Zone Encryption with Anonymous Authentication for V2V Communication

Jan Camenisch, Manu Drijvers, Anja Lehmann, Gregory Neven and Patrick Towa

Gautam Singh

Indian Institute of Technology Hyderabad

April 27, 2024

V2X Related Terminology

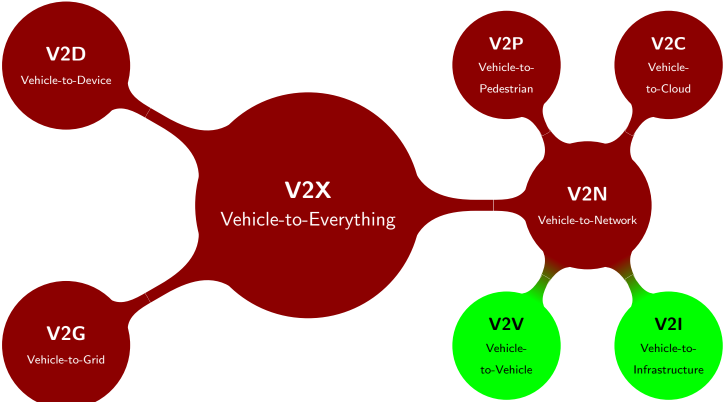

Figure 1: A breakdown of V2X.

Message Types in V2X

1. **Cooperative Awareness Messages** (CAMs)[1] and **Basic Safety Messages** (BSMs)[2].
    1. Exchanged between vehicles to create awareness and support cooperative performance of vehicles in the road network.
    2. Includes status information such as time, position, speed, active systems, vehicle dimensions, etc.

[1] European Telecommunications Standards Institute. "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service". In: ETSI EN 302 637-2 V1.4.1 (2019). URL: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf.

[2] *J2735_202309: V2X Communications Message Set Dictionary - SAE International*. URL: https://www.sae.org/standards/content/j2735_202309/ (visited on 04/15/2024).

Message Types in V2X

1. **Cooperative Awareness Messages** (CAMs)[1] and **Basic Safety Messages** (BSMs)[2].
    1. Exchanged between vehicles to create awareness and support cooperative performance of vehicles in the road network.
    2. Includes status information such as time, position, speed, active systems, vehicle dimensions, etc.
2. Other types of messages
    1. **Signal Phase and Timing** (SPaT)
    2. **Roadside Infrastructure Information** (MAP)

[1] European Telecommunications Standards Institute, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service".

[2] *J2735_202309*.

V2X and Cryptology

1. CAMs broadcasted unencrypted in 5.9 GHz channel (ETSI ITS-G5).

V2X and Cryptology

1. CAMs broadcasted unencrypted in 5.9 GHz channel (ETSI ITS-G5).
    - Frequently broadcast: 1 CAM per second in US, 10 per second in EU.
    - Easy to intercept.
    - Leak sensitive information about the vehicle owners.

V2X and Cryptology

1. CAMs broadcasted unencrypted in 5.9 GHz channel (ETSI ITS-G5).
    - Frequently broadcast: 1 CAM per second in US, 10 per second in EU.
    - Easy to intercept.
    - Leak sensitive information about the vehicle owners.
    - **Huge privacy concerns and threats!**

V2X and Cryptology

1. CAMs broadcasted unencrypted in 5.9 GHz channel (ETSI ITS-G5).
    - Frequently broadcast: 1 CAM per second in US, 10 per second in EU.
    - Easy to intercept.
    - Leak sensitive information about the vehicle owners.
    - **Huge privacy concerns and threats!**
2. Encryption impractical, since CAMs *must* be decrypted by nearby vehicles in a highly dynamic environment.
    - But CAMs *have to* be encrypted because of the data they carry!

V2X and Cryptology

1. CAMs broadcasted unencrypted in 5.9 GHz channel (ETSI ITS-G5).
    - Frequently broadcast: 1 CAM per second in US, 10 per second in EU.
    - Easy to intercept.
    - Leak sensitive information about the vehicle owners.
    - **Huge privacy concerns and threats!**
2. Encryption impractical, since CAMs *must* be decrypted by nearby vehicles in a highly dynamic environment.
    - But CAMs *have to* be encrypted because of the data they carry!
3. Instead, focus on *privacy-preserving authentication*.
    - Ensuring a message is issued by a “genuine” vehicle.
    - “Genuine” vehicles must be untraceable.

