PRIVACY PRESERVING TRUST MODEL BASED ON BLOCKCHAIN FOR VANETS

Problem statement

VANETS are the ad hoc systems that were appointed to improve the effectiveness and activities of vehicles of the network .

These networks are volatile in nature also they are extremely rapid in comparewith other wired and remote systems .

Are the two major concerns in VANET

1. security
2.privacy

MOTIVATION

Sensitive info is being broadcasted in VANET which in turn attracts various attackers.

No Authentication and association measures are provided in WAVE standard due to fast network establishment needs

Easy to attack due to infrastructure less model

Very high chances of threat to privacy

Motivated by these challenges,we propose a privacy-preserving authentication scheme for VANETs based on consortium blockchain,

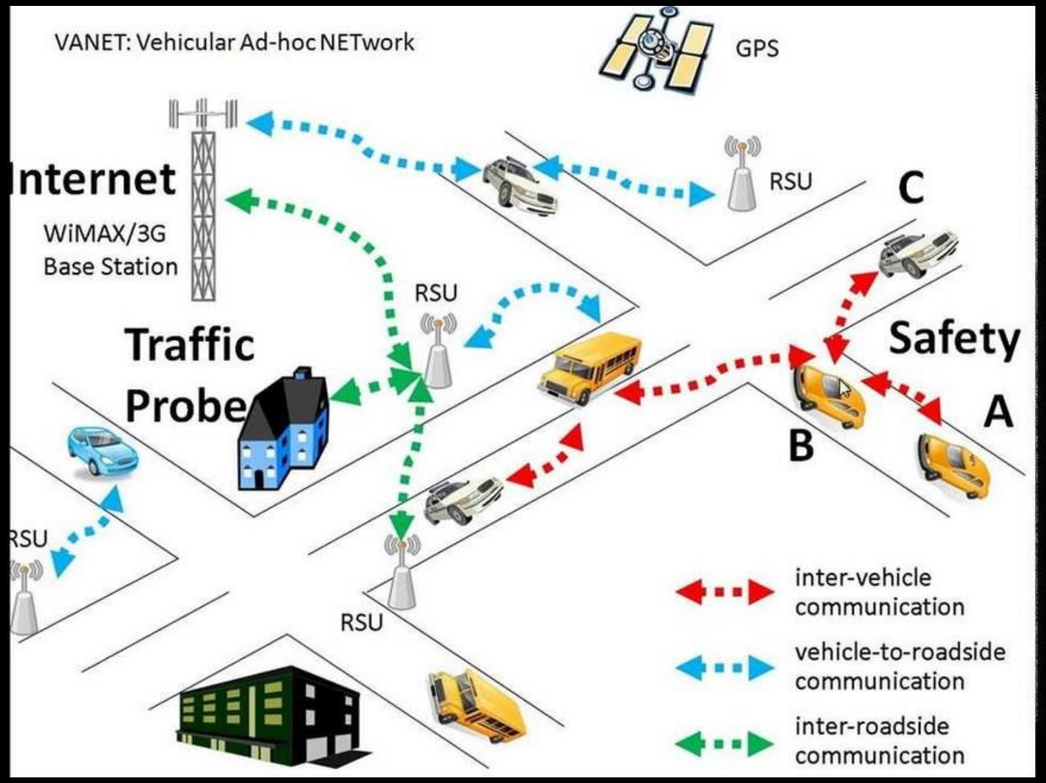
INSPIRATION

Pairing based cryptography - Diverse groups are framed developing their geographical area and groups forward messages to eachother

Failure model -Because group leader is chosen haphazardly and via grouping, leader messages are passed.

So the leader's security is under threat

VANETS



Wide attention in the field of intelligent transportation systems (ITSs)

1.Communicate with each other in vehicle-to-vehicle (V2V) m

2.Communicate directly with road-side units (RSUs) through vehicle to-infrastructure (V2I) mode

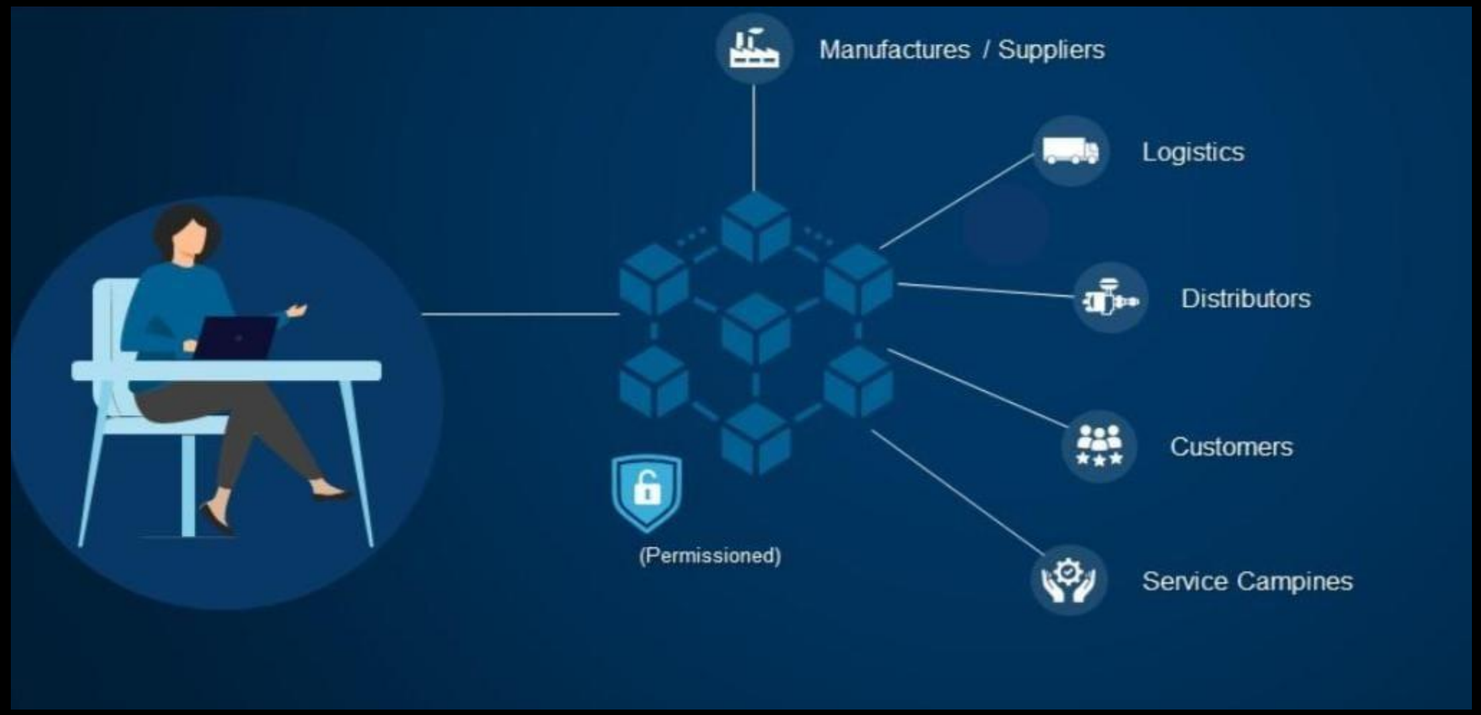
GOALS

SECURITY - related cryptography and mechanisms should always be the first priority of any authentication protocol.

- **PRIVACY -A vehicle's private information should be required as less as possible EFFICIENCY -The computational cost and storage requirement for keys and certificates**

