A Security Credential Management System for V2V Communications

William Whyte (Security Innovation); André Weimerskirch (ESCRYPT); Virendra Kumar (ESCRYPT); Thorsten Hehn (Volkswagen of America)
Outline of presentation

- Significance of this design
  - There are lots of papers written every year about certificate management for V2V safety, why is this special?
  - If V2V safety communications happen, the design in this presentation is the leading candidate for real-world deployment in the US.
- Overall architecture + privacy by design
- Original features of the design
  - Linkage authorities and linkage values
  - Butterfly keys
Who are we and what are we doing?

• Crash Avoidance Metrics Partnership (CAMP)
  • Founded by Ford and GM, forms and manages project teams for pre-competitive technical research
  • Partner organization, Vehicle Infrastructure Integration Consortium (VIIC), provides coordinated policy statements from automotive OEMs

• CAMP Vehicle Safety Communications 3 (VSC3) Consortium: Ford, General Motors, Honda, Hyundai-Kia, Mercedes-Benz, Nissan, Toyota*, and Volkswagen / Audi

• VSCS Aim: Develop a design for a Secure Credential Management System (SCMS) suitable for deployment across 300 million vehicles
  • Plus potentially aftermarket and nomadic devices
  • Identify full set of functionality that must be supported in day 1 devices

* Toyota is not part of the VSCS Study Team developing the SCMS
Background

- 32,000 deaths on the road in the US in 2012
- Significant reduction may be possible from V2V wireless communications for 360° warning applications.
  - 300 m range, 802.11-derived medium access
  - Basic Safety Message (BSM)
    - Location, velocity, steering angle…
  - Allows receiving unit to predict collisions
    - Forward, longitudinal, intersection
  - Warn driver, driver action can prevent or reduce impact of collision
  - Spectrum reserved for these communications since 1999
- USDOT (NHTSA) currently considering mandating this system for inclusion in new light vehicles
  - Decision on mandate to be made 2013, some years before it takes effect
Security considerations

• Risk of false messages
  • Reduce users’ faith in system and cause warnings to be ignored
  • (not safety-related): Messages may affect choice of route or have other mobility/efficiency impacts
  • Requirement: must be able to detect untrustworthy senders or messages and let receivers know not to trust them

• Impact on privacy
  • Don’t want the system to be used as a tracking system
    • Tracking is always possible, don’t want this option to be the cheapest
  • Prevent eavesdroppers or insiders from collecting Personally Identifiable Information (PII)
  • Conflict with requirement to detect and remove untrustworthy senders

• Design constraints
  • Constraints on available data rate using current V2V system (6 MBps under ideal conditions)
  • Cost-sensitive suppliers: limits on processing power, storage, connectivity, number of 5.9 GHz radios, …
Security concept of operations

- Protect against false messages:
  - Messages are signed and not encrypted
    - Signed using ECDSA over the NISTp256 curve
  - Signed message includes (or references) a certificate that specifies permissions (not identity) of holder
  - Misbehaving units can have their certificates identified and revoked
- Protect privacy:
  - Don’t directly reveal information: No personal information included in broadcast messages
  - Prevent tracking: “Identifiers” at application, network and other levels should be transient
    - Eavesdropper can only track from place to place if they record all your messages
  - Vehicles have a number of simultaneously valid certificates, can choose which certificate to use to sign each message
    - Baseline number of certs =20 per week
    - When cert changes, all other identifiers change too
      - Currently no standardized algorithm for cert change
  - SCMS is split into a number of components so that no individual component knows the full set of certificates that belong to a single device
  - Policy: out of scope for this presentation (and CAMP). Could consider
    - Restricting law enforcement use of the system
    - Data retention rules for storage of BSMs
Privacy by Design, an OEM perspective

• Privacy from attacks by an SCMS insider
  • Don’t link certificates to VIN or require legal process
  • Separate operation of SCMS components:
    Two or more components should not be run by the same organization without “proper” separation if the combined information held by the components would allow the organization to track* a vehicle

*predict next pseudonym certificate based on current one or find out whether two certificates belong to the same device

• Divide functionality between SCMS components as necessary to satisfy this approach
Overview / Standard PKI Hierarchy

