IEEE 1609.2 and Connected Vehicle Security:

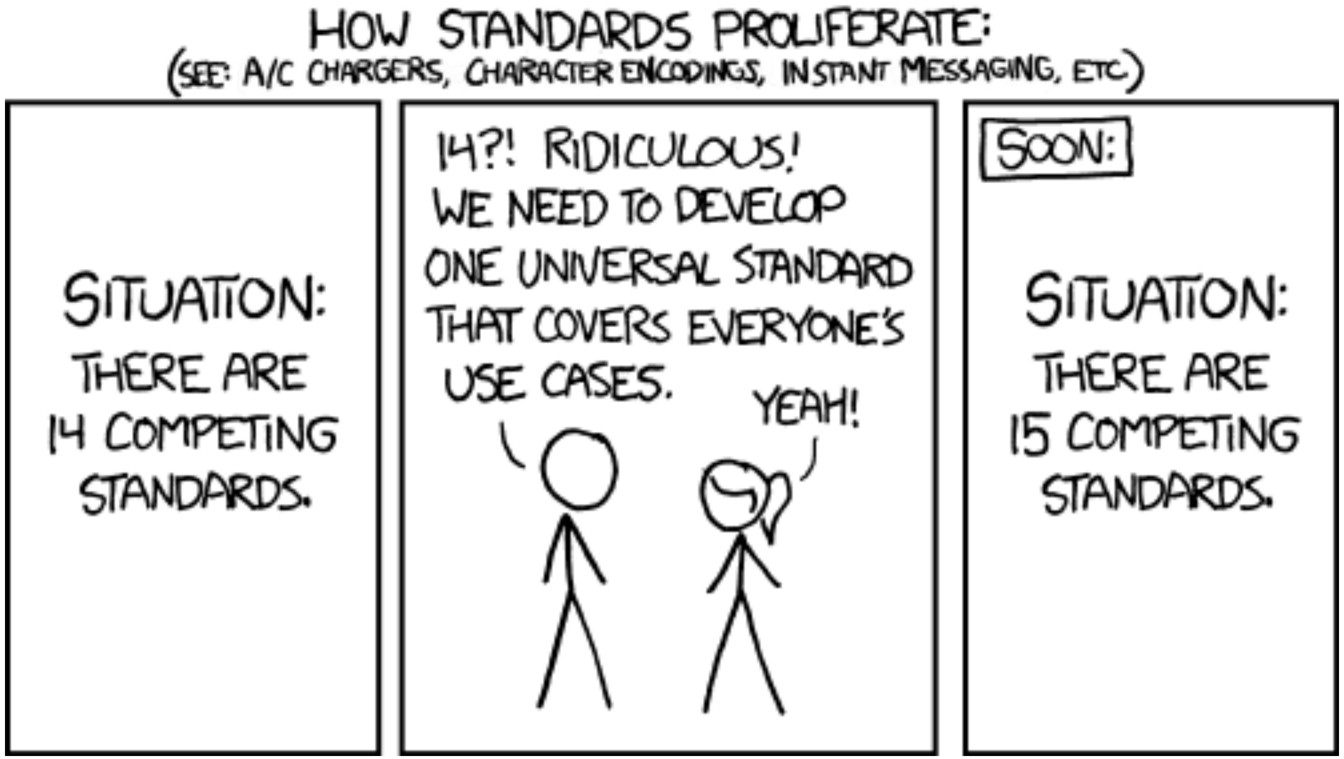
Standards Making in a Pocket Universe

William Whyte

Chief Scientist, Security Innovation

Security Standardization Research Workshop
December 6, 2016

SECURITY INNOVATION


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

- https://imgs.xkcd.com/comics/standards.png

- The nice thing about standards is that you have so many to choose from. – Andrew Tanenbaum (or Grace Hopper?)

- The nice thing about standards is that you have so many to choose from. – Andrew Tanenbaum
- However, as every parent of a small child knows, converting a large object into small fragments is considerably easier than the reverse process.

Abstract

- The Connected Vehicle system is going to result in more than 300 million devices with needs for secure communication.
- In 2003, the decision was taken to create a standard for security for this communication that was distinct from existing standards
  - X.509, S/MIME-CMS, etc
- The standard, IEEE 1609.2, contains a number of crypto and protocol design decisions that are different from protocols attempting to do similar things
  - Although it was developed in public, and although it will go into cars, it has not received the level of scrutiny that higher profile standards have
- In 2012, the US and European versions of this standard diverged and are now incompatible
  - Creates a dilemma for, e.g., Australia

Goals

- Explore how successful we were in developing a security standard with only piecemeal contributions from recognized experts
- Go in-depth on some decisions and see if they stand up
    - Small-scale: individual design choices
    - Large-scale: the choice to have a new standard in the first place
- For improvable decisions, understand how they came about
- Examine how the divergence in standards between US and EU happened and whether it could or should have been avoided
- Lessons learned: what would we do differently next time? What resources helped?

Goals

- Explore how successful we were in developing a security standard with only piecemeal contributions from recognized experts
- Go in-depth on some decisions and see if they stand up
  - Small-scale: individual design choices
  - Large-scale: the choice to have a new standard in the first place
- For improvable decisions, understand how they came about
- Examine how the divergence in standards between US and EU happened and whether it could or should have been avoided
- Lessons learned: what would we do differently next time?

Do a worked example of John Kelsey's presentation from yesterday

How can we make security standards better?

Full agreement that these are important

- Risks to standardization: pushback, doing the minimum
- Ease of review and clearness of rationale
- Transparency
- Institutional memory / robustness against losing key contributor
- Crypto agility

Needed more emphasis

- Clarity of requirements
- How requirements might not be purely technical
  - Line between requirements / assumptions / stubbornness is not always clear
  - A big villain of this piece is different preferences among secondary requirements
- Time pressure and coordination problems

Background

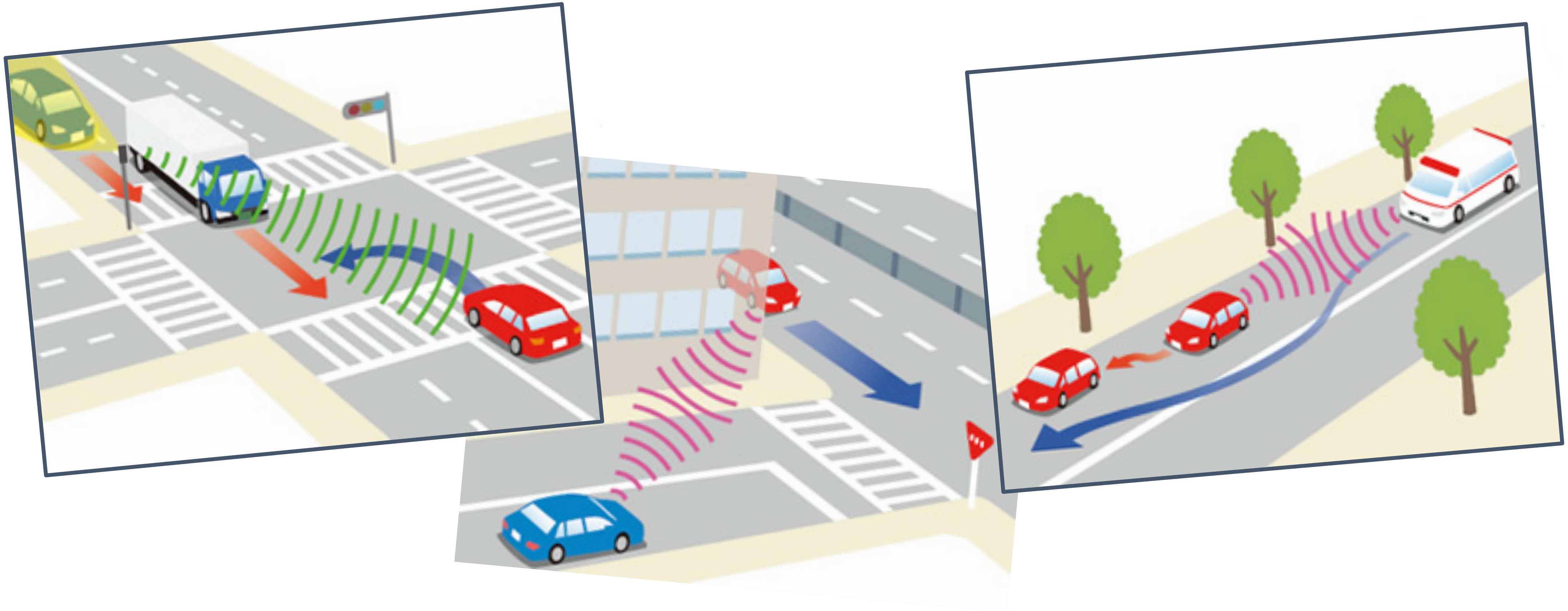
Traffic Safety

- 32,000 US road deaths, and 3,800,000 injuries
- Fatalities and injuries = $300B/year
- Congestion = $230B/year
- Leading cause of death for ages 15-34 in US

Technology Evolution

Passive → *Active* → *Proactive*

Our main setting: Vehicle-to-Anything (V2X)



Illustrations from https://www.itsconnect-pc.org/en/about_its_connect/service.html

V2X

- New technology but almost 20 years in the making
- Allows cars to avoid invisible danger
- Uses short-range DSRC radio
- 10 situational messages per second/car
- Plan for mandatory adoption by 202x

Source: U.S.DOT

V2X Pilots

- Ann Arbor Safety Pilot extended to 30,000 vehicles
- New York, 8,000 vehicles testing city safety
- Tampa, better freeway management
- Wyoming, improving I80 trucking safety and efficiency
- Many EU and Asia Pacific pilots
- All major manufacturers engaged

Ann Arbor
Huron
Washtenaw
Instrumented Signal
Curve Speed Warning
Instrumented Freeway
W 96th St
Jacqueline Kennedy Onassis Reservoir
Manhattan
Yorkville
Theodore Roosevelt Park
Upper East Side
Central Park
Lincoln Square
Broadway
W 59th St
E 59th St
Sutton Place
Midtown East
Clinton
W 42nd St
W 40th St
E 42nd St
E 40th St
W 34th St
E 34th St
W 23rd St
E 23rd St
Kips Bay
Riverside Dr
12th Ave
10th Ave
9th Ave
8th Ave
7th Ave
5th Ave
Madison Ave
Park Ave
3rd Ave
2nd Ave
1st Ave
Avenue of the Americas

Privacy and security

- Privacy and security are critical success factors for the system
- Robustness
- Effectiveness
- Public confidence

IEEE 1609.2 process: goal

- Produce a single security standard for a set of Connected Vehicle messages that is:
  - High quality
  - Suitable for use in all vehicles
  - Done!
  - As few options as possible
  - … but as flexible as necessary to support other messages not in the initial set

Security requirements

- All the usual ones
  - (Sometimes) confidentiality, integrity, authenticity, authorization, (sometimes) non-repudiation
  - Security and cryptography requirements depend on application setting
    - Focus for now is on broadcast messages
- Plus
  - Privacy: don't want tracking / traffic analysis to be easy
  - Channel congestion: 3-6Mbps channels
  - Constrained devices due to cost of automotive quality equipment – affects connectivity, hardware security, …
- Plus!
  - Security management: distributing security management information to devices that have intermittent Internet connectivity

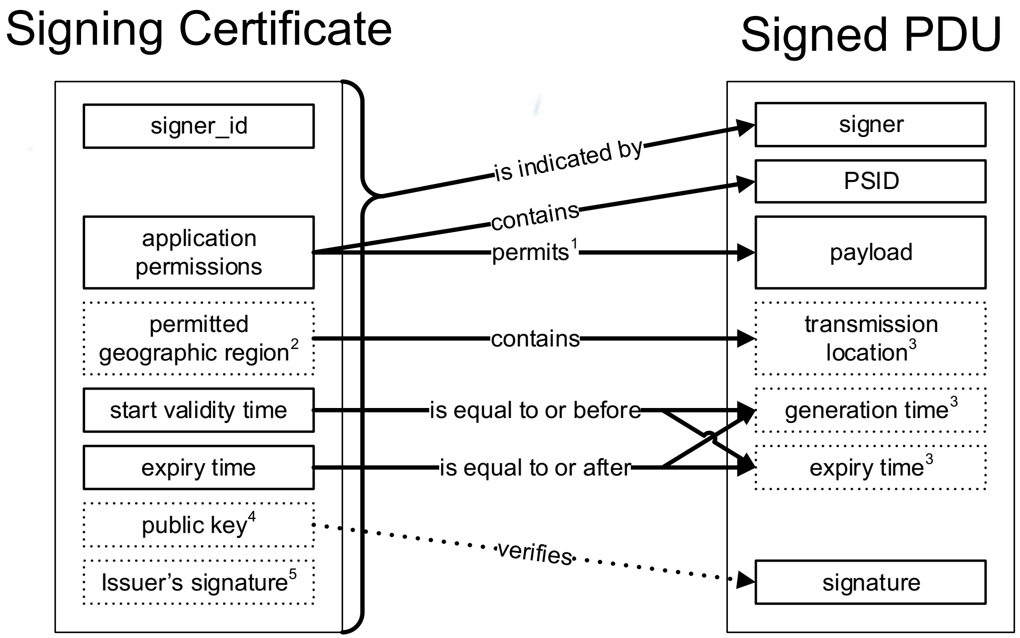
Why not X.509? Why not S/MIME (CMS)?

- Because of the packet size conventional X.509 certificates are unsuitable for use with VSC Security. They are excessively large and there is a large semantic gap between X.509 distinguished names and the names which make sense here. Although it is in principle possible to shoehorn VSC information into X.509 certificates, the result would be fairly unwieldy. In addition, because interoperability with conventional PKIs is not a priority and we do not anticipate using the signature algorithms that are commonly used with X.509 certificates, there is no significant advantage to using X.509 rather than a custom format.

WAP Forum, yesterday

Sign messages for AuthN/AuthZ

- Custom certificate and message format allows us to constrain:
  - Permissions (indicated by PSID)
  - Geographic region
- Role-based access control: **devices' permissions, not identities**, are stated in their certificates

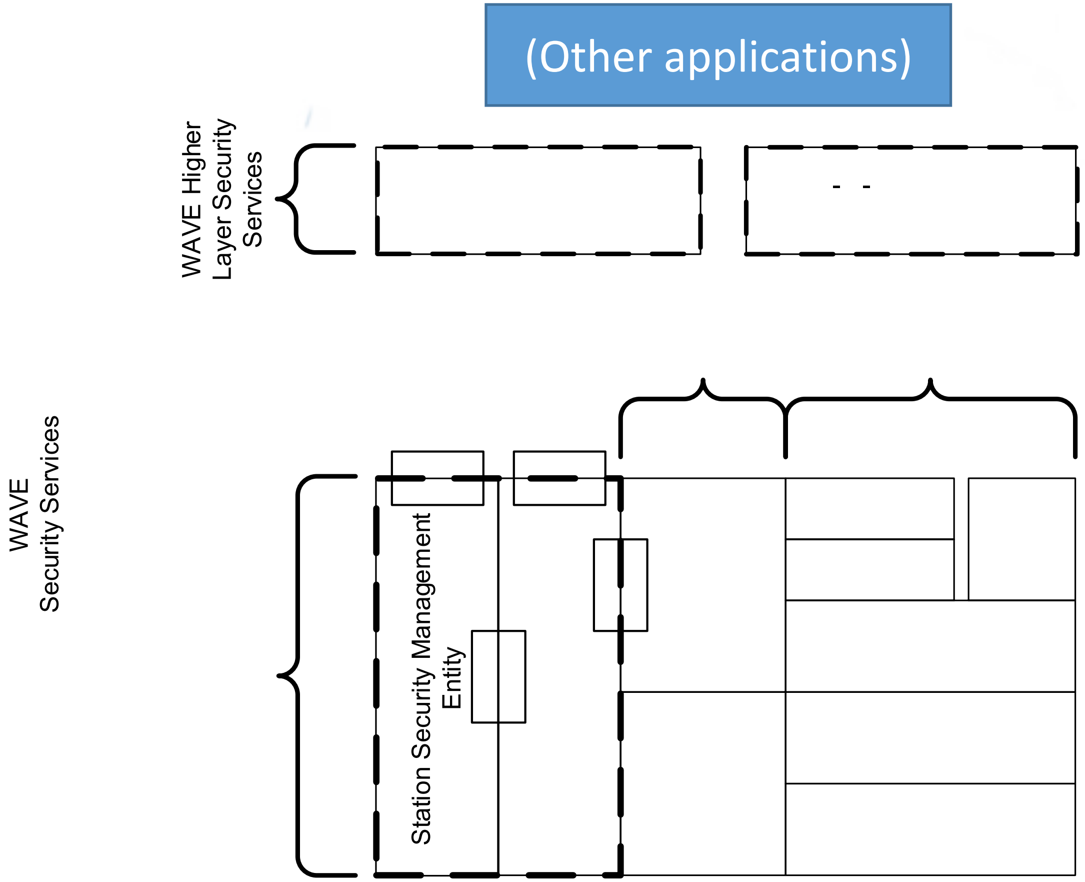


NOTES:

1. Determined using the PSID and SSP. The process to determine whether the operational permissions permit the message payload is specified by the organization reserving the PSID and is out of scope for this standard.
2. Included per policy set by the appropriate authority for the region where the certificate is being used.
3. Optional. Inclusion of this data is as determined by the organization reserving the PSID. This data may be contained in the payload or within the security header fields.
4. For implicit certificates, the public key is derived rather than explicitly stated within the certificate.
5. Not included in an implicit certificate.

Two interfaces

- Security services
  - Over the air – normative
  - Internal interfaces – informative
- Fits in the WAVE Device archictecture
- Interaction with application
  - What are responsibilities of application and what are responsibilities of security services?
  - Where is the line drawn?



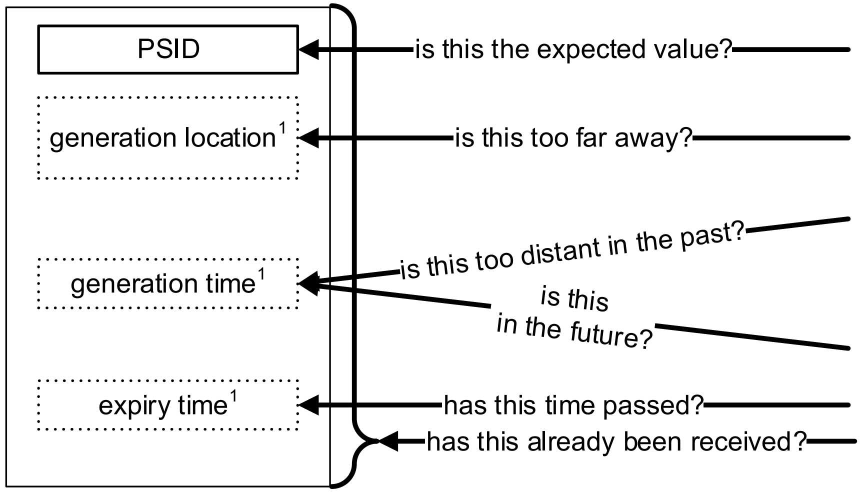
(Other applications)
WAVE Higher Layer Security Services
- -
WAVE Security Services
Station Security Management Entity

SECURITY INNOVATION

Relevance checks and application-specific checks

- Security services can carry out crude relevance checks
- But these are application dependent, e.g.:
  - Signing location doesn't matter for CRLs
  - Replay doesn't matter for messages that are meant to be repeated – "Traffic signal will go red at time T"
  - No need to include explicit expiry time for instantaneous messages
- Additionally, the application has to carry out specific permissions tests
  - Does the (PSID, SSP) combination from the cert allow this action?
- $The processing in 1609.2 is not on its own enough to guarantee security – application specifier education needed
  - $"Security profile"

Signed Data



PSID ← is this the expected value?

generation location[1] ← is this too far away?

generation time[1] ← is this too distant in the past? ← is this in the future?

expiry time[1] ← has this time passed?

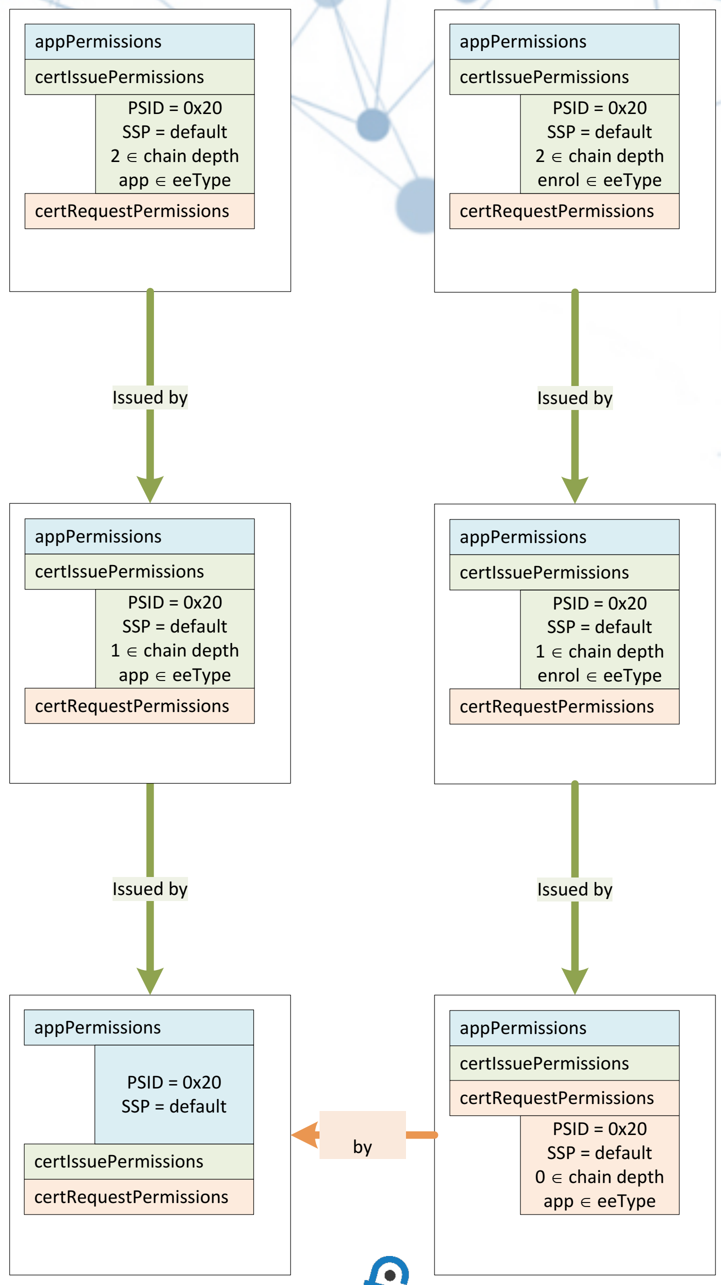
← has this already been received?

NOTES:

1. This data may be contained in the payload or within the security header fields.

Issue and enrol permissions

- Issue lets a CA issue certs with chain depth range contained within the CA’s chain depth range
  - App Permissions counts as depth 0 for these purposes
  - eeType app ==> chain ends in application cert
  - eeType enrol ==> chain ends in enrolment cert, i.e. cert with Request permissions and chain depth = 1
- Request lets you request certs with the same PSID, SSP, chain depth, eeType
- Verbs are better than nouns!



appPermissions
certIssuePermissions
PSID = 0x20
SSP = default
2 ∊ chain depth
app ∊ eeType
certRequestPermissions

Issued by

appPermissions
certIssuePermissions
PSID = 0x20
SSP = default
1 ∊ chain depth
app ∊ eeType
certRequestPermissions

Issued by

appPermissions
PSID = 0x20
SSP = default
certIssuePermissions
certRequestPermissions

by

appPermissions
certIssuePermissions
PSID = 0x20
SSP = default
2 ∊ chain depth
enrol ∊ eeType
certRequestPermissions

Issued by

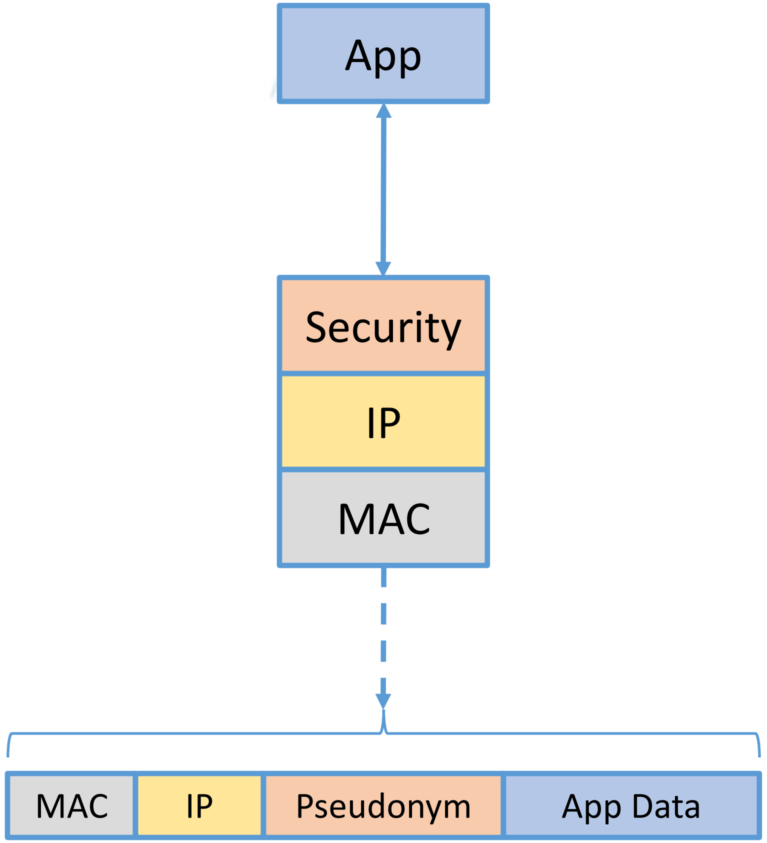
appPermissions
certIssuePermissions
PSID = 0x20
SSP = default
1 ∊ chain depth
enrol ∊ eeType
certRequestPermissions

Issued by

appPermissions
certIssuePermissions
certRequestPermissions
PSID = 0x20
SSP = default
0 ∊ chain depth
app ∊ eeType

Privacy

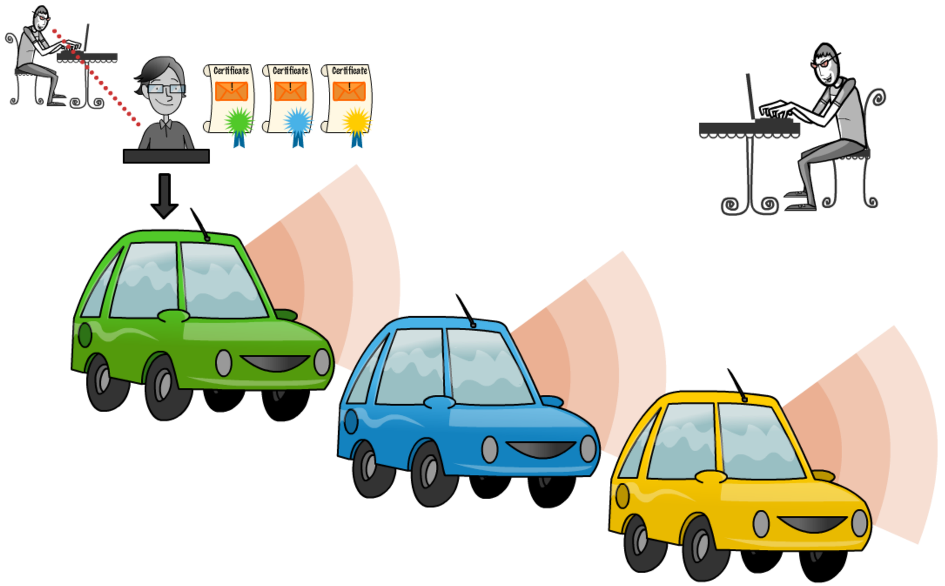
- A listener who records all Basic Safety Messages (BSMs) can track a vehicle
  - By design!
- System design provides privacy protection against a "mid-size" attacker
  - Multiple certificates for an application (20+ per week)
  - Change all identifiers in the stack simultaneously
- Considered group signatures but too large & slow
- Need policy measures to prevent automatic speeding tickets etc



SECURITY INNOVATION

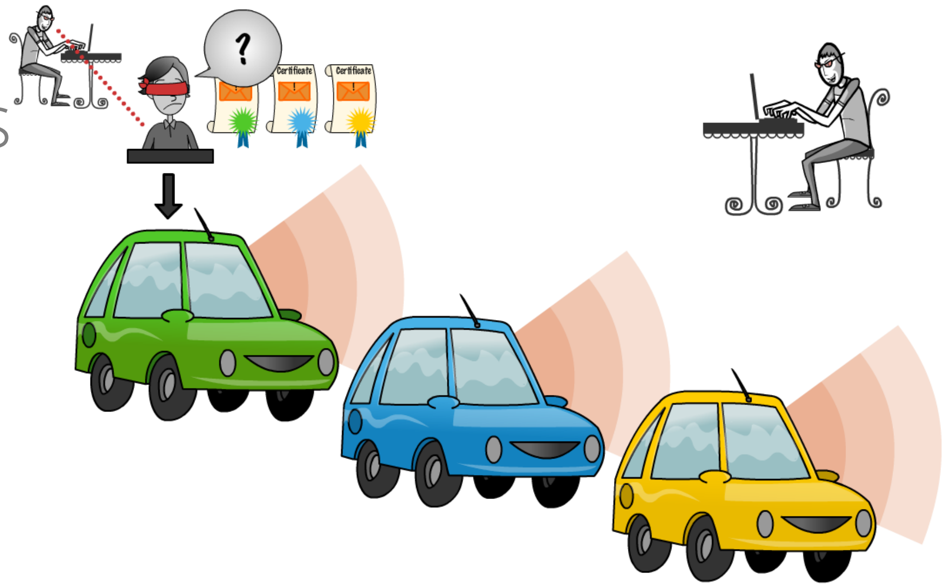
Privacy

- Of course, this means a CA could link if it knows which certificates go to which device

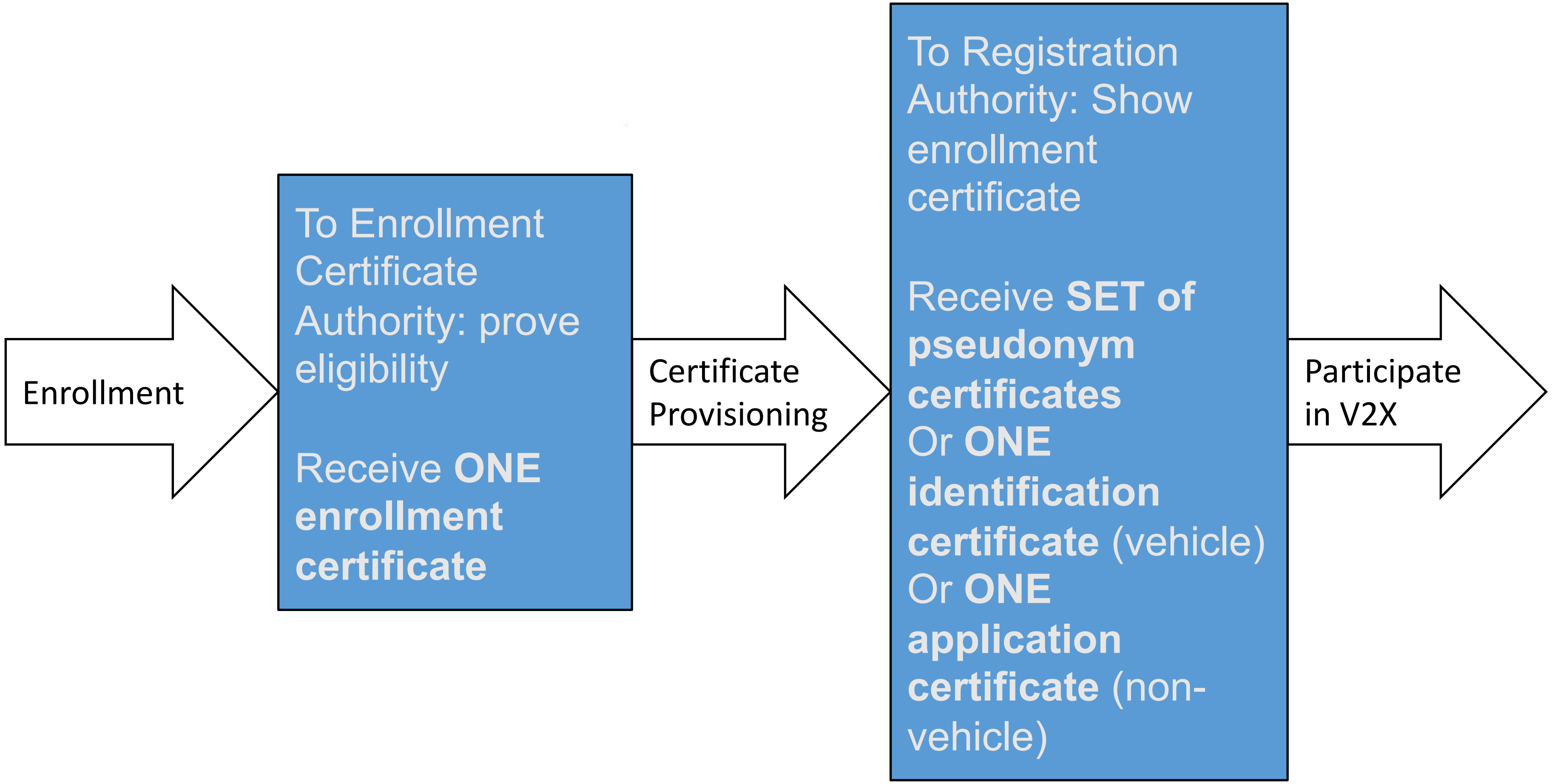


Privacy

- Of course, this means a CA could link if it knows which certificates go to which device
- … so the (US) system “blinds” the CA, preventing insiders as well as outsiders from linking



Certificate lifecycle



Enrollment

To Enrollment Certificate Authority: prove eligibility

Receive **ONE enrollment certificate**

Certificate Provisioning

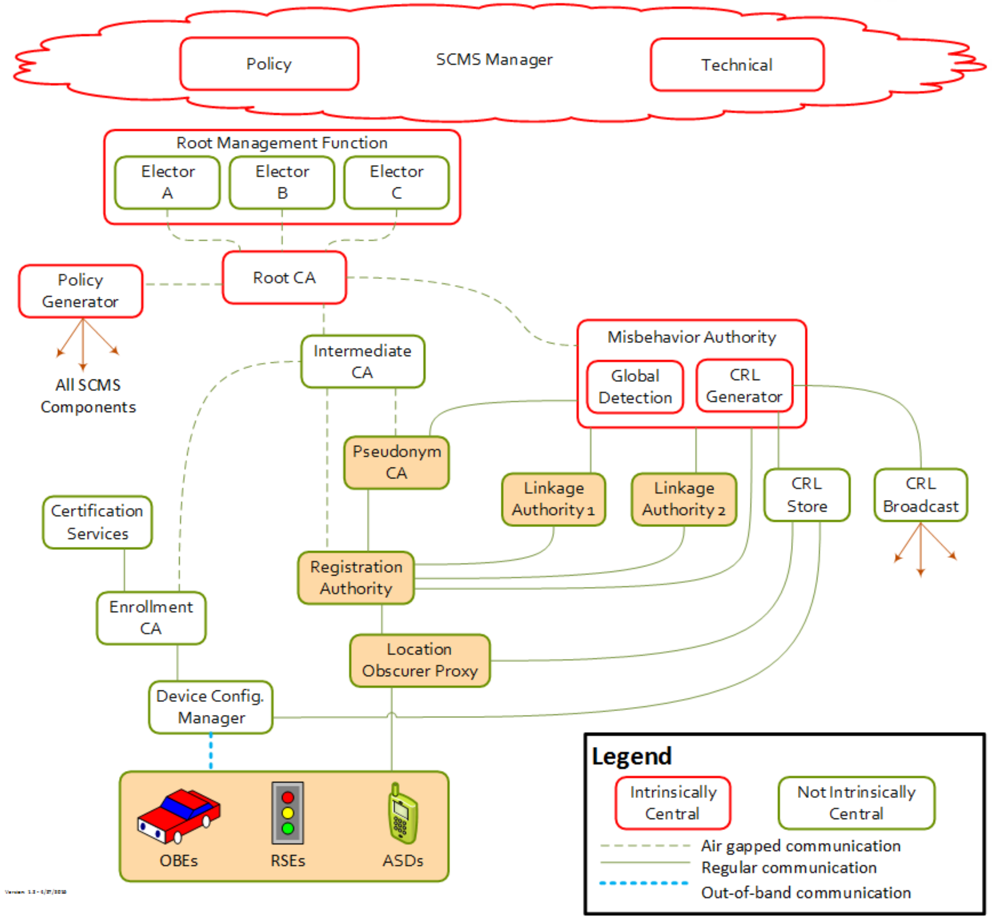
To Registration Authority: Show enrollment certificate

Receive **SET of pseudonym certificates**
Or **ONE identification certificate** (vehicle)
Or **ONE application certificate** (non-vehicle)

Participate in V2X

Certificate issuance

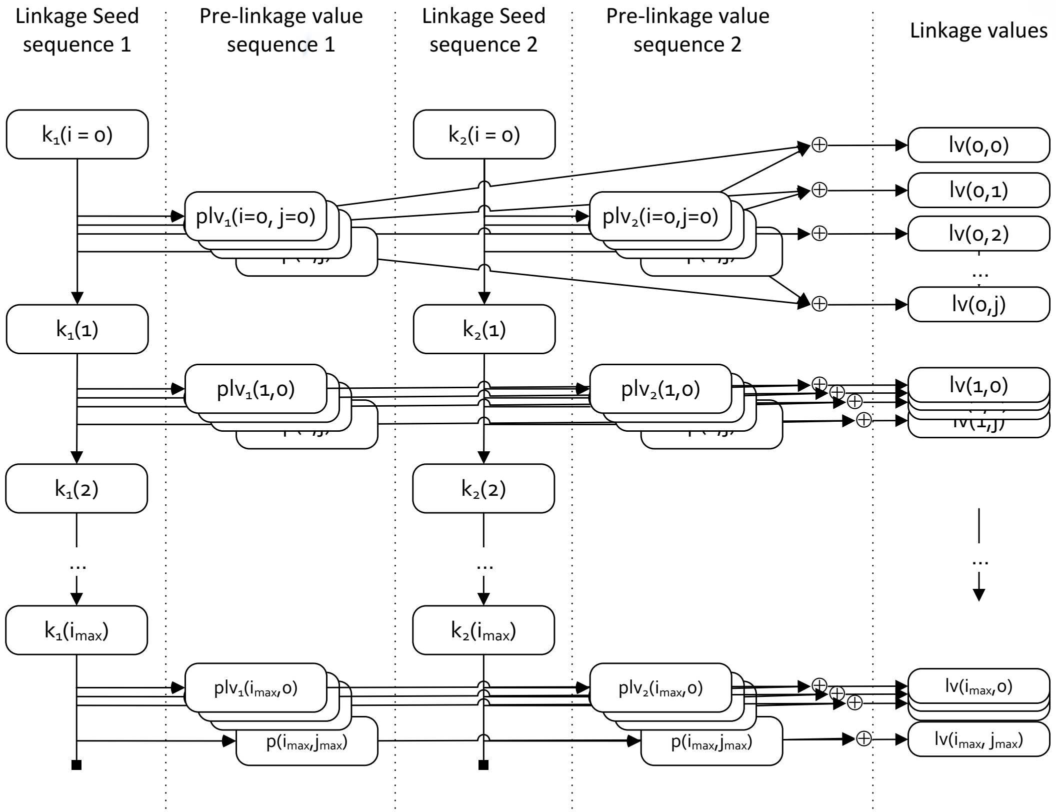
- Secure Credential Management System (SCMS – think PKI-on-steroids) for V2V includes privacy-preserving mechanisms
- Shuffle at RA to protect against CA learning certificates
- Linkage authorities to allow tracing misbehaving devices without revealing their identity, and revoking in a way that only allows them to be tracked after revocation
- Organization separation ensures no single insider / no single database breach can track any car



SECURITY INNOVATION

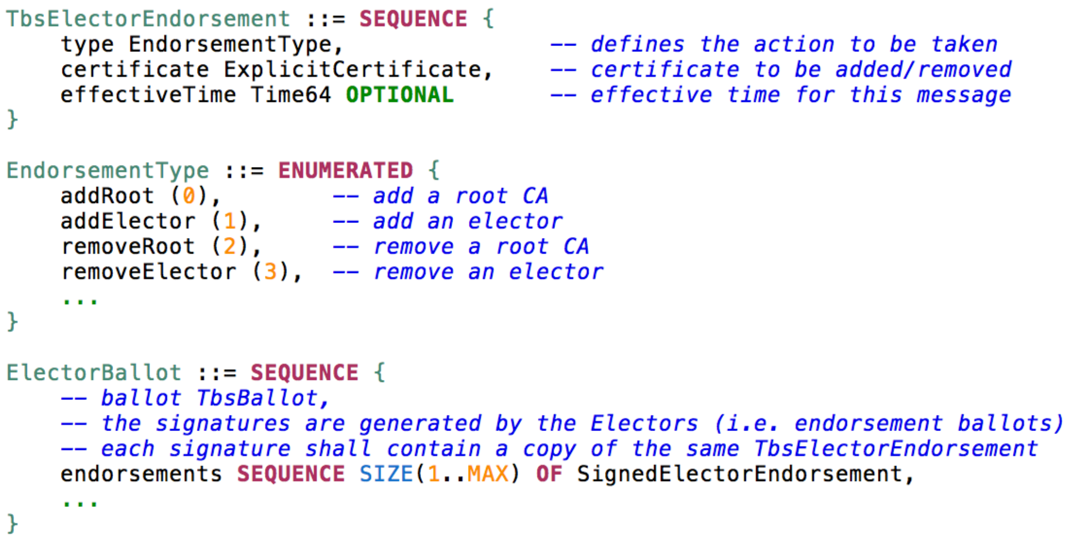
Revocation and misbehavior reporting

- Certificates contain "linkage values"
  - These are generated by XORing together two pre-linkage values from series generated by two LAs
  - Pre-linkage values are generated via a hash from a linkage seed
  - Linkage seed is itself generated via a hash chain
- To determine if two misbehaving messages came from the same vehicle, need only consult one LA
  - Interfaces being defined for this that preserve privacy
- To revoke, reveal both seeds
  - Allows receivers to calculate all linkage values going forward in time, but not backwards



Electors

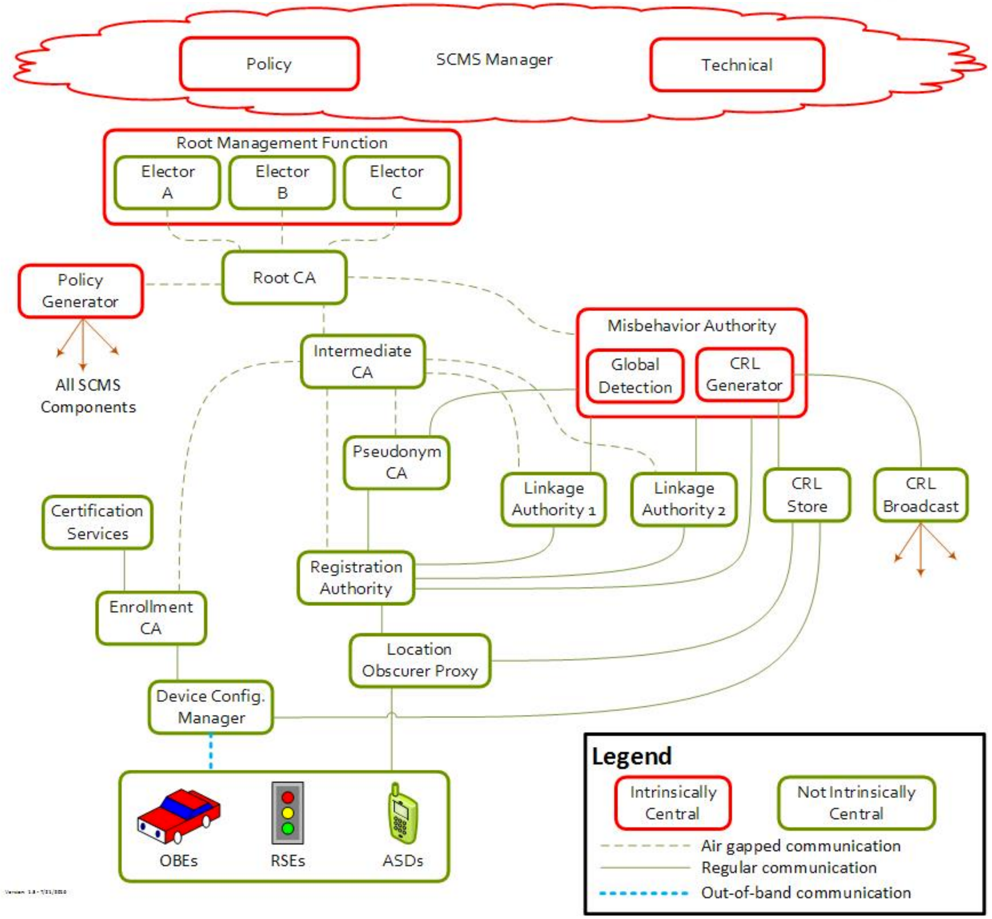
- Electors issue ballots that <VERB> a <NOUN>:
  - VERB: Endorse or revoke
  - NOUN: Root CA or Elector



```
TbsElectorEndorsement ::= SEQUENCE {
    type EndorsementType,                -- defines the action to be taken
    certificate ExplicitCertificate,     -- certificate to be added/removed
    effectiveTime Time64 OPTIONAL        -- effective time for this message
}

EndorsementType ::= ENUMERATED {
    addRoot (0),            -- add a root CA
    addElector (1),         -- add an elector
    removeRoot (2),         -- remove a root CA
    removeElector (3),      -- remove an elector
    ...
}

ElectorBallot ::= SEQUENCE {
    -- ballot TbsBallot,
    -- the signatures are generated by the Electors (i.e. endorsement ballots)
    -- each signature shall contain a copy of the same TbsElectorEndorsement
    endorsements SEQUENCE SIZE(1..MAX) OF SignedElectorEndorsement,
    ...
}
```



Now: how well did we do?

A worked example of John Kelsey's presentation from yesterday

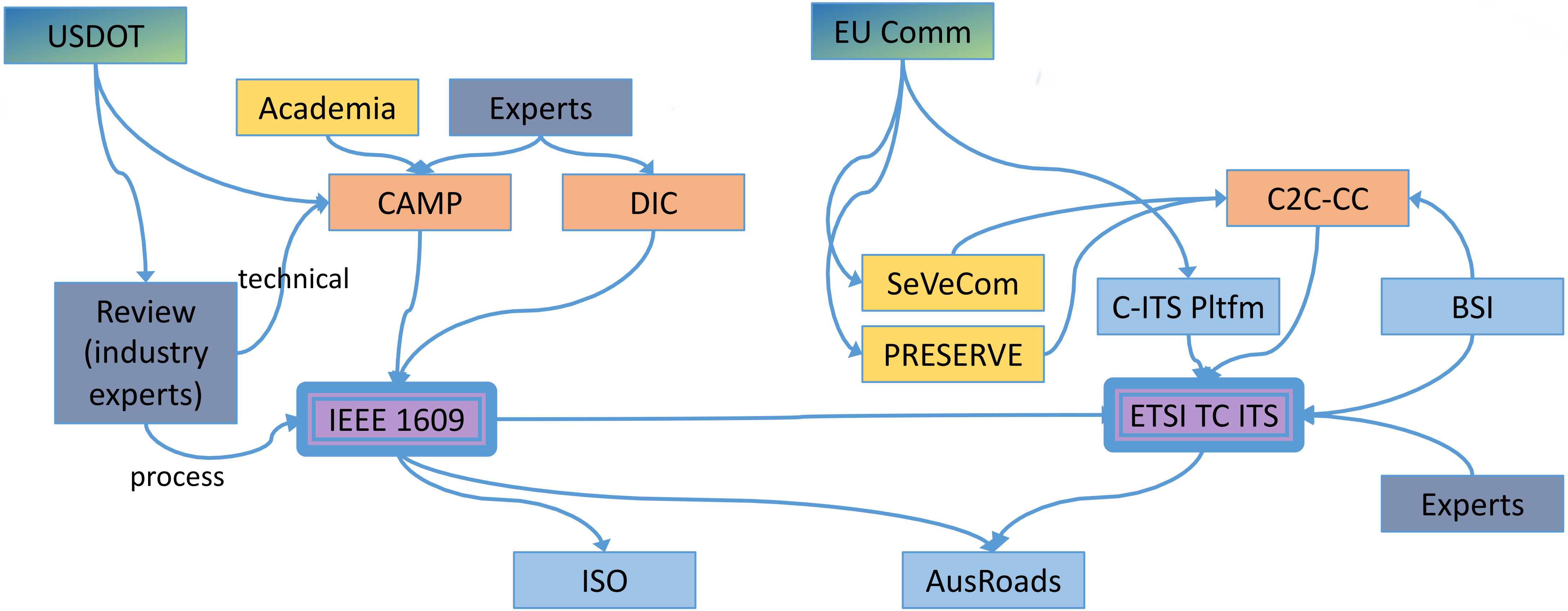
- Risks to standardization
  - Pushback on requirements
  - Doing the minimum required
- Quality of standard
  - Closed standards processes – going to be hard in future to do security standards in the closed model
  - Institutional memory and losing key people
  - Good review?
  - What errors will users of standards make?
  - Public confidence
  - Methods to enable feedback?
  - Transparency – make reasoning as open as possible

Dynamics of standardization decisions

- Speed versus participation
- Control versus legitimacy
- … often framed as ”consortium v standardization”

- Different priorities of secondary requirements, especially intangible ones
  - “Flexibility” / “Future-proofness” / “Complexity” / “Human-readability” / “Simplicity” / “Machine-readability” / “Availability of tools” / “Stability”
  - May not be an objective way to rank these requirements

Stakeholders in our case



SECURITY INNOVATION

Openness / transparency

- 1609 Mailing lists and meetings open to public
  - But not so focused on security
- C2C-CC WG Sec – pay to play
- ETSI ITS WG5 – pay to play
- CAMP – invite only and publication restrictions
  - SCMS design paper was presented in December 2013 after key concepts were developed in March 2011

- SeVeCom, Preserve, etc, published a large number of papers
  - But in general these were at vehicular networking conferences, not crypto conferences
  - Not clear review came from the proper community

Contributions from experts

- $Eric Rescorla, Russ Housley – initial draft
  - $Input from John Kelsey
- $WW (IEEE 1363 chair)
- $Andre Weimerskirch + others from Escrypt (academic family tree: Christof Paar)
- $Rene Struik, voluntary contributor
- $Briefly, review from Scott Vanstone, David Kravitz (when at Certicom) and others from Certicom team
- $More in-depth review from Trustpoint Innovation (Rob Lambert, ex Certicom), Green Hills Software (Bill Lattin, also partly ex Certicom)
- $Very brief review from Alfred Menezes

- $Adrian Perrig and team from Carnegie Mellon
- $Yi-Chun Huh and team from UIUC
- $Frank Kargl
- $Panos Papadmitratos
- $Jonathan Petit
- $Ongoing review via congressional subcommittee

- $Implementers from AutoTalks, Escrypt, Kapsch, Security Innovation, Southwest Research Institute
  - $Mainly off-list

Level of confidence – what has been reviewed?

- CAMP/Escrypt: 2 year project to establish that ECDSA had acceptable performance in automotive setting
  - $Rejected fancier constructions like TESLA
- $1-year discussion for WG to agree that ASN.1 was the right presentation language even though original 1609.2 was in TLS-like syntax
  - $Included detailed performance analysis
- 1-year discussion for WG to agree to include implicit certificates

- Reviewed well
  - Butterfly keys
  - Linkage values
  - Implicit certificates
  - Permissions model for messages
- $Not reviewed so well
  - $Hash for signing
  - $Canonicalization

- $No use of analysis tools and requirements are complicated

How well did we do in mutually educating the user community?

- Security -> Apps:
  - App specifiers have taken on board the job of defining an SSP
  - Generation time in many common messages rolls over after a minute
    - Trivial replay attack
    - Countermeasure: include much longer rollover time in security header
    - Better countermeasure: only include one generation time and make it robust
- Apps -> Security:
  - Generation location in message can be checked against validity region in cert…
    - … Or not checked at all, e.g. for CRLs…
  - … But some messages apply to multiple locations, which may be points or areas
    - Signal Phase and Timing message can give info about multiple signals at once
  - API in 1609.2 doesn't allow passing a set of locations down to security services or geographic region from cert up.
    - Noted during prep for New York City deployment
- Conclusion: two-way outreach to user community needs to be stronger

Specific regrets and concerns

Regret: ECDSA and NISTp256 / Brainpool rather than ed25519

- $Provenance of NISTp256 is unclear, selecting it is rewarding bad behavior
  - $Also, any reason to prefer it to Brainpool r256 is a reason to prefer ed25519 to it and vice versa
- $ECDSA is suboptimally efficient (inverses), and malleable!
  - $If (r, s) signs m, so does (r, -s)
- $Ed25519 allows:
  - $Batch signature verification
  - $Faster verifications in software on single signatures
- $Brainpool r256 may (it's unclear to me) be faster at private key operations in blinded hardware
  - $… but you only sign 10 times a second, you verify many more times than that; ed25519 may let you omit verification acceleration altogether with little change in signing hardware cost

- $Decision was to go with ECDSA and NIST p256 because:
  - $Hardware exists
  - $It's the US standard
  - $Carmakers didn't want to be the first significant industry body to select ed25519
- $Subsequently BSI stated requirement for Brainpool r256 for traffic controllers etc and Brainpool r384 for root CA
  - $If you can create facts on the ground you can put them in the standard

- $Impact is minor but regrettable

Insufficiently reviewed?: hash for signing

- Signed data signatures include the signing cert as well as the signed data
  - Prevents cert misbinding attacks
  - Also prevents multi-key attacks on ECDSA
- Specific implementation
  - H = Hash (Hash (ToBeSignedData) | | Hash (Cert))
    - Or Hash (Hash (ToBeSignedData) | | Hash (“”)) for self-signed messages
- Is this okay?

Insufficiently reviewed?: Canonicalization

- $To allow senders to choose between compressed points (for size) and uncompressed points (for speed), signatures are generated on the **canonicalized** form of PDUs
  - $… i.e. whether points in a PDU are *transmitted* compressed or uncompressed, they're *input to the hash* in compressed form.
- $Typically hash is carried out on encoded text, not on "semantic" text
  - $… Though see XmlDigSig
  - $C2C/ETSI decided this isn't functionality worth supporting

- $There are a number of cases where signed objects are hashed
  - $In particular for replay detection
- $But ECDSA signatures are malleable!
  - $If (r, s) signs m, so does (r, -s)
- $Canonicalize before hashing by setting s to the smaller of the allowed values
- $C2C/ETSI decided just to hash the encoded data, don't directly address replay

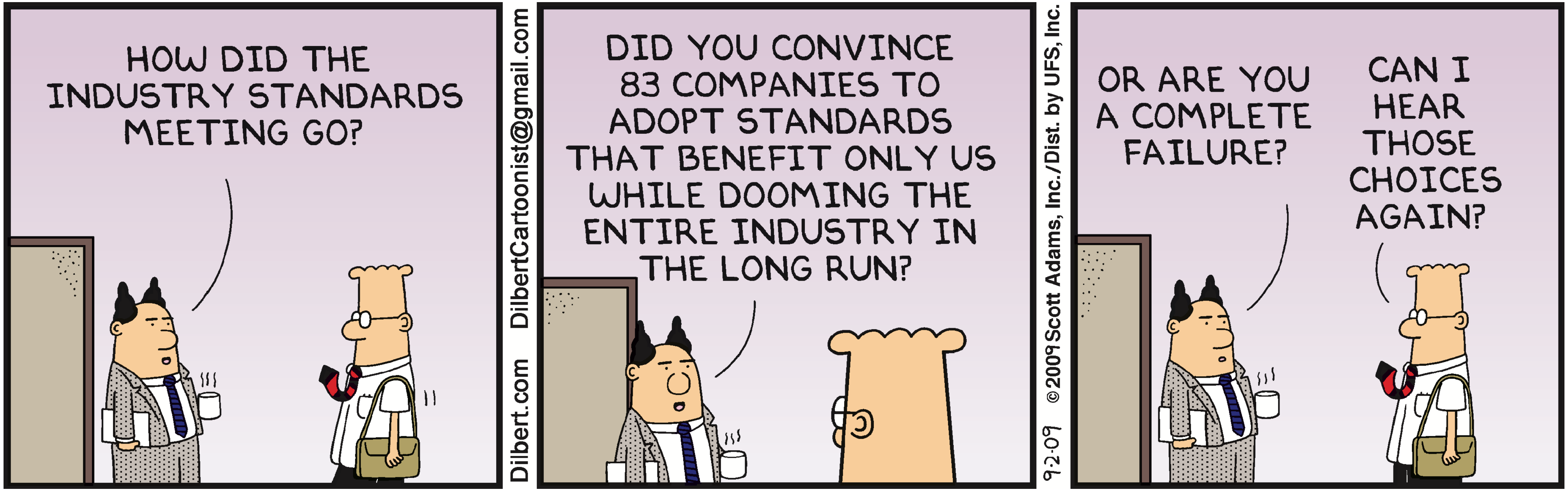
How did these decisions get made?

- $Initial observation
- $Internal discussions within Security Innovation
- $Issue brought to CAMP with proposed solution for sign-off
- $Issue brought to 1609 Working Group
- $Resolution included in 1609.2 working draft

- $Common thread: one person with a strong opinion walking into a room of people without strong opinions
- $Tried to mitigate this risk by:
  - $Clearly spelling out alternatives when presenting to CAMP and to 1609
  - $Running CAMP discussion over a period of weeks, by mail and face-to-face
  - $Revisiting question in two 1609 meetings in a row
- $These are still unorthodox choices and perhaps not right
  - $You can run a good-faith process and not get the right outcome

Standards divergence

Different players have different goals


HOW DID THE INDUSTRY STANDARDS MEETING GO?
Dilbert.com DilbertCartoonist@gmail.com
DID YOU CONVINCE 83 COMPANIES TO ADOPT STANDARDS THAT BENEFIT ONLY US WHILE DOOMING THE ENTIRE INDUSTRY IN THE LONG RUN?
9-2-09 © 2009 Scott Adams, Inc./Dist. by UFS, Inc.
OR ARE YOU A COMPLETE FAILURE?
CAN I HEAR THOSE CHOICES AGAIN?

- DILBERT © 2009 Scott Adams. Used By permission of UNIVERSAL UCLICK. All rights reserved.

Differences between EU and US approach

- $Mandated system (for automakers to build in) v opt-in system initially
    - $Privacy concerns are even greater – driven largely by OEMs
        - $People who are concerned about privacy opt out of the system by opting out of buying new cars – OEMs don't want that!
    - $Much larger population of day-1 devices than in the European setting
        - $More price-sensitive, affects ability to provide
            - $Crypto hardware
            - $Connectivity
        - $More difficult to make a change from the day-1 system
            - $Design may seem more complex, but we need to get it right first time.
- $Infrequent connectivity changes approach to removing bad actors
    - $Frequent certificate reissue (EU) v revocation (US)
- $Focus on single-hop safety (US) v broader environmental awareness (EU)
    - $EU uses "geonetworking" to support multi-hop, changes the security model somewhat

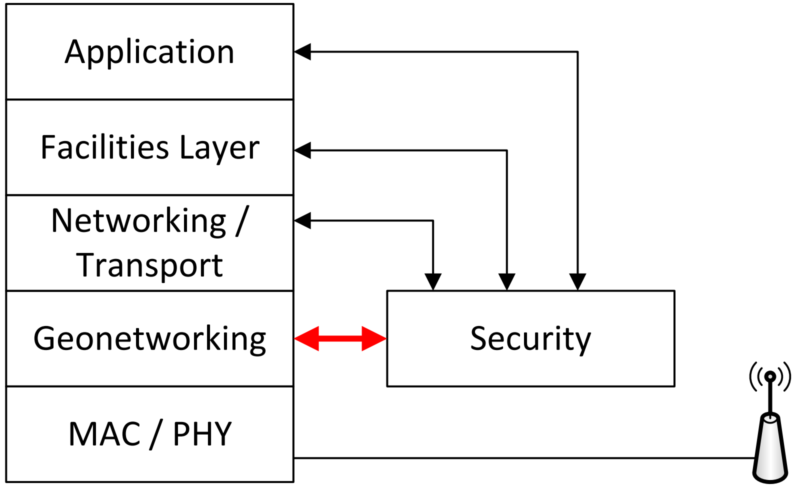
SECURITY INNOVATION

Geonetworking within VANET

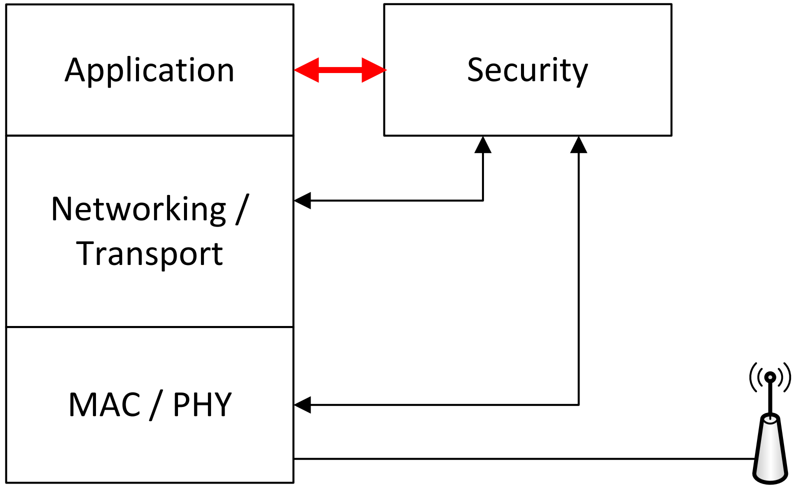
- ETSI model
  - All packets sent over geonetworking are signed at the geonetworking layer
  - Indicates that the sender has permissions to ask that a packet is forwarded
  - Packets are verified before forwarding
  - Prevents unauthorized requests for forwarding, reduces congestion
- Packet size optimization: application messages signed at the geonetworking layer do not need to also be signed at the application layer
  - So long as they are not forwarded over a different medium

