CHAPTER 5

PROXY BLIND SIGNATURE SCHEME FOR AUTHENTICATING NON-SAFETY APPLICATION MESSAGES

5.1 INTRODUCTION

Non-safety applications in VANET aim to create new commercial opportunities and ensure travelling comfort and entertainment through boosted access with cost-effective technology. They offer the required informational support and entertainment access to the road travellers in making their journey more comfortable. They are varied and encompass conventional IP-based applications such as media playing and web browsing along with applications pertaining to the vehicular scenario like map downloading, travelling information as per upcoming locations, toll service and relevant advertisements. The non-safety applications are also known as comfort and general information services and that they encompass traffic efficiency details such as better route selection and shorter driving time (Casteigts et al 2011).

The non-safety applications, unlike safety ones, possess different communication requirements, ranging right from no dedicated real-time demand to assured QoS for interactive amusement. Usually, this VANET application class is inspired by the wish of the travellers to interact with other vehicles or with road-based infrastructure like internet hosts. The focus of these applications is not only on transmitting information to maximum
possible vehicles over a big area but also on applications that reach up to small areas for ensuring cooperative driving (Wischhof et al 2005 and Nadeem et al 2006).

In practice, non-safety messages such as inquiry, routing and traffic management information are forwarded between the V2V and V2I on the DSRC medium. However, it is essential that the non-safety messages should not disturb the highly critical safety messages. Nevertheless, although not as critical as safety messages, the non-safety ones need to maintain the security requirements of trust, message’s integrity and non-repudiation.

Just as safety messages, non-safety ones are also susceptible to security attacks in VANET. As several companies set up their business near the highway or a street road to offer non-safety application services, the rivals can exploit the resources to trigger issues for other parties for ruining their services with different types of attacks. Furthermore, non-safety related messages also consume a significant percentage of resources in vehicular communication, which can considerably increase with higher traffic density. This means that the security-based scalability issue also needs to be addressed.

The importance of authenticating non-safety messages can be understood through an example of car parking application, a common and major non-safety application in VANET. In this scenario, the concerned RSU sends information about the available parking slots at a nearby shopping mall to the authentic user X (vehicle or node), which has the message ‘Parking slot is available’. Now, the user X forwards this message to other user B, an attacker node then changes the message to ‘No parking slot’ and transmits it to other vehicle C, as shown in Figure 5.1.
Therefore, it becomes necessary to authenticate the non-safety messages in VANET. This chapter explores authenticating these messages through the proxy blind signature scheme.

5.2 BLIND SIGNATURE

In the world of security, digital signature has proved itself as an effective solution to keep identity and integrity threats at bay. It is a digital version of a handwritten signature that is essential for ensuring that only a genuine user has sent a message that has not been changed during its course of transmission. In short, a digital signature takes care of authentication, non-repudiation and message integrity. However, severe limitations of digital signature are reflected when the user has to identify itself in transactions such as getting a service, which results in privacy breach. While a digital signature exposes the user’s identity in any communication, a blind signature retains the sender’s privacy by allowing it to get the message signed from the signer without sharing the actual message or its signature.

An associate of digital signature, blind signature is a two-party protocol that involves a signer along with a cluster of requesters. A requestor
requests and obtains a legitimate signature from a signer without exposing the message contents to the signer. The generated signature by the signer is known as a blind signature. The signer adds the signature with the help of her/his private key, which is then verified by the requester with the help of the signer’s public key. The requester un-blinds the obtained blind signature and verifies it to make the message signature pair to the public, as shown in Figure 5.2.

![Figure 5.2 Generic blind signing procedure](image)

Blind signature helps in communicating securely and anonymously without risking the identity (Chaum 1983). It possesses a major property known as unlinkability or untraceability, which means that once the signature is generated, the signer is unable to trace or associate the signature with its corresponding message. This is termed as blindness property via which the requesters avert the signer from obtaining the exact association between the signer’s signing process and the signature known later.

In short, in a blind signature mechanism, the signer cannot identify the content of the message and that the requester exposes the signature later even after which the signer is unable to relate the blind message and blind
signature. According to Mohammed et al (2000) and Mohanty & Majhi (2010), the blind signature must satisfy the following properties or security requirements apart from unlinkability.

**Blindness**: The signer is unaware of the message for which a valid signature is generated.

**Correctness**: Any requester or verifier can test the preciseness of a signature using the public keys of signer.

**Integrity**: The message contents are intact.

**Authenticity**: A valid signature means a genuine signer signed it knowingly.

**Non-Reusability**: The requester can use the valid signature only once.

**Unforgeability**: A valid signer produces a valid signature.

**Non-repudiation**: The signer agrees to have signed a message with a valid signature.

**Confidentiality**: Only the authenticated user can change the message contents.

A blind signature also facilitates revocable anonymity. Apart from protecting the message contents, it is also necessary to keep the recipient’s identity private. In the context of digital transactions where there is no anonymity, it becomes easy to figure out the users’ preferences that act as critical information for commercial benefitters. This creates an issue in the case of anonymous transactions. The transactions can be anonymous via a blind signature. However, if there is a dispute, the identity of the wicked user
is revealed by the third party authority, which is termed as revocable anonymity - to withdraw secrecy when needed.

