# CHAPTER 5

# MODIFIED ELLIPTIC CURVE SIGNCRYPTION APPLICATIONS

## 5.1. Introduction

In this chapter, We will discuss the proposed applications on the Modified EC Signcryption:

1. Multi – Recipients
2. IPSec Modifications

## 5.2. Modified EC Signcryption for Multi- Recipients

Chapter 3 discussed Zheng's signcryption for Multiple Recipients (SCS1M and SCS2M) and computed saving in communication overhead and saving in computational cost with respect to the traditional approach based on DL and RSA. Now, applying the modified signcryption scheme for multiple recipients and compute saving in communication overhead and computational cost and compare it with Zheng's scheme for multi – recipients as show in the following stages.

Key generation:

Using the same EC parameters and public / private keys as mentioned in chapter 3. Alice needs to send message $m$ to $t$ number of recipients each one has its public and private keys ($P_i$ , $v_i$) where $P_i = v_i.G$

Signcryption:

1. Alice picks a random value $u \in_R [1, \ldots\ldots, q-1]$, compute the message encryption key $h = hash\,(m, u)$, the encrypted message $c = E_h(m)$
2. Pick a random value $v \in_R [1, \ldots\ldots, q-1]$ and calculates $R = (vG) = (x_R, y_R)$, calculate $r = hash\,(c, x_R)$
3. Create a signcrypted text of $h$ for each recipient $i = 1, \ldots\ldots, t$
   - Calculates key encryption to each recipient $k_1 = (v + x_R.v_a)P_1 = (x_{k_1}, y_{k_1})$, $k_2 = (v + x_R.v_a)P_2 = (x_{k_2}, y_{k_2})$, ..... $k_t = (v + x_R.v_a)P_t = (x_{k_t}, y_{k_t})$
   - $c_1 = E_{x_{k_1}}(h)$ , $c_2 = E_{x_{k_2}}(h)$ , ..... .... $c_t = E_{x_{k_t}}(h)$
4. Create a signature
   - Compute $s = (v_a.r - v)\ mod\ q$

Alice broadcasts to all recipients ( $c, c_1, \ldots\ldots\ldots\ldots\ldots, c_t, r, s$)

Note, each recipient can verify the signature of the message and use its private key to encrypt the own ciphertext to obtain the secure message encryption key and decrypt the message.

Unsigncryption at Recipient $R_i$

The signcrypted text $(c, c_1, \ldots\ldots, c_t, r, s)$ received through a broadcast channel, a recipient $R_i$'s private key $v_i$ where $1 \leq i \leq t$, Alice's public key $P_a$, $G$ and $p$.

1. Find out $(c, c_i, r, s)$ in $(c, c_1, \ldots\ldots, c_t, r, s)$.
2. Verify the signature
   - Compute $R = rP_a - sG = (x_R, y_R)$
   - Compute $r' = hash\,(c, x_R)$

   if $\begin{cases} r' = r & \text{Accept } c \text{ and complete decryption process} \\ r' \neq r & \text{Reject } c \end{cases}$
3. Unsigncrypt ciphertext $c_i$ to obtain message encryption key
   - Compute $k = v_i(R + x_R P_a) = (x_{k_i}, y_{k_i})$
   - $h = D_{x_{k_i}}(c_i)$
4. Decrypt the message $m = D_h(c)$

## 5.2.1. Saving in Communication Overhead

The comparison between the proposed scheme for multiple recipients with the signcryption scheme (SCS1M and SCS2M). As shown in table 5.1, saving by the proposed scheme in communication overhead for the sender reduce from $t.(|k| + |KH| + |q|) + |KH|$ to $t.|k| + 2|KH| + |q|$. So, the saving in communication overhead is more significant when compared with signcryption scheme (SCS1M and SCS2M).