CONSORTIUM BLOCKCHAIN

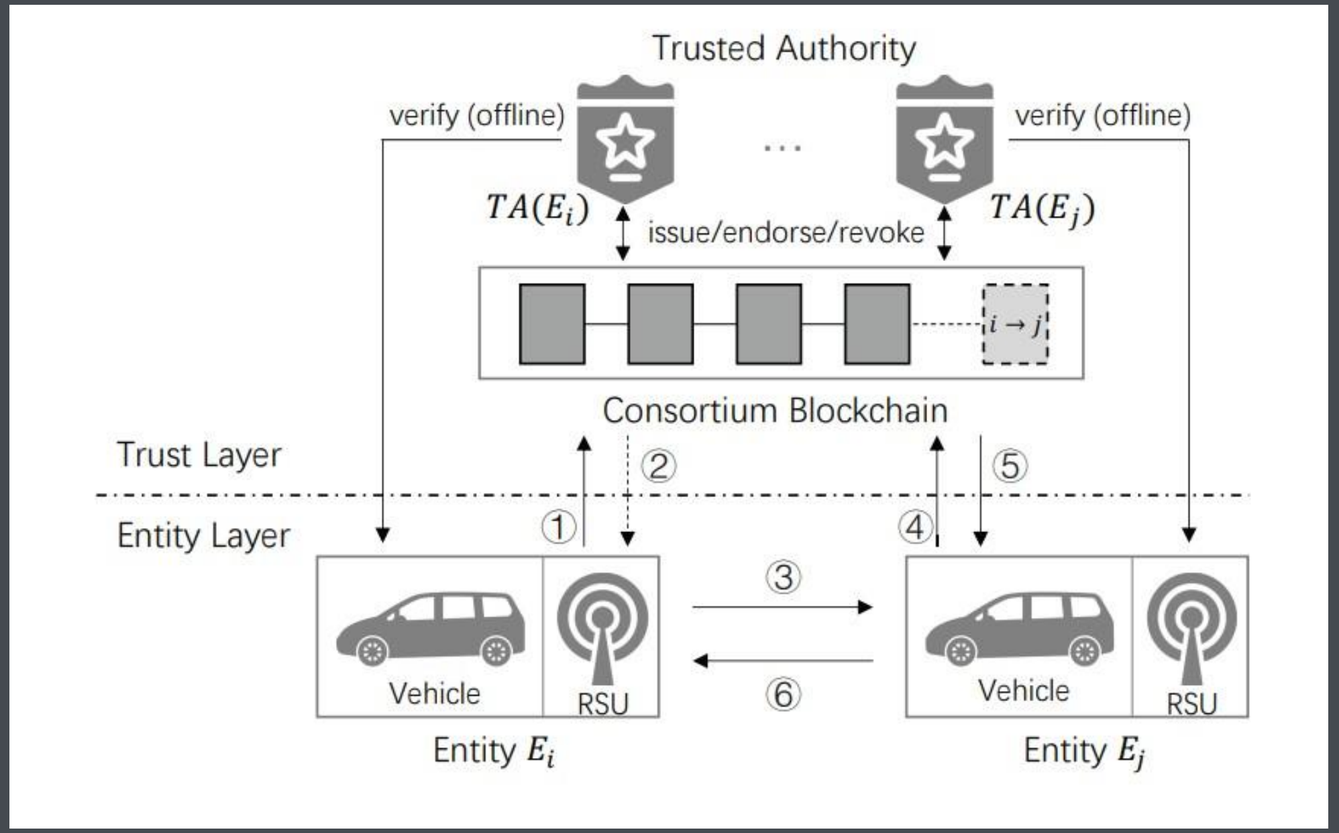
A consortium blockchain is a type of semi-decentralized network in which members are not granted to a single entity.



FRAMWORK

1. ENTITY LAYER
   A vehicle or a RSU is regarded as an entity of VANETs which needs authentication service.

2. TRUST LAYER
   A consortium blockchain is deployed along with its native trusted authorities



UTXO Data Structure

- Unspent transaction output (UTXO), successfully used in Bitcoin
- A token is introduced to represent a one-time guarantee for

