Secure Vehicle Communication

Identification in VANETs

Frank Kargl
frank.kargl@uni-ulm.de

Media Informatics Dep.
Ulm University

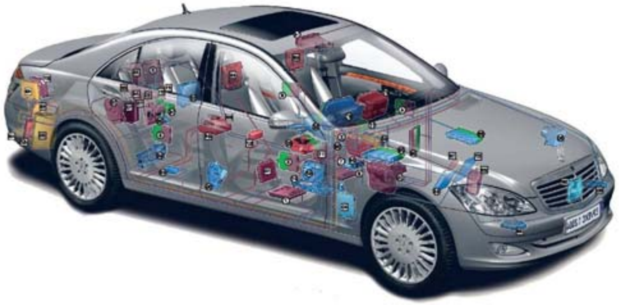
Budapest Meeting, 4./5. Sept. 2006

Definition: Authentication

- “... the process by which a computer, computer program, or another user attempts to confirm that the computer, computer program, or user from whom the second party received some communication is, or is not, the claimed first party.” [wikipedia]
- So what should we authenticate in a VANET?

Driver

Car



Car Component

Definitions

- Definition Identity:
  “The ***identity*** of an object is a ***unique*** property that is ***tied irreversibly to this single object***. The identity is ***unchangeable*** throughout the lifetime of this object and ***cannot be transferred*** to other objects”.
- Definition Identifier:
  “An ***identifier*** is a property (or group of properties) which is suited to identify an object, i.e. doubtlessly determine its identity according to the properties defined above.”
- Real World Example: Car Serial number
  - Unique: together with brand
  - Irreversibly tied to car: embossed into car frame
  - Unchangeable: tampering will be detected
  - Cannot be transferred: tampering will be detected

Different Forms of Authentication

- Identity-Authentication
- Property-Authentication
  - "Verifying some property of an entity"
  - Type of node, max. speed, dimensions, …
- Geo-Authentication
  - "Verifying the position of an entity"
  - Moving objects: changing property
  - Stationary objects: static property

- First focus: Identity-Authentication

How to choose VANET identifiers?

- What objects should be identified?
- How to guarantee unique IDs?
- How is the creation of new or change/transfer of existing IDs prevented?
- How are identifiers and addresses linked to each other?

What objects should be identified?

- Drivers? Cars? Components/Interfaces?
  - Users:
    Multiple drivers per car? Car stolen or sold?
    Exclusion of all cars from one user?
  - Interfaces: Multiple identities per car?

➔ One identifier per vehicle

How to guarantee unique IDs?

- Unique hardware property?
  - Needs tamper-resistant/-proof hardware
  - Hard to communicate correctness to other nodes
- Self-generated generic UUIDs?
  - Statistically unique
  - Can be cloned easily by attackers ☹
  - Nodes can create them in arbitrary number ☹
- Asymmetric Cryptographic keys?
  - Statistically unique
  - Nodes can create them in arbitrary number ☹
  - Cannot be cloned easily by attackers ☺
  - Can be transferred from one node to another ☹
  - Eliminates the need for binding identifiers to keys
    ➔ no need for classical PKI ☺

Solution?

➔ Use public keys as identifiers

Preventing creation/change/transfer of IDs

- Need for a Trusted Third Party (TTP)
- Certifying authority (CA) issuing certificates
  - For what precisely do we need the CA?
    - Uniqueness of nodes
    - Revocation possibility in case e.g. of hardware tampering
  - CA-based approaches
    - Centralized CAs need online connection ☹
    - Hierarchical CAs need online connection ☹
    - Distributed CAs
      - Joined creation of signatures
      - Complex & resource consuming ☹
      - Bootstrapping problems ☹
      - Lot of unanswered questions in literature
  - 'Web of Trust' based approaches
    - "Distributed CA" where each node can generate signatures individually
    - Bootstrapping problems ☹
    - Users involved: To sign or not to sign? ☹
    - Existing work mixes up concept of validity and trust ☹

VANET CAs

➔ Build a centralized/hierarchical CA

- try to make it operate as economically as possible
- prevent need for communication during VANET operation
- Only certify validity of a PK – no need to link ID and PK!
- Single/Hierarchical/Distributed?
  - E.g. manufacturer or gov. CAs,
    flat hierarchy + cross-certification

- When will the CA certify what?
  - Public key at production time
  - Immutable Properties (in a second certificate)?
  - Identifier of the car? Yes, Serial-#, License Plate? No
  - Identifier of the user?
    - Multiple users per car? Separate cert. for drivers
    - Problems when directly linking car and driver?
- What cryptosystem to use?
  - ECDSA: signature verification very costly
  - Other options?
- Who will be funding/operating the CA?
  - Not our business?
  - Minimize effort and therefore cost of operation

Certificate Revocation

- Number of revocations?
  - Number of expected certificates in total
    - Germany: 60 Mio. cars
  - Number of expected certificates issued per year
    - Keep low: 1 cert. per car per lifetime minimum
    - Needs very strong crypto to protect secret key this long!
    - Problem with revocation
    - → long-lived certs and keys (at least 1 year)
    - → automatic key-/cert-renewal
  - Number of expected revocations issued per year
    - Depends on conditions for key revocation
    - Might be low: 10.000 per year?
- How to revoke certificates?
  - Classical CRLs? Bad idea, long lists
  - Optimized CRLs? Compressed/Incremental CRLs
  - Short certificate lifetimes? Many renewals generate overhead
  - Verifiers → see HICSS-39 paper
  - Other approaches?

MANET-IDs

- MANET-ID for node X
  - Asymmetric keypair (PK_X,SK_X)
  - PK_X is used as identifier
  - Crypto-based address $CBA_X = h_{64}(PK_X)$ used for routing
- TTP/CA generates certificate to make MANET-ID valid
  - cert = $E_{SKCA}(PK_X$, serial, Y, valid_until)
  - Certificate proves validity –
    no binding of public key to name or other identifier!
    ⇒ TTP generates signatures without identity checks, etc.
  - Certificate revocation may be used for disabling malicious MANET nodes permanently or until manual check

See F. Kargl, S.Schlott, M. Weber: Identification in Ad hoc Networks, HICSS-39, January 2006

MANET CRS

- Based on Certificate Revocation System by Micali
- Nodes prove the validity of their certificate
- Hashchain: CA chooses random start value Y0