**Table 5.1 Saving in Comm. Overhead of Signcryption for Multi-Recipients**

| Scheme | Communication Overhead |
|---|---|
| **Signcryption (SCS1M – SCS2M)** | $t.(\|k\| + \|KH\| + \|q\|) + \|KH\|$ |
| **Proposed Scheme** | $t.\|k\| + 2\|KH\| + \|q\|$ |

## 5.2.2. Saving in Computational Cost

When comparing the proposed scheme for multiple recipients with the signcryption scheme (SCS1M and SCS2M). As can be seen in table 5.2, saving by the proposed scheme in computational cost can be represented as the number of modular exponentiations, is approximately the same for the sender. But, for each recipient increase from 1.17 to 2.34, the operation is duplicated .

**Table 5.2 Saving in Comp. Cost of Signcryption for Multi-Recipients**

| Operation / Scheme | Participant | EXP | DIV | MUL | ADD | ENC | $ENC_{short}$ | DEC | $DEC_{short}$ | HASH |
|---|---|---|---|---|---|---|---|---|---|---|
| **Signcryption** | **Alice** | $t$ | $t$ | – | $t$ | 1 | $t$ | – | – | $2t$+1 |
| | **Recipient $R_i$** | 1.17 | – | 2 | – | – | – | 1 | 1 | 3 |
| **Proposed Scheme** | **Alice** | – | 1 | $t$+1 | $t$+1 | 1 | $t$ | – | – | 3 |
| | **Recipient $R_i$** | – | – | 2.34 | 1 | – | – | 1 | 1 | 2 |

Where

EXP : The number of Modular Exponentiations .

DIV : The number of Modular Divisions.

MUL : The number of Modular Multiplications.

ADD : The number of Modular Addition or Subtraction.

HASH : The number of One-way or Keyed Hash Operations.

ENC : The number of Encryptions using a private key cipher.

$ENC_{short}$ : The number of Short Encryptions using a private key cipher.

DEC : The number of Decryptions using a private key cipher.

$DEC_{short}$ : The number of Short Decryptions using a private key cipher.

Then, the modified EC signcryption for multi – recipients achieves the security attributes; unforgeability, confidentiality, integrity and non – repudiation in addition to forward security and public variability. Also, achieves saving in communication overhead higher than Zheng's signcryption scheme for multiple recipients (SCS1M and SCS2M) but on the other side, the computational cost is doubled for each recipient compared with Zheng's signcryption scheme for multiple recipients (SCS1M and SCS2M).

## 5.3. Modified EC Signcryption on IPSec

As shown in chapter 1 IPSec is an open source mechanism consists of some protocols work together to ensure the security and integrity of transmitted data between two points over internet and WAN. IPSec depends on two main protocols to encrypt and authenticate the transmitted data; AH which responsible for the signature of the message using hash function SHA-1or MD, ESP which responsible for encrypt data using cryptographic algorithms (Symmetric algorithms – Asymmetric algorithms) but in practice ESP uses symmetric algorithms; because it is easy for hardware implementation but its main problem is how to transfer the share secret key in a secure channel over internet (untrusted network). So, using asymmetric algorithms like Diffie – Hellman Asymmetric Key Exchange to transfer the share secret key of the used symmetric algorithm and in sometimes using RSA to encrypt data.

In practical, IPSec not with all features according to the network required; where using AH protocols only to authenticate the data without encrypted, using ESP protocol only to ensure the security and confidentiality of data without authenticated it, using AH & ESP for ensure the security, confidentiality, anti – replay, and authentication of transmitted data, or using ESP protocol to encrypt and authenticate the transmitted data. Applying the IPSec mechanism is managed according to the administrator and the network and data requirements.

Since, IPSec mechanism is more useful and needed in computer networks to transfer the urgent data with keeping confidentiality, integrity, Anti–replay, and origin authentication of transmitted data. Then to improve the security and performance of IPSec mechanism, improve the encryption algorithms to increase its security attributes and reduce computational cost and communication overhead.