- SCMS Manager
  - Policy
  - Technical
- Root CA
- Intermediate CA
- Pseudonym CA
- Request Coordination
- Registration Authority
- Enrollment CA
- Device Config. Manager
- Location Obscurer Proxy
- Device 1
- Device 2
- Device 3
- Misbehavior Authority
  - Internal Blacklist Manager
  - Global Detection
  - CRL Generator
  - CRL Store
  - CRL Broadcast
- Linkage Authority 1
- Linkage Authority 2
Unique Features

- RA shuffle for privacy
- Certificate request: Butterfly keys
  - Allows more responsiveness & robustness, less work on OBE
- Certificate issuance and revocation: Linkage authorities and linkage values
  - Allows efficient revocation while preventing any SCMS component from tracking non-revoked vehicle
- Misbehavior analysis and revocation
  - Allows certs from misbehaving vehicles to be linked while respecting the privacy of correctly behaving vehicles
Shuffle at the RA

- RA receives requests from multiple end-entity devices
- Combines requests so that PCA doesn’t know that two individual cert requests received at the same time come from the same vehicle
Butterfly keys: Certificate generation goals

• OBE could simply generate a large number of cert requests and send them encrypted to the PCA, but:
  • OBE is constrained
    • Minimum processing on the OBE
    • Minimum wasted processing on the OBE
  • Connectivity is not guaranteed
    • Small uploads
    • Want to request as many certificates as possible at a given time
  • What if the PCA goes out of business?
• Butterfly keys address all these issues
  • Performance and robustness enhancement, not security enhancement as such
Butterfly keys: concept

- Device generates
  - A seed or “caterpillar” keypair
  - An expansion function
  - Cost: ~1 key generation
Butterfly keys: concept

- Device generates
  - A seed or “caterpillar” keypair
  - An expansion function
  - Cost: ~1 key generation
- RA runs the expansion function to generate “cocoon” public keys from the caterpillar public key
  - Cocoon public keys from the same caterpillar keys are not correlated
  - Expansion function lets you generate arbitrarily many cocoon keys
  - RA submits cocoon keys to CA for certification
Butterfly keys: concept

- Device generates
  - A seed or “caterpillar” keypair
  - An expansion function
  - Cost: ~1 key generation
- RA runs the expansion function to generate “cocoon” public keys from the caterpillar public key
  - Cocoon public keys from the same caterpillar keys are not correlated
  - Expansion function lets you generate arbitrarily many cocoon keys
  - RA submits cocoon keys to CA for certification
- CA randomizes each public key separately so the RA can’t recognize them
  - Certs contain the resulting “butterfly” keys
  - CA returns certs and private randomization values to the OBE
- Result: Large number of certs generated from a single initial keypair
  - OBE is the only device that knows private keys
  - Public keys cannot be correlated by any entity
  - Low computational burden on OBE at request time
  - Request once, generate keys for the entire lifetime of the vehicle
Butterfly keys vs goals

- Minimum processing on the OBE:
  - One cert request from OBE allows generation of arbitrary number of individual certs
- Minimum wasted processing on the OBE:
  - Certs that are not used need not be decrypted
- Small uploads:
  - Upload is two public ECC keys + two expansion functions (= AES keys)
- Want to request as many certificates as possible at a given time
  - One cert request from OBE allows generation of arbitrary number of individual certs
- What if PCA goes out of business?
  - Requests are not encrypted for a particular PCA; any PCA change can be handled on the backend by the RA
Revocation and Linkage Authorities

• Why do we need revocation?
  • Why not just choose not to issue new certs to a misbehaving vehicle?
• Not all vehicles will have good data connection
  • Even vehicles that do may be out of coverage
  • Vehicles need to be provisioned with a minimum number of certs in case they are turned off for some time and turned on in an area with no coverage
• If you have a month’s worth of certs, you can misbehave for a month
  • If you have three months’ worth of certs, you can misbehave for three months
  • If you have three years’ worth of certs…
• Revocation must be supported to reduce potential disruption within system, even if in practice it isn’t used.
• Need efficient, privacy-preserving revocation
Revocation and Linkage Authorities

- Revoke all $n$ of a device’s certs with just one entry on the CRL
- Multiple certs valid in one time period
- Backwards unlinkability
- No component in the SCMS knows the chain
Revocation and Linkage Authorities

