

Interactive SMS

Plain HTTP Connection (Content_Subscription)

9/7/2011
v1.1



THIS DOCUMENT IS CONFIDENTIAL. IT IS INTENDED ONLY FOR THE INDIVIDUAL OR ENTITY TO WHICH IT IS ADDRESSED. DO NOT COPY, FORWARD, DISSEMINATE OR RELEASE TO THE PUBLIC OR DEFENDANTS THIS DOCUMENT OR ANY ATTACHMENTS WITHOUT OBTAINING PRIOR APPROVAL FROM THE SENDER. IF YOU ARE NOT THE NAMED RECIPIENT OR HAVE RECEIVED THIS DOCUMENT OR COMMUNICATION IN ERROR, PLEASE NOTIFY THE SENDER IMMEDIATELY BY EMAIL.

Table of Contents

Scope.....	3	2
Introduction	4	
Description.....	4	
Protocol and Parameters	5	
Samples.....	6	
ANNEXES	7	
Annex 1 – Operator IDs.....	8	
Annex 2 – MD5 encryption function.....	10	
VB.Net	10	
PHP	10	
Java.....	11	
VB 6.0.....	12	
Annex 3 –Actions description	20	
sub.....	20	
unsub.....	20	
Annex 4 – Convert from UCS2 to Arabic (or UTF8) function	21	
VB Script and VB.net :	21	
PHP 5	21	
Annex 5 – Convert from Arabic (or UTF8) to UCS2 function	22	
VB Script and VB.net:	22	
PHP 5.....	22	

Scope

This document has been designed for ACTEL's partners.

The partner could be a connection provider, a fellow VAS operator or simply a content provider owner willing to create subscription service.

Introduction

This document explains the messaging integration points between Actel's Large Scale Competition systems and the content owner's system. It goes into the details of system integration topology, connection protocols, Actel's requirements, and step by step guidelines for safe and successful integration.

Description

Connectivity to content provider messaging platform will be established by using Actel's HTTP API. Actel will provide aggregators with access to its gateway. The following diagram describes how the connection flow is arranged:

