Lightweight Privacy Preserving Authentication and Encryption Scheme for V2X With Self Revocation

Presentation By Prashanth Sriram S For Mini-project under Dr.Maria Francis

Indian Institute of Technology Hyderabad

December 16, 2022

Outline

Introduction

Introduction

- V2X (Vehicle-To-Everything) : Communication between Vehicles and between Vehicles and Traffic Infrastructure - integral part in C-ITS (Cooperative Intelligent Transportation Systems).
- V2X relies on regular continuous transmission of CAMs (Cooperative Awareness Messages) - transmits on-road traffic information
- V2X provides efficient solutions for reducing traffic congestion, improving road safety, etc.
- **Constraints on V2X:**
  - Communication occurs over insecure open networks
  - Conventional Authentication Schemes - impractical due to Bandwidth constraints on CAMs
  - Privacy issues in authentication: Need to prevent vehicles getting profiled- leading to tracking the location or identifying the user

Existing Solutions

- European Telecommunications Standards Institute (ETSI) and the U.S. Department of Transportation (USDOT) use pseudonym certificates for anonymous authentication in V2X.
- **Pseudonym based models:**
  - Each vehicle has a pool of pseudonyms and messages are signed with different pseudonyms from the pool.
  - Since, each of these pseudonyms are short lived, changing them frequently can prevent tracking the vehicle over a period of time.
- **Drawbacks of Pseudonym Based Models:**
  - Comes with additional storage and download costs
  - Does not provide protection against colluding adversaries or malicious infrastructure.
  - Even though the pseudonyms are changing, since the CAMs are sent as clear text, an eavesdropping adversary can eavesdrop on static information like the vehicle's dimensions and then correlate the messages coming from a specific vehicle over a long period of time.
  - Locating the eavesdropping adversary is difficult

Existing Solutions

- Still, Pseudonyms are extensively used, because they are relatively lightweight and faster than the alternatives that may provide stronger anonymity guarantees.
- Since many of the drawbacks arise from the plaintext nature of CAMs, encrypting them may be a solution, But, Since, the group of vehicles are continuously changing and the C-ITSs is open, It is a challenging task.

The Originally Proposed Scheme

- A bilinear pairing based privacy preserving authentication and encryption scheme fine tuned to V2X with more robust privacy and anonymity guarantees than pseudonym-based mechanisms.
- **Assumptions of the Environment :** The V2X Environment is geographically partitioned into zones based on the location of roadside units (RSUs).
- **Basic Setup:** There are 3 main entities.
  1. Enrollment authority as the issuer of credentials
  2. Vehicles as the users
  3. RSUs as the verifiers
- **Main Cryptographic Primitives:**
  1. **Anonymous authentication using randomisable signatures** - Vehicles prove the possession of the credentials (issued by the enrollment authority) to the verifier(RSUs) anonymously using randomisable signatures.
  2. **Symmetric Key Authenticated Encryption:** CAM data is encrypted, thus preventing the misuse of it by eavesdroppers.

Anonymous Authentication

- The issuing authority (say, the Department of Transport) issues a single long-term credential to the vehicle on registration.
- Everytime, a vehicle enters an RSU zone, it randomised the long term credential to generate a fresh short-term credential and uses this to authenticate itself to the corresponding RSU. After authentication, it can communicate with the other vehicles
- Randomisation helps preserve anonymity across different RSU zones.
- **Unlinkability:** These short-term credentials are unlinkable across different zones and unlinkable to the long-term credential.
- Since the short-term credentials are generated by the vehicle itself, even if an adversary colludes with the issuing authority, the vehicle cannot be identified.

Symmetric Key Authenticated Encryption

- After the RSU authenticates a vehicle, it ties an attribute to the vehicle, using which it will derive the secret key for an encrypted communication session in that zone. This secret key is specific to that zone
- Knowledge of vehicle's secret on which the long term credential is generated is required to derive a valid zone key.
- **Performance benefits:**
  - Since only an authenticated vehicle can derive the secret zone key required for communication, the vehicles do not have to authenticate every CAM.
  - This scheme does not require the vehicle to present a Zero Knowledge Proof of Knowing (ZKPoK) of the secret corresponding to the long-term credential. Even though, a vehicle without the credential may get wrongfully authenticated by the RSU, since it is needed to derive the secret key, it cannot communicate with the other entities.