- Revoke all $n$ of a device’s certs with just one entry on the CRL
- Multiple certs valid in one time period
- Backwards unlinkability
- No component in the SCMS knows the chain
  - LAs encrypt chain for PCA
    - Send to RA
  - RA groups
  - PCA decrypts, XORs

\[ k_0 \]
\[ k_1 \]
\[ k_2 \]
\[ k_{imax} \]
Revocation

SCMS Manager

Policy

Technical

Root CA

Intermediate CA

Pseudonym CA

Request Coordination

Registration Authority

Enrollment CA

Certification Services

Device Config. Manager

Location Obscuer Proxy

Misbehavior Authority

Linkage Authority 1

Linkage Authority 2

CRL Store

CRL Generator

CRL Broadcast

Legend

Directly acts in this use case

Provides information before execution

Device 1

Device 2

Device 3
Misbehavior investigation

- Misbehavior reporting:
  - OBE -> MA
- Misbehavior analysis:
  - MA by itself
- Misbehavior investigation:
  - MA asks PCA if two certs belong to same vehicle
  - PCA asks LAs
  - Yes/no answer
  - Interfaces can be defined to require evidence to be presented at each stage
  - Interfaces protect privacy – only yes/no answer, linkage seeds are not revealed
  - If a vehicle misbehaves often enough it can be revoked
- Revocation:
  - Linkage seed from each LA goes on the CRL
  - CRL recipients at each time period:
    - Hash linkage seeds forward to that time period
    - Calculate 20 pre-linkage values for each
    - XOR to get linkage value
    - Compare to received cert and reject if match
Outlook and Ongoing Projects

- **VSCS Study One: Design Optimization and Cost Analysis of Connected Vehicle Security System**


- **Activities:**
  - Define baseline security model and baseline OBE requirements
  - Develop security system cost model
  - Perform cost analysis on baseline security model
  - Analyze potential simplifications to the deployment model
  - Analyze alternative device-SCMS connectivity approaches
  - Identify technical approaches to linking enrollment certificates to batches of devices to aid defect investigations
  - Provide design recommendations for V2V Security System
Extra slides
Butterfly Keys: Elliptic Curve background

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, A = aG</td>
<td>G, A</td>
</tr>
<tr>
<td>a = private key, A = public key, G = base point</td>
<td></td>
</tr>
<tr>
<td>Alice uses a to sign</td>
<td></td>
</tr>
<tr>
<td>Bob knows A and G but can’t find a</td>
<td></td>
</tr>
<tr>
<td>Bob can use A to verify Alice’s signatures</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>b, B = bG</th>
</tr>
</thead>
<tbody>
<tr>
<td>“ephemeral keypair”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a+b, A+B</th>
<th>b, A+B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+B = (a+b) G</td>
<td></td>
</tr>
<tr>
<td>Only Alice knows a+b although Bob has contributed to key</td>
<td></td>
</tr>
<tr>
<td>Alice can sign with (a+b) just as with any private key; no-one else can</td>
<td></td>
</tr>
<tr>
<td>Bob and others can verify with A+B just as with any public key</td>
<td></td>
</tr>
</tbody>
</table>

Why does this matter?
Butterfly key process

(Notation is different from paper for space reasons)

<table>
<thead>
<tr>
<th>OBE</th>
<th>RA</th>
<th>PCA</th>
</tr>
</thead>
</table>
| a, A = aG, $f_s(i, j)$ | $A, f_s$ | $f_s = \text{“pseudorandom permutation”}$
| | | $= \text{AES}_k(i \parallel j)$ for some k |
| | $B_{1,1} = A + f_s(1,1)G$ | $a + f_s(1,1)$ is private key for $B_{1,1}$ |
| | $B_{1,2} = A + f_s(1,2)G$ | $a + f_s(1,2)$ is private key for $B_{1,2}$ |
| | $B_{1,3} = A + f_s(1,3)G$ | $a + f_s(1,3)$ is private key for $B_{1,3}$ |
| | $\ldots$ | $\ldots$ |
| | $B_{1,1}$ | c is randomly generated & distinct for each received B |
| $\text{Issue Cert}(B_{1,1} + C)$ | $c, C = cG$ | $E = \text{Enc}_{\text{OBE}}(\text{Cert}, c, \text{“1,1”})$ |
| | $\text{Enc}_{\text{OBE}}$ | Encrypt response so that RA can’t see cert contents |
| | Response encryption key is butterfly key formed from $(H, f_e)$ | |
| (Cert, c, “1,1”) | $\text{Sign}_{\text{CA}}(E)$ | $\text{Sign}_{\text{CA}}(E)$ |
| | | Signing proves that CA encrypted message, not RA |
| $a + f_s(1,1)$ | | $a + f_s(1,1) + c$ is private key for |
Butterfly keys: OBE to RA