V2X and Cryptology

1. Deployed systems
    - Use short-term **pseudonym certificates** (100 per week in EU, 20 per week in US), rotate between them.
    - Trade-off between security (Sybil resistance), privacy and efficiency (storage and bandwidth costs).

V2X and Cryptology

1. Deployed systems
    - Use short-term **pseudonym certificates** (100 per week in EU, 20 per week in US), rotate between them.
    - Trade-off between security (Sybil resistance), privacy and efficiency (storage and bandwidth costs).
2. Proposed systems
    - Stronger privacy and security guarantees.
    - Do not meet the *stringent bandwidth constraint* of **300 bytes per CAM**, thus impractical.

Motivation and Goals

1. Unlimited privacy.
2. Address problems of authenticity and confidentiality in combination *for the first time*.

Motivation and Goals

1. Unlimited privacy.
2. Address problems of authenticity and confidentiality in combination *for the first time*.
3. Meet (bandwidth) requirements.
4. Negligible storage and bandwidth overheads.

Motivation and Goals

1. Unlimited privacy.
2. Address problems of authenticity and confidentiality in combination *for the first time*.
3. Meet (bandwidth) requirements.
4. Negligible storage and bandwidth overheads.
5. Efficient encryption scheme (symmetric-key crypto).
6. Better security guarantees (privacy, authenticity, confidentiality).

Preliminaries

1. Pairing-based Cryptography
2. Hardness Assumptions
   1. Symmetric Discrete Logarithm (SDL) assumption
   2. Modified q-Strong Diffie-Hellman (q-MSDH-1) assumption
3. Deterministic Authenticated Encryption (DAE)
4. PS Signatures
5. Dynamic Group Signatures with Attributes (DGS+A)

Preliminaries

1. Pairing-based Cryptography
2. Hardness Assumptions
   1. Symmetric Discrete Logarithm (SDL) assumption
   2. Modified q-Strong Diffie-Hellman (q-MSDH-1) assumption
3. Deterministic Authenticated Encryption (DAE)
4. PS Signatures
5. Dynamic Group Signatures with Attributes (DGS+A)
6. **CS6190**

Overall Flow of Zone Encryption

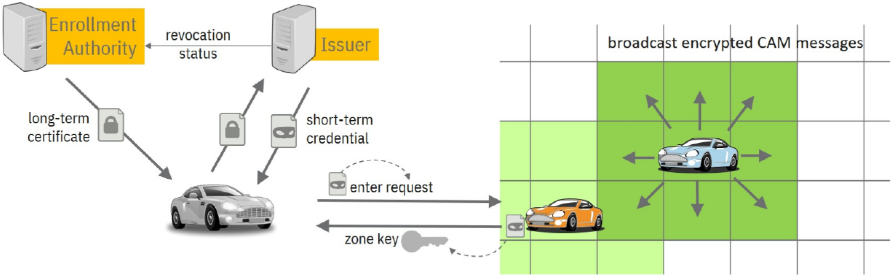

Figure 2: Illustration of Zone Encryption with its Anonymous-Authentication Approach.

Notation

Notation	Meaning
Z	Set of zones covering the road network
$\mathcal{P}$	Payload/message space
$Epoch$	Set of epochs
T	Set of timestamps
$K_{z,t}$	Zone key for zone z at time t
L_K	List of zone keys known to a vehicle, stored as $(z, t, K_{z,t})$
$\mathcal{E}$	Enrollment authority
$\mathcal{I}$	Issuer
$\mathcal{V} \in \{0,1\}^*$	Vehicle identity
$cert_{\mathcal{V}}$	Long-term certificate of $\mathcal{V}$
$cred_{\mathcal{V}}$	Short-term credential of $\mathcal{V}$

Zones, Epochs, Zone Keys

1. A *zone z* is a continuous geographical area covering part of a road network (shown as squares alongside).

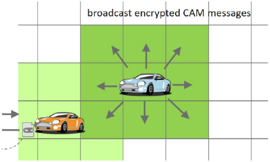

Figure 3: A vehicle must have the zone keys of zones adjacent to it. It can communicate with another vehicle if they share a zone key.

Zones, Epochs, Zone Keys

1. A *zone* z is a continuous geographical area covering part of a road network (shown as squares alongside).
2. Each zone has a *zone key* $K_{z,t}$ periodically refreshed after a time interval called an *epoch*.
   - An epoch is denoted by $[e, e+1)$. Each time instance t satisfies $e \leq t < e+1$ for a unique e. This is denoted as $e(t)$.
   - Vehicles need $K_{z,t}$ for secure communication when they are in zone z at time t.

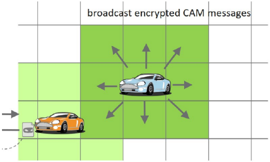

Figure 3: A vehicle must have the zone keys of zones adjacent to it. It can communicate with another vehicle if they share a zone key.

Zones, Epochs, Zone Keys

1. A *zone* z is a continuous geographical area covering part of a road network (shown as squares alongside).
2. Each zone has a *zone key* $K_{z,t}$ periodically refreshed after a time interval called an *epoch*.
   - An epoch is denoted by $[e, e+1)$. Each time instance t satisfies $e \le t < e+1$ for a unique e. This is denoted as $e(t)$.
   - Vehicles need $K_{z,t}$ for secure communication when they are in zone z at time t.
3. Vechicles can communicate securely with other vehicles in surrounding zones also.

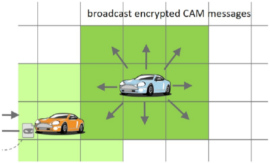

Figure 3: A vehicle must have the zone keys of zones adjacent to it. It can communicate with another vehicle if they share a zone key.

Entities and Credentials

1. An *enrollment authority* $\mathcal{E}$ issues *long-term certificates* to vehicle $\mathcal{V} \in \{0,1\}^*$.
   1. Long-term certificate $cert_{\mathcal{V}}$ obtained.
   2. Can be used to check revocation status.

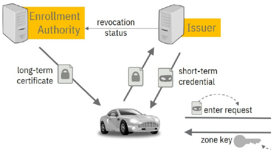

Figure 4: Various entities and exchanged credentials in ZE.

Entities and Credentials

1. An *enrollment authority* $\mathcal{E}$ issues *long-term certificates* to vehicle $\mathcal{V} \in \{0, 1\}^*$.
   1. Long-term certificate $cert_{\mathcal{V}}$ obtained.
   2. Can be used to check revocation status.
2. An *issuer* $\mathcal{I}$ issues *short-term credentials* to vehicles every epoch.
   1. Long-term credential $cert_{\mathcal{V}}$ used here.
   2. Short-term credential $cred_{\mathcal{V}}$ obtained.
   3. $cred_{\mathcal{V}}$ is valid only for the epoch e in which it was issued.

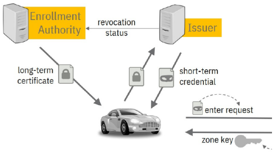

Figure 4: Various entities and exchanged credentials in ZE.

Syntax of ZE

Setup and Key Generation

1. Setup $\left(1^{\lambda}, Z, Epoch, T\right) \rightarrow pp$
2. KG.E $(pp) \rightarrow (pk_{\mathcal{E}}, (sk_{\mathcal{E}}, st_{\mathcal{E}}))$
   - State keeps track of enrolled vehicles.
3. KG.I $(pp) \rightarrow (pk_{\mathcal{I}}, (sk_{\mathcal{I}}, st_{\mathcal{I}}))$
   - State keeps track of open messages sent during key requests.

Syntax of ZE

Setup and Key Generation

1. Setup $\left(1^{\lambda}, Z, Epoch, T\right) \rightarrow pp$
2. KG.E $(pp) \rightarrow (pk_{\mathcal{E}}, (sk_{\mathcal{E}}, st_{\mathcal{E}}))$
   - State keeps track of enrolled vehicles.
3. KG.I $(pp) \rightarrow (pk_{\mathcal{I}}, (sk_{\mathcal{I}}, st_{\mathcal{I}}))$
   - State keeps track of open messages sent during key requests.

Credential Issuance

1. $\langle \text{Enroll.V}\,(pk_{\mathcal{E}}, \mathcal{V}) \leftrightharpoons \text{Enroll.E}\,(sk_{\mathcal{E}}, st_{\mathcal{E}}, \mathcal{V})\rangle \rightarrow \langle cert_{\mathcal{V}}, st'_{\mathcal{E}}\rangle$
2. $\langle \text{Authorize.V}\,(cert_{\mathcal{V}}, e, pk_{\mathcal{I}}) \leftrightharpoons \text{Authorize.I}\,(sk_{\mathcal{I}}, st_{\mathcal{I}}, \mathcal{V}, e, pk_{\mathcal{E}})\rangle \rightarrow \langle cred_{\mathcal{V}}, st'_{\mathcal{I}}\rangle$

Syntax of ZE

Setup and Key Generation

1. Setup $\left(1^{\lambda}, Z, Epoch, T\right) \rightarrow pp$
2. KG.E $(pp) \rightarrow (pk_{\mathcal{E}}, (sk_{\mathcal{E}}, st_{\mathcal{E}}))$
   - State keeps track of enrolled vehicles.
3. KG.I $(pp) \rightarrow (pk_{\mathcal{I}}, (sk_{\mathcal{I}}, st_{\mathcal{I}}))$
   - State keeps track of open messages sent during key requests.

Credential Issuance

1. $\langle \text{Enroll.V}\,(pk_{\mathcal{E}}, \mathcal{V}) \leftrightharpoons \text{Enroll.E}\,(sk_{\mathcal{E}}, st_{\mathcal{E}}, \mathcal{V})\rangle \rightarrow \langle cert_{\mathcal{V}}, st'_{\mathcal{E}}\rangle$
2. $\langle \text{Authorize.V}\,(cert_{\mathcal{V}}, e, pk_{\mathcal{I}}) \leftrightharpoons \text{Authorize.I}\,(sk_{\mathcal{I}}, st_{\mathcal{I}}, \mathcal{V}, e, pk_{\mathcal{E}})\rangle \rightarrow \langle cred_{\mathcal{V}}, st'_{\mathcal{I}}\rangle$
   - Vehicle uses certificate to obtain credentials.
   - Issuer checks certificate using public key of enrollment authority.

Syntax of ZE

Entering and Exiting Zones

1. $\langle \text{Enter.V}\left(cred_{\mathcal{V}}, L_K, pk_{\mathcal{I}}, z, t, requester\right) \leftrightharpoons$ $\text{Enter.W}\left(cred_{\mathcal{W}_i}, L_{K_i}, pk_{\mathcal{I}}, z, t, responder_i\right)_{i \geq 0}\rangle \rightarrow \langle L_K, \perp\rangle$
   - Why $i \geq 0$?
2. $\text{Exit}\left(L_K, z, t\right) \rightarrow L'_K$

Syntax of ZE

Entering and Exiting Zones

1. $\langle \mathrm{Enter.V}\left(cred_{\mathcal{V}}, L_K, pk_{\mathcal{I}}, z, t, requester\right) \leftrightharpoons \mathrm{Enter.W}\left(cred_{\mathcal{W}_i}, L_{K_i}, pk_{\mathcal{I}}, z, t, responder_i\right)_{i\geq 0}\rangle \rightarrow \langle L_K, \perp\rangle$
   - Why $i \geq 0$?
2. $\mathrm{Exit}\left(L_K, z, t\right) \rightarrow L'_K$

Sending and Receiving Payloads

1. $\mathrm{Send}\left(L_K, P, Y \subseteq Z, t\right) \rightarrow \gamma / \perp$
2. $\mathrm{Receive}\left(L_K, \gamma\right) \rightarrow P / \perp$
3. It's all symmteric key cryptography!

Syntax of ZE

Identity Escrow

1. Open $(sk_{\mathcal{I}}, st_{\mathcal{I}}, m) \rightarrow \mathcal{V}/\perp$
2. m is a message that was sent during an execution of Enter.

Syntax of ZE

Identity Escrow

1. Open $(sk_{\mathcal{I}}, st_{\mathcal{I}}, m) \rightarrow \mathcal{V}/\perp$
2. m is a message that was sent during an execution of Enter.
3. Only $\mathcal{I}$ can find which vehicle sent m. Use cases
   - To revoke certificates of misbehaving vehicles.
   - To provide concrete court evidence.

Syntax of ZE

Identity Escrow

1. Open $(sk_{\mathcal{I}}, st_{\mathcal{I}}, m) \rightarrow \mathcal{V}/\perp$
2. m is a message that was sent during an execution of Enter.
3. Only $\mathcal{I}$ can find which vehicle sent m. Use cases
   - To revoke certificates of misbehaving vehicles.
   - To provide concrete court evidence.
4. Assuming identity escrow is rare, Open need not be efficient in terms of time/storage complexity.

Security of ZE

1. **Anonymity**: Ciphertexts and messages during Enter do not reveal info about the vehicle that sent them.
    - Not necessary for messages associated with Authorize. (Why?)