Related Work

Pseudonym Certificates

- Pseudonym certificates are widely preferred for authentication of vehicles in X2X, primarily because of their efficiency.
- ETSI and USDOT standards employ Elliptic Curve Digital Signature Algorithm (ECDSA) based pseduonym certificates due to its short signatures and efficient verification
- Our scheme uses encrypted messages to communicate rather than signed plaintext messages. And the derivation of this encryption key requires authentication, thus preventing the use of Zero Knowledge proofs, making it more efficient.

Design Goals

Design Goals

Our proposed scheme has the following features with respect to pseudonym based mechanisms.

- **Anonymous Authentication:** Randomisable signatures are used to generate credentials for the vehicles during enrollment. The unforgeability and unlinkability property of randomized signatures maintains the vehicle's anonymity
- **Collusion Resistance:**
  1. Unlike in the pseudonym based mechanisms where the credentials are generated by the pseudonym authority, here, the vehicles themselves generate the short-term credentials, so there is no short-term credential issuing authority to collude with.
  2. Also, due to the fact that this is unlinkable to the long-term credential issued by the issuing authority, The user's privacy is protected even when the adversary colludes with the issuing authority.

Design Goals

- **Confidentiality:** Since the CAMs are encrypted and not clear text messages, misuse of the messages (such as profiling, mass surveillance, etc.) can be prevented.
- **Efficiency:**
  - Unlike the schemes, where the vehicles has to compute and present a ZKPoK of its secret for verifying its credentials (which is computationally expensive), here the vehicle will need the secret to derive the key after getting authenticated by the RSU to then communicate with the other entities.
  - Since the vehicles are authenticated by the RSUs and the CAMs are encrypted, there is no need to sign and authenticate each CAM using temporary certificates, thus further decreasing the overhead for ensuring confidentiality.

Preliminaries

Notations

- $a \xleftarrow{R} X$: a is chosen uniformly at random from the set X
- $\mathbb{Z}_q$: Group of integers modulo q
- $\mathbb{Z}_q^* = \{x | x \in \mathbb{Z}_q \text{ and } \gcd(x, q) = 1\} = \mathbb{Z}_q \backslash \{0\}$ when q is prime

Bilinear Pairings:

- Let $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ be two additive cyclic groups of prime order q
- $G = \langle g \rangle$ be a multiplicative cyclic group of order q.
- A bilinear pairing is defined as an efficiently computable map $e : (G_1 \times G_2) \rightarrow \mathbf{G}$ satisfying the following properties:
  1. **Bilinearity:** $\forall x, y \in \mathbb{Z}_q^*, P \in G_1, Q, Q_1, Q_2 \in G_2, e(xP, yQ) = e(P, Q)^{xy}$ and $e(P, Q_1 + Q_2) = e(P, Q_1).e(P, Q_2)$
  2. **Non-degeneracy:** $e(g_1, g_2) \neq 1_{\mathbf{G}}$ i.e. $e(g_1, g_2)$ is a generator of $\mathbf{G}$
- $\mathcal{G}en(1^k)$: bilinear pairing group generator for security parameter k and prime $q = \Theta(k)$.
- Let $(q, e, g_1, g_2, \mathbf{g}, G_1, G_2, \mathbf{G}) \xleftarrow{R} \mathcal{G}en(1^k)$ be the asymmetric bilinear pairing group generated uniformly at random by $\mathcal{G}en(1^k)$.
- Our scheme requires a Type-3 pairing group - asymmetirc pairing groups where there is no known efficiently computable homomorphism from G_1 to G_2

Public Key Encryption Scheme

- A public key encryption scheme is a triplet of PPT algorithms $(GenPKE, Enc, Dec)$ s.t.
  - $GenPKE(1^k)$: outputs pair of public and secret keys (pk, sk)
  - The Encryption algorithm $c \leftarrow Enc(pk, m)$:
  - The Deterministic Decryption algorithm $m = Dec(sk, c)$

  such that $Dec(sk, Enc(pk, m)) = m$ except for a negligible probability.
- $GenPKEScheme(1^k)$: Generates a $(GenPKE, Enc, Dec)$ triplet such that the resulting PKE Scheme is secure.

Hardness Assumptions

We rely on the following hardness assumptions to prove the security of our scheme.