A blind signature scheme is applied wherein the requester does not want the signee to link a message and a signature to a specific request. Such a scheme is crucial in an electric cash transaction wherein a message may contain a monetary value of the customer (requester). When the bank (signee) receives this message and signature for payment, it cannot infer which party originally generated the signature. The requester obtains the signature by unblinding function and the signee is unable to associate the signature and the blind message. Through this untraceable scheme, the user is allowed to withdraw a legitimate coin from a bank and use it anonymously at a shop (Tan et al 2002).

Since proxy signature contributes greatly to the security aspects in a few applications, it has grabbed attention of several developers. The mix of blind signature and proxy signature triggers some crucial advantages due to which it is proposed to apply them concurrently in a few real situations, for instance, in anonymous parking situations or toll collection applications in VANET.

5.3 PROXY BLIND SIGNATURE

Lin & Jan (2000) first introduced a proxy blind signature scheme based on DLP and ECDLP to take the advantage of security properties of both the blind and proxy signature mechanisms. It was derived from the Schnorr blind signature scheme. Later, Tan et al (2002) introduced a new a proxy blind signature mechanism. Awasthi & Lal (2005) proved that Tan et al scheme was susceptible to a kind of forgery attack and also introduced a new proxy blind mechanism based on Mambo’s scheme which was found to be more efficient and secure one.
In 2005, Wang & Wang introduced a proxy blind signature scheme by using ECDLP. However, Yang & Yu (2008) proved that the above scheme did not meet the security properties and therefore, introduced an improved scheme. Nevertheless, this scheme did not satisfy the unforgeability property. Later, Qi & Wang (2009) introduced a proxy blind signature mechanism that implemented factoring and ECDLP but again this scheme did not fulfill the unlinkability and unforgeability properties. Later, Alghazzawi et al (2011) and Pradhan & Mohapatra (2011) introduced a new proxy blind signature mechanism that implemented ECDLP, which was asserted to be secure and efficient.

Table 5.1 shows the computational overhead of the various proxy blind signature schemes. The notations M, A and H denote modulo multiplication, modulo addition and hash operation respectively.

**Table 5.1 Comparison of efficiency of proxy blind signature schemes**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Delegation Phase</th>
<th>Blind Signing Phase</th>
<th>Verification Phase</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alghazzawi et al’s (2011)</td>
<td>3M+2A</td>
<td>4M+3A+H</td>
<td>2M+A+H</td>
<td>9M+6A+2H</td>
</tr>
<tr>
<td>Pradhan and Mohapatra’s (2011)</td>
<td>4M+3A+2H</td>
<td>4M+4A+H</td>
<td>2M+H</td>
<td>10M+7A+4H</td>
</tr>
</tbody>
</table>

From the above table, it is clear that Alghazzawi et al (2011) scheme has the lowest total computational overhead of all the schemes.
However, in terms of fulfilling the security properties, this scheme is not recommended because it does not fulfill the signature unlinkability requirement. Therefore, Pradhan & Mohapatra (2011) proxy blind signature scheme can be implemented, where users’ privacy and proxy signature are essential.

5.3.1 Definition

A proxy blind signature refers to a digital signature mechanism that fulfils the security properties of proxy as well as blind signatures. In such a scheme, the original signer assigns his signing right to the proxy signer due to special reasons. This means that the proxy signer produces a blind signature on behalf of the actual signer. Therefore, a proxy blind signature scheme is a special kind of signature that enables a designated user termed as a proxy signer to sign on behalf of multiple actual signers without viewing the message content. The delegation relationship is thus created between the original and proxy signers. The blindness property here signifies that the proxy signer is unaware of the content of the message to be signed. The scheme blends the advantages of blind, proxy and multi-signature schemes. A majority of proxy blind signature schemes were established on the basis of the hard problems such as Integer factorization problem (IFP) and DLP that feature sub-exponential time.

The proxy blind signature mechanism is useful in ensuring the security of e-commerce transactions. Because it focuses on both authentication and privacy, the scheme should fulfil the following security properties:

**Verifiability**: Any arbitrary verifier or the signature’s receiver can accurately validate the proxy blind signature.
**Distinguishability**: The normal signature made by the actual signer should be different and distinguishable from the proxy blind signature made by the proxy signer.

**Unforgeability**: Only the proxy signer has the right to produce a legitimate proxy blind signature.

**Unlinkability**: Once the requester exposes the unblinded form of the signature for verification, the proxy or the actual signer cannot link the relation between the blinded message she or he signed and the exposed signature.

**Non-repudiation**: The actual signer as well as the proxy signer cannot later deny that they were not involved in the signing procedure.

**Identifiability**: Anyone can verify the identity of the original as well as of the proxy signer from the corresponding signature.

**Prevention of Misuse**: The proxy key pair is available only for producing proxy signature.

### 5.3.2 Network Assumptions

The RSUs in a particular area (streets, highways) are connected to the designated RSCs. Thus a number of RSCs are set up throughout the VANET, which are in turn connected to the Internet. The principal authority in VANET system is the DOT which works as the CA that protects an exhaustive dataset containing all necessary information of each RSU under an RSC. For instance, the RSU’s location data, stationing history of RSUs along with the public key of the RSC are stored. DOT’s public key is openly available to all the members including the vehicles in the VANET. The DOT may in turn be assisted by the local transportation authorities with
information necessary to negotiate any dispute, including issuing licensing materials for a vehicle and/or any commercial aspects.

