Developing a New Cryptic Communication Protocol by Quantum Tunnelling over Classic Computer Logic

Mesut KAVAK*

I have been working for a time about basic laws of directing the universe [1,2]. It seems that the most basic and impressive principle which causes any physical phenomenon is the Uncertainty Principle of Heisenberg [3], that existence have any property because of the uncertainty. During this process, while I was thinking about conservation of information I noticed, that information cannot be lost; but at a point, it becomes completely unrecognizable according to us as there is no alternative. Any information and the information searched for become the same after a point relatively to us. The sensitivity increases forever but its loss. Each sensitivity level also has higher level; so actually an absolute protection seems possible.

1 Introduction

In accordance with the uncertainty, any measurable physical value of matter increases or decreases between absoluteness and absolute absence but its loss. No physical value can take certain value. They can only be expressed as some approaches.

As a result of this condition, for example, if you want to measure process number of a CPU, you can only get some different numbers at least for the numbers after the comma each time of measurement. When ever you want to measure more sensitive, new different numbers emerge after the fixed numbers after the comma. Except this, you can never get a certain number. Additionally for example, if you make a division, for the same operation the processor always performs the operation in different times. Yes it performs by some packets like 32 or 64 and always gives to you the same number that actually this also is not exact, it does not perform at the same time for each repeat. If you set the interval to find out this and if you ask to it over a program, sometimes you can get some results like 33 or 65 by unpredictable changing frequencies like 1 of 50 or 20 for 1 second.

Actually this physical principle and condition can be used as a perfect unpredictable cyberspace security element; because it creates a way to select incalculable and unpredictable codes like the below.

2 Cryptobits

Let us use rounding the decimal numbers into closest one. For example, if the number is like x,xxxx6 it is rounded into x,xxxxx and this condition is called as 1. If the number is like x,xxxx4 then it is rounded into x,xxxx and the condition is called as 0. If the number is like x,xxxx5 try again. These 0 and 1 are not bits.

Also there is another way from many other ways that I used this second one. For example while a timer performs countdown, ask the time of counting until a number. If you repeat again and again the countdown, many times you get the same number but some times you get different number which is closest number to the actual number. The following VB codes are for this.

```
'Just use 3 text boxes, 2 timers and 1 button, and then copy the
      code below. It counts from 1000 down to 998 by the interval
      of 1. Textbox3.Text is set as 3 by the interval of 10. You
      can change them; but over these values, Textbox3.Text
      becomes 1 many times but sometimes becomes 2 if you repeat
      the same operation by the same button. You can use any
      frequency for the repeats over the same button. Especially
      use long interval clicks to be sure. You can also generate
      auto-repeat program. You can download the following program
      called Single Counter.exe from
      https://zenodo.org/record/1450385#.W7iZanszaM8
Public Class Form1
    Private Sub Timer1_Tick(sender As Object, e As EventArgs)
        Handles Timer1.Tick
        TextBox1.Text = Val(TextBox1.Text) - 1
        If TextBox1.Text = Val(TextBox2.Text) Then
            Timer1.Enabled = False
            Timer2.Enabled = False
        End If
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs)
        Handles Button1.Click
        TextBox1.Text = 1000
        TextBox2.Text = 998
        TextBox3.Text = 3
        Timer1.Interval = 1
        Timer2.Interval = 10
        Timer1.Start()
        Timer2.Start()
    End Sub
    Private Sub Timer2_Tick(sender As Object, e As EventArgs)
        Handles Timer2.Tick
        TextBox3.Text = Val(TