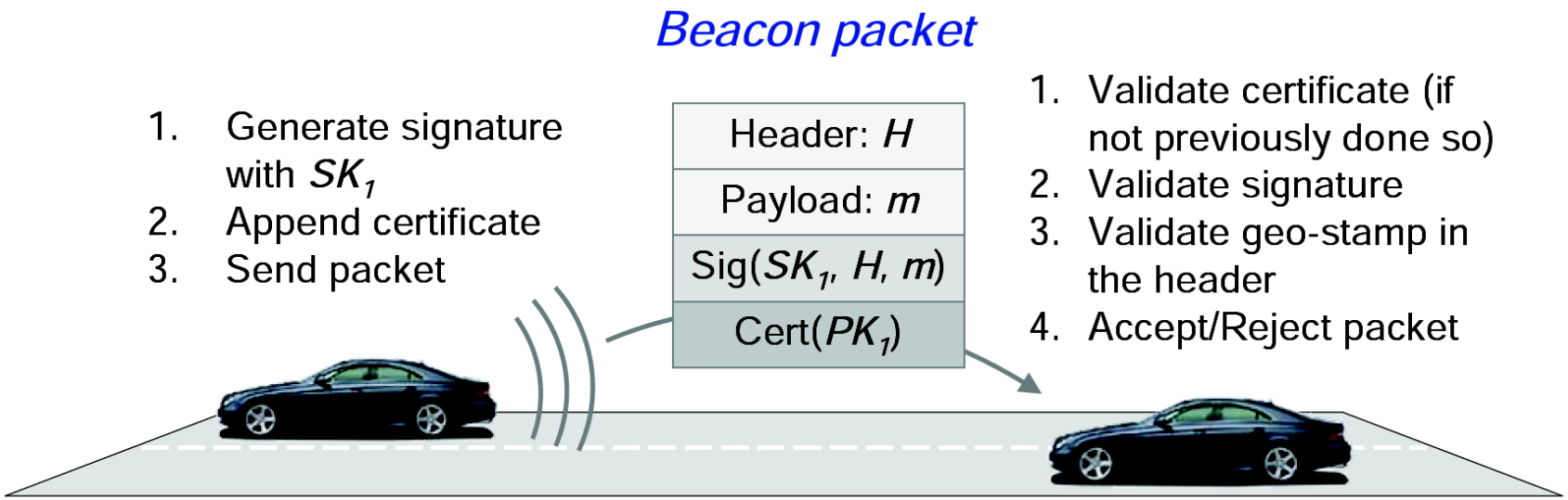




Vehicular Public-Key Infrastructure (VPKI)

- Pseudonymous authentication
- Trusted Third Party (TTP):
  - Certification Authority (CA)
  - Issues credentials & binds users to their pseudonyms

Security and Privacy for VC Systems (cont'd)



- Sign packets with the private key, corresponding to the current valid pseudonym
- Verify packets with the valid pseudonym
- Cryptographic operations in a Hardware Security Module (HSM)

Challenges and Motivation
Traditional PKI vs. Vehicular PKI

- Dimensions (5 orders of magnitude more credentials)
- Complexity and constraints
  - Balancing act: security, privacy, and efficiency
    - *Honest-but-curious* VPKI entities
    - Performance constraints: safety- and time-critical operations (rates of 10 safety beacons per second)
  - Multiple and diverse entities, global deployment, long-lived entities
  - Cost-driven platform resource constraints
- Mechanics of revocation
  - Highly dynamic environment
  - Short-lived pseudonyms, multiple per entity
  - Need for efficient and timely distribution of Certificate Revocation Lists (CRLs)

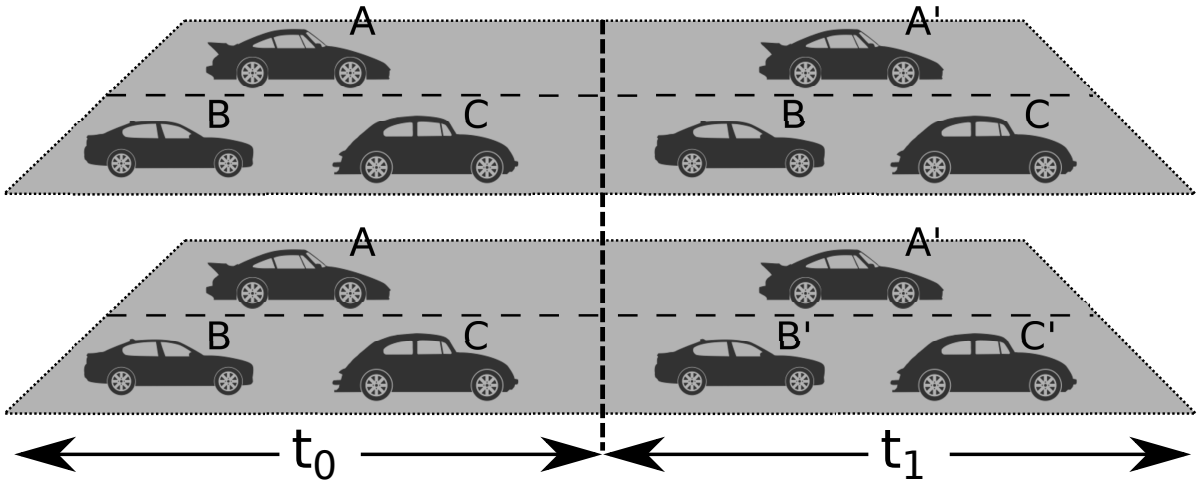
Challenges and Motivation (cont'd)

- Efficient and timely distribution of CRLs to every legitimate vehicle in the system
- Strong privacy for vehicles prior to revocation events to every vehicle
- Computation and communication constraints of On-Board Units (OBUs) with intermittent connectivity to the infrastructure
- Peer-to-peer distribution is a double-edged sword: abusive peers could "pollute" the process, thus degrading the timely CRL distribution

Challenges and Motivation (cont'd)

Attacks on location privacy (traceability): Openness of wireless communication and dissemination of basic safety messages in plaintext

- *Syntactic linking:* "joining the dots" between two Cooperative Awareness Messages (CAMs) by looking at the pseudo-identifier attributes, i.e., time of changing pseudonyms.
- *Semantic linking:* constructing a trajectory through a consistent series of (position, velocity, etc.) pairs.



Secure VC System

- Root Certification Authority (RCA)
- Long Term CA (LTCA)
- Pseudonym CA (PCA)
- Resolution Authority (RA)
- Lightweight Directory Access Protocol (LDAP)
- Roadside Unit (RSU)
- Trust established with RCA, or through cross certification

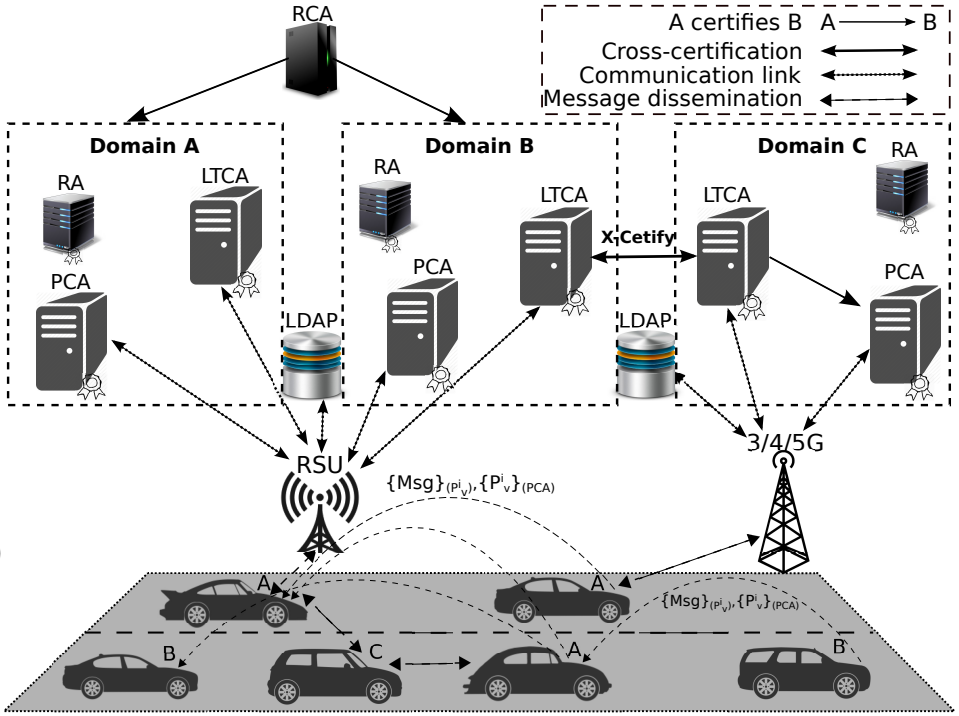


Figure: VPKI Overview

Adversarial Model

- *Honest-but-curious* service providers, i.e., they can attempt to gain advantages towards its goal, e.g., profiling users
- In addition, malicious PCAs could try to:
  - issue multiple sets of (simultaneously valid) pseudonyms for a legitimate vehicle
  - issue a set of pseudonyms for a non-existing vehicle
  - fraudulently accuse different vehicles (users) during a pseudonym resolution process
- A deviant LTCA could attempt to:
  - map a different Long Term Certificate (LTC) during the resolution process
  - issue fake authorization tickets, to be used during pseudonym acquisition process

Adversarial Model (cont'd)

- Malicious (compromised) entities:
  - Internal adversaries, i.e., OBUs, could try to:
    - repeatedly request multiple simultaneously valid pseudonyms, thus misbehaving each as multiple registered legitimate-looking vehicles
    - degrade the operations of the system by mounting a clogging Denial of Service (DoS) attack against the VPKI servers
  - External adversaries, i.e., unauthorized entities, could try to:
    - harm the system operations by launching a DoS attack, thus degrading the availability of the system

Objectives

- Design, analyze, implement and evaluate the VPKI
  - Management of credentials: provisioning, revocation, resolution
  - Standard-compliant implementation
- Resilience to *honest-but-curious* and *malicious* VPKI entities
- Eradication of Sybil-based misbehavior (without degrading performance)
- Handling unexpected demanding loads while being cost-effective
- Scalability
- Efficient revocation and resolution

System Model

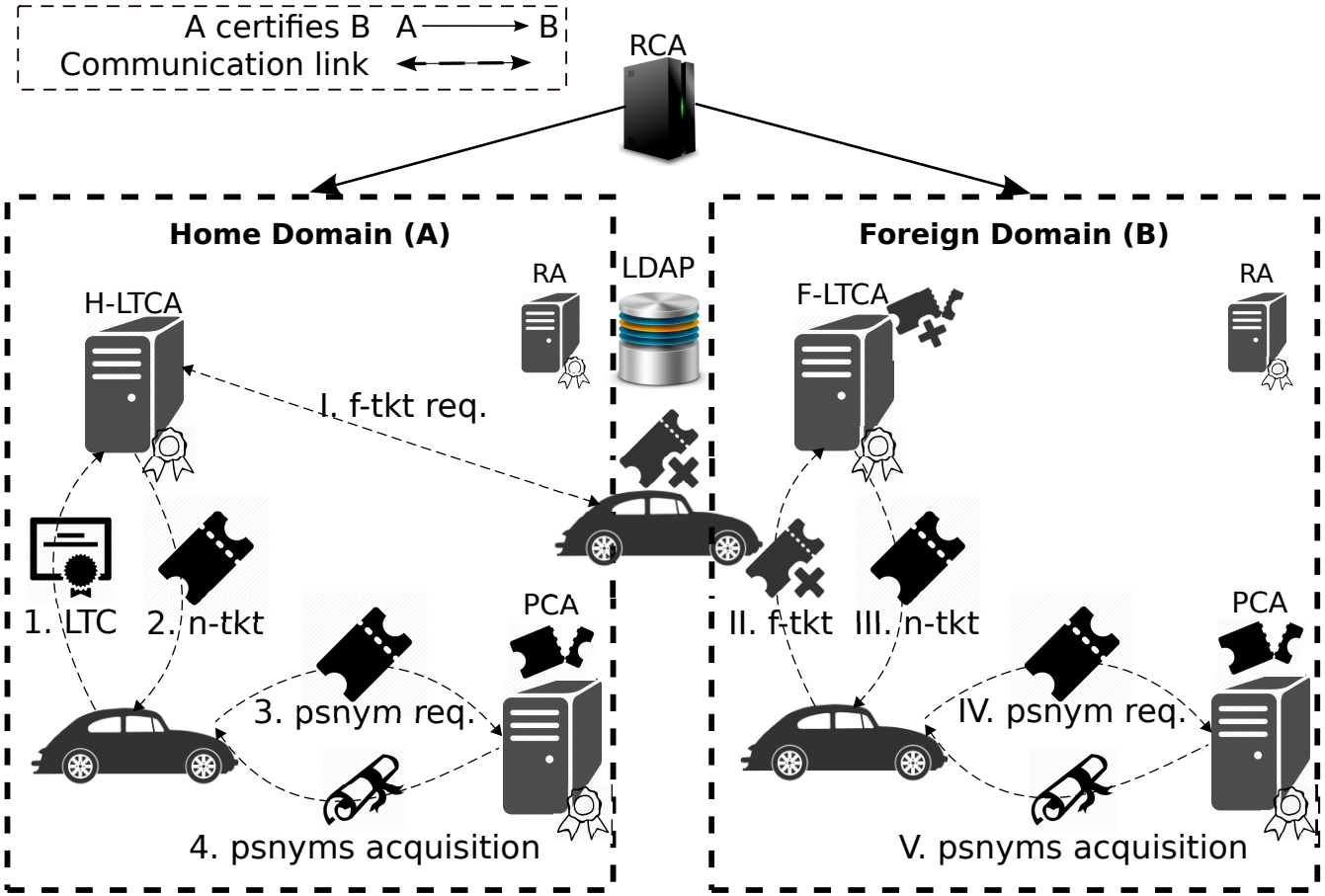
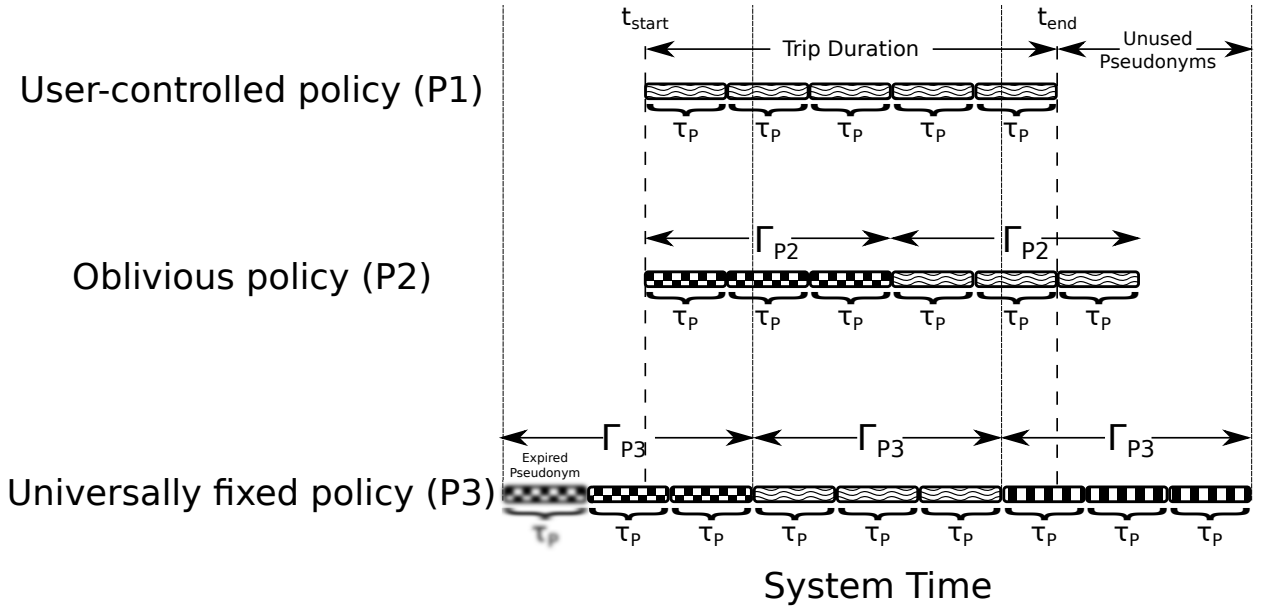


Figure: VPKI Architecture

Pseudonym Acquisition Policies



- P1 & P2: Requests could act as user *"fingerprints"*; the exact time of requests and all subsequent requests until the end of trip could be unique, or one of few
- P3: Requesting intervals fall within *"universally"* fixed interval Γ_{P3}, and pseudonym lifetimes are aligned with PCA clock

VPKI as a Service (VPKIaaS)

- Refactoring a state-of-the-art VPKI source code
- Fully automated all procedures of deployment
- Migrating VPKI to the cloud, e.g., Google Cloud Platform (GCP), Amazon Web Service (AWS), Microsoft Azure
- Enhancing its functionalities towards a highly-available, dynamically-scalable, and fault-tolerant design
- Providing health and load metric publishing feature to be used by an orchestration service to scale in/out accordingly
- Eradicating Sybil-based misbehavior when deploying such a system on the cloud with multiple replicas of a microservice without diminishing the efficiency of the pseudonym acquisition process

VPKI as a Service (VPKIaaS) Architecture

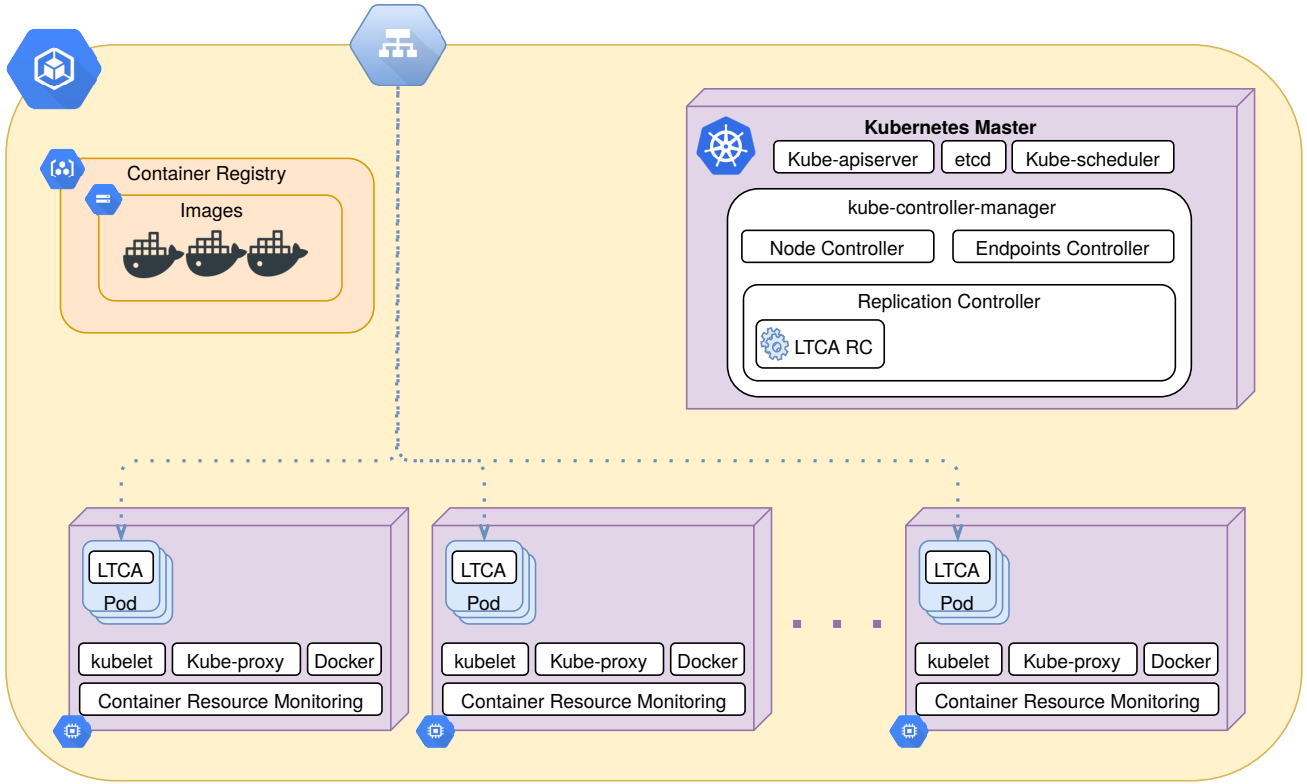


Figure: A High-level Overview of VPKIaaS Architecture on the Cloud

VPKI as a Service (VPKIaaS) Architecture

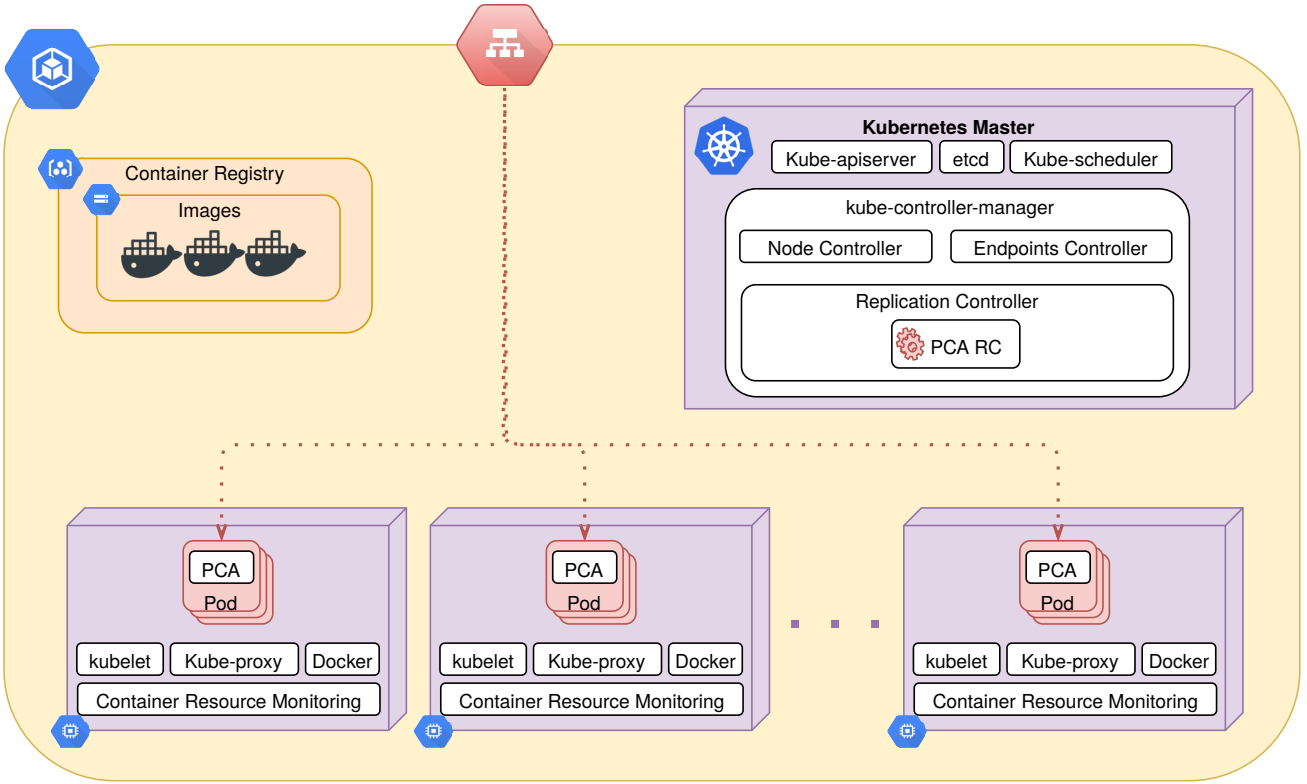


Figure: A High-level Overview of VPKIaaS Architecture on the Cloud

VPKI as a Service (VPKIaaS) Architecture

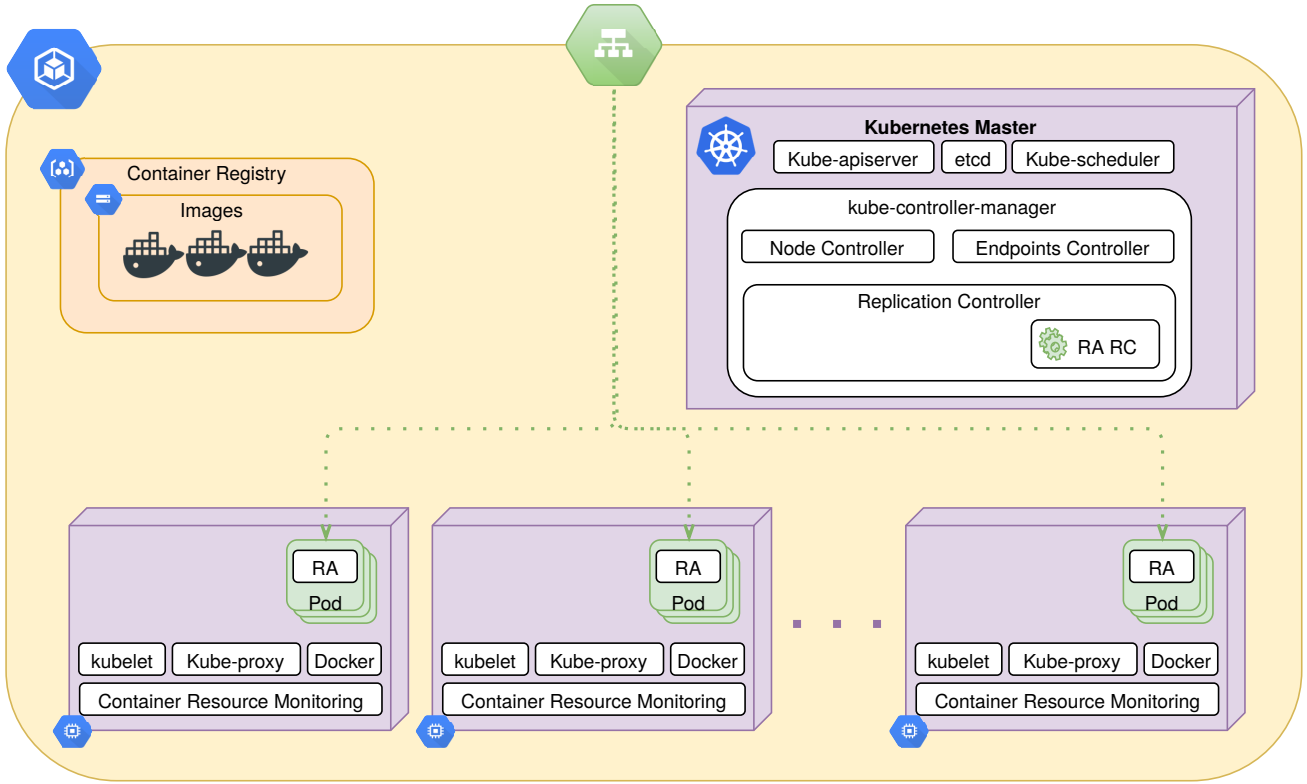
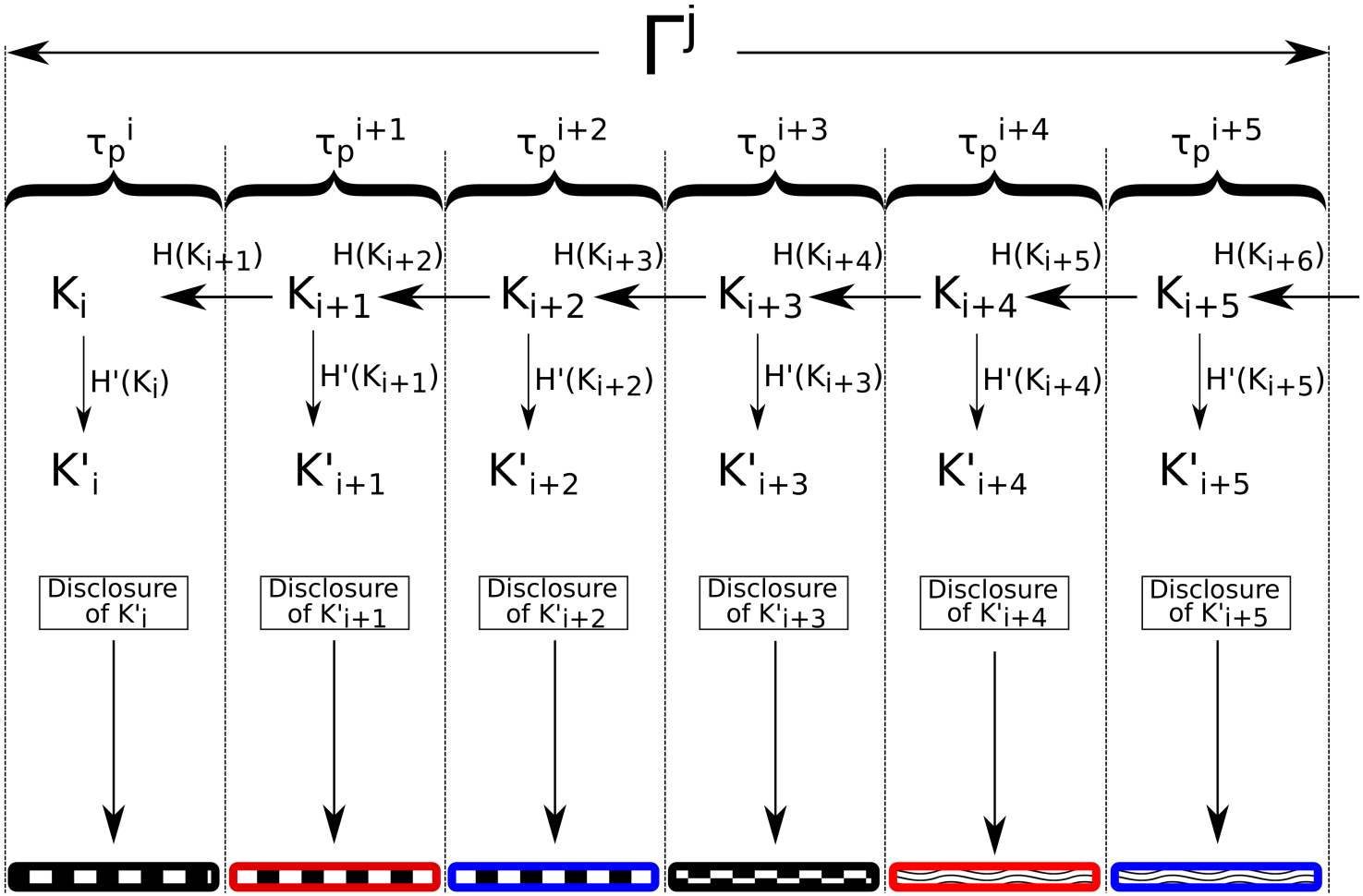


Figure: A High-level Overview of VPKIaaS Architecture on the Cloud

Issuing Multiple Pseudonyms in a Γ Interval



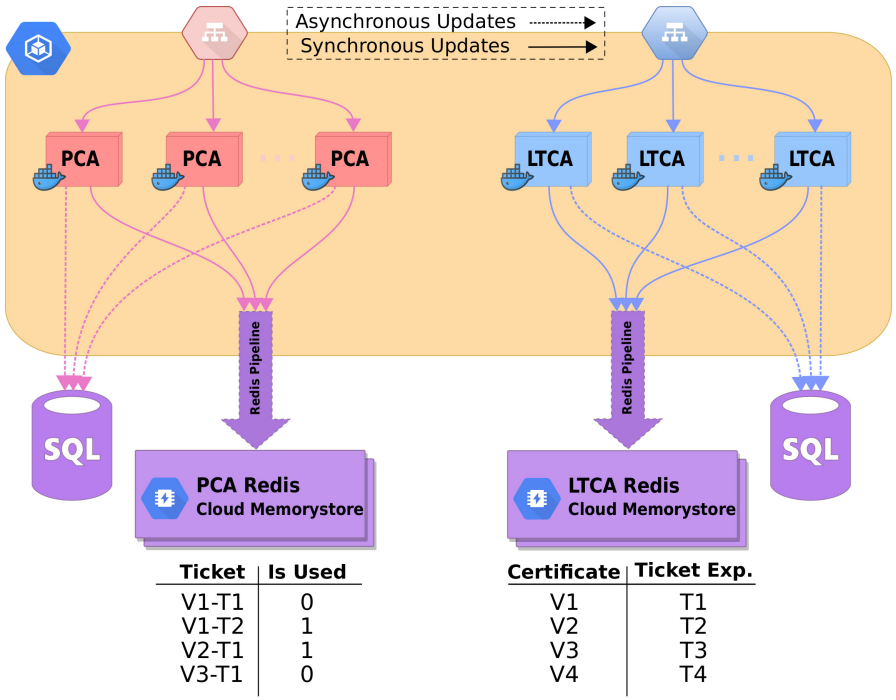
VPKIaaS Memorystore with Redis and MySQL

LTCA Sybil Attack Mitigation:

- Checking if a ticket was issued to the requester during that period
- Storing the serial number of the vehicle's LTC (as the key) and the expiration time of its current ticket (as the value) on the Redis database
- Invoking ticket issuance procedure

PCA Sybil Attack Mitigation:

- Checking if pseudonyms were issued to the requester of a given ticket
- Updating the Redis database with the value of true (i.e., used)
- Invoking pseudonym issuance procedure



VPKIaaS Memorystore with Redis & MySQL

VPKI Secrecy Analysis for Dolev-Yao Adversaries

User-sensitive Piece of Information	Entity	Secrecy	Strong Secrecy (Unlinkability)
Vehicle Id (LTC)	V, LTCA	✓	✓
Ticket ($tkt/n\text{-}tkt$)	V, LTCA, PCA	✓	✓
Pseudonym Certificate Signing Request (CSR) $((K_v^1, \ldots, K_v^n)_{\sigma_{k_v^i}})$	V, PCA	✓	✓
Pseudonym $((P_v^i)_{\sigma_{pca}})$	V, LTCA, PCA	✓	✓
Timestamps (t_s, t_e)	V, LTCA, PCA	✓	✓
Random number (Rnd_{tkt})	V, LTCA	✓	✓
Random number ($Rnd_{CK_{tkt}}$)	V, LTCA, PCA	✓	✓
Random number (Rnd_{psnym})	V, PCA	✓	✓
Ticket Commitment Key (CK_{tkt})	V, LTCA, PCA	✓	✓
Pseudonym Commitment Key (CK_P)	V, PCA	✓	✓

Experimental Setup

- **VPKI testbed**
  - Implementation in C++, OpenSSL for cryptographic protocols & primitives, TLS and Elliptic Curve Digital Signature Algorithm (ECDSA)-256.
  - FastCGI to interface Apache web-server; we use XML-RPC & Google Protocol Buffers
- **VPKIaaS system**
  - Built and pushed Docker images for LTCA, PCA, RA, MySQL, and Locust, *an open source load testing tool*, to the Google Container Registry
  - Google Kubernetes Engine (GKE) v1.10.11
  - Configured a cluster of five Virtual Machines (VMs) (n1-highcpu-32), each with 32 vCPUs and 28.8GB of memory
- **VPKIaaS Memorystore**
  - Redis, in-memory key-value data store, and MySQL

Table: Experiment Parameters

Parameters	Config-1	Config-2	Config-3
total number of vehicles	1000	100, 50,000	5000
hatch rate	1	1, 100	5, 10, 15, 20, 25
interval between requests	1000-5000 ms	1000-5000 ms	30000-60000 ms
pseudonyms per request	100, 200, 300, 400, 500	100, 200, 500	100, 200
LTCA memory request	128 MiB	128 MiB	128 MiB
LTCA memory limit	256 MiB	256 MiB	256 MiB
LTCA CPU request	500 m	500 m	500 m
LTCA CPU limit	1000 m	1000 m	1000 m
LTCA HPA	1-40; CPU 60%	1-40; CPU 60%	1-40; CPU 60%
PCA memory request	128 MiB	128 MiB	128 MiB
PCA memory limit	256 MiB	256 MiB	256 MiB
PCA CPU request	700 m	700 m	700 m
PCA CPU limit	1000 m	1000 m	1000 m
PCA HPA	1-120; CPU 60%	1-120; CPU 60%	1-120; CPU 60%

- Config-1: normal vehicle arrival rate; every 1-5 sec, a new vehicle joins the system, requesting 100-500 pseudonyms
- Config-2: for a flash crowd scenario; beyond having vehicles joining the system based on Config-1, 100 new vehicles join the system every 1-5 sec, requesting 100-200 psnyms.

Experimental Setup (cont'd)

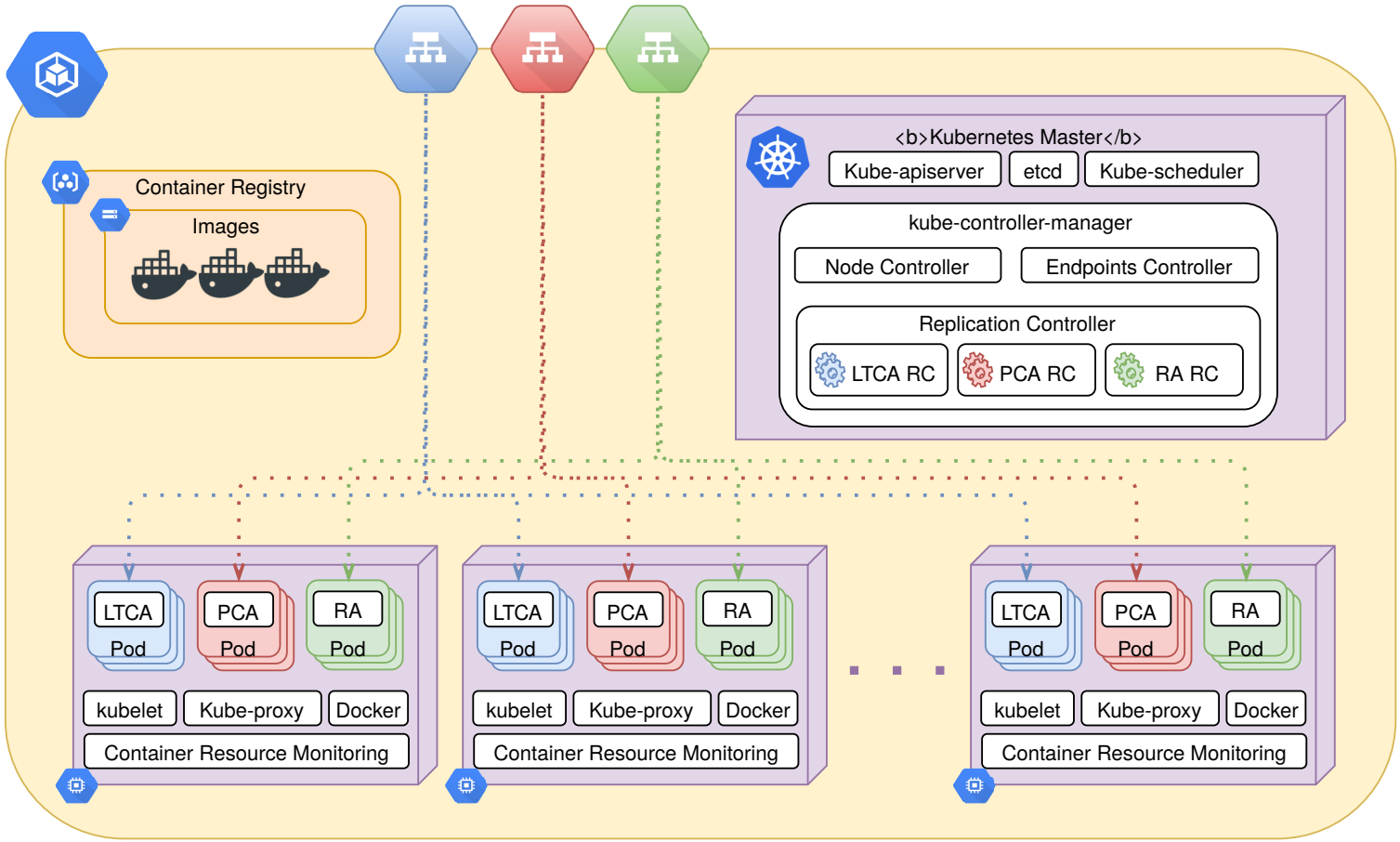
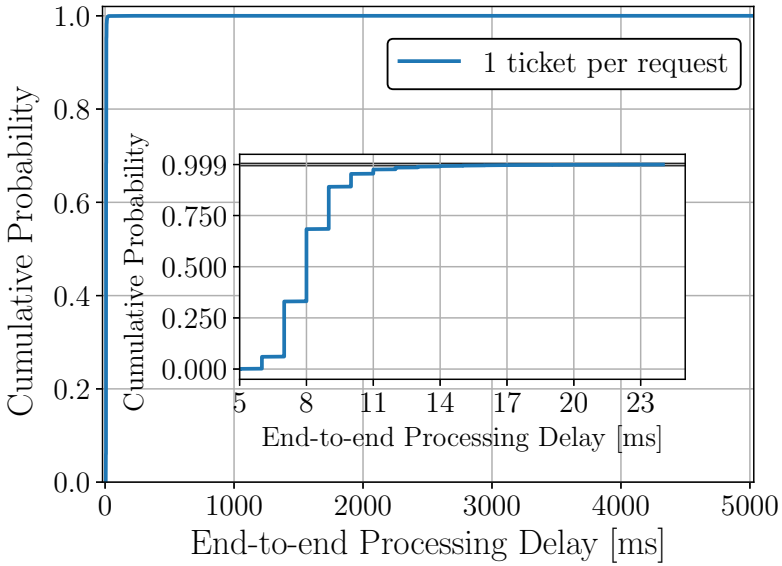


Figure: A High-level Overview of VPKIaaS Architecture on the Cloud

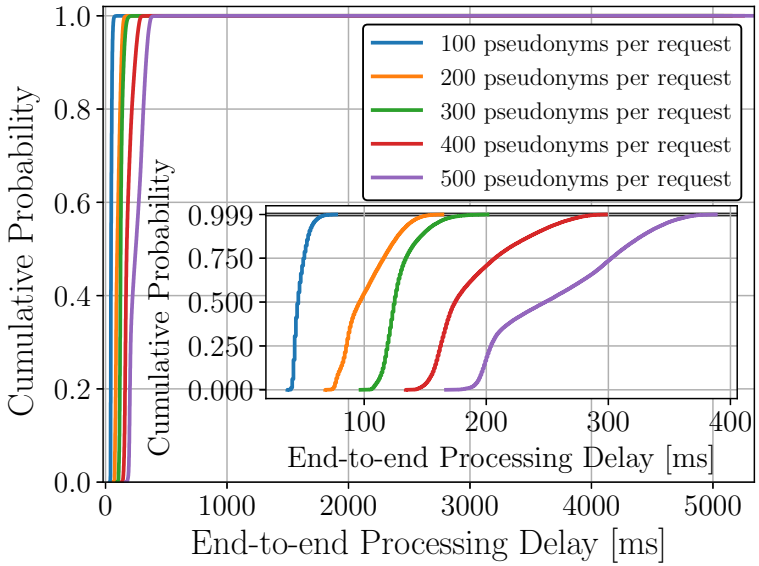
Experimental Setup (cont'd)

- **Network connectivity**
  - Varies depending on the actual OBU-VPKI connectivity
  - Reliable connectivity to the VPKI (e.g., RSU, Cellular, opportunistic WiFi)
- **Metrics**
  - **End-to-end processing delay to issue tickets and pseudonyms**
  - **High-availability and dynamic-scalability**
- **Use cases**
  - Large-scale pseudonym provision
  - VPKIaaS with flash crowd load pattern
  - Dynamic-scalability of the VPKIaaS
- **Remark**
  - Pseudonyms are issued with non-over-lapping intervals, to mitigate Sybil-based misbehavior
  - **Average daily commute time is 10-30 min. (actual urban vehicular mobility dataset), or 1 hour (according to the US DoT)**
  - **Obtaining 100 and 500 pseudonyms per day implies pseudonyms lifetimes of 14.4 min. ($\tau_P = 14.4$ min.) or 3 min. (τ_P =172.8 sec), respectively, covering 24 hours trip duration**
  - **Requesting pseudonyms based on Config-2, i.e., VPKIaaS system would serve 720,000 vehicles joining the system within an hour**

Performance Evaluation



(a) E2E latency to issue a ticket

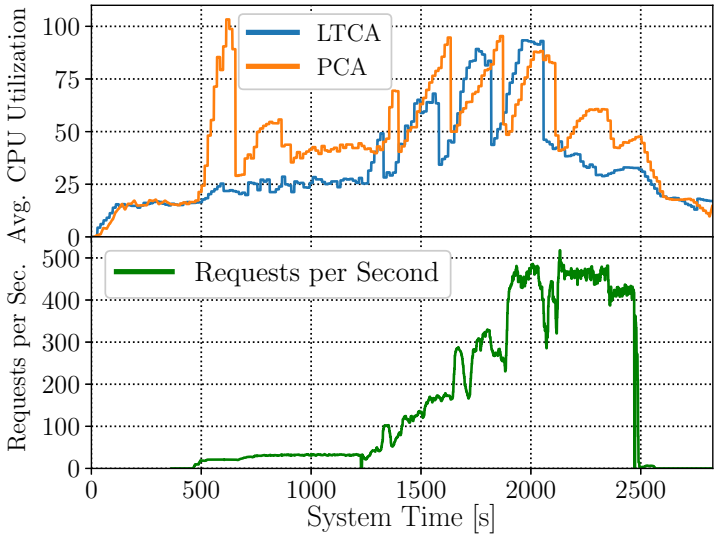


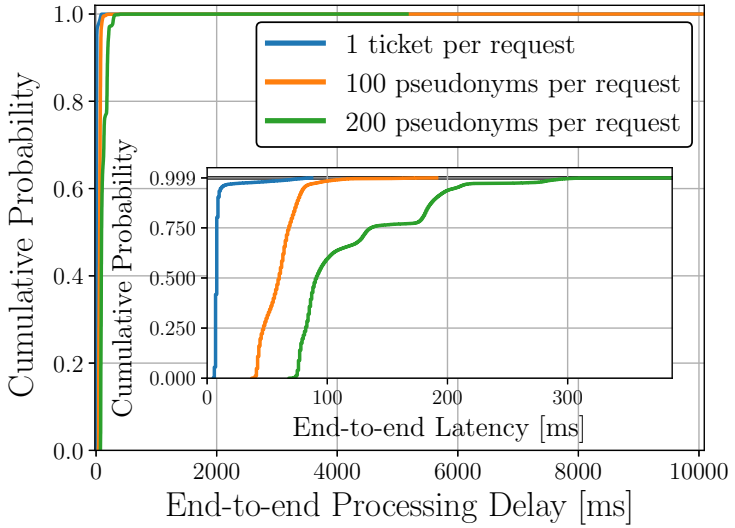
(b) E2E processing delay to issue psnyms

Large-scale pseudonym acquisition (based on Config-1):

- End-to-end Latency for ticket: $F_x(t = 24\ ms) = 0.999$
- With a batch of 100 pseudonyms per request, 99.9% of the vehicles are served within less than 77 ms ($F_x(t = 77\ ms) = 0.999$)
- With a batch of 500 pseudonyms per request, the VPKIaaS system efficiently issues pseudonyms: $F_x(t = 388\ ms) = 0.999$

Performance Evaluation (cont'd)





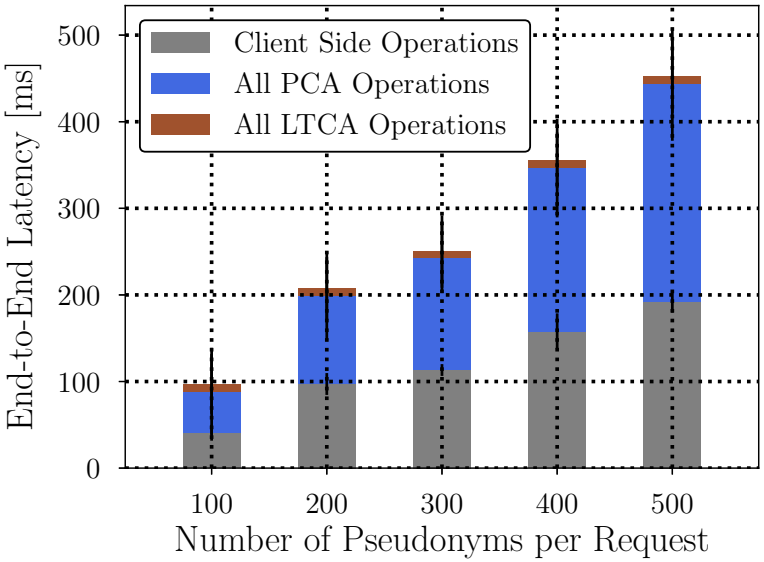
(a) CPU utilization and the number of requests per second (100 psnyms/req)

(b) CDF of processing latency to issue tickets and pseudonyms

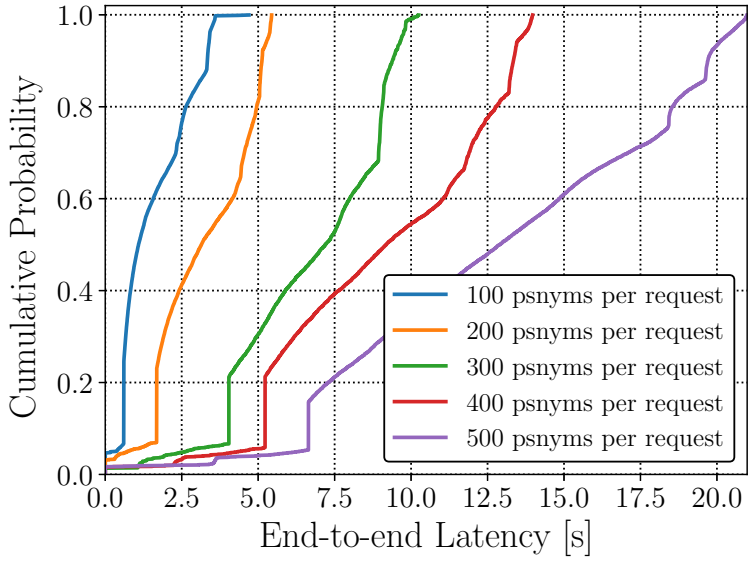
VPKIaaS system in a flash crowd load situation (based on Config-2):

- CPU utilization hits 60% threshold, services scale out, CPU utilization drops
- The processing latency to issue a single ticket is: $F_X(t = 87\ ms) = 0.999$
- Issuing a batch of 100 pseudonyms per request: $F_X(t = 192\ ms) = 0.999$
- 'normal' conditions vs. flash crowd: processing latency of issuing a single ticket increases from 24 ms to 87ms; the processing latency to issue a batch of 100 psnyms increased from 77ms to 192ms

Performance Evaluation (cont'd)



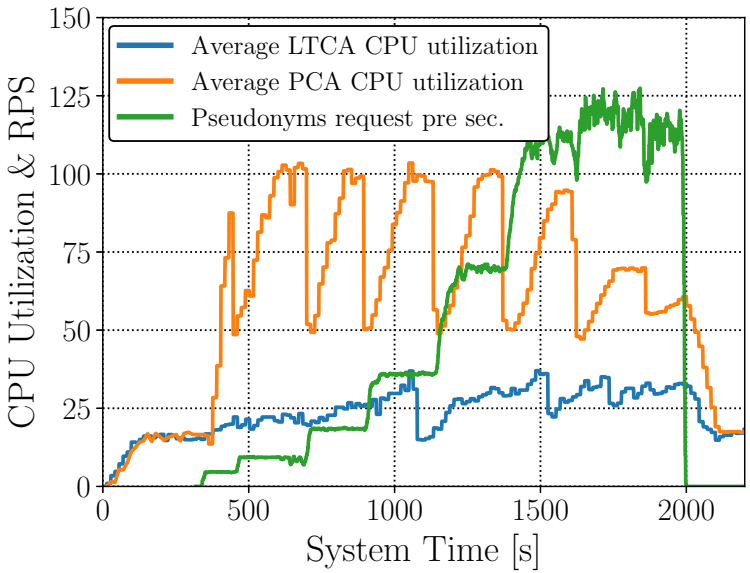
(a) E2E latency

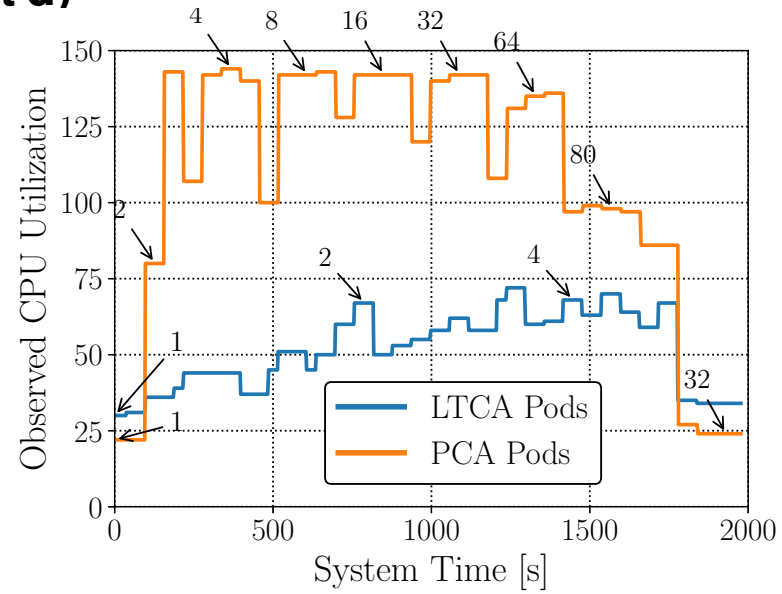


(b) CDF of E2E latency

VPKIaaS system with flash crowd load pattern. (a) Average end-to-end latency to obtain pseudonyms. (b) CDF of end-to-end latency, observed by clients.

Performance Evaluation (cont'd)



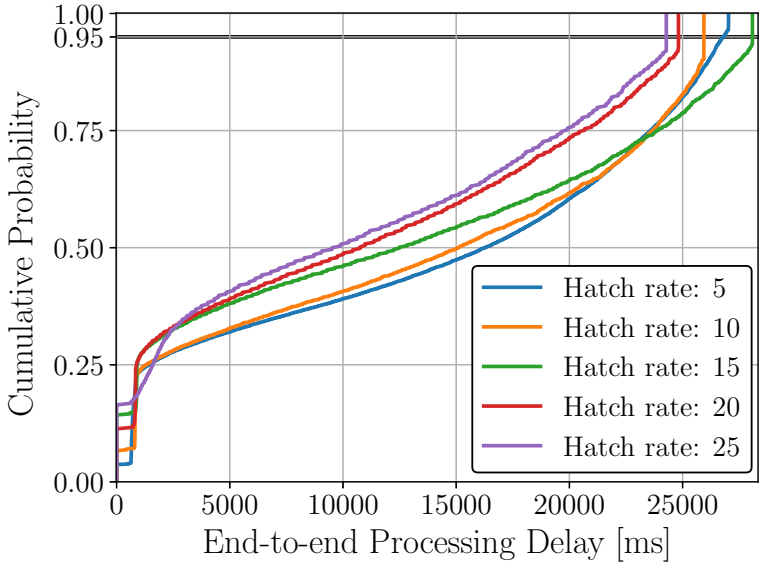


(a) Active vehicles and CPU utilization **(b)** Dynamic scalability of VPKIaaS system

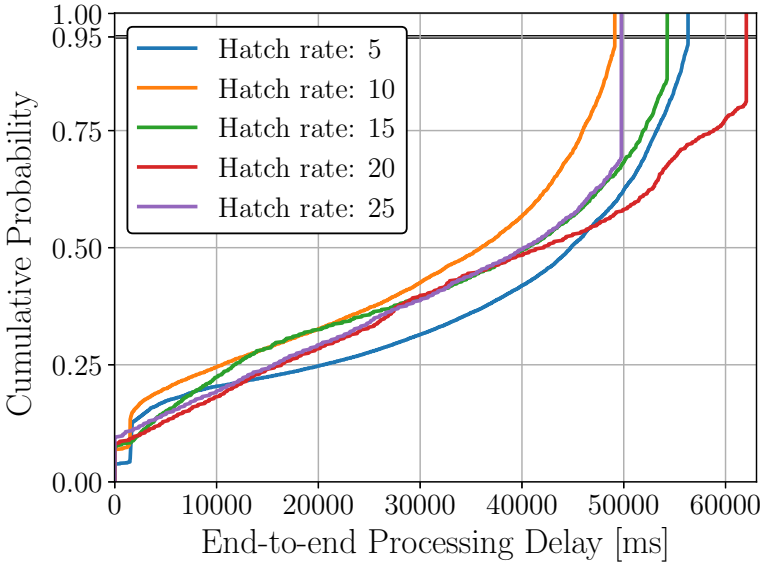
Reliability and dynamic scalability of the VPKIaaS system (with flash crowd load pattern, based on Config-2):

- Each vehicle requests 500 pseudonyms (CPU utilization observed by HPA)
- Synthetic workload generated using 30 containers, each with 1 vCPU and 1GB of memory (based on Config-2)

Performance Evaluation (cont'd)



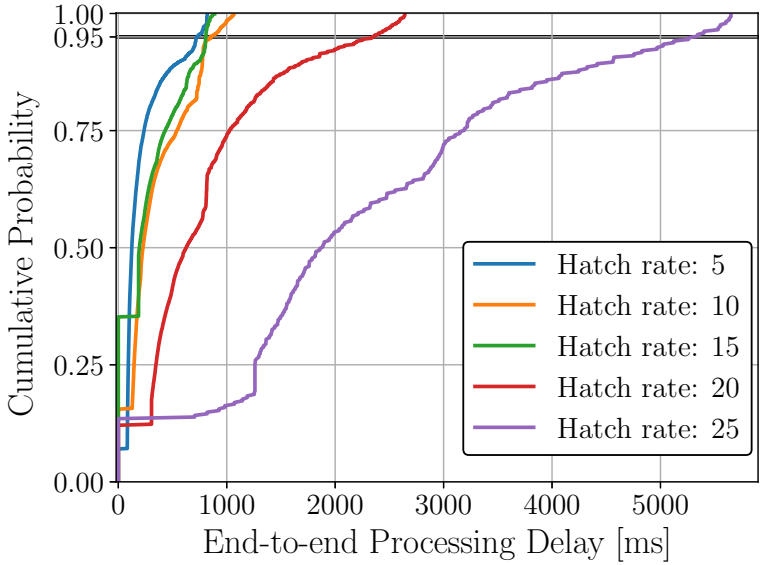
(a) 100 pseudonyms per request



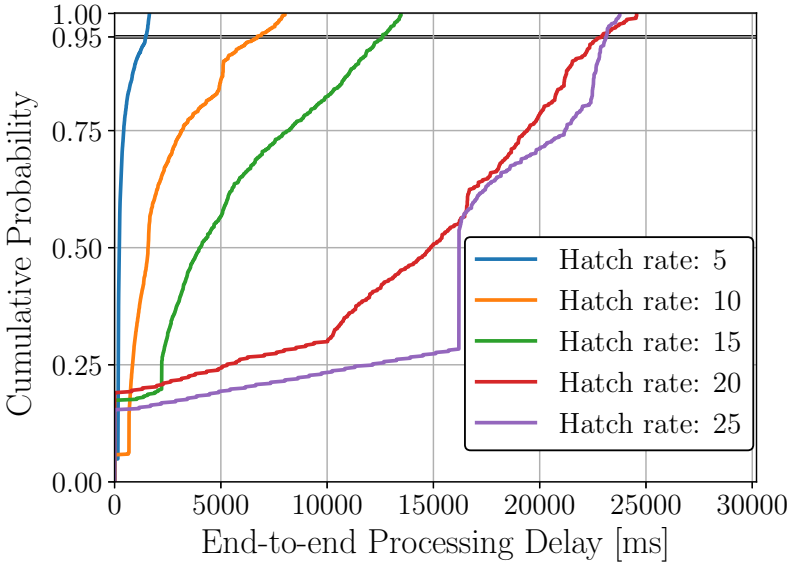
(b) 200 pseudonyms per request

Pseudonym acquisition with SECMACE.

Performance Evaluation (cont'd)



(a) 100 pseudonym per request



(b) 200 pseudonym per request

Pseudonym acquisition with SECMACE+ (VPKIaaS).

Experimental Setup (cont'd)

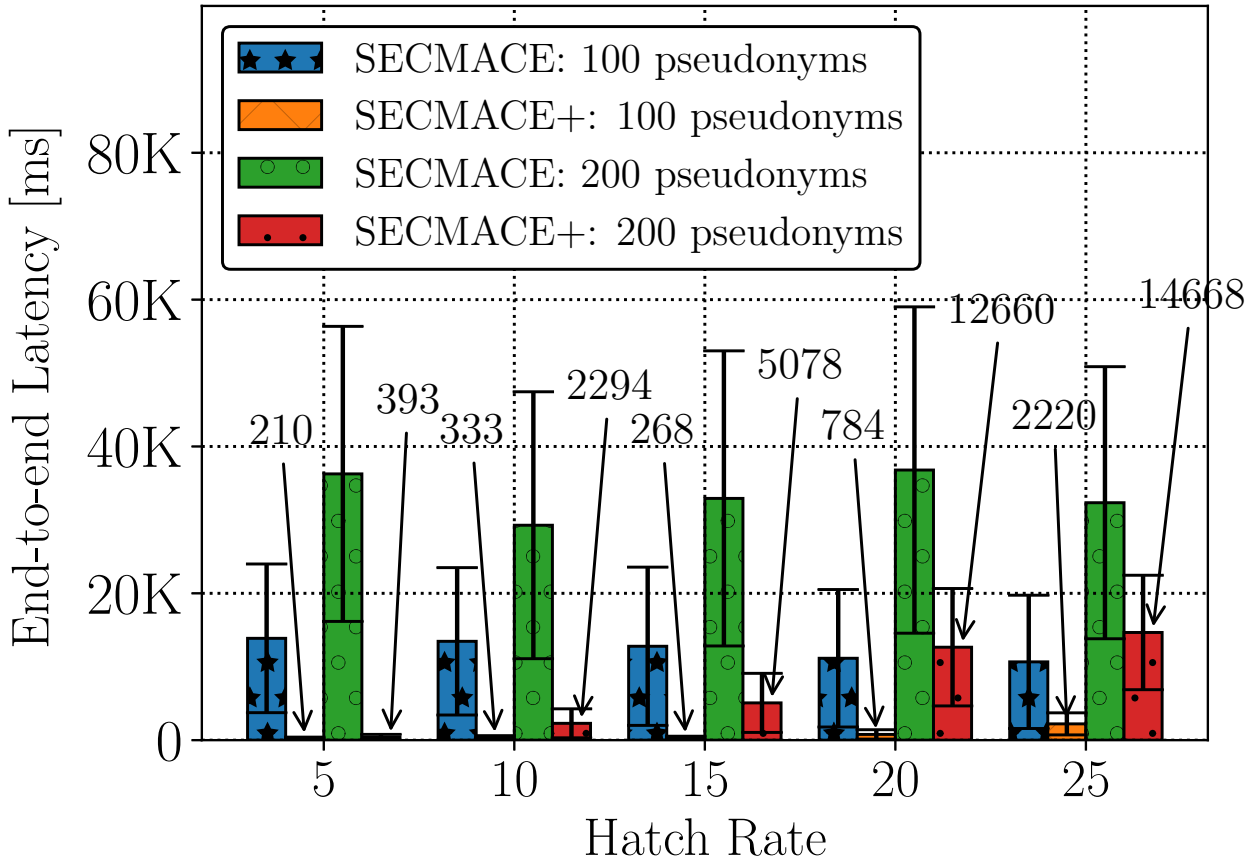


Figure: Pseudonym acquisition comparison between SECMACE and SECMACE+ (VPKIaaS with Redis Enabled).

SECMACE+: Upscaling Pseudonymous Authentication for Large Mobile Systems

Mohammad Khodaei, Hamid Noroozi,
and Panos Papadimitratos

Networked Systems Security Group

March, 2023