When applying the modified signcryption scheme "*Modified Elliptic Curve Signcryption*" on IPSec mechanism, will benefit with its security attributes, and its bandwidth saving. Also, combing the signature and encryption processes into one step. The main advantage of the modified signcryption scheme, not depends on a certain encryption algorithm so, can be used symmetric encryption algorithm to keep the easiness of hardware implementation, and solve the problem of exchanging a share secret key of symmetric algorithms.

In the following, discuss how to apply the modified signcryption scheme on the IPSec mechanism to achieve the required features according to the requirements of system, where applying the modified signcryption scheme on IPSec for authentication only, for encryption only, or for both encryption and authentication and how the communication overhead will change in each case.

### 5.3.1. Modified EC Signcryption on IPSec for Authentication Only

When using the modified signcryption scheme on IPSec to only authenticate the transmitted data, the sender need to signature the data payload only by the following procedures:

1. Compute $R = (vG) = (x_R, y_R)$
2. Compute $r = hash\ (m, x_R)$
3. Compute $s = (v_a.r\ - v)\ mod\ q$

the sender transmits ( $r, s$ ) with the packet of data, then any one can verify the signature and authentication of packet.

*Verify the signature at the recipient:*

1. Compute $R = rP_a - sG = (x_R, y_R)$
2. Compute $r' = hash\ (m, x_R)$

if $\begin{cases} r' = r & \text{Packet authenticated} \\ r' \neq r & \text{Packet not authenticated} \end{cases}$

the recipient ease verify the signature and ensure the integrity of data where, no one have the sender's private key.

## 5.3.2. Modified EC Signcryption on IPSec for Encryption Only

When using the modified signcryption scheme on IPSec to only encrypt the transmitted data, the sender will create a session key encryption and use the any encryption algorithm as preferred by the following procedures:

1. Compute $R = (vG) = (x_R, y_R)$
2. Compute session key encryption $k = (v + x_R.v_a)P_b = (x_k, y_k)$
3. Encrypt the message $c = E_{y_k}[E_{x_k}(M)]$

the sender send $R$ with the ciphertext or encrypted data payload to enable the recipient to recover the session key encryption, and it is considered as a communication overhead.

*Decryption at the recipient:*

1. Compute $k = v_b(R + x_R P_a) = (x_k, y_k)$
2. Decrypt the message $M = D_{y_k}\left[D_{x_k}(c)\right]$

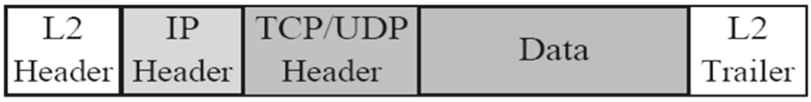
the modified signcryption scheme increase the confidentiality and security of the mechanism because any one to recover the session key encryption must have the recipient's private key or if he could have the sender's private key, he must find the corresponding random chosen value $v$.

### 5.3.3. Modified EC Signcryption on IPSec for Authentication and Encryption

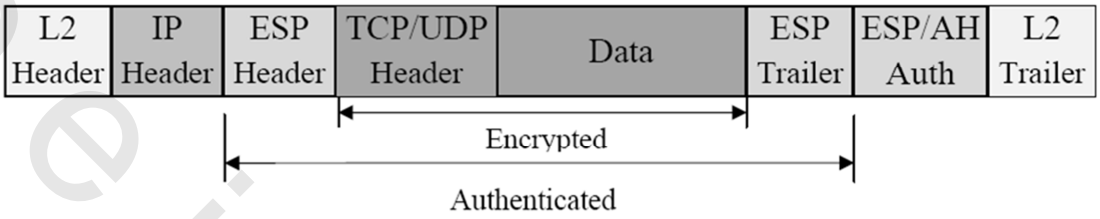
When using the modified signcryption scheme on IPSec to authenticate and encrypt the transmitted data, will be used the full procedures of the modified signcryption scheme as mentioned in chapter 4, and the sender will send only $(r, s)$ which enable the recipient to authenticate the transmitted data and recover the session encryption key to decrypt data. Although, $(r, s)$ are communication overhead but less than the communication overhead of the traditional approach of encryption – then – signature.

