

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

- 1 Introduction
- 2 Preliminaries
- 3 Zone Encryption
- 4 Dynamic Group Signatures with Attributes
- 5 Conclusion
- 6 Summary

V2X Related Terminology

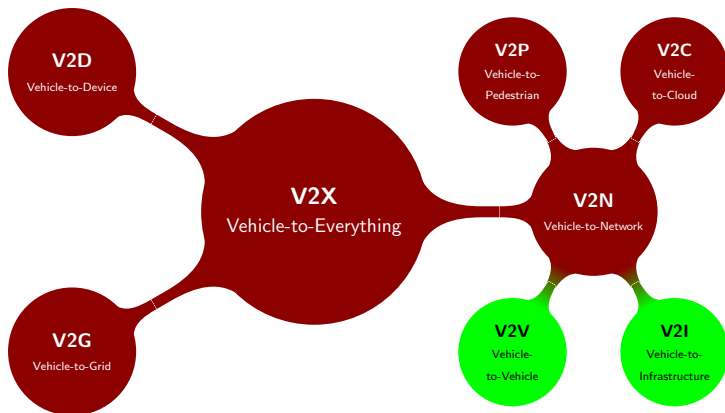


Figure 1: A breakdown of V2X.

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.

¹European 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.

²J2735-202309: 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)**

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

²J2735_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

- ❶ 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!**
- ❷ 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!
- ❸ 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

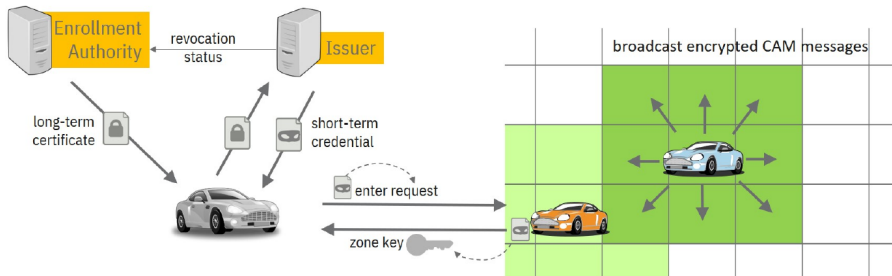


Figure 2: Illustration of Zone Encryption with its Anonymous-Authentication Approach.

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).

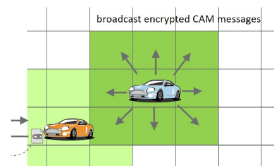


Figure 3: A vehicle must have the zone keys of zones adjacent to it. It can communicate with another vehicle if they share a zone key.

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 .

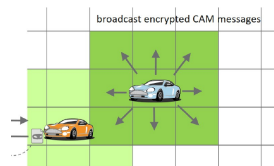


Figure 3: A vehicle must have the zone keys of zones adjacent to it. It can communicate with another vehicle if they share a zone key.

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 $e \leq t < e + 1$ 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 .
- ③ Vehicles can communicate securely with other vehicles in surrounding zones also.

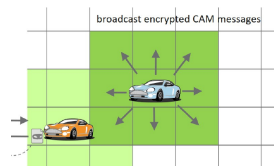


Figure 3: A vehicle must have the zone keys of zones adjacent to it. It can communicate with another vehicle if they share a zone key.

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.

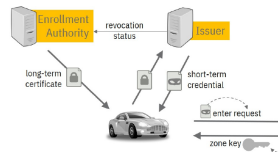


Figure 4: Various entities and exchanged credentials in ZE.

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.

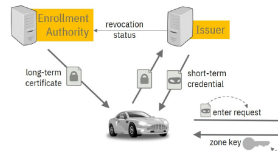


Figure 4: Various entities and exchanged credentials in ZE.