Architectural comparison: OBE

EU



US



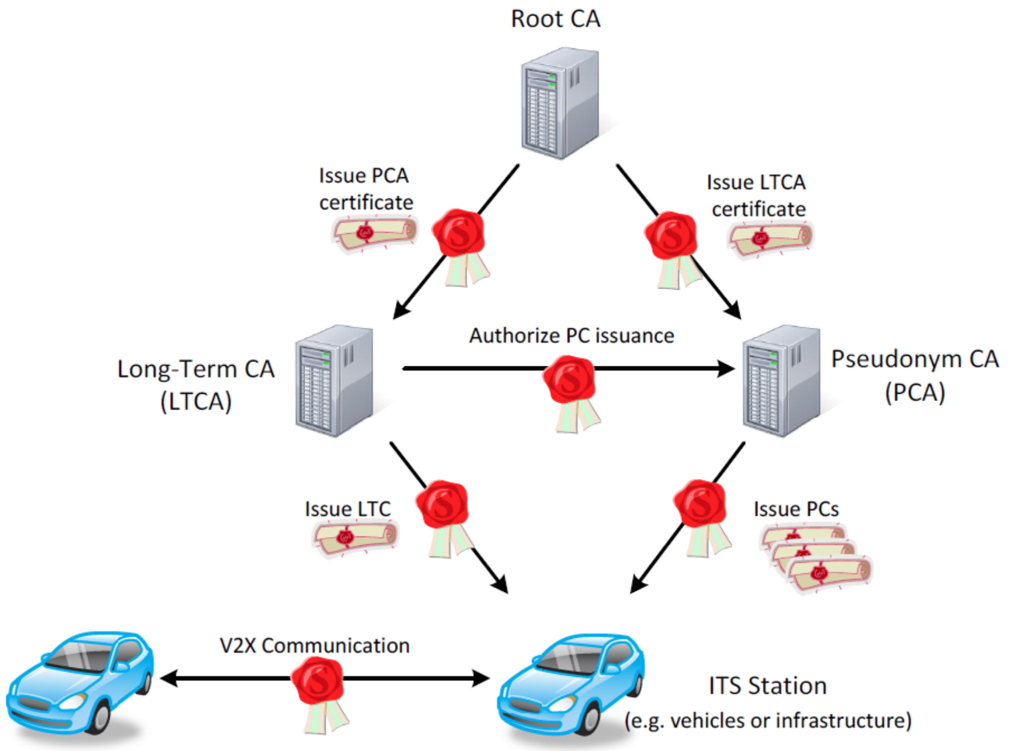
🟥 ↔ Crypto operations

↔ Other operations

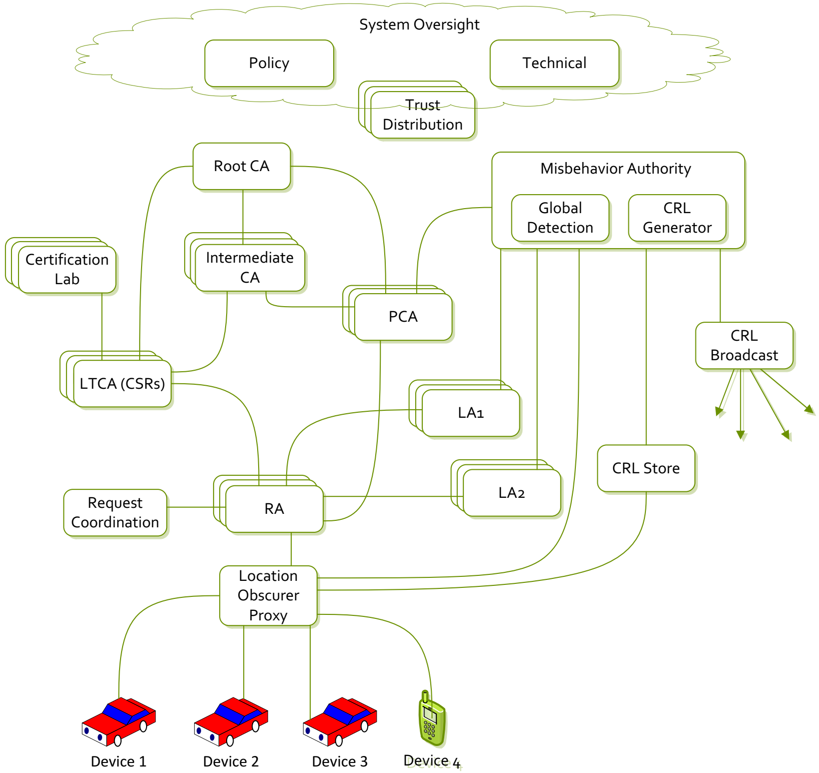
SECURITY INNOVATION

Architectural comparison: PKI

EU



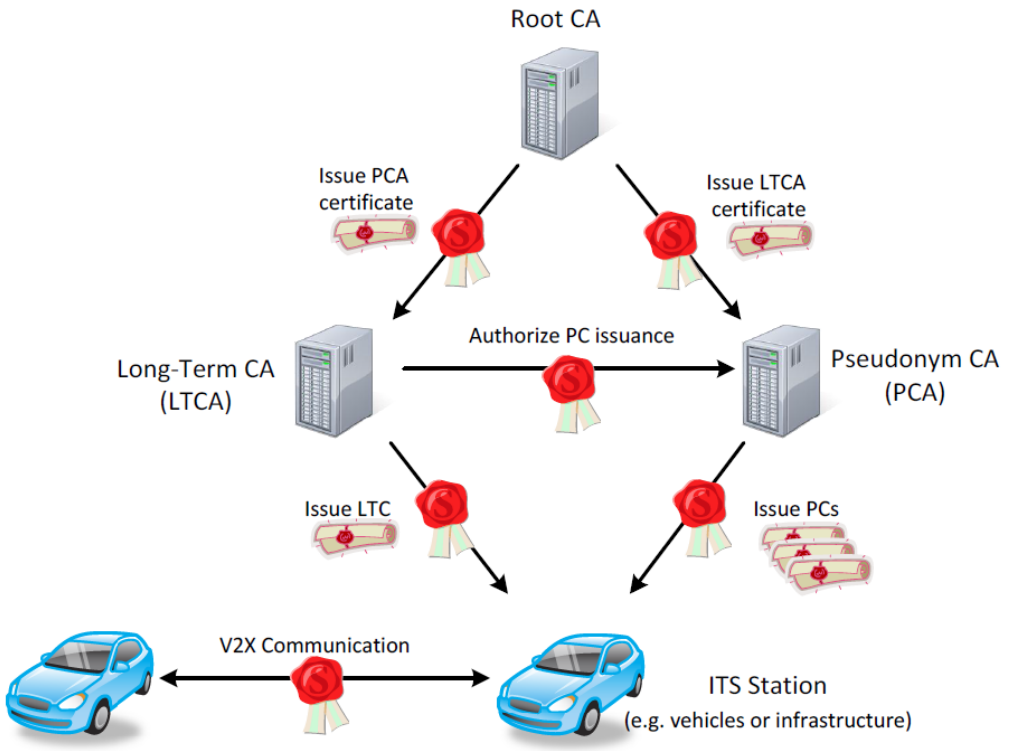
US



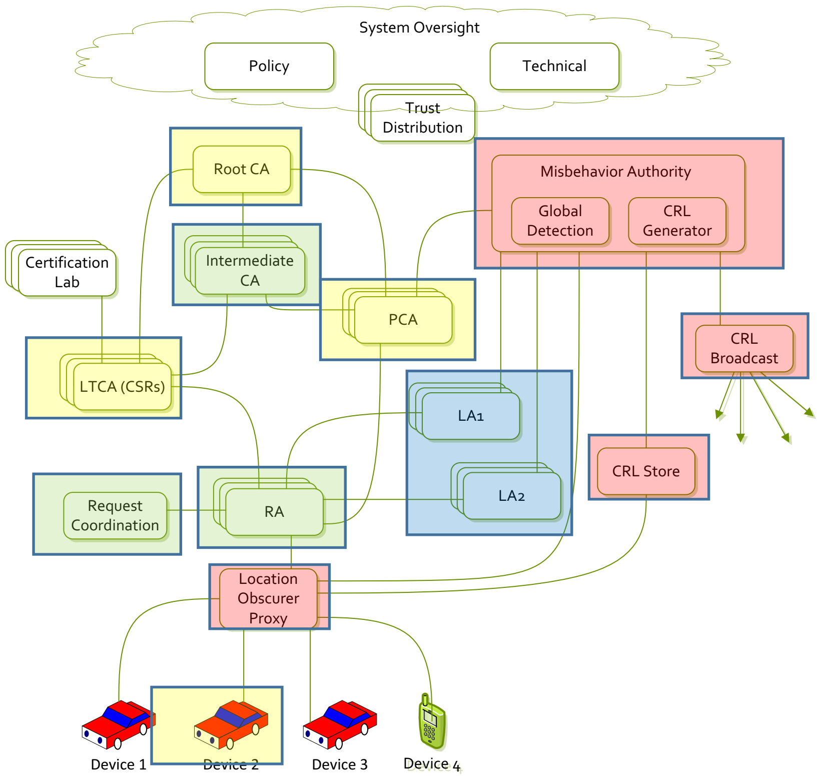
SECURITY INNOVATION

Architectural comparison: PKI

EU



US

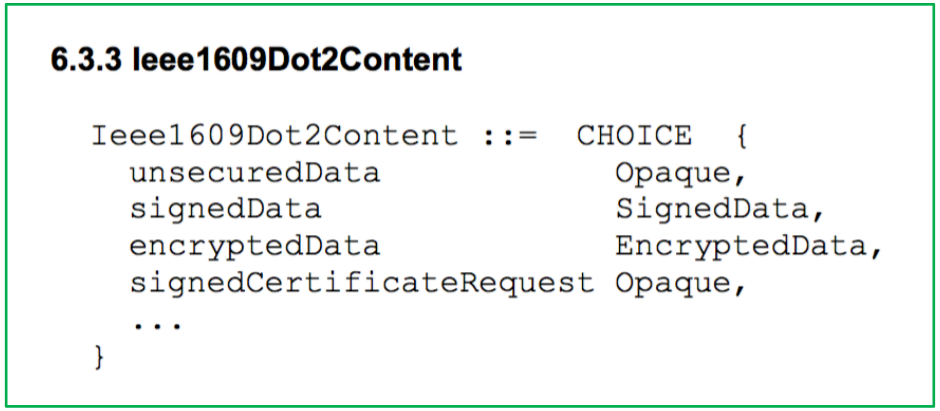


Until 2011 ETSI and C2C-CC's plan was to profile 1609.2. Then…

- $6/11: 1609.2 sponsor ballot begins
- $6/11: C2C-CC releases Memorandum of Understanding committing each OEM to having one model on the road by 2016 with V2X technology
- $3/12: liaison statement, ETSI->1609, requesting two new features
- $5/12: 1609 response: We intend to do this but we're nearly finished, so we'll put it in the next version.
- $5/12, C2C-CC response: We'll draft this ourselves as an ETSI TS. "We only want to use this also to show something from the EU side, and possibly withdraw the TS later on, when all extensions are integrated in 1609.2."
- $7/12, C2C-CC: "As you know, we have some concerns about the current complexity of the security formats in 1609.2, and would like to make this more flexible."
- $11/12: C2C-CC: Standardization coordination has taken too long.
- $12/12: final recirculation ballot of 1609.2 in IEEE
- $2/13: Attempt to harmonize, unsuccessful

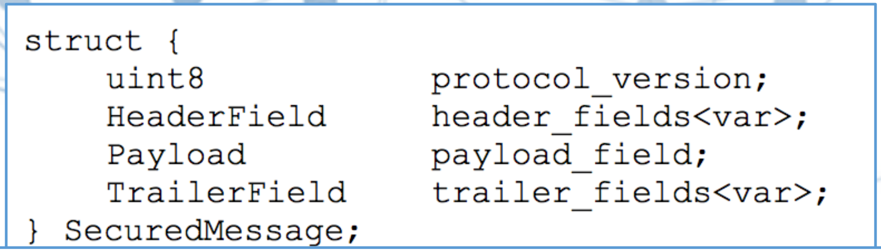
1609.2 v 103 097

- 1609.2 distinguishes between signed and encrypted data structures, 103 097 doesn't.
- A lot of consistency logic needs to go in the English text of 103 097, 1609.2 is compilable
- Deal-breaker!