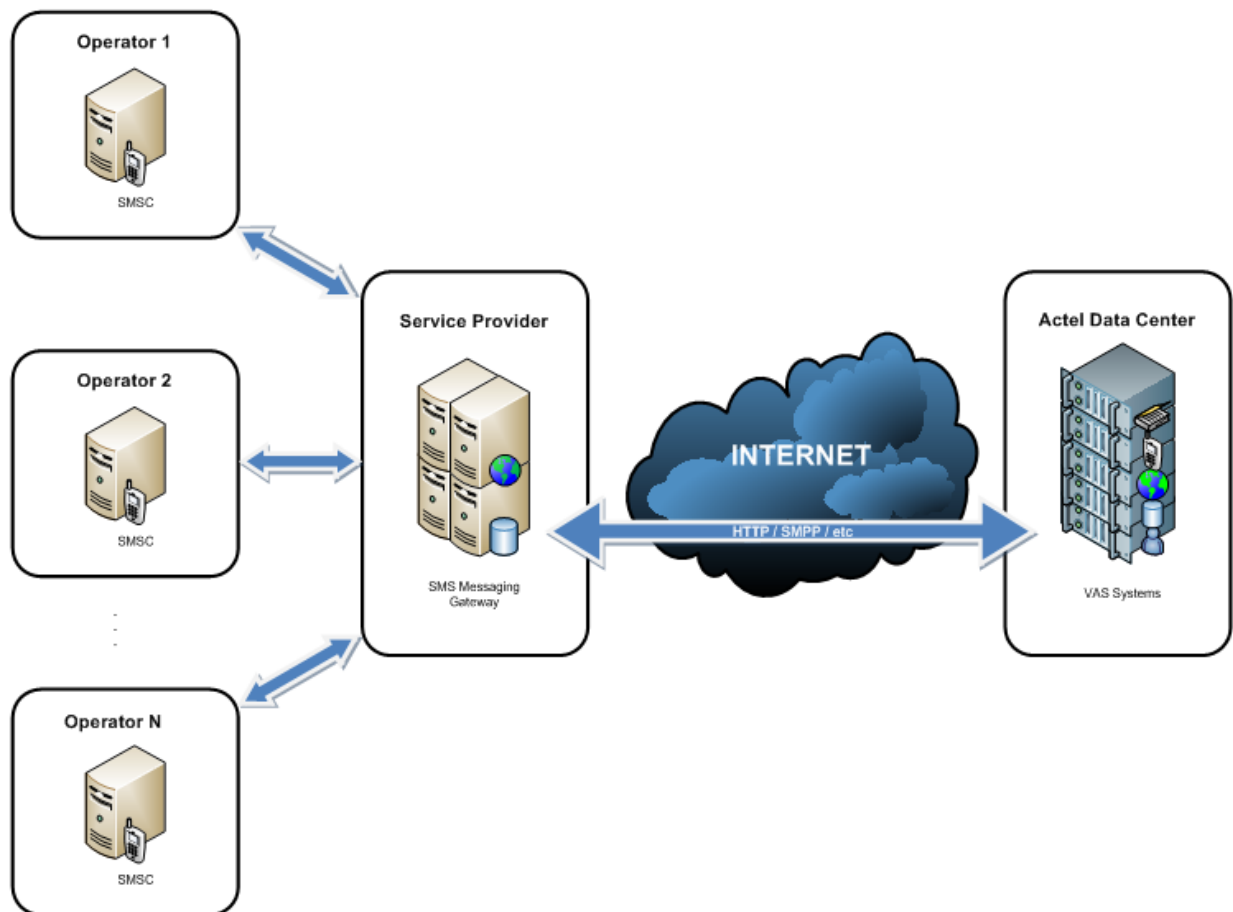


Figure 1 - Connection flow between Actel, Operators & Service Providers

Protocol and Parameters

Protocol	HTTP
URL	http://clients.actelme.com/mainsms/Subscription_Driver.aspx

HTTP Parameters		Type
RequestID	Unique ID	Alphanumeric String
UserName	Fix Alphanumeric value	Alphanumeric String
ServiceID	Fix numeric value	Integer
sender	Phone number	Integer
IDOperator	Operator ID refer to the Annex 1	Integer
signature	MD5 signature refer to the Annex 2	Alphanumeric String (Letters are Uppercase)
Action	refer to the Annex 3	Alphabetic String
idlang	0 for UCS2 (Arabic messages) , 1 for ACSII	Integer

Samples

Below are the acceptable request formats/structures including all valid HTTP parameters:

- 1) Valid format including all HTTP parameters for a new subscription request:

Text Request	http://clients.actelme.com/mainsms/Subscription_Driver.aspx?sender=9613793475&ServiceID=2075&RequestID=f5etytyefef&Action=sub&UserName=mbc!&idlang=0&IDOperator=2&signature=88AE708E98738416819790408C7A9914
---------------------	---

Return codes

Error Table		
Returned Values	Description	Type
Invalid sender	Invalid mobile number	Alphabetic String
Invalid ServiceID	Invalid or empty ServiceID	Alphabetic String
Invalid RequestID	Invalid or empty RequestID	Alphabetic String
Invalid Action	Empty message or invalid action	Alphabetic String
Invalid UserName	Invalid or empty username	Alphabetic String
Invalid IDoperator	Invalid or empty operator ID	Alphabetic String
Invalid signature	Invalid signature	Alphabetic String
Invalid Request, Duplicates	Duplicate content id posting, only it's change resolve this issue.	Alphabetic String
Invalid idlang	Invalid or empty language ID	Alphabetic String
ok	Message Sent Successfully	Alphabetic String

ANNEXES

Annex 1 - Operator IDs

opid	Operator Name	Country Name
1	Alfa	Lebanon
2	MTCTouch	Lebanon
3	Djezzy	Algeria
4	batelco	Bahrain
5	Zain	Bahrain
6	Mobinil	Egypt
7	Vodafone	Egypt
8	asiacell	Iraq
9	Iraqna	Iraq
10	Zain	Iraq
11	Zain	Jordan
12	Orange	Jordan
13	Umniah	Jordan
14	Xpress	Jordan
15	MOBILY	KSA
16	STC	KSA
17	Zain	kuwait
18	wataniya	kuwait
19	MAROCTEL	Morocco
20	MEDITEL	Morocco
21	Nawras	Oman
22	omanMobile	Oman
23	Jawal	Palestine
24	qtel	Qatar
25	Zain	Sudan
26	AREEBA	Syria
27	MTN	Syria
28	syriatel	Syria
29	Tunisiana	Tunisia
30	Etisalat	UAE
31	sabafon	yemen
32	MTN	yemen
33	yemenMobile	yemen
34	Etisalat	Egypt
41	DU	UAE
42	korectel	Iraq
43	Sanatel	Iraq
44	Smartcom	International
48	Tunistelecom	Tunisia

49	MTN	Sudan
50	Mobilis	Algeria
51	Nedjma	Algeria
52	Wana	Morocco
53	Libyana	Libya
54	elmadar	Libya
55	Etisaluna	Iraq
56	Sudatel	Sudan
57	Y	yemen
83	Zain	KSA
84	Jordan Telecom	Jordan
85	viva	kuwait
86	ZainIQ	Iraq
87	OmanTel	Oman
88	Zain	Palestine
92	Vodafone	Qatar
93	Wataniya	Palestine

Annex 2 – MD5 encryption function

DISCLAIMER: Please be advised that the provided code samples are provided for your convenience. Use them at your own risk.

VB.Net

```
Public Function MD5Encrypt(ByVal EncString As String) As String
    'Variable Declarations
    Dim MD5String As String
    Dim EncStringBytes() As Byte
    Dim Encoder As New UTF8Encoding
    Dim MD5Hasher As New MD5CryptoServiceProvider
    'Converts the String to bytes
    EncStringBytes = Encoder.GetBytes(EncString)
    'Generates the MD5 Byte Array
    EncStringBytes = MD5Hasher.ComputeHash(EncStringBytes)
    'Create MD5 hash
    MD5String = BitConverter.ToString(EncStringBytes)
    MD5String = MD5String.Replace("-", "")
    'Returns the MD5 encrypted string to the calling object
    Return MD5String
End Function
```

Usage

```
Signature=MD5Encrypt("ClientPassword@" & smssender)
```

PHP

```
$Signature= md5("clientpassword@".$smssender);
```

Java

```
import java.security.MessageDigest;

public class Main {
    public static void main(String[] args) {
        try{
            MessageDigest digest = java.security.MessageDigest.getInstance("MD5");
            //Put in here the String that you want to has its MD5 sig
            String sig = new String("actel");
            digest.update(sig.getBytes());
            byte[] hash = digest.digest();
            String result = "";
            for (int i=0; i < hash.length; i++) {
                result += Integer.toString( ( hash[i] & 0xff ) + 0x100, 16).substring( 1 );
            }
            System.out.println(result);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

VB 6.0

```
Private Const BITS_TO_A_BYTE = 8
Private Const BYTES_TO_A_WORD = 4
Private Const BITS_TO_A_WORD = 32
```

```
Private m_lOnBits(30)
Private m_l2Power(30)
```

```
m_lOnBits(0) = CLng(1)
m_lOnBits(1) = CLng(3)
m_lOnBits(2) = CLng(7)
m_lOnBits(3) = CLng(15)
m_lOnBits(4) = CLng(31)
m_lOnBits(5) = CLng(63)
m_lOnBits(6) = CLng(127)
m_lOnBits(7) = CLng(255)
m_lOnBits(8) = CLng(511)
m_lOnBits(9) = CLng(1023)
m_lOnBits(10) = CLng(2047)
m_lOnBits(11) = CLng(4095)
m_lOnBits(12) = CLng(8191)
m_lOnBits(13) = CLng(16383)
m_lOnBits(14) = CLng(32767)
m_lOnBits(15) = CLng(65535)
m_lOnBits(16) = CLng(131071)
m_lOnBits(17) = CLng(262143)
m_lOnBits(18) = CLng(524287)
m_lOnBits(19) = CLng(1048575)
m_lOnBits(20) = CLng(2097151)
m_lOnBits(21) = CLng(4194303)
m_lOnBits(22) = CLng(8388607)
m_lOnBits(23) = CLng(16777215)
m_lOnBits(24) = CLng(33554431)
m_lOnBits(25) = CLng(67108863)
m_lOnBits(26) = CLng(134217727)
m_lOnBits(27) = CLng(268435455)
m_lOnBits(28) = CLng(536870911)
m_lOnBits(29) = CLng(1073741823)
m_lOnBits(30) = CLng(2147483647)
```

```
m_l2Power(0) = CLng(1)
m_l2Power(1) = CLng(2)
m_l2Power(2) = CLng(4)
m_l2Power(3) = CLng(8)
m_l2Power(4) = CLng(16)
m_l2Power(5) = CLng(32)
m_l2Power(6) = CLng(64)
m_l2Power(7) = CLng(128)
```

```
m_l2Power(8) = CLng(256)
m_l2Power(9) = CLng(512)
m_l2Power(10) = CLng(1024)
m_l2Power(11) = CLng(2048)
m_l2Power(12) = CLng(4096)
m_l2Power(13) = CLng(8192)
m_l2Power(14) = CLng(16384)
m_l2Power(15) = CLng(32768)
m_l2Power(16) = CLng(65536)
m_l2Power(17) = CLng(131072)
m_l2Power(18) = CLng(262144)
m_l2Power(19) = CLng(524288)
m_l2Power(20) = CLng(1048576)
m_l2Power(21) = CLng(2097152)
m_l2Power(22) = CLng(4194304)
m_l2Power(23) = CLng(8388608)
m_l2Power(24) = CLng(16777216)
m_l2Power(25) = CLng(33554432)
m_l2Power(26) = CLng(67108864)
m_l2Power(27) = CLng(134217728)
m_l2Power(28) = CLng(268435456)
m_l2Power(29) = CLng(536870912)
m_l2Power(30) = CLng(1073741824)
```

Private Function LShift(IValue, iShiftBits)

 If iShiftBits = 0 Then

 LShift = IValue

 Exit Function

 Elseif iShiftBits = 31 Then

 If IValue And 1 Then

 LShift = &H80000000

 Else

 LShift = 0

 End If

 Exit Function

 Elseif iShiftBits < 0 Or iShiftBits > 31 Then

 Err.Raise 6

 End If

 If (IValue And m_l2Power(31 - iShiftBits)) Then

 LShift = ((IValue And m_lOnBits(31 - (iShiftBits + 1))) * m_l2Power(iShiftBits)) Or &H80000000

 Else

 LShift = ((IValue And m_lOnBits(31 - iShiftBits)) * m_l2Power(iShiftBits))

 End If

End Function

Private Function RShift(IValue, iShiftBits)

 If iShiftBits = 0 Then

```
RShift = IValue
Exit Function
Elseif iShiftBits = 31 Then
  If IValue And &H80000000 Then
    RShift = 1
  Else
    RShift = 0
  End If
  Exit Function
Elseif iShiftBits < 0 Or iShiftBits > 31 Then
  Err.Raise 6
End If

RShift = (IValue And &H7FFFFFFE) \ m_l2Power(iShiftBits)

If (IValue And &H80000000) Then
  RShift = (RShift Or (&H40000000 \ m_l2Power(iShiftBits - 1)))
End If
End Function

Private Function RotateLeft(IValue, iShiftBits)
  RotateLeft = LShift(IValue, iShiftBits) Or RShift(IValue, (32 - iShiftBits))
End Function