- **Co-computational Diffie-Hellman (Co-CDH) assumption:** Given $(q, e, g_1, g_2, \mathbf{g}, G_1, G_2, \mathbf{G}) \xleftarrow{R} \mathcal{G}en(1^k), g_1^x, g_2^y$, where $x, y \xleftarrow{R} \mathbb{Z}_q^*$, for every PPT adversary $\mathcal{A}$ there exists a negligible function $\epsilon(k)$ such that $Pr[\mathcal{A}((q, e, g_1, g_2, \mathbf{g}, G_1, G_2, \mathbf{G}), g_1^x, g_2^y) \rightarrow g_1^{xy}] \leq \epsilon(k)$

Hardness Assumptions

- **Decisional Bilinear Diffie-Hellman (DBDH) assumption:** Given $(q, e, g_1, g_2, \mathbf{g}, G_1, G_2, \mathbf{G}) \xleftarrow{R} \mathcal{G}en(1^k), g_1^x, g_2^y, g_2^z$ where $x, y, z \xleftarrow{R} \mathbb{Z}_q^*$, Let $D_0 = e(g_1, g_2)^{xyz}$ and $D_1 = e(g_1, g_2)^c$, where $c \xleftarrow{R} \mathbb{Z}_q^*$, for every PPT adversary $\mathcal{A}$ there exists a negligible function $\epsilon(k)$ such that $|Pr[\mathcal{A}((q, e, g_1, g_2, \mathbf{g}, G_1, G_2, \mathbf{G}), g_1^x, g_2^y, g_2^z, D_0) = 1] - Pr[\mathcal{A}((q, e, g_1, g_2, \mathbf{g}, G_1, G_2, \mathbf{G}), g_1^x, g_2^y, g_2^z, D_1) = 1]| \leq \epsilon(k)$

Randomisable Signatures

- Given a randomisable signature σ on a message α, we can create another fresh valid signature $\sigma^{'}$ for α, indistinguishable from σ by raising σ to a scalar r.
- Thus, a user can get a signature σ on some secret s from the issuer and then for proving to a verifier, a fresh signature can ge generated like this and so, the proof remains unlinkable from all the previous times

Originally Proposed Scheme

Authentication Mechanism

- Setup $\mathcal{S}(1^k)$: With the security parameter k, output the type-3 bilinear pairing parameters $(q, e, g_1, g_2, \mathbf{g}, G_1, G_2, \mathbf{G}) \xleftarrow{R} \mathcal{G}en(1^k)$
- Keygen $\mathcal{K}(1^k)$:
  1. The enrollment authority generates $sk = (x, y) \xleftarrow{R} \mathbb{Z}_q^*$ as its secret keys and publishes $pk = (X = g_2^x, Y = g_2^y)$ as its public keys
  2. The user generates $(\alpha, \beta) \xleftarrow{R} \mathbb{Z}_q^*$ as its secrets and sends $req = (a = g_1^\beta, b = a^\alpha)$ to the issuer for obtaining the credential
- GenCred $\mathcal{G}(sk, req)$: The issuer verifies the vehicle and uses its secret keys (x, y) to compute $c = a^x, d = (a^\alpha.c)^y = a^{y(\alpha+x)}$ and then outputs the signature $\sigma = (a, b, c, d)$

Authentication Mechanism

- Verify $\mathcal{V}(\sigma, pk)$: Uses the public keys of the issuer and the credential of the user as input and verifies whether $e(a, X) \stackrel{?}{=} e(c, g_2)$ and $e(d, g_2) \stackrel{?}{=} e(b.c, Y)$. On successful verification, it accepts (outputs 1), else rejects (outputs 0)

Correctness of Verification

If $\sigma = (a, b = a^{\alpha}, c = a^{x}, d = a^{y(\alpha+x)},$

- $e(c = a^{x}, g_2) = e(a, g_2^{x}) = e(a, X)$
- $e(d = a^{y.(\alpha+x)}, g_2) = e(a, g_2)^{y.(\alpha+x)}$ And, $e(b.c, Y) = e(a^{\alpha}.a^{x}, g_2^{y}) = e(a, g_2)^{y(\alpha+x)}$

Randomisability and Unlinkablity

From a credential σ, By choosing $r \xleftarrow{R} \mathbb{Z}_q^*$, we can generate $\bar{\sigma} = (a^r, b^r, c^r, d^r)$ which is also a valid credential and it is unlikable from σ

Unforgeability

- Only the enrollment authority can issue a valid credential. No efficient adversary can forge a credential and authenticate itself as a valid user.
- We say that a credential scheme $(\mathcal{S},\mathcal{K},\mathcal{G},\mathcal{V})$ is unforgeable if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage in $\mathbf{Exp}^{\mathbf{unforgeable}}_{\mathcal{O}_{(a,b)},\mathcal{V},\mathcal{Q},k}(\mathcal{A})$