Security of ZE

1. **Anonymity**: Ciphertexts and messages during Enter do not reveal info about the vehicle that sent them.
    - Not necessary for messages associated with Authorize. (Why?)
2. **Traceability**: If a vehicle knows $K_{z,t}$, it must have entered zone z at time t.
    - Issuer can trace the messages during Enter to long-term identity.

Security of ZE

1. **Anonymity**: Ciphertexts and messages during Enter do not reveal info about the vehicle that sent them.
    - Not necessary for messages associated with Authorize. (Why?)
2. **Traceability**: If a vehicle knows $K_{z,t}$, it must have entered zone z at time t.
    - Issuer can trace the messages during Enter to long-term identity.
3. **Ciphertext Integrity**: An efficient adversary cannot compute a valid ciphertext γ for a given (z, t) without knowing $K_{z,t}$.

Security of ZE

1. **Anonymity**: Ciphertexts and messages during Enter do not reveal info about the vehicle that sent them.
    - Not necessary for messages associated with Authorize. (Why?)
2. **Traceability**: If a vehicle knows $K_{z,t}$, it must have entered zone z at time t.
    - Issuer can trace the messages during Enter to long-term identity.
3. **Ciphertext Integrity**: An efficient adversary cannot compute a valid ciphertext γ for a given (z, t) without knowing $K_{z,t}$.
4. **Payload-Hiding security against Chosen-Ciphertext Attacks (PH-CCA)**: No efficient adversary can infer about the underlying payload without knowing $K_{z,t}$.

Building Blocks of ZE

1. SIG: Signature scheme for long-term certificates.

Building Blocks of ZE

1. SIG: Signature scheme for long-term certificates.
2. DGSA: Group signature scheme for short-term credentials.

Building Blocks of ZE

1. SIG: Signature scheme for long-term certificates.
2. DGSA: Group signature scheme for short-term credentials.
3. PKE: Public-key encryption for symmetric key exchange.

Building Blocks of ZE

1. SIG: Signature scheme for long-term certificates.
2. DGSA: Group signature scheme for short-term credentials.
3. PKE: Public-key encryption for symmetric key exchange.
4. SE: Symmetric-key encryption scheme for fast encryption of larger payloads.

Building Blocks of ZE

1. SIG: Signature scheme for long-term certificates.
2. DGSA: Group signature scheme for short-term credentials.
3. PKE: Public-key encryption for symmetric key exchange.
4. SE: Symmetric-key encryption scheme for fast encryption of larger payloads.
5. DAE: Deterministic Authenticated Encryption for wrapping symmetric payload keys with zone keys.
    - Why not do DAE on payloads?

Building Blocks of ZE

1. SIG: Signature scheme for long-term certificates.
2. DGSA: Group signature scheme for short-term credentials.
3. PKE: Public-key encryption for symmetric key exchange.
4. SE: Symmetric-key encryption scheme for fast encryption of larger payloads.
5. DAE: Deterministic Authenticated Encryption for wrapping symmetric payload keys with zone keys.
    - Why not do DAE on payloads?
    - *Remember the CAM length constraint!*

Summary of ZE

Parameter	Zone Encryption	C-ITS Proposal
Encrypted CAM	Yes	No
Anonymity	Yes	No
Pseudonyms per Week	Unlimited	100 (EU) / 20 (US)
CAM Authentication	DAE	ECDSA
Overhead per CAM	224 Bytes	160 Bytes
+ per entered Zones	284 (Request) / 300 (Response) Bytes	N/A

Table 1: Comparison of zone encryption to current C-ITS proposals at a 128-bit security level.

Introduction to DGS+A

1. **Group Signatures**[3]: A scheme where a user can sign a message anonymously on behalf of the group.

[3] David Chaum and Eugène van Heyst. "Group Signatures". In: *Advances in Cryptology — EUROCRYPT '91*. Ed. by Donald W. Davies. Berlin, Heidelberg: Springer, 1991, pp. 257–265. ISBN: 978-3-540-46416-7. DOI: 10.1007/3-540-46416-6_22.

[4] Dan Boneh, Xavier Boyen, and Hovav Shacham. *Short Group Signatures*. 2004. URL: https://eprint.iacr.org/2004/174 (visited on 04/26/2024). preprint.