5.3.3 Mechanism

Initially, the RSU announces the certificate containing $ID_{RSC}, ID_{RSU}, ADDR_{RSU}$ (MAC address of the RSU) and the $LOC_{RSU}$ (location information of RSU). An OBU obtains the public key of the RSC, the original MAC addresses of the RSU and the designated RSU location. The initial (beacon) message has the following certificate:

$\langle ID_{RSC}, ID_{RSU}, LOC_{RSU} \rangle, H(ID_{RSC}, ID_{RSU}, LOC_{RSU})Sign_{CA}$

where $H(.)$ is a one-way hash function and $(\cdot), H(\cdot)Sign_{CA}$ indicates a signature using CA’s secret key. The CA’s signature endorses the message integrity and that the RSU is a valid member of the affiliating RSU group governed by the particular RSC.

By accepting the beacon frame the OBU checks the received MAC address with the MAC address of the transmitting RSU. The OBU joins the RSU group after the RSU’s MAC address is validated. The position and time of the OBU are synchronized with the position and time information from the RSU.

In order to accomplish the message integrity and trust requirements of RSU-to-OBU communications, proxy blind signature can be used that would authorize a RSU to sign a message on behalf of the message originator while in the process, the RSU cannot alter the message or replay the expired messages.
Furthermore, delegation with warrant has been used in proxy blind signature for the message integrity and privacy of OBU message delivery that covers OBU-to-RSU and OBU-to-OBU message delivery.

Two large prime numbers, $p$ and $q$ ($q$ is a prime factor of $p - 1$), are conglomerated with VANET inception. Both $p, q$ are attached to a large geographic region such as $p$ to a country and $q$ to a state or province in that country. Then a generator $g$ for $Z_p$, is picked to be associated with a comparatively small area (for example a city, or a town). The system variables with their scope in the network are detailed in Table 13.

**Table 5.2 List of parameters and their extensiveness for message delivery in VANET**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Vastness in the Network</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RSC (Original Signer)</td>
</tr>
<tr>
<td></td>
<td>Public</td>
</tr>
<tr>
<td>$p, q, g, r, T$</td>
<td>✓</td>
</tr>
<tr>
<td>$s, v, r_s$</td>
<td></td>
</tr>
<tr>
<td>$g, t, s'$</td>
<td></td>
</tr>
<tr>
<td>$s_t, k$</td>
<td></td>
</tr>
<tr>
<td>$e^<em>, s^</em>$</td>
<td></td>
</tr>
<tr>
<td>$a, b, r', e$</td>
<td></td>
</tr>
</tbody>
</table>

**System Parameter Initialization**

The parameters used in the proposed scheme are:

**A**: Original Signer - RSC

**B**: Proxy Signer

- RSU (in case of RSU message delivery)
- OBU (in case of OBU message delivery)
\( R \): Signature Requester \[\text{RSU (in case of OBU message delivery)}\] \[\text{OBU (in case of RSU message delivery)}\]

\( CA \): Central Authority or DOT

\( p, q \): Two large prime numbers such that, \( q \mid q - 1 \)

\( g \): An element of order \( q \) in \( Z_p^* \)

\( x_A, x_B, x_R \in Z_q^* \): Secret key of \( A, B, R \) respectively.

\( y_A = g^{x_A} (mod p) \): RSC’s public key

\( y_B = g^{x_B} (mod p) \): Proxy signer’s public key

\( y_R = g^{x_R} (mod p) \): Receiver R’s public key

\( t_s \): Message timestamp

\( H(\cdot) \): Cryptographically secure one way hash function

\( || \): Concatenation of two strings

\( m_w \): Message warrant

\( m \): Message

**Proxy Delegation**

The RSC randomly picks out \( v \in Z_q^* \) and computes,

\[ r = g^v (mod p) \quad (5.1) \]

\[ s = x_A + v.H(m_w||r) (mod q) \quad (5.2) \]

RSC sends \((r, s)\) along with the message warrant \( m_w \) to the proxy signer \( B \) and \( CA \), via a secure channel.

Now the proxy signer calculates,
\[ s_{pr} = s + x_B y_A \]  \hspace{1cm} (5.3)

The value of \( s_{pr} \) obtained is the secret identity of an individual proxy signer; hence it is usually kept within its RAM and is obscured from other parties.

**Blind Signing**

Proxy signer, \( B \) randomly selects an integer \( k \in \mathbb{Z}_q \) and computes

\[ t = g^{k+s_B} (mod \, p) \]  \hspace{1cm} (5.4)

The tuple \( (r, t, m_w) \) is sent to the receiving OBU/RSU \( R \).

\( R \) checks \( A \)’s (i.e. RSC’s) and \( B \)’s identities and the delegation lifetime of the warrant \( m_w \). This checking helps to prevent the attacks such as message forgery, impersonation if somehow the proxy signer is compromised.

If the above checking is successful,

\( R \) selects two random numbers \( a, b \in \mathbb{Z}_q \) and computes

\[ r' = t, g^{a+x_R} y_{pr}^b (mod \, p) \]  \hspace{1cm} (5.5)

where \( y_{pr} = g^{s_{pr}} (mod \, p) \) is the public key for the proxy blind signature.

\[ e = H(r'||m) (mod \, q) \]  \hspace{1cm} (5.6)

\[ e^* = b - e (mod \, q) \]  \hspace{1cm} (5.7)

If \( r' = 0 \), then \( R \) needs to select a new tuple \( (a, b) \) otherwise, \( R \) sends \( e^* \) to proxy signer and \( CA \).
For signing blinded message, \( B \) must request a time stamp for the message.