Syntax of ZE

Setup and Key Generation

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

Syntax of ZE

Setup and Key Generation

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

Credential Issuance

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

Syntax of ZE

Setup and Key Generation

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

Credential Issuance

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

Syntax of ZE

Entering and Exiting Zones

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

Syntax of ZE

Entering and Exiting Zones

- ① $\langle \text{Enter.V}(cred_V, L_K, pk_I, z, t, requester) \Leftrightarrow \text{Enter.W}(cred_{W_i}, L_{K_i}, pk_I, z, t, responder_i)_{i \geq 0} \rangle \rightarrow \langle L_K, \perp \rangle$
 - Why $i \geq 0$?
- ② $\text{Exit}(L_K, z, t) \rightarrow L'_K$

Sending and Receiving Payloads

- ① $\text{Send}(L_K, P, Y \subseteq Z, t) \rightarrow \gamma / \perp$
- ② $\text{Receive}(L_K, \gamma) \rightarrow P / \perp$
- ③ It's all symmetric key cryptography!

Syntax of ZE

Identity Escrow

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

Syntax of ZE

Identity Escrow

- ① $\text{Open}(sk_{\mathcal{I}}, st_{\mathcal{I}}, m) \rightarrow \mathcal{V} / \perp$
- 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 $\text{Open}(sk_{\mathcal{I}}, st_{\mathcal{I}}, m) \rightarrow \mathcal{V} / \perp$
- 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.

³David 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.

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

⁵David 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.

³Chaum and van Heyst, "Group Signatures".

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

⁵Pointcheval 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.

³Chaum and van Heyst, "Group Signatures".

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

⁵Pointcheval 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.

³Chaum and van Heyst, "Group Signatures".

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

⁵Pointcheval 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.

³Chaum and van Heyst, "Group Signatures".

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

⁵Pointcheval 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 .

³Chaum and van Heyst, "Group Signatures".

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

⁵Pointcheval 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 $\text{Setup}(1^\lambda, k) \rightarrow pp$: Generate public parameters.
- ② $\text{KG.I}(pp) \rightarrow (pk, (sk, st))$: Generate key pair for \mathcal{I} .

Credential Issuance

$\langle \text{Issue.I}(sk, st, id, A = (a_i)_{i=1}^k) \rightleftharpoons \text{Issue.U}(id, A, pk) \rangle \rightarrow cred$:

Interactive protocol between a user \mathcal{U} and issuer \mathcal{I} to obtain credentials $cred$.

Syntax of DGS+A

Signing and Verification

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

Opening

$\text{Open}(sk, st, m, A, tok) \rightarrow id / \perp$: Check whether tok was generated properly and recover the identity id of the user that generated tok .

Note: Time complexity of Open is $\mathcal{O}(|ID|)$.

- 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 $\Gamma = (q, \mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, e : \mathbb{G}_0 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T) \leftarrow G(1^\lambda)$, 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 $g_1 \in_R \mathbb{G}_1$, $x, y_1, \dots, y_{k+1} \in_R \mathbb{Z}_q$.
 - 2: Compute $X \leftarrow g_1^x$ and $Y_j \leftarrow g_1^{y_j}$ for $j \in \{1, \dots, k+1\}$.
 - 3: **return** $sk \leftarrow (x, y_1, \dots, y_{k+1})$ and $vk \leftarrow (X, Y_1, \dots, Y_{k+1})$.
-

Pointcheval-Sanders Signatures

Algorithm 2 PS.Sign

Input: Signing key sk and message $m = (m_1, \dots, m_k)$.

Output: Signature σ on m .

- 1: Generate $h \in_R \mathbb{G}_1$, $m' \in_R \mathbb{Z}_q$.
 - 2: **return** $\sigma \leftarrow (m', h, h^{x + \sum_{j=1}^k y_j m_j + y_{k+1} m'})$.
-

Algorithm 3 PS.Vf

Input: Verification key vk , message $m = (m_1, \dots, m_k)$, signature $\sigma = (m', \sigma_1, \sigma_2)$ on m .

Output: $b \in \{0, 1\}$.

- 1: **return** $b \leftarrow e\left(\sigma_1, X \prod_{j=1}^k Y_j^{m_j} Y_{k+1}^{m'}\right) \stackrel{?}{=} e(\sigma_2, g_1)$
-