[5] David Pointcheval and Olivier Sanders. *Short Randomizable Signatures*. 2015. URL: https://eprint.iacr.org/2015/525 (visited on 04/26/2024). preprint.

Introduction to DGS+A

1. **Group Signatures**[3]: A scheme where a user can sign a message anonymously on behalf of the group.
   - Group size and composition is fixed, thus impractical.

[3] Chaum and van Heyst, "Group Signatures".

[4] Boneh, Boyen, and Shacham, *Short Group Signatures*.

[5] Pointcheval and Sanders, *Short Randomizable Signatures*.

Introduction to DGS+A

1. **Group Signatures**[3]: A scheme where a user can sign a message anonymously on behalf of the group.
   - Group size and composition is fixed, thus impractical.
2. **Dynamic Group Signatures**[4]: A scheme where users can additionally join and leave the group at any time.

[3] Chaum and van Heyst, "Group Signatures".

[4] Boneh, Boyen, and Shacham, *Short Group Signatures*.

[5] Pointcheval and Sanders, *Short Randomizable Signatures*.

Introduction to DGS+A

1. **Group Signatures**[3]: A scheme where a user can sign a message anonymously on behalf of the group.
    - Group size and composition is fixed, thus impractical.
2. **Dynamic Group Signatures**[4]: A scheme where users can additionally join and leave the group at any time.
3. **Dynamic Group Signatures with Attributes**: Users obtain membership credentials corresponding to a set of their attributes by interacting with an issuer. Signatures are verified w.r.t. attributes.

[3] Chaum and van Heyst, "Group Signatures".

[4] Boneh, Boyen, and Shacham, *Short Group Signatures*.

[5] Pointcheval and Sanders, *Short Randomizable Signatures*.

Introduction to DGS+A

1. **Group Signatures**[3]: A scheme where a user can sign a message anonymously on behalf of the group.
    - Group size and composition is fixed, thus impractical.
2. **Dynamic Group Signatures**[4]: A scheme where users can additionally join and leave the group at any time.
3. **Dynamic Group Signatures with Attributes**: Users obtain membership credentials corresponding to a set of their attributes by interacting with an issuer. Signatures are verified w.r.t. attributes.
    - Other attributes of the user need not be revealed.

[3] Chaum and van Heyst, "Group Signatures".

[4] Boneh, Boyen, and Shacham, *Short Group Signatures*.

[5] Pointcheval and Sanders, *Short Randomizable Signatures*.

Introduction to DGS+A

1. **Group Signatures**[3]: A scheme where a user can sign a message anonymously on behalf of the group.
   - Group size and composition is fixed, thus impractical.
2. **Dynamic Group Signatures**[4]: A scheme where users can additionally join and leave the group at any time.
3. **Dynamic Group Signatures with Attributes**: Users obtain membership credentials corresponding to a set of their attributes by interacting with an issuer. Signatures are verified w.r.t. attributes.
   - Other attributes of the user need not be revealed.
4. DGS+A using PS[5] scheme.
   - Can sign k message blocks at once.
   - No hash functions needed and signatures are randomizable.
   - Also doubles up as a ZKPoK of σ on m.

[3] Chaum and van Heyst, "Group Signatures".

[4] Boneh, Boyen, and Shacham, *Short Group Signatures*.

[5] Pointcheval and Sanders, *Short Randomizable Signatures*.

Syntax of DGS+A

Note: An *issuer* $\mathcal{I}$ is a trusted party that issues credentials to users and can find the user that generated a given signature for a given message.

Setup and Key Generation

1. $\text{Setup}\left(1^\lambda, k\right) \to pp$: Generate public parameters.
2. $\text{KG.I}\left(pp\right) \to \left(pk, \left(sk, st\right)\right)$: Generate key pair for $\mathcal{I}$.

Credential Issuance

$\left\langle \text{Issue.I}\left(sk, st, id, A = \left(a_i\right)_{i=1}^{k}\right) \leftrightharpoons \text{Issue.U}\left(id, A, pk\right)\right\rangle \to cred$: Interactive protocol between a user $\mathcal{U}$ and issuer $\mathcal{I}$ to obtain credentials $cred$.

Syntax of DGS+A

Signing and Verification

1. $\mathrm{Auth}\left(pk, cred, m\right) \rightarrow tok$: Generate an *authentication token* or signature on m.
2. $\mathrm{Vf}\left(pk, m, A, tok\right) \rightarrow b \in \{0, 1\}$: Verify whether tok has been properly generated for the given m and A.