When there is a message to be transported over the VANET, either for some road-safety application, or, for some other need (e.g. a commercial advertisement, weather update etc.), the RSC must supplement the message content \( m \) with a message expiry time \( t_s \). It is crucial for the VANET system to thwart the RSU from abusing the proxy blind signature by posting invalid messages, or replaying the old messages. The message \( m \) is thus jointly signed by the RSC and the subsequent RSUs before it is delivered to the vehicles on road.

RSC chooses a random number \( k_s \in \mathbb{Z}_q^* \) and computes

\[
\begin{align*}
    r_s &= g^{k_s (mod\ p)} \\
    T &= H(r_s || t_s || e^*) (mod\ p)
\end{align*}
\]

(5.8)
(5.9)

RSC sends \( T \) to the proxy signer \( B \) and receiver \( R \).

The proxy signer applies the gained \( e^* \) and \( T \) values to estimate the final signed message as

\[
s' = k + e^* s_{pr} + T
\]

(5.10)

The proxy blind signature \((m, s')\) can now be delivered to the receiver.

**Verification**

Upon acquiring \( s' \) from \( B \), the receiving node computes,

\[
s'^* = g^{a + s' - t} (mod\ p)
\]

(5.11)
Thus, the proxy blind signature on message $m$ is the tuple $(m, m_w, s^*, e)$. Verifier can now verify the proxy blind signature by checking whether

$$e = H(s^*y_B y_R y_{pr}^e || m) \pmod{q} \quad (5.12)$$

where $y_{pr} = g^{s_{pr}} \pmod{p}$ is the public key for the proxy blind signature.

**Revocation Phase**

Under any circumstances if the RSC wants to revoke the delegation before the specified delegation period, then that particular RSC looks up in its revocation list. The CA/DOT maintains the entire list of the revoked nodes (RSU/ OBU). On demand from the RSC, CA provides the revocation list from its repository. During the computation of $T$, the RSC checks the validity of delegation period specified in the proxy warrant $m_w$ and the revocation list. If it is within the valid delegation period and the proxy signer is not found in the revocation list, RSC computes $T$, sends it to proxy signer $B$ and receiver $R$ for the message. If $B$ is in the revocation list then RSC does not compute $T$. Hence, the proxy signer cannot sign the message. Also, suspicious node will be communicated to the CA. Soon after, the CA will update its revocation database which may be passed down to subsequent RSCs.

**System Accuracy**

Consider the verification equation given in Equation (5.12). The main component of the equation is $(s^* y_B y_R y_{pr}^e || m)$

Now, $s^* y_B y_R y_{pr}^e || m = s^* g^{x_B} g^{x_R} y_{pr}^e || m$

$$= g^{x_B + s^* e_{pr} + x_R + x_{pr} y_{pr}^e} || m$$

$$= g^{x_B + (b - e) s_{pr} + x_R + x_{pr} y_{pr}^e} || m$$
5.3.4 Overhead Calculation

Consider the proxy blind signature on the message \( m \), \((m_m, m_w, s^e, e)\). To calculate the signature payload on the message the quantities \( m_w, s^e, e \) are considered. The prime numbers \( p \) and \( q \) are of 512 and 140 bits respectively, the total size of the signature payload in the authentication of the proxy blind signature would amount to 102 bytes with the standard SHA-1 hash operation. Table 5.3 gives the overhead for the message authentication in VANET using proxy blind signatures.

**Table 5.3 Overhead using proxy blind signature**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Size (in Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_w )</td>
<td>20</td>
</tr>
<tr>
<td>( s^e )</td>
<td>64</td>
</tr>
<tr>
<td>( h )</td>
<td>18</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>102</strong></td>
</tr>
</tbody>
</table>

5.3.5 Investigation of the Scheme

The security of the proposed scheme relies mainly on the inherent difficulty of solving discrete logarithm problem of proxy signature scheme.
For the message signature, the proxy signer uses a new secret which is derived from the actual secret key of the original signer. The intractability of the discrete logarithm problem from proxy signature scheme assumes that an adversary can’t reverse the process to generate the actual secret from the knowledge of a proxy key.

The first part of the security analysis focuses on the secure RSU-to-OBU message delivery approach of the VANET, while in the next part the anonymous OBU message delivery has been analyzed.

RSU Message Delivery

False Message Injection: The original signer, RSC produces a message to be delivered to the OBUs while it allows its corresponding subordinate RSUs to sign on behalf of it. In proxy blind signature scheme over VANET, a new proxy tuple has to be generated and delivered to the proxy signer for every single new message. The RSU cannot voluntarily input malevolent messages into the network. Since the key $x_A$ of the original signer (RSC) is attached to the secret key of the proxy blind signature $s_{pr}$, only the RSC can produce a valid proxy key pair for which the message will be accepted by an OBU.

There is a possibility that an adversary can successfully get a false message verified by an OBU. The probability for a false or modified message to be verified by an OBU is $1/(q - 1)$. The probability that an OBU will be deceived by a false message is $1/(p - 1)$. Hence, the overall chance that an OBU would be misled by an adversary is $(1/(p - 1) + 1/(q - 1))$. Therefore, the values for both $p$ and $q$ should be large enough in order to avoid such scenarios.