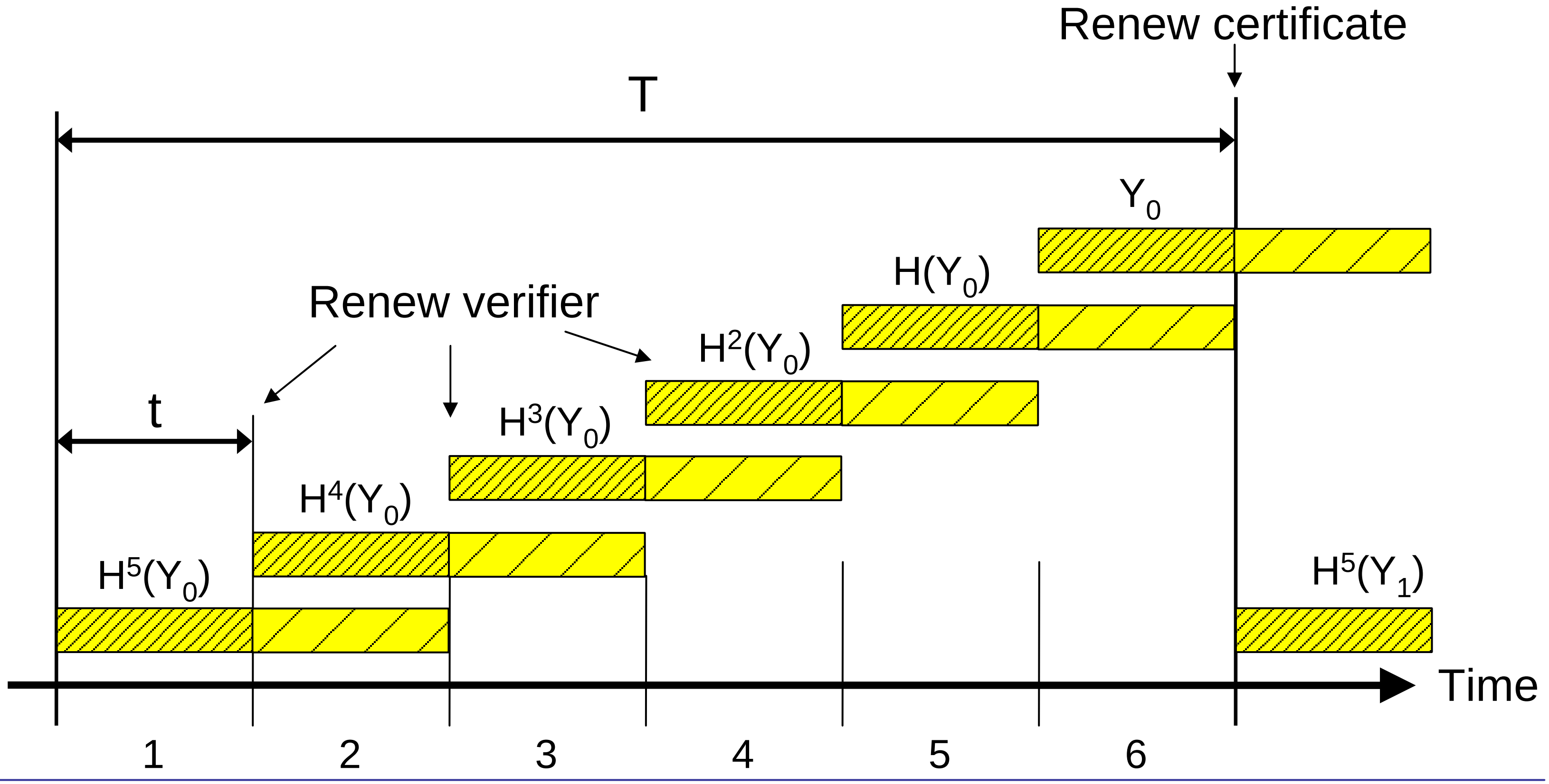
$$\begin{tikzcd}
H(Y_0), \arrow[r, bend left] & H(H(Y_0)), \arrow[l, bend left] \arrow[r, bend left] & H(H(H(Y_0))), \arrow[l, bend left] & \ldots, \arrow[r, bend left] & H^n(Y_0)=Y \arrow[l, bend left]
\end{tikzcd}$$

- Validity period of certificate T partitioned in n sections
- When establishing communication, nodes need to provide $V_i = H^{n-i}(Y_0)$ („Verifier“) to prove the validity of their certificate in the current section:

