

| | |
|--|---------------------|
| Name of Experiment: Digital Signature | Exp No: MC14 |
| Background: Student should have a basic knowledge of C#. | |
| Summary: Digital Signature is mainly for the authentication of Message. People used to sign their digital data with digital signature that prove their authenticity during message transfer. This aspect of security is very much needed in present life. | |
| Learning Objective: Student is able to learn the way of authentication through digital signature. | |
| Target Platforms: This experiment is tested on Windows Phone Emulator and Nokia Lumia 800. | |
| <p>Procedure:</p> <p>Step1. Repeat the steps [1-4] as in experiment no MC1. [refer experiment Hello World]</p> <p>Step2. Drag and drop the UI controls as 3 Buttons, 2 Textboxes and 1 Textblock in the MainPage.xaml.</p> <p>Step3. Add new class by right click on the experiment name in the Solution Explore and named as Signature.cs.</p> <p>Step4. Add few namespaces as using System, using System.Security.Cryptography and using System.Text.</p> <p>Step5. Define two static methods as GetSignature() and VerifyHash() in class Signature.cs.[refer source code section]</p> <p>Step6. For each button, there is a button event handler defined in MainPage.xaml.cs.</p> <p>Step7. Define the body of each event handler inside the MainPage.xaml.cs.[refer source code section]</p> <p>Step8. Write pre-processors for Validation. [refer source code section]</p> <p>Step9. Save all changes made to the project by pressing ctrl + S.</p> <p>Step10. Press F5, to debug this experiment on the Windows Phone Emulator.</p> | |
| Source Code | Comments |
| <p>MainPage.xaml</p> <pre> <!--LayoutRoot is the root grid where all page content is placed--> <Grid x:Name="LayoutRoot" Background="Transparent"> <Grid.RowDefinitions> <RowDefinition Height="Auto"/> <RowDefinition Height="*/> </Grid.RowDefinitions> <!--TitlePanel contains the name of the </pre> | |

```

application and page title-->
    <StackPanel x:Name="TitlePanel" Grid.Row="0"
Margin="12,17,0,28">
        <TextBlock x:Name="ExperimentTitle"
Text="ExpNo MC14" TextAlignment="Right"
Style="{StaticResource PhoneTextNormalStyle}"/>
        <TextBlock x:Name="ApplicationTitle"
Text="Mobile Computing" Style="{StaticResource
PhoneTextNormalStyle}"/>
        <TextBlock x:Name="PageTitle"
Text="Digital Signature" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
    </StackPanel>

    <!--ContentPanel - place additional content
here-->
    <Grid x:Name="ContentPanel" Grid.Row="1"
Margin="12,0,12,0">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="Auto"/>
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <TextBlock x:Name="tbMessage"
Text="Message" Height="50" Margin="12" Grid.Row="1"
Grid.Column="0" Width="100"/>
        <TextBox x:Name="txtMessg" Height="72"
Grid.Row="1" Grid.Column="1" Width="330"
TextWrapping="Wrap" />
        <Button Content="GetHash"
Click="Button_Click" Grid.Row="2" Grid.Column="1"
Width="330" Height="72" Background="Black"/>
        <TextBox x:Name="txtHash" Height="127"
Grid.Row="3" Grid.ColumnSpan="2" TextWrapping="Wrap"
Width="448" BorderBrush="Green" Background="Black"
Foreground="White" FontSize="{StaticResource
PhoneFontSizeSmall}" />
        <Button x:Name="bVerify" Content="Verify"
Grid.Row="4" Grid.Column="1" Height="72"
Click="bVerify_Click"/>
        <Button x:Name="bReset" Content="Reset"
Grid.Row="4" Grid.Column="0" Height="72"
Click="bReset_Click"/>

    </Grid>
</Grid>

```

← ExpNo MC14(Experiment Title)

← Mobile Computing (Application Title)

← Digital Signature (PageTitle)

Signature.cs

```

using System;
using System.Security.Cryptography;
using System.Text;

namespace DigitalSignature

```

← Signature class

```

{
    public class Signature
    {
        public static string GetSignature(string
plaintxt){
            StringBuilder str = new StringBuilder();
            SHA1Managed sha1 = new SHA1Managed();
            byte[] hashValue =
sha1.ComputeHash(Encoding.Unicode.GetBytes(plaintxt));
            for (int counter = 0; counter <
hashValue.Length; counter++) {
                str.Append(hashValue[counter].ToString("X1"));
            }
            return str.ToString();
        }
        public static bool VerifyHash(string
plaintxt, string prevHashValue ) {
            string hashValue2 =
GetSignature(plaintxt);
            StringComparer strComparer =
StringComparer.OrdinalIgnoreCase;
            if (strComparer.Compare(hashValue2,
prevHashValue).Equals(0))
            {
                return true;
            }
            else {
                return false;
            }
        }
    }
}

```

←GetSignature Method Body

VerifyHash Method Body

MainPage.xaml.cs

```

using System.Windows;
using Microsoft.Phone.Controls;

namespace DigitalSignature
{
    public partial class MainPage :
PhoneApplicationPage
    {
        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender,
RoutedEventArgs e)
        {
            if(Validate()){
                string result =
Signature.GetSignature(txtMessg.Text);
                txtHash.Text = result;}
        }
    }
}