Private Function AddUnsigned(IX, IY)
  Dim IX4
  Dim IY4
  Dim IX8
  Dim IY8
  Dim IResult

  IX8 = IX And &H80000000
  IY8 = IY And &H80000000
  IX4 = IX And &H40000000
  IY4 = IY And &H40000000

  IResult = (IX And &H3FFFFFFF) + (IY And &H3FFFFFFF)

  If IX4 And IY4 Then
    IResult = IResult Xor &H80000000 Xor IX8 Xor IY8
  Elseif IX4 Or IY4 Then
    If IResult And &H40000000 Then
      IResult = IResult Xor &HC0000000 Xor IX8 Xor IY8
    Else
      IResult = IResult Xor &H40000000 Xor IX8 Xor IY8
    End If
  Else
    IResult = IResult Xor IX8 Xor IY8
  End If
End Function
```

```
End If

AddUnsigned = IResult
End Function

Private Function F(x, y, z)
    F = (x And y) Or ((Not x) And z)
End Function

Private Function G(x, y, z)
    G = (x And z) Or (y And (Not z))
End Function

Private Function H(x, y, z)
    H = (x Xor y Xor z)
End Function

Private Function I(x, y, z)
    I = (y Xor (x Or (Not z)))
End Function

Private Sub FF(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(F(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub

Private Sub GG(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(G(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub

Private Sub HH(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(H(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub

Private Sub II(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(I(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub

Private Function ConvertToWorldArray(sMessage)
    Dim IMessageLength
    Dim INumberOfWords
```

```

Dim IWordArray()
Dim IBytePosition
Dim IByteCount
Dim IWordCount

Const MODULUS_BITS = 512
Const CONGRUENT_BITS = 448

IMessageLength = Len(sMessage)

INumberOfWords = (((IMessageLength + ((MODULUS_BITS - CONGRUENT_BITS) \ BITS_TO_A_BYTE)) \
(MODULUS_BITS \ BITS_TO_A_BYTE)) + 1) * (MODULUS_BITS \ BITS_TO_A_WORD)
ReDim IWordArray(INumberOfWords - 1)

IBytePosition = 0
IByteCount = 0
Do Until IByteCount >= IMessageLength
    IWordCount = IByteCount \ BYTES_TO_A_WORD
    IBytePosition = (IByteCount Mod BYTES_TO_A_WORD) * BITS_TO_A_BYTE
    IWordArray(IWordCount) = IWordArray(IWordCount) Or LShift(Asc(Mid(sMessage, IByteCount + 1,
1)), IBytePosition)
    IByteCount = IByteCount + 1
Loop

IWordCount = IByteCount \ BYTES_TO_A_WORD
IBytePosition = (IByteCount Mod BYTES_TO_A_WORD) * BITS_TO_A_BYTE

IWordArray(IWordCount) = IWordArray(IWordCount) Or LShift(&H80, IBytePosition)

IWordArray(INumberOfWords - 2) = LShift(IMessageLength, 3)
IWordArray(INumberOfWords - 1) = RShift(IMessageLength, 29)

ConvertToWorldArray = IWordArray
End Function

Private Function WordToHex(IValue)
    Dim IByte
    Dim ICount

    For ICount = 0 To 3
        IByte = RShift(IValue, ICount * BITS_TO_A_BYTE) And m_lOnBits(BITS_TO_A_BYTE - 1)
        WordToHex = WordToHex & Right("0" & Hex(IByte), 2)
    Next
End Function