Unforgeability: Only a valid proxy-signer RSU can create a given signature on behalf of the RSC. The receiver OBU cannot forge the signature after
receiving \((m, m_w, s^*, e)\) on message \(m\). When an adversary with the new modified message \(m'\) tries to forge a signature \((m', s^*, e')\) on message \(m'\), it must verify that the equation given below is correct.

\[
s^*y_{B}y_{R}y_{pr}' (mod\ p) = tg^{a+x_{R}y_{pr}'} (mod\ p) \tag{5.13}
\]

By using the Equations (5.4) to (5.8)

\[
s^*y_{B}y_{R}y_{pr}' (mod\ p) = g^{a+s'-t} g^{x_{B}g^{x_{R}y_{pr}'}} (mod\ p) \tag{5.14}
\]

\[
= g^{a+s'-t+x_{B}+x_{R}e'} (mod\ p) \tag{5.15}
\]

\[
= tg^{a+(v-e)s_{pr}+x_{R}g^{e's_{pr}}} (mod\ p) \tag{5.16}
\]

\[
= tg^{a} + x_{R}y_{pr}' \tag{5.17}
\]

From the above,

\[
g^{(b-e)s_{pr}} g^{s_{pr}e'} (mod\ p) = g^{b_{pr}} (mod\ p) \tag{5.18}
\]

This cannot hold true, as \(e \neq e'\). Therefore the OBU fails to forge a valid proxy blind signature on message \(m'\).

**Non-repudiation and Impersonation:** As a RSU is strictly assigned to only one proxy, it cannot generate any valid proxy signature which would not be identified as a signature of only that particular RSU. The \(s_{pr}\) value of a valid signature for a given session is unique and can only be generated by a particular RSU. An adversary cannot generate a valid proxy signature from the public parameters, since \(s_{pr}\), the derived secret key of the proxy blind signature is dependent on the RSC’s private key.

Even if an adversary succeeds in generating a new proxy key pair \((s_{pr}', y_{pr}')\) launching an impersonation attack is not possible, since a malicious
RSU cannot provide its exact identity to the receiver with a considerable probability for computing a valid $e$ using Equation (5.6) in the stipulated time $t_x$ for the message $m$.

Due to the inclusion of the original signer and proxy signer identities information, message type to be signed by the proxy signer, delegation period, etc. in the warrant itself, it is capable of preventing proxy key pair misuse.

**Revocation:** An adversary may successfully compromise a RSU to get the possession of its designated proxy. Upon detection of the compromise, the RSC must revoke the proxy as the adversary may attempt to use the proxy to sign a malicious message. The revocation process starts at the RSC when it informs the CA about the corrupt node in the network.

Although, the compromised proxy is still a valid one and can be used by the adversary, it cannot harm the system by signing an illegitimate or expired message. This is due to the fact that $e$ requiring the original message $m$ itself, the expiry information $t_x$ and the primary secret $\chi_A$ generated only by the RSC. Nevertheless, the misbehaving RSUs must be replaced once identified, after conducting an investigation by the VANET administrator.

**OBU Message Delivery**

Some of the security issues concerning the proxy blind signature scheme for OBU message broadcasts in VANET are:

**Anonymity:** At the time of registration/license renewal an OBU is preloaded with $n$ different delegations. The size of $n$ is important for the vehicle’s anonymity and may vary according to the owner’s preference of privacy. The OBU uses one of them while sending a new message in VANET. This proxy
is chosen randomly from the preloaded set of proxies. Thus, the original identity of the vehicle is not exposed to other parties during the message communication. Generally, the proxy blind signature is un-linkable at the receiving end which provides an adaptive anonymity and privacy to a VANET user while the original MAC address of the sender is also undisclosed as indicated by the standards. Thus, the original identity of the vehicle is not exposed to other entities during an OBU message transmission.

**Accountability:** Under a critical situation when it is necessary and permitted by the appropriate law enforcement authorities, a vehicle’s identity can be traced by investigating a sent message. From the warrant $m_w$, anyone can mark original signer and proxy signer. On the other hand, as the verification equation contains the public key of the proxy signer and original signer, one can determine them. The message is reconstructed at the DOT using the identity assigned to that particular vehicle, the parameters, $m_w, t_x$ from the signed message and $s^t$ from the reporting RSU. While the reconstructed message matches, the complete identity of the vehicle is retrieved by the DOT.

**False Message Injection:** A malicious OBU may try to transmit a false or modified message $m'$ in the VANET. The only necessity for an adversary is to compute a valid public key $y_{pr}$ for the message $m'$. As $y_{pr}$ is modulo $p$ operation, the probability that the false (or, modified) message would get through is $1/(p - 1)$, meaning that a large (usually, at least 512 bit for a proxy signature) $p$ would be required.

**Replay Attacks:** A malicious party may attempt to replay a valid message at the same location where the signed message was originally delivered. However, the expiry information of the message is associated with the main message content which would make the signed message invalid once the
validity expires. As proxy blind signature requires a new proxy tuple to be generated securely delivered to the proxy signer for every single new message, replay attacks are impractical in this system.

**Node Compromise and Sybil Attacks:** An adversary may launch several useless and misleading messages to distract a VANET upon an OBU compromise. The malicious behaviour of a vehicle must be reported to the DOT as soon as identified. The DOT would release a revocation order for the tainted vehicle over the VANET if it is confirmed about the malicious act. It would then incorporate that vehicle in the revocation list which should be published to all the RSCs. Later the RSU generates an alert, so that the other vehicles can ignore the vehicle. This process would continue till the issue is resolved and the DOT further notifies the VANET about it.