Opening

$\mathrm{Open}\left(sk, st, m, A, tok\right) \rightarrow id/\perp$: Check whether tok was generated properly and recover the identity id of the user that generated tok.
Note: Time complexity of Open is $\mathcal{O}\left(|ID|\right)$.

1. **Security Properties**: Correctness, Traceability, Anonymity.
2. **Application to ZE**: 216 Byte token size at 128-bit security level.
3. Extension to threshold opening.

Challenges and Future Improvements

1. **Key Agreement Strategy**
   - Which vehicle should reply to an entering vehicle?
   - How to handle key clusters due to transmission loss?
   - How to refresh zone keys (and who will generate them)?

Challenges and Future Improvements

1. **Key Agreement Strategy**
    - Which vehicle should reply to an entering vehicle?
    - How to handle key clusters due to transmission loss?
    - How to refresh zone keys (and who will generate them)?
2. **Robustness / Implementation Details**
    - Encrypting payloads under zone keys in a region.
    - Overlapping time periods for smooth transition.
    - Robust communication medium and retransmission mechanisms.

Challenges and Future Improvements

1. **Key Agreement Strategy**
   - Which vehicle should reply to an entering vehicle?
   - How to handle key clusters due to transmission loss?
   - How to refresh zone keys (and who will generate them)?
2. **Robustness / Implementation Details**
   - Encrypting payloads under zone keys in a region.
   - Overlapping time periods for smooth transition.
   - Robust communication medium and retransmission mechanisms.
3. Do we *really* need to encrypt CAMs?
   - Google (Maps) may already be profiling us!
   - Focus on more sensitive messages and information sent less frequently.
   - Avoid complexities in implementation of ZE.

Summary

1. Brief Introduction on V2X.
    - Services of V2X
    - Standards involved in V2X
    - V2X and cryptography. *Huge discrepancies*

Summary

1. Brief Introduction on V2X.
   - Services of V2X
   - Standards involved in V2X
   - V2X and cryptography. *Huge discrepancies*
2. Zone Encryption
   - Motivation and goals
   - Overall flow
   - Syntax
   - Building blocks
   - Security properties
   - Comparison to other proposals

Summary

1. Brief Introduction on V2X.
    - Services of V2X
    - Standards involved in V2X
    - V2X and cryptography. *Huge discrepancies*
2. Zone Encryption
    - Motivation and goals
    - Overall flow
    - Syntax
    - Building blocks
    - Security properties
    - Comparison to other proposals
3. DGS+A
    - Syntax
    - Instantiation from PS
    - Application to ZE

Pointcheval-Sanders Signatures

Consider $\Gamma = (q, \mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, e : \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_T) \leftarrow \mathsf{G}\left(1^\lambda\right)$, where G is a type-3 pairing group generator and λ is the *security parameter*, PS consists of the following algorithms.

Algorithm 1 PS.KG

Input: Pairing group Γ and number of message blocks k.
Output: Signing and verification key (vk, sk).

1: Generate $g_1 \in_R \mathbb{G}_1$, $x, y_1, \ldots, y_{k+1} \in_R \mathbb{Z}_q$.
2: Compute $X \leftarrow g_1^x$ and $Y_j \leftarrow g_1^j$ for $j \in \{1, \ldots, k+1\}$.
3: **return** $sk \leftarrow (x, y_1, \ldots, y_{k+1})$ and $vk \leftarrow (X, Y_1, \ldots, Y_{k+1})$.

Pointcheval-Sanders Signatures

Algorithm 2 PS.Sign

Input: Signing key sk and message $m = (m_1, \ldots, m_k)$.
Output: Signature σ on m.
1: Generate $h \in_R \mathbb{G}_1, m' \in_R \mathbb{Z}_q$.
2: **return** $\sigma \leftarrow \left(m', h, h^{x+\sum_{j=1}^{k} y_j m_j + y_{k+1} m'}\right)$.

Algorithm 3 PS.Vf

Input: Verification key vk, message $m = (m_1, \ldots, m_k)$, signature $\sigma = (m', \sigma_1, \sigma_2)$ on m.
Output: $b \in \{0, 1\}$.
1: **return** $b \leftarrow e\left(\sigma_1, X \prod_{j=1}^{k} Y_j^{m_j} Y_{k+1}^{m'}\right) \stackrel{?}{=} e\left(\sigma_2, g_1\right)$