- Start with a single “caterpillar” public key $A$ in a cert request
  - $A = aG$, $a$ = private key (integer) mod $p$, $G$ = Elliptic Curve Base Point
  - Given $A$ & $G$, very hard to find the value $a$
  - $(a+b)*G = aG + bG$

- Want to expand this to certs for time period $(i, j)$
  - OBE defines *expansion function* $f_s(i, j)$ that takes $(i, j)$ to (pseudo)random integer mod $p$
    - Pick AES key $k$
    - $f_s(i, j) = \text{AES}_k(0^{128} \text{ XOR } [i_{32} \| j_{32}]) \| \text{AES}_k(1^{128} \text{ XOR } [i_{32} \| j_{32}])$
  - Shares $f_s(i, j)$ with RA (i.e. shares $k$)
  - Then RA can calculate $B_{ij} = A + f_s(i, j)*G$
    - $f_s$ is pseudorandom, so the PCA cannot determine that $B_{ij}$s from the same $A$ are related
    - Corresponding private key is $a + f_s(i, j)$ which *only OBE knows*

- So:
  - A single cert request from the OBE to the RA leads to…
  - Multiple individual uncorrelated public keys from the RA to the PCA
  - These can be shuffled together, protecting OBE privacy against PCA
Butterfly keys: RA to PCA

- One more requirement: RA must not know the public keys in the certs
  - But RA has put the public keys in the requests
- PCA generates an offset
  - Integer $c$, point $C = cG$, generated freshly at random for each request
  - PCA receives request containing $B_{ij}$, signs cert containing $B_{ij} + C$
    - $B_{ij} = \text{“cococon” public key}$, $B_{ij} + C = \text{“butterfly” public key}$
  - PCA returns $(c, \text{Cert})$ to RA to return to OBE
    - Encrypted under a separate butterfly encryption key
    - Ciphertext signed by PCA to prevent MITM attack by RA
    - Encrypted response includes indication of the request it is associated with so RA can return it to the right OBE
- Now:
  - Shuffle prevents PCA from knowing which certs go together
  - Offset prevents RA from knowing which certs go together
  - Only the OBE knows the contents of its certs
  - OBE knows $a$, $f_s(i,j)$, receives $c$:
    - $(a + f_s(i,j) + c) \cdot G = A + f_s(i,j)G + C = B_{ij} + C \leftarrow \text{public key in cert}$
    - … so $a + f_s(i,j) + c = \text{private key for cert}$
Revocation and Linkage Authorities

- Revoke all $n$ of a device’s certs with just one entry on the CRL
  - Include linkage value $l(i) = E_k(i)$ in the cert
  - Include key $k$ on CRL; in each time period $i$, vehicles calculate $E_k(i)$ for all entries and compare to the linkage value in the cert.
Revocation and Linkage Authorities

- Revoke all $n$ of a device’s certs with just one entry on the CRL
  - Include linkage value $l(i) = E_k(i)$ in the cert
  - Include key $k$ on CRL; in each time period $i$, vehicles calculate $E_k(i)$ for all entries and compare to the linkage value in the cert.
Revocation and Linkage Authorities

- Revoke all $n$ of a device’s certs with just one entry on the CRL
- Multiple certs valid in one time period
Revocation and Linkage Authorities

- Revoke all $n$ of a device’s certs with just one entry on the CRL
- Multiple certs valid in one time period
Revocation and Linkage Authorities

- Revoke all $n$ of a device’s certs with just one entry on the CRL
- Multiple certs valid in one time period
- Backwards unlinkability
Revocation and Linkage Authorities

- Revoke all \( n \) of a device’s certs with just one entry on the CRL
- Multiple certs valid in one time period
- Backwards unlinkability
Revocation and Linkage Authorities

- Revoke all \( n \) of a device’s certs with just one entry on the CRL
- Multiple certs valid in one time period
- Backwards unlinkability
- No component in the SCMS knows the chain
Revocation and Linkage Authorities

- Revoke all $n$ of a device’s certs with just one entry on the CRL
- Multiple certs valid in one time period
- Backwards unlinkability
- No component in the SCMS knows the chain
  - LAs encrypt chain for PCA
    - Send to RA
  - RA groups
  - PCA decrypts, XORs