A malicious vehicle may want to launch a Sybil attack where a vehicle sends out several identities usually to misdirect a VANET. To thwart such an attack, an entity would not be allowed to create or store pseudonymous identities.

### 5.4 ID-BASED PROXY BLIND SIGNATURE SCHEME USING ECPVS NOTATIONS

Tan (2013) has proposed an efficient identity-based pairing-free proxy blind signature scheme based upon the discrete logarithm assumptions in the random oracle model. This next section outlines the implementation details of the ID-based proxy blind signature scheme using ECPVS that consumes fully exponential time. Although ECDLP is one of the latest signature schemes on proposal, it has been found to consume less power and yet increase the probability of successful delivery of messages with the increase in number of recipients. This makes it useful in devices having small processors and low power.
The notations used in ECPVS-based proxy blind signature scheme certainly differs than a simple proxy blind signature mechanism. Thus, for a simplified illustration, Table 5.4 lists the notations used in this scheme.

**Table 5.4 Notations for proxy blind signature mechanism**

<table>
<thead>
<tr>
<th>Element Used</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RSC$</td>
<td>Original Signer</td>
</tr>
<tr>
<td>$OBU_i$</td>
<td>Proxy Signer - $OBU_i$ for communicating with $RSU_j$</td>
</tr>
<tr>
<td>$RSU$</td>
<td>Signature requester</td>
</tr>
<tr>
<td>$Q$</td>
<td>Public Key</td>
</tr>
<tr>
<td>$G$</td>
<td>Base Point</td>
</tr>
<tr>
<td>$H(\cdot)$</td>
<td>Hash Function</td>
</tr>
<tr>
<td>$s$</td>
<td>Original Signer’s Secret key</td>
</tr>
<tr>
<td>$y$</td>
<td>Original Signer’s Public key, $y = sQ$</td>
</tr>
<tr>
<td>$x_o$</td>
<td>Proxy Signer’s Secret key</td>
</tr>
<tr>
<td>$y_o$</td>
<td>Proxy Signer’s Public key, $y_o = x_oQ$</td>
</tr>
<tr>
<td>$m$</td>
<td>Message to be signed</td>
</tr>
<tr>
<td>$m_w$</td>
<td>Proxy warrant, which contains the identity information of the original signer and the proxy signer, validation periods of delegation, limits of authority.</td>
</tr>
</tbody>
</table>

### 5.5 MECHANISM

The ECPVS-based proxy signature mechanism comprises of four distinct phases namely, key setup, proxy allocation, blind signing and verification. The ID based proxy blind signature has been demonstrated in Figure 5.3.
Figure 5.3 ID-based proxy blind signature scheme

Key Setup Phase

i. During the key initialization phase, the CA chooses a random number $k_C$ such that $1 < k_C < q$ and computes:

$$R_C = k_C Q \quad (5.19)$$

i.e, $R_C(xR_C, yR_C)$

$$r_C = xR_C \mod q \quad (5.20)$$

$$s_C = s + k_C h(m_w \parallel r_C) \mod q \quad (5.21)$$

ii. The CA preloads $(R_C, s_C, m_w)$ into the disk of the OBU. By this way all the vehicles with the proxy blind signature scheme enabled gets different values of $R_C, s_C$ and $m_w$. 


Proxy Delegation Phase

i. Once when the OBU wants to blind sign the message, it splits the message into $c$ and $v$. It performs the symmetric encryption on $c$ to produce $e$ as:

$$e = T_c(c) \quad (5.22)$$

ii. For each vehicle OBU picks $k_o$ such that $1 < k_o < q$ to determine $R_o$ as:

$$R_o = k_o Q \quad (5.23)$$

iii. OBU computes proxy system’s secret key $s_p$

$$s_p = x_o + s_c \quad (5.24)$$

iv. The corresponding proxy public key is

$$y_p = y + y_o + R_c h(m_w || r_c) = s_p Q$$

$$y_p = s_p Q \quad (5.25)$$

Blind Signing Phase

i. OBU now sends $(R_o, y_p, m_w)$ to signature requester RSU.

The RSU chooses three blinding factors $a, b, c$ arbitrarily for ensuring a more robust security protection and computes

$$r = R_o + b Q - y_p (a + c) (mod q) \quad (5.26)$$
This step is known as blinding. If this equation results in 0, the RSU chooses the blinding factors again. Otherwise, it computes:

\[ h^* = H(r \| t) \]  \hspace{1cm} (5.27)
\[ h = h^* - c - a \]  \hspace{1cm} (5.28)

ii. Based on the hash value produced, OBU creates the signature as

\[ S' = (H(e) + s_\rho x_\rho) mod q \]  \hspace{1cm} (5.29)

where \( x_\rho \) is the session parameter computed as:

\[ (x_\rho, y_\rho) = h, k_\rho \ mod q \]  \hspace{1cm} (5.30)

iii. As RSU is prone to attacks, \( S' \) is sent to the signature requester RSU which then calculates

\[ S = S' + b \]  \hspace{1cm} (5.31)

iv. The final proxy blind signature on the message is \( (R_o, R_C, e, h^*, S, t) \).