authenticity

- **Once a receiver gets a token from a sender in the ledger**

1. It means an authentication request has been sent from the latter entity and

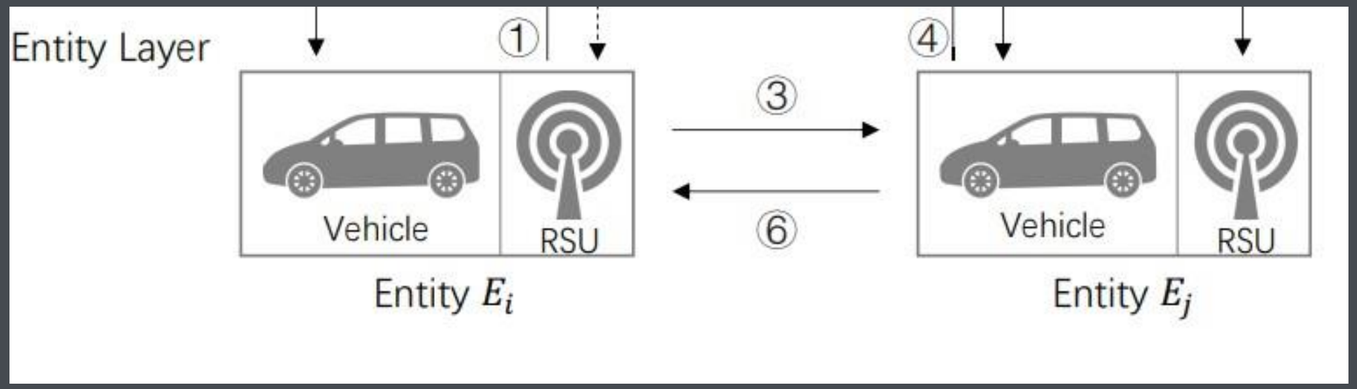
It represents the proof of the dedicated sender and its authentication activity

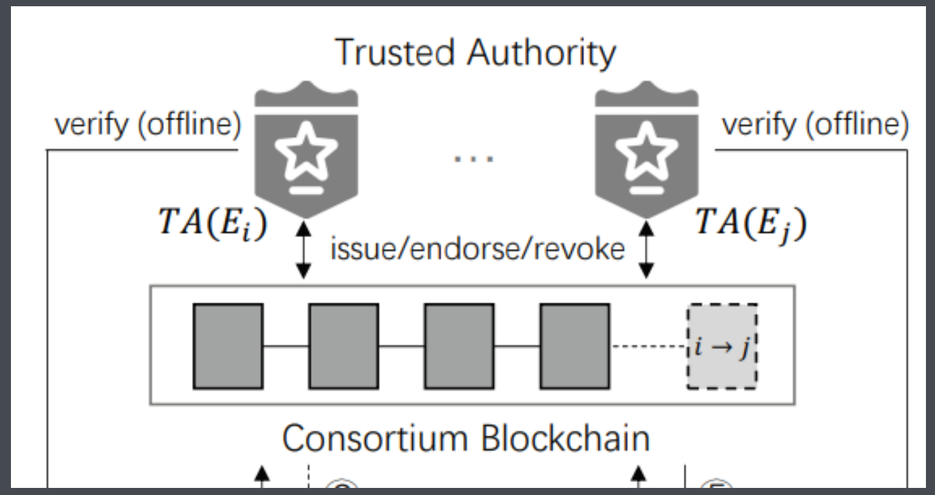
KEY ELEMENTS OF UTXO DATA STRUCTURE

Item	Name	Description
basic	Tx_id	Transaction ID
	$Timestamp$	Request time
	$Script$	Operation name
	Sig	Signature of requester to prove ownership
in	$Sender\_id$	TA/Entity ID
	$Type$	*Token* type (signature of TA on $Send\_id$)
out	$Recipient\_id$	TA/Entity ID
	$Quantity$	*Token* amount (for authentication, 1)