Public Function MD5(sMessage)
    Dim x
    Dim k

```


Dim AA
Dim BB
Dim CC
Dim DD
Dim a
Dim b
Dim c
Dim d

Const S11 = 7
Const S12 = 12
Const S13 = 17
Const S14 = 22
Const S21 = 5
Const S22 = 9
Const S23 = 14
Const S24 = 20
Const S31 = 4
Const S32 = 11
Const S33 = 16
Const S34 = 23
Const S41 = 6
Const S42 = 10
Const S43 = 15
Const S44 = 21

x = ConvertToWordArray(sMessage)

a = &H67452301
b = &HEFCDAB89
c = &H98BADCFE
d = &H10325476

For k = 0 To UBound(x) Step 16

AA = a
BB = b
CC = c
DD = d

FF a, b, c, d, x(k + 0), S11, &HD76AA478
FF d, a, b, c, x(k + 1), S12, &HE8C7B756
FF c, d, a, b, x(k + 2), S13, &H242070DB
FF b, c, d, a, x(k + 3), S14, &HC1BDCEEE
FF a, b, c, d, x(k + 4), S11, &HF57C0FAF
FF d, a, b, c, x(k + 5), S12, &H4787C62A
FF c, d, a, b, x(k + 6), S13, &HA8304613
FF b, c, d, a, x(k + 7), S14, &HFD469501
FF a, b, c, d, x(k + 8), S11, &H698098D8

FF d, a, b, c, $x(k + 9)$, S12, &H8B44F7AF
FF c, d, a, b, $x(k + 10)$, S13, &HFFFF5BB1
FF b, c, d, a, $x(k + 11)$, S14, &H895CD7BE
FF a, b, c, d, $x(k + 12)$, S11, &H6B901122
FF d, a, b, c, $x(k + 13)$, S12, &HFD987193
FF c, d, a, b, $x(k + 14)$, S13, &HA679438E
FF b, c, d, a, $x(k + 15)$, S14, &H49B40821

GG a, b, c, d, $x(k + 1)$, S21, &HF61E2562
GG d, a, b, c, $x(k + 6)$, S22, &HC040B340
GG c, d, a, b, $x(k + 11)$, S23, &H265E5A51
GG b, c, d, a, $x(k + 0)$, S24, &HE9B6C7AA
GG a, b, c, d, $x(k + 5)$, S21, &HD62F105D
GG d, a, b, c, $x(k + 10)$, S22, &H2441453
GG c, d, a, b, $x(k + 15)$, S23, &HD8A1E681
GG b, c, d, a, $x(k + 4)$, S24, &HE7D3FBC8
GG a, b, c, d, $x(k + 9)$, S21, &H21E1CDE6
GG d, a, b, c, $x(k + 14)$, S22, &HC33707D6
GG c, d, a, b, $x(k + 3)$, S23, &HF4D50D87
GG b, c, d, a, $x(k + 8)$, S24, &H455A14ED
GG a, b, c, d, $x(k + 13)$, S21, &HA9E3E905
GG d, a, b, c, $x(k + 2)$, S22, &HFCEFA3F8
GG c, d, a, b, $x(k + 7)$, S23, &H676F02D9
GG b, c, d, a, $x(k + 12)$, S24, &H8D2A4C8A

HH a, b, c, d, $x(k + 5)$, S31, &HFFFA3942
HH d, a, b, c, $x(k + 8)$, S32, &H8771F681
HH c, d, a, b, $x(k + 11)$, S33, &H6D9D6122
HH b, c, d, a, $x(k + 14)$, S34, &HFDE5380C
HH a, b, c, d, $x(k + 1)$, S31, &HA4BEEA44
HH d, a, b, c, $x(k + 4)$, S32, &H4BDECFA9
HH c, d, a, b, $x(k + 7)$, S33, &HF6BB4B60
HH b, c, d, a, $x(k + 10)$, S34, &HBEBFBC70
HH a, b, c, d, $x(k + 13)$, S31, &H289B7EC6
HH d, a, b, c, $x(k + 0)$, S32, &HEAA127FA
HH c, d, a, b, $x(k + 3)$, S33, &HD4EF3085
HH b, c, d, a, $x(k + 6)$, S34, &H4881D05
HH a, b, c, d, $x(k + 9)$, S31, &HD9D4D039
HH d, a, b, c, $x(k + 12)$, S32, &HE6DB99E5
HH c, d, a, b, $x(k + 15)$, S33, &H1FA27CF8
HH b, c, d, a, $x(k + 2)$, S34, &HC4AC5665

II a, b, c, d, $x(k + 0)$, S41, &HF4292244
II d, a, b, c, $x(k + 7)$, S42, &H432AFF97
II c, d, a, b, $x(k + 14)$, S43, &HAB9423A7
II b, c, d, a, $x(k + 5)$, S44, &HFC93A039
II a, b, c, d, $x(k + 12)$, S41, &H655B59C3
II d, a, b, c, $x(k + 3)$, S42, &H8F0CCC92