**Verification Phase**

The receiver of the signature, generally RSC checks the legitimacy of the signature \( (R_o, R_C, e, h^*, S, t) \) by computing

\[ h_c = H(e \| t) \]

Finally if the following equation holds, then the signature is a valid one
\[(x_p, y_p) = (H(e)(RC + h_cQ) + x_p)S^{-1} \mod q \quad (5.32)\]

**Message Recovery**

i. The OBU finds \( u \) as

\[ u = s_i e G - h_{j,c}Q \quad (5.33) \]

ii. It recovers the confidential part of the message via inverse bijective transformation on \( e \) as

\[ c = T^{-1}_u(e) \quad (5.34) \]

iii. Perform redundancy test on \( c \). If successful, recover the plaintext or else discard the message.

The algorithm for anonymous authentication has been presented in Table 5.5. The complete signature process is shown in Figure 5.4

![Figure 5.4 Proxy blind signing procedure in VANET](image-url)
Table 5.5 Non-safety application messages authentication

<table>
<thead>
<tr>
<th>Algorithm 3. Message authentication using proxy blind signatures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I. Key Initialization (at OBU):</strong> Initialized with public parameters ( { n, y_o, Q } ) and secret parameters ( { x_o, k_C } ).</td>
</tr>
<tr>
<td>1: ( R_c = k_cQ )</td>
</tr>
<tr>
<td>2: ( r_c = xR_c \mod q )</td>
</tr>
<tr>
<td>3: ( s_c = s + k_c h(m_w | r_c) \mod q )</td>
</tr>
<tr>
<td><strong>II. Proxy Key Pair Generation (at OBU):</strong></td>
</tr>
<tr>
<td>1: if ( s_cQ = R_ch(m_w</td>
</tr>
<tr>
<td>2: ( S_p = x_o + s_c )</td>
</tr>
<tr>
<td>3: ( Y_p = QS_p )</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>4: ( \text{REJECT} )</td>
</tr>
<tr>
<td>5: ( \text{endif} )</td>
</tr>
<tr>
<td><strong>III. Preprocessing (at RSU_i):</strong></td>
</tr>
<tr>
<td>1: ( R_o = k_oQ )</td>
</tr>
<tr>
<td>2: ( r_o = xR_o )</td>
</tr>
<tr>
<td><strong>IV. Blinding (at RSU_i):</strong> Chooses three blinding factors ( a, b, c )</td>
</tr>
<tr>
<td>1: ( r = R_o + hQ - Y_p(a + c) (mod q) )</td>
</tr>
<tr>
<td>2: ( h^* = h(r</td>
</tr>
<tr>
<td><strong>V. Signature Generation:</strong></td>
</tr>
<tr>
<td>1: At OBU, ( S' = (h(e) + s_p x_p) \mod q )</td>
</tr>
<tr>
<td>2: At RSU_i, ( S = S' + b )</td>
</tr>
<tr>
<td><strong>VI. Verification (at RSC):</strong></td>
</tr>
<tr>
<td>1: if ( h^* = H((SQ - h^*y_p) \mod q</td>
</tr>
<tr>
<td>2: ( \text{Accept e} )</td>
</tr>
<tr>
<td>3: ( \text{else} )</td>
</tr>
<tr>
<td>4: ( \text{REJECT} )</td>
</tr>
<tr>
<td>5: ( \text{endif} )</td>
</tr>
<tr>
<td>6: ( u = s_{iE}G - h_{iC}Q )</td>
</tr>
<tr>
<td>7: ( c = T_u^{-1}(e) )</td>
</tr>
<tr>
<td>8: if redundancy checks on ( c = \text{valid} ) then</td>
</tr>
<tr>
<td>9: ( \text{Accept c} )</td>
</tr>
<tr>
<td>10: ( \text{else} )</td>
</tr>
<tr>
<td>11: ( \text{REJECT} )</td>
</tr>
<tr>
<td>12: ( \text{endif} )</td>
</tr>
</tbody>
</table>
5.6 INVESTIGATION OF THE PROPOSED SCHEME

Now, the prospective ECPVS-based proxy blind signature mechanism fulfils all the security requirements of proxy blind signature as stated under. This proves that the scheme is resistant to the various security attacks.

5.6.1 Signature Linking

When the signature \((R_0, R_C, e, h^*, S, t)\) is generated, the proxy signer \(OBU\) is only aware of the \((s_C, r_C, m_w, S', m)\). Further, \(OBU\) is permitted to provide the signature on \(RSC\)'s behalf when it receives the message \((r_C, s_C, m_w)\). Further, \(RSU\) or the signature requester takes help of three arbitrary secret keys \(a, b\) and \(c\) to turn the message blind. Therefore, unblinding can only happen via the requester, not through anybody else involved in the process. Because both the proxy and original signers are unaware of these three secret parameters, the scheme is said to satisfy the unlinkability property.