- For the purpose of authentication, one token is sufficient

WORKING MECHANISM



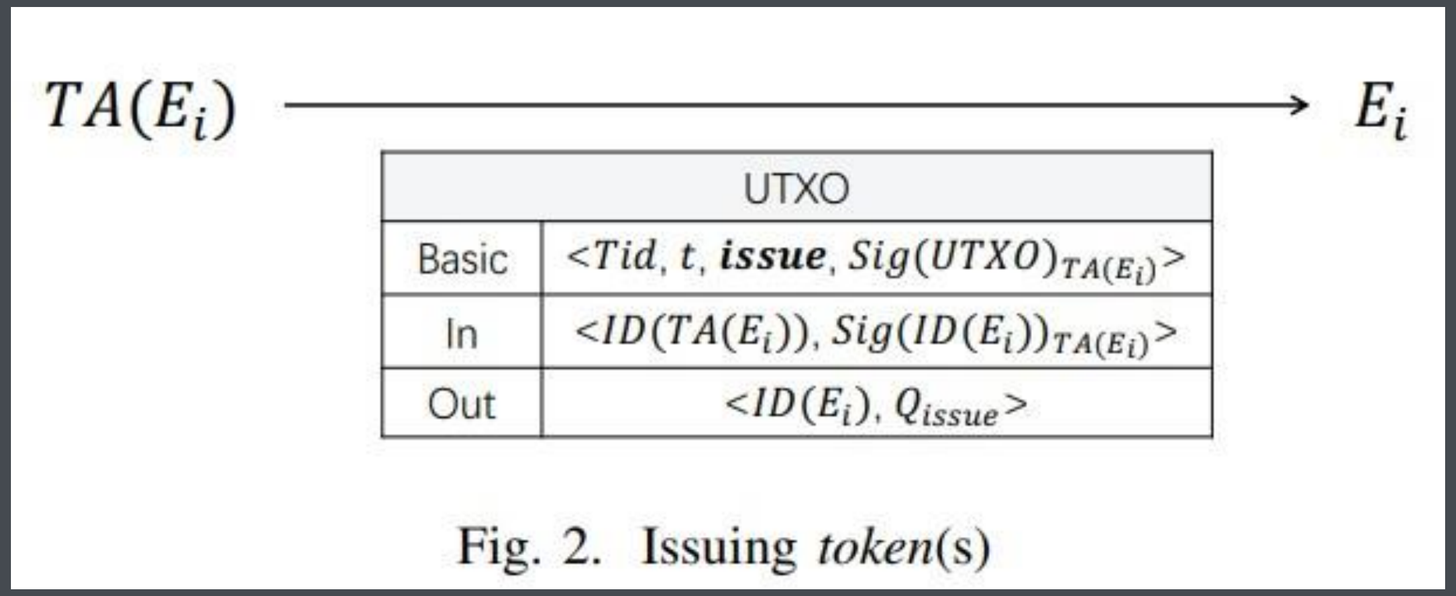


- Entity Ei and Ej create their public/private keys locally and register in their dedicated authorities, namely T A(Ei) and T A(Ej), by submitting their real identities and public keys.
- Once an entity (Ei) is successfully verified by a TA
- The entity enrolls in the consortium blockchain is assigned with a unique address or an ID, namely ID(Ei).

After the enrollment, operations of the entity are

1. Issue

- This operation is the exclusive way to generate new tokens Which is issued to an entity Ei by its T A(Ei) under two circumstance
-