```
ll c, d, a, b, x(k + 10), S43, &HFFEFF47D
ll b, c, d, a, x(k + 1), S44, &H85845DD1
ll a, b, c, d, x(k + 8), S41, &H6FA87E4F
ll d, a, b, c, x(k + 15), S42, &HFE2CE6E0
ll c, d, a, b, x(k + 6), S43, &HA3014314
ll b, c, d, a, x(k + 13), S44, &H4E0811A1
ll a, b, c, d, x(k + 4), S41, &HF7537E82
ll d, a, b, c, x(k + 11), S42, &HBD3AF235
ll c, d, a, b, x(k + 2), S43, &H2AD7D2BB
ll b, c, d, a, x(k + 9), S44, &HEB86D391
```

```
a = AddUnsigned(a, AA)
```

```
b = AddUnsigned(b, BB)
```

```
c = AddUnsigned(c, CC)
```

```
d = AddUnsigned(d, DD)
```

```
Next
```

```
MD5 = UCase(WordToHex(a) & WordToHex(b) & WordToHex(c) & WordToHex(d))
End Function
```

Annex 3 –Actions description

20

sub

Subscribe a user

unsub

Unsubscribe a user

Annex 4 – Convert from UCS2 to Arabic (or UTF8) function

VB Script and VB.net :

```
Function Arabize(ByVal text As String) As String
    Try
        Dim thetext, char1, arabictext As String
        arabictext = ""
        thetext = text
        For i As Integer = 1 To Len(thetext) / 4
            char1 = Left(thetext, 4)
            arabictext = arabictext & ChrW(CInt("&H" & char1))
            thetext = Right(thetext, Len(thetext) - 4)
        Next
        Arabize = arabictext
    Catch ex As Exception
        Arabize = " invalid message"
    End Try
End Function
```

PHP 5

```
header('Content-Type: text/html; charset=UTF-8');
mb_http_output('UTF-8');
echo '<html><head>';
echo '<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />';
echo '</head><body>';

$UCS2 = "06230643062A064A064400200647064A0020062706440623064206480649";
$ucs2string = pack("H*" , $_REQUEST['ucs2']);
// $ucs2string = pack("H*" , $UCS2);
$utf8string = mb_convert_encoding($ucs2string , 'UTF-8' , 'UCS-2');
echo 'UTF8: '.$utf8string.'<br />';

echo '</body></html>';
```

Annex 5 – Convert from Arabic (or UTF8) to UCS2 function

VB Script and VB.net:

```

Public Function UniEnc(ByVal text As String) As String
    Dim i, j, count_char As Integer, nchar, hexval, arabicunicode As
String
    arabicunicode = ""
    For i = 1 To Len(text)
        count_char = count_char + 1
        nchar = Left(text, 1)
        hexval = Hex(AscW(nchar))
        If Len(hexval) < 4 Then
            For j = 1 To 4 - Len(hexval)
                hexval = "0" & hexval
            Next
        End If
        arabicunicode = arabicunicode & hexval
        text = Right(text, Len(text) - 1)
    Next
    UniEnc = arabicunicode
End Function

```

PHP 5

```

header('Content-Type: text/html; charset=UTF-8');
mb_http_output('UTF-8');
echo '<html><head>';
echo '<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />';
echo '</head><body>';

$bodyar = "الأقوى هي أكتيل";
$arr = unpack('H*hex', @iconv('UTF-8', 'UCS-2BE', $bodyar));
echo "UCS2 : ".$arr['hex'];
echo "<br><br>";

echo '</body></html>';

```