```

←GetHash button event handler

← Apply Validation

```

    }

    private void bVerify_Click(object sender,
RoutedEventArgs e)
    {
        string
prvHash=Signature.GetSignature(txtMessg.Text);
        bool res =
Signature.VerifyHash(txtMessg.Text,prvHash);
        if (res)
        {
            MessageBox.Show("Verified
Successfully");
        }
        else {
            MessageBox.Show("Unsuccessful");
        }
    }

    private void bReset_Click(object sender,
RoutedEventArgs e)
    {
        txtMessg.Text = string.Empty;
        txtHash.Text = string.Empty;
    }
    #region UI
    private bool Validate(){
    if(string.IsNullOrEmpty(txtMessg.Text)){
    MessageBox.Show("Please Enter the Message");
    return false;}
    return true;}
    #endregion
    }
}

```

← Verify button event handler

←Message Box

←Reset button event handler

← pre-processor for Validation

Screenshots:



Fig. No 1 Home Screen

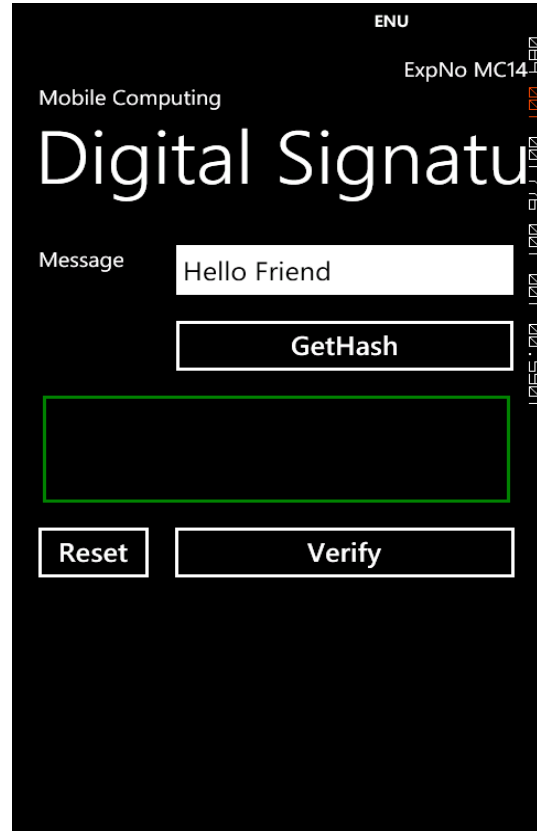


Fig. No 2 Message Typed

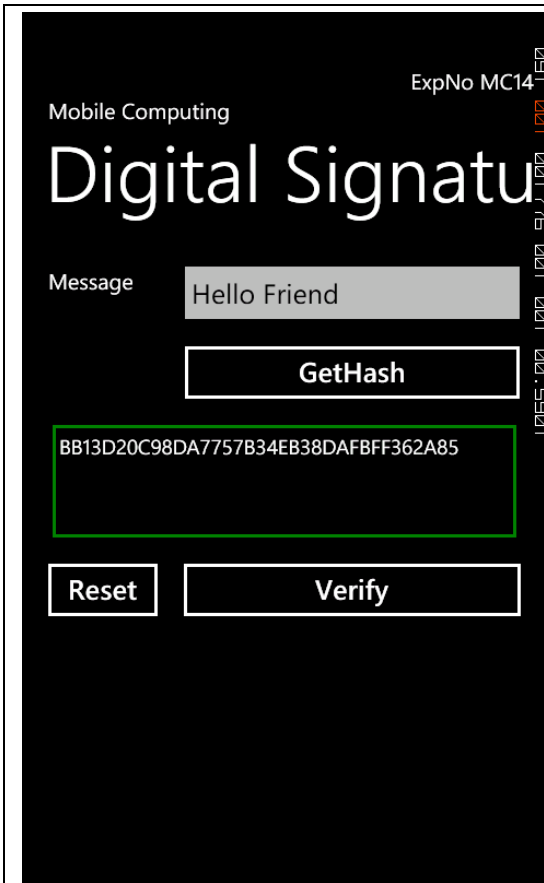


Fig.No 3 Hash Code for the Message

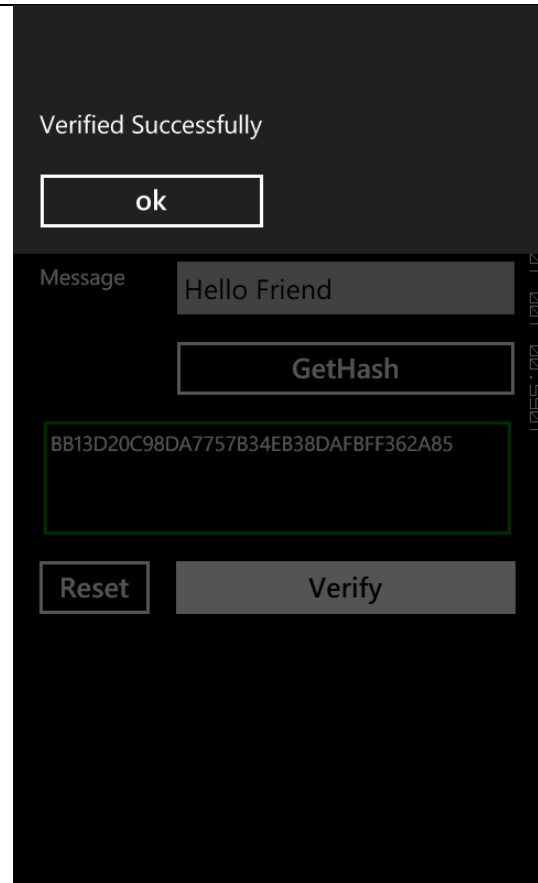


Fig. No 4 Verification of Hash Code

Observations: It is observed by developer that Digital Signature can be implemented in variety of application.