5.6.2 False Signing

Suppose an adversary desires to imitate the proxy signer for signing the message. Therefore, he forges a legitimate proxy blind signature \((R_0, R_C, e, h^*, S)\) to form \((R_0, R_C, e, h^*, S')\) such that it passes the verification equation \(h^* \equiv H((SQ - h^*y_p) \mod q \| e \mod q)\). But the adversary needs to solve \(S'\). It is hard to do so because it is difficult to crack ECDLP that is assumed to be non-viable. Although the adversary can capture the delegation information \((R_C, s_C, m_w)\), it is impossible to fetch \(S_p\), which is the proxy signature secret key. The scheme is resistant to forgery.
5.6.3 Signature is Distinguishable

In this scheme, message warrant $m_w$ is present in proxy blind signature $(m_w, r_c, m, e^*, S)$. Further, the public key of the actual as well as of the proxy signer is present in the proxy signature public key $Y_p = y + y_o + R_o h(m_w || r_c) = Q * S_p$. This means that both the proxy signature and normal signature are distinguishable. Because the proxy key differs from the actual signer’s private key, any proxy signature is discernible from the normal signature of original signer.

5.6.4 Identifiability

The proxy blind signature $(m_w, r_c, m, e^*, S)$ has the message warrant. Further, the verification equation $Y_p = y + y_o + R_c h(m_w || r_c)$ has the public key of actual as well as of the proxy signer. This indicates that it is possible to know about the corresponding signer from its proxy signature.

5.6.5 Non Repudiation

In the proposed scheme, the actual signer is unaware of the secret key of proxy signer. Further, at the same time, the proxy signer is unaware of the secret key of actual signer. Moreover, only the proxy signer knows $S_p$, if the proxy signer’s secret key is not compromised. Nobody else in the process can generate $S'$ by solving $S' = e S_p + k_o$, except for the proxy signer. Therefore, the proxy signer cannot deny the fact that it has signed the message on the original signer’s behalf.
5.6.6 Correctness of the Scheme

The verification equation used in the scheme to verify the proxy blind signature is \( h^* \equiv H((SQ - h^*y_p)mod q || e)mod q \). The precision of the signature is proved as follows:

\[
SQ - h^*y_p = (S^* + b)Q - h^*y_p \\
= (h* s_p + k_o)Q + bQ - h^*y_p \\
= h* s_p Q + k_o Q + bQ - h^*y_p \\
= y_p(h^* - c - a) + R_o + bQ - e^*y_p \\
= h^*y_p - cy_p - ay_p + R_o + bQ - h^*y_p \\
= R_o + bQ - y_p(a + c) \\
= r
\]

5.6.7 Key Pair Misuse

The proposed scheme has the ability to prevent the misuse of proxy key pair. As mentioned before, the warrant \( m_w \) contains the identity information of both the original as well as of the proxy signer, delegation duration and the type of message that a proxy signer can sign. By using the proxy key, the proxy signer is unable to sign a message that the actual signer has not allowed. Therefore, it is proved that the proxy signer cannot use the proxy key to sign any message but only to sign those messages that the original signer has permitted.
5.7 COMPUTATIONAL OVERHEAD

The security of the scheme is stronger because ECDLP is much more difficult than DLP. The final signature components in the formulated ID-based proxy blind signature is \((R_o, R_c, e, h^*, S, t)\). In P-256 curve, the payload of signature is 32 bytes. Each of the three components is of 32 bytes, whereas \(t\) is of 8 bytes. Because there is no need of a third-party certificate for verification, the total length of the message is reduced by subtracting 126 bytes for certificate as well as the signer’s type field. Nevertheless, \(R_o, h^*\) and \(t_c\) of 64 bytes adds to the overhead (32, 24 and 8 bytes respectively). Thus, the overall size of the message payload is reduced to 168 bytes that is lesser than the ECDSA proxy signature scheme wherein the WSM length is of 184 bytes.

5.8 SIMULATION RESULTS

The simulation area is set to 500m x 100m on a bi-directional road with four lanes. The number of nodes can range between 10 and 100. In order to predict the path loss along the link the radio propagation model 2-ray ground has been used. The MAC protocol IEEE 1609.2P WAVE for security and management of applications and messages has been considered. The speed of vehicles in the network ranges from 10 to 100 km/hr with the transmission rate of 6 Mbps.

The simulation has been carried out for about 200 seconds. The average outcome of 10 samples with different random seeds for each experiment is taken. The results produced are plotted as shown in Figure 5.5. The message delivery ratio of ID based proxy signature is higher compared to the presented ID based proxy blind signature scheme. The signature overhead
of ID based proxy signature is lesser than ID based proxy blind signature scheme. Also, in a proxy blind signature scheme a new tuple has to be generated for every single new message.

Although the proxy blind signature requires a new tuple every time, the blinding property makes its use desirable in VANET. Without comparing it to the proxy signature scheme, this proxy blind signature mechanism performs relatively better than the conventional ECDSA scheme (Figure 5.6).

![Message Delivery Ratio using ID Based Proxy Blind Signature Scheme](image)

**Figure 5.5 Successful message delivery using proxy blind signature**

It can be clearly noted that when the vehicle population is lesser, the message delivery ratio for both the schemes are almost the same. As the number of participating nodes increases, the proxy blind signature scheme outperforms the conventional ECDSA based scheme.
5.9 SUMMARY

In the introduction of this chapter, the susceptibility of non-safety application messages to security attacks have been discussed in detail. These attacks can be kept at bay with the proxy blind signature mechanism. Such a scheme accomplishes the security requirements namely, distinguishability, non-repudiation, strong unforgeability, verifiability, identifiability and unlinkability. The simulation results have delineated the effectiveness of proxy blind signature based method. Albeit system is slower than the ID based proxy signature scheme, it is better the ID based proxy signature using ECDSA. Hence proxy blind signature based approaches can be effectively used in accomplishing VANET security.