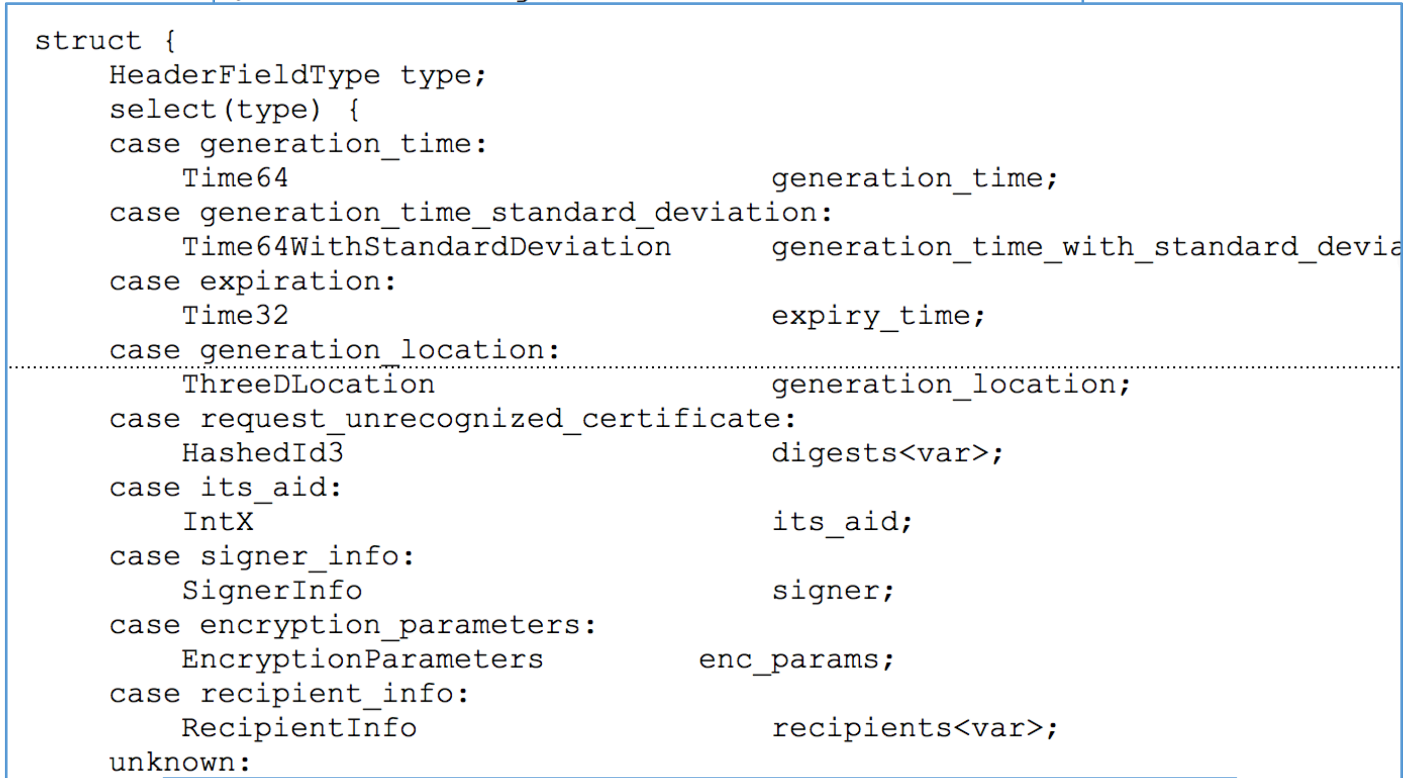


6.3.3 Ieee1609Dot2Content

```
Ieee1609Dot2Content ::=  CHOICE  {
  unsecuredData             Opaque,
  signedData                SignedData,
  encryptedData             EncryptedData,
  signedCertificateRequest  Opaque,
  ...
}
```

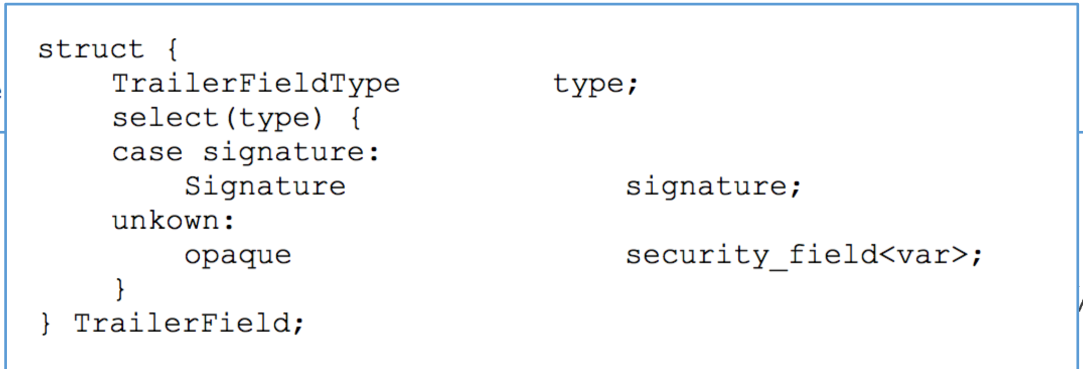


```
struct {
    uint8           protocol_version;
    HeaderField     header_fields<var>;
    Payload         payload_field;
    TrailerField    trailer_fields<var>;
} SecuredMessage;
```



```
struct {
    HeaderFieldType type;
    select(type) {
    case generation_time:
        Time64                            generation_time;
    case generation_time_standard_deviation:
        Time64WithStandardDeviation       generation_time_with_standard_devia
    case expiration:
        Time32                            expiry_time;
    case generation_location:
        ThreeDLocation                    generation_location;
    case request_unrecognized_certificate:
        HashedId3                         digests<var>;
    case its_aid:
        IntX                              its_aid;
    case signer_info:
        SignerInfo                        signer;
    case encryption_parameters:
        EncryptionParameters          enc_params;
    case recipient_info:
        RecipientInfo                     recipients<var>;
    unknown:

    }
} Heade
```



```
struct {
    TrailerFieldType        type;
    select(type) {
    case signature:
        Signature               signature;
    unkown:
        opaque                  security_field<var>;
    }
} TrailerField;
```

ATION

Requirements -> design

- 1609.2:
  - Flexibility: Canonical encoding
  - Rigidity: order of fields within message, distinction between signed and encrypted data structures
- 103 097:
  - Rigidity: Hash the encoding you saw
  - Flexibility: order of fields within message, no distinction between signed and encrypted data structures
- In a lot of cases you suspect the requirements were retrofitted to the design that happened to be there

Until 2011 ETSI and C2C-CC's plan was to profile 1609.2. Then…

- $6/11: 1609.2 sponsor ballot begins
- $6/11: C2C-CC releases Memorandum of Understanding committing each OEM to having one model on the road by 2016 with V2X technology
- $3/12: liaison statement, ETSI->1609, requesting two new features
- $5/12: 1609 response: We intend to do this but we're nearly finished, so we'll put it in the next version.
- $5/12, C2C-CC response: We'll draft this ourselves as an ETSI TS. "We only want to use this also to show something from the EU side, and possibly withdraw the TS later on, when all extensions are integrated in 1609.2."
- $7/12, C2C-CC: "As you know, we have some concerns about the current complexityof the security formats in 1609.2, and would like to make this more flexible."
- $11/12: C2C-CC: Standardization coordination has taken too long.
- $12/12: final recirculation ballot of 1609.2 in IEEE
- $2/13: Attempt to harmonize, unsuccessful
- $12/2016: no cars

Goal of a single standard

- Need to be seen as trusted and responsive by all
    - Prioritize all requirements
- "I agree with your requirements but is this the right mechanism?"
- Different deadlines can cause divergence as much as different requirements can
    - CAMP wasn't willing to commit to C2C design in the time that C2C had
- Facts on the ground may win
- Hard to make changes when release is "imminent"
    - Resistance to change unless practical attack
    - … if it affects hardware

On the other side of “release is imminent”

Twice!

Ed25519 selection by CFRG

- $Ed25519 takes public key as input into hash
- $1609.2 takes certificate as input into hash
- $WW -> CFRG: propose that signature algorithm takes “public key identifier field” that SHOULD be public key itself but MAY be certificate
  - Note: although in general you have the public key if you have the certificate, 1609.2 uses “implicit certs” where there’s a processing cost to recover the public key
- $Stephen Farrell: If CFRG start down that road and fail to define a scheme without any certificate handling first, (and *very* soon!) my confident prediction is that the signature scheme resulting will be overtaken by events before it is finished. I currently have about 5 internet-drafts that are looking to use Ed25519 and where I'm asking folks to hold off until this work is done. I reckon that's what'll end up being used if this process takes more than a handful of weeks longer. Your proposal falls squarely into the kind of late change that would guarantee that kind of failure.
- $CFRG successfully decided on Ed25519

QSH TLS

- (Not a CV project)
- July 2015, WW proposes to TLS WG that they support quantum-safe ”mix-in” for TLS handshake
- No pushback in principle but general feeling that the WG is nearly done with TLS 1.3 and doesn’t want to defocus
- Late 2015, TLS 1.3 handshake is significantly refactored
- Late 2016, TLS Last Call starts

Group dynamics

- Standards have in-groups and out-groups and out-groups find it hard to get listened to
  - Assumption is that even if their proposals have technical merit they're lower priority
- This creates a tension that makes it hard to
  - (a) produce a single standard
  - (b) with the right level of scrutiny

Conclusion

What’s worked, what hasn’t?

- Lots of resources out there
  - Academic research
  - CFRG mailing list
- Open discussion helps
  - Clear analysis of problems
  - Thorough and creative initial enumeration of possible solutions
  - Open discussion to allow modification
- The problem of getting sufficient review for a niche standard, even a high-impact one, is real
  - 1609.2 is still modifiable, though only just: please have a look!

Thank you!

- Questions?
- wwhyte@securityinnovation.com