The following figures display the IP frame in the original form with the encapsulation of OSI model ( headers and trailer of 7 layers ), IPSec frame of transmitted data using AH / ESP headers and IPSec frame of transmitted data after applying the modified signcryption scheme for authentication and encryption of transmitted data in the transport mode and the tunnel mode.

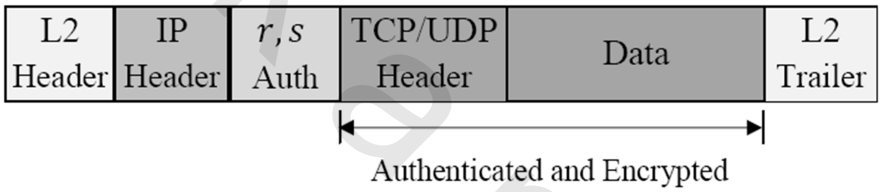
- Figure $(a)$ displays the original IP frame without applying IPSec mechanism, while the transmitted data is encapsulated by the headers of OSI model.
- Figure $(b)$ displays the IPSec frame in transport mode using ESP protocol to encrypt the transmitted data and AH / ESP protocol to authenticate the encrypted data.
- Figure $(c)$ displays IPSec frame in transport mode using the modified scheme to encrypt and authenticate transmitted data in one step reducing bandwidth and computational cost.
- Figure $(d)$ displays IPSec frame in tunnel mode using ESP protocol to encrypt data including original IP header and AH / ESP protocol to authenticate the encrypted data.
- Figure $(e)$ displays IPSec frame in tunnel mode using the modified scheme to encrypt and authenticate transmitted data including original IP header in one step reducing bandwidth and computational cost.



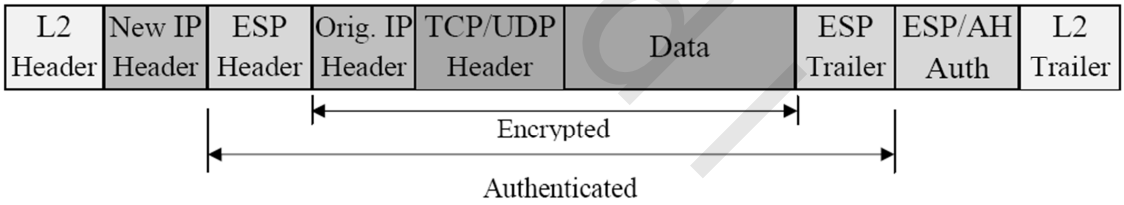

**(*a*) Original IP Frame**



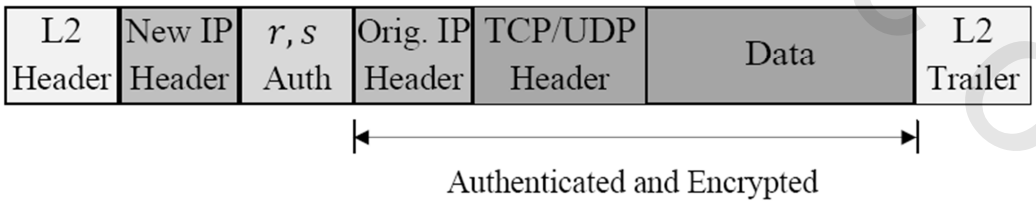

**(*b*) IPSec Frame using ESP and AH headers in Transport Mode**




**(*c*) IPSec Frame using Modified Signcryption in Transport Mode**




**(*d*) IPSec Frame using ESP and AH Headers in Tunnel Mode**




**(*e*) IPSec Frame using Modified Signcryption in Tunnel Mode**

**Figure 5.1 IPSec before and after Applying Modified Scheme**