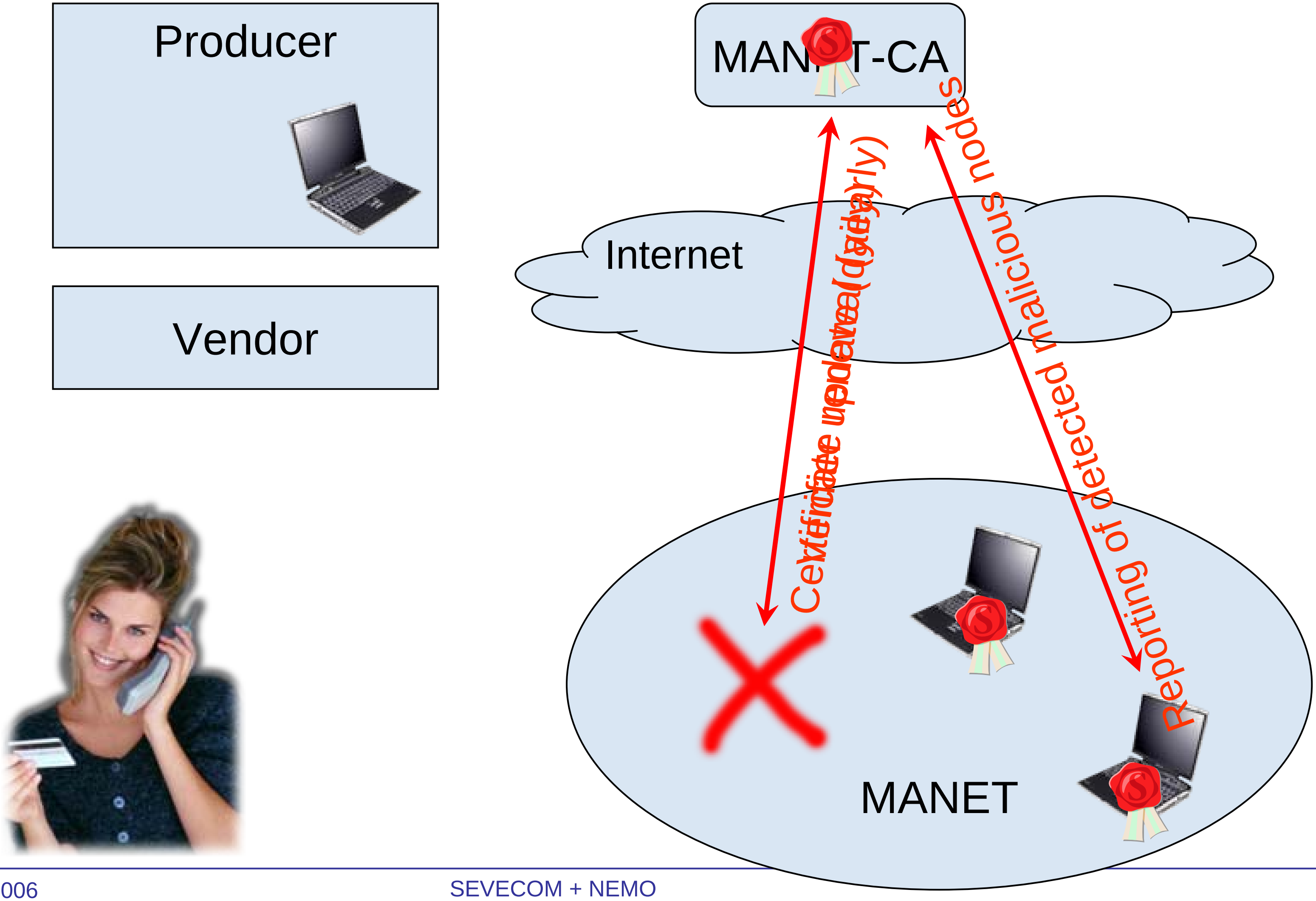
$$H^i(V_i) = H^i(H^{n-i}(Y_0)) = H^{i+n-i}(Y_0) = H^n(Y_0) = Y$$

Validity

- When section expires, node retrieves its new verifier from CA
- After period expires, nodes get new certificates
- Typical values: T = 1 year, n = 365, t = 1 day



MANET-ID application


Producer
Vendor
MANET-CA
Internet
Certificate renewal (regularly)
update (daily)
Reporting of detected malicious nodes
MANET

Performance

- MANET-ID: 1024 Bit RSA keys
- Certificate Size: 282 Bytes
- CA database for 10^9 users: ~ 260 Gigabytes
- MANET node storage requirements
  - 100 communication partners
  - Own cert. + Verifier + Verifiers + certs of other nodes
  - ~ 31 kByte
- Communication overhead
  - Verifier update: 24 Bytes (per day)
  - Certificate update: 422 Bytes (per year)

Summary + Conclusion

- MANET-IDs provide unique identifiers per node that cannot be forged
- Can be used to invalidate malicious nodes in MANETs
- Drawbacks/open questions
  - Needs CA infrastructure
  - Needs regular communication with CA (daily)
  - Moderate resource requirements
  - How to prevent transfer of identifiers?
    Embed in Secure Hardware?
- Privacy/pseudonym Issues not considered yet!!!