Key Generation Mechanism

Authenticated vehicles use a symmetric key k_v which is derived with the help of the RSU, for encrypted communication

- Keygen $\mathcal{K}(1^k, c = a^x)$: The key issuing authority generates attribute $v \xleftarrow{R} \mathbb{Z}_q^*$. The authority updates the credential σ by updating c to $c' = c^v$. $V = g_1^v$ is also published for the verification of the derived key.
- Derive $\mathcal{D}(c', \beta)$: This algorithm derives the symmetric key g_1^{xv} as follows: $(c')^{\beta^{-1}} = (a^{xv})^{\beta^{-1}} = (g_1^{\beta xv})^{\beta^{-1}} = g_1^{xv}$
- Verify $\mathcal{V}(V, X, g_1^{xv})$: Verifies whether $e(V = g_1^v, X = g_2^x) \stackrel{?}{=} e(g_1^{xv}, g_2)$ is satisfied.

Modified Scheme

Additional Design Goals

- **Self Revocation:** Users should be able to revoke themselves without revealing their secrets (α,β). This is achieved by limiting the lifetime of the credentials and maintaining a list of non-revoked credentials. The credentials of only those users in the list are updated, and a user can be removed from the list to revoke that user.
- **Efficient Updates of Credentials:** To avoid expensive interactions between each user and the issuing authority to update the credentials, the issuing authority posts the $update$ of each credential in the list publicly online, before the next epoch begins and users can download their $update$ while online and then update their credentials offline. Each user can only make use of their appropriate $update$, since it is encrypted with each of the user's public key.

Modified Authentication Scheme

- Setup $\mathcal{S}(1^k)$: With the security parameter k, output the type-3 bilinear pairing parameters $(q, e, g_1, g_2, \mathbf{g}, G_1, G_2, \mathbf{G}) \xleftarrow{R} \mathcal{G}en(1^k)$
  Output the PKE scheme: $(GenPKE, Enc, Dec) \leftarrow GenPKEScheme(1^k)$
- Keygen $\mathcal{K}(1^k)$:
  1. The enrollment authority generates $sk = (x_0, y) \xleftarrow{R} \mathbb{Z}_q^*$ as its secret keys and publishes $pk = (X_0 = g_2^{x_0}, Y = g_2^y)$ as its public keys
  2. The user generates $(\alpha, \beta) \xleftarrow{R} \mathbb{Z}_q^*$ as its secrets. User also generates a $(sk_{user}, pk_{user}) \leftarrow GenPKE(1^k)$. User sends $req = (a = g_1^\beta, b = a^\alpha, pk_{user})$ to the issuer for obtaining the credential
- InitIA $\mathcal{I}(1^k)$: The issuing authority creates an empty table T with $A, B, pk, C_{enc}, D_{enc}$ as its columns. This Table will contain the entries for each of the valid users. (The issuing authority publishes the pk, C_{enc}, D_{enc} part of T before every epoch begins)

Modified Authentication Scheme

- GenCred $\mathcal{G}(sk, req, T)$, where $sk = (x, y)$ is the secret key of the issuing authority, this epoch:
  1. The issuer verifies the vehicle and uses its secret keys (x, y) to compute $c = a^x, d = (a^\alpha.c)^y = a^{y(\alpha+x)}$ and then outputs the signature $\sigma = (a, b, c, d)$.
  2. It computes $C_{enc} = Enc(pk_{user}, c)$ and $D_{enc} = Enc(pk_{user}, d)$
  3. It then adds $(a, b, pk_{user}, C_{enc}, D_{end})$ to the table T.

Modified Authentication Scheme

- **IssuerUpdate** $\mathcal{U}(1^k, sk, T)$:, run by the issuing authority before the next epoch begins,
  1. Issuing authority picks a new $x' \xleftarrow{R} \mathbb{Z}_q^*$ and publishes $X^{'} = g_2^{x'}$, thus making the secret and public keys for the next epoch as $sk' = (x', y)$ and $pk' = (X', Y)$ and publishes the new public keys.
  2. For each entry in T:
     1. Let the entry be $(A_i, B_i, pk_{user_i}, C_{enc_i}, D_{enc_i})$
     2. The issuing authority computes new $c = A_i^x, d = (B.c)^y$
     3. The I.A. updates $C_{enc_i} = Enc(pk_{user_i}, c)$ and $D_{enc_i} = Enc(pk_{user_i}, d)$ in T
  3. The issuing authority publishes the $pk_{user_i}, C_{enc_i}, D_{enc_i}$ columns of T online, let this be called T_{pub}

  (Note, the I.A. only needs to go online after completing step 2 and publish the updates all at once.)