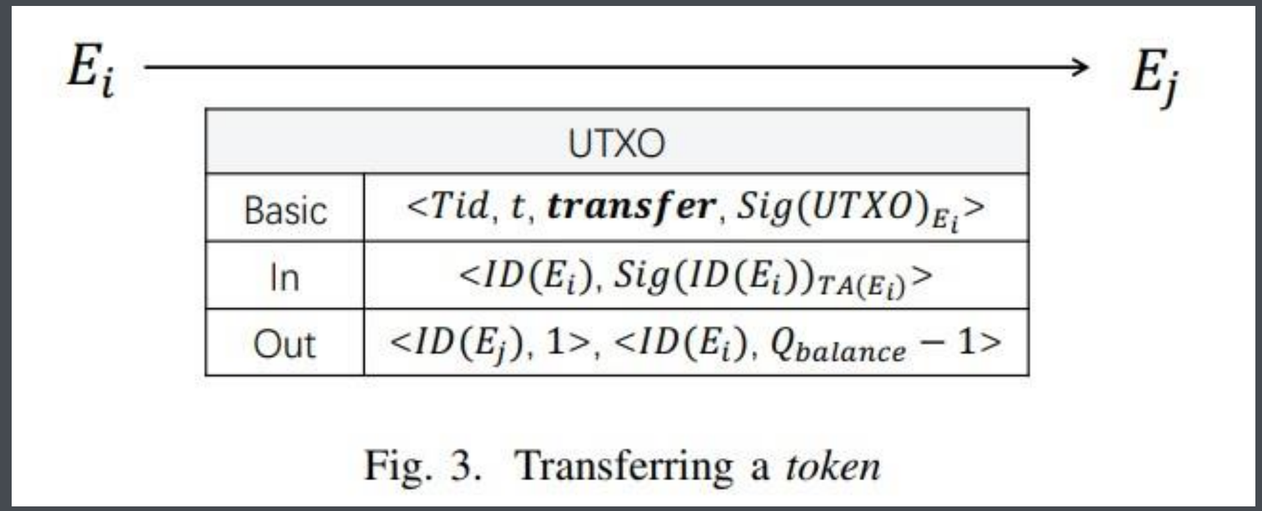


UTXO	
Basic	$<Tid, t, \textbf{issue}, Sig(UTXO)_{TA(E_i)}>$
In	$<ID(TA(E_i)), Sig(ID(E_i))_{TA(E_i)}>$
Out	$<ID(E_i), Q_{issue}>$

Fig. 2. Issuing *token*(s)

2. Transfer:

- A basic operation for authentication,where entity Ei transfers one token to entity Ej



UTXO	
Basic	$<Tid, t, \boldsymbol{transfer}, Sig(UTXO)_{E_i}>$
In	$<ID(E_i), Sig(ID(E_i))_{TA(E_i)}>$
Out	$<ID(E_j), 1>, <ID(E_i), Q_{balance} - 1>$

Fig. 3. Transferring a *token*

3. Query:

- The query operation is a straightforward
- Given an index (e.g., T x id), the operation provides the UTXO of the transaction on the blockchain. It is fairly useful for the recipient to check if the sender is trustworthy or not

AUTHENTICATION ALGORITHM

Algorithm 1 Ingoing Authentication Algorithm

Input: $UTXO$ from E_i without a Tx_id.
Output: Tx_id or $error\_code$.

1: **if** Formal check on the $UTXO$ is ok **then**
2: Retrieve E_i's public key $K_{pub}(E_i)$ by $TA(E_i)$
3: Verify $UTXO.Sig(UTXO)_{E_i}$ by using $K_{pub}(E_i)$
4: **if** $UTXO.Sig(UTXO)_{E_i}$ is valid **then**
5: $Type \leftarrow$signature of $TA(E_i)$ on $UTXO.ID(E_i)$
6: **if** $Type == UTXO.Sig(ID(E_i))_{TA(E_i)}$ **then**
7: $Tx\_id \leftarrow generateTransaction(UXTO)$
8: **return** Tx_id
9: **end if**
10: **end if**
11: **end if**
12: **return** $error\_code$

Algorithm 2 Outgoing Authentication Algorithm

Input: $ID(E_i)$ and a transaction $index$ (from E_i to E_j)
Output: $true$ or $false$

1: $UTXO \leftarrow queryTransaction(index)$
2: **if** $UTXO.Sender\_id == ID(E_i)$
&& $UTXO.Recipient\_id == ID(E_j)$ **then**
3: **if** $UTXO$ has never been used for E_j **then**
4: Mark $UXTO$ as used in E_j's local database
5: **return** $true$
6: **end if**
7: **end if**
8: **return** $false$

EVALUATION :

1. Security and privacy analysis
2. Man in the middle attack
3. Identity revealing attack
4. Authority abuse attack

REFERENCES

1Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 2, pp. 760–776, 2019.

2 S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," Journal of Network and Computer Applications, vol. 37, pp. 380–392, 2014.

3 M. Azeez, P. Vijayakumar, and L. J. Deborah, "Comprehensive survey on security services in vehicular ad-hoc networks," IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 6, pp. 379–388, 2016.

4 M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," Journal of Computer Security, vol. 15, no. 1, pp. 39–68, 2007.