Modified Authentication Scheme

- UserUpdate $(sk_{user}, pk_{user}, pk', , T_{pub})$:, run by each user to update their credentials for the next epoch, (run after the issuer has updated T_{pub})
  1. The user goes online and searches for the entry with their pk_{user} in T_{pub} and download that entry - $(pk_{user}, C_{enc}, D_{enc})$
  2. User uses their secret key to decrypt to get the updated credentials, $c' = Dec(sk_{user}, C_{enc})$, $d' = Dec(sk_{user}, D_{enc})$. User now uses $\sigma' = (a, b, c', d')$ as their credentials for the next epoch.
  3. User runs $\mathcal{V}(\sigma', pk')$ to verify whether the new credentials are correct.

Modified Authentication Scheme

- Verify $\mathcal{V}(\sigma, pk)$: Uses the public keys of the issuer, this epoch and the credential of the user as input and verifies whether $e(a, X) \stackrel{?}{=} e(c, g_2)$ and $e(d, g_2) \stackrel{?}{=} e(b.c, Y)$. On successful verification, it accepts (outputs 1), else rejects (outputs 0)

Self Revocation

- Revoke $\mathcal{R}$ is an interactive protocol between the issuer and the user.
  1. The User first sends their non-randomised original credential $\sigma = (a, b, c, d)$ to the issuer.
  2. The Issuer first runs $\mathcal{V}(\sigma, pk)$ to verify whether it is a valid signature, if it fails, the revocation also fails
  3. The Issuer then wants to verify if the user knows the secret(α, β) behind their signature, so, the issuer generates $u \xleftarrow{R} \mathbb{Z}_q^*$ and sends $c' = c^u$ and g_1^u to the user.
  4. If the user knew the secrets, the user can run $\mathcal{D}(c', \beta)$ to get g_1^{xu} and send this back to the issuer, let this be R
  5. The issuer simply verifies $e(g_1^u, X = g_2^x) \stackrel{?}{=} e(R = g_1^{xu}, g_2)$ and thus checks if the user has sent the correct value. If not, the revocation fails. (Note: Since this verification part does not require the knowledge of the private key x, revocation can be moved from the issuing authority to a separate revocation authority too)
  6. Now, the issuer checks for an entry in T with $A = a$ and $B = b$. If found, that entry is removed and the revocation is done, since that user's credentials will no longer be updated and hence cannot be used from the next epoch.
  7. If such an entry was not found in T, it means the user sent a randomised version of their credential and hence the revocation will simply fail.

Comparison to the original scheme

- The Key Generation Mechanism at the RSU remains the same except for the fact that the RSU must use the correct pk of the issuing authority of that epoch.
- The modified scheme retains the randomisability and Unlinkability of the original scheme. Since, For a valid updated credential σ, $\bar{\sigma} = (a^r, b^r, c^r, d^r)$, for a randomly chosen r in $\mathbb{Z}_q^*$, is also a valid credential and unlinkable from σ.
- If the original scheme is unforegeable, then the modified scheme is also unforgeable, i.e no efficient adversary can forge a valid credential, even if it also had access to valid credentials of previous epochs.
- Since, the key generation mechanism and the encryption methods at the RSU zone level after authentication is unchanged, all the security properties of this encryption scheme are retained in the modified scheme.

Conclusion

Conclusion

- We present a privacy preserving authentication scheme to enable anonymous authentication and resolve confidentiality issues in V2X.
- We also present a way to add self revocation capabilities to this scheme by maintaining a list of valid credentials and limiting the lifetime of each credential, along with an efficient way to update them.
- Further, we can also move the revocation from the issuing authority to a separate Revocation Authority, if we wanted.

Limitations

- Our scheme has the limitation that for communication, a vehicle has to rely on an RSU for authentication and key derivation.
- Our scheme currently lacks a revocation mechanism to revoke the credentials of malicious vehicles, i.e. currently there is no way for either the RSU or the issuing authority to revoke an user based on the randomised credential used by the user. The scheme only supports self-revocation currently.
- However, we believe that we can support revocation of malicious users using their randomised credentials by looking at other anonymous credential schemes that support such revocation and integrating it to our scheme.

Thank you!

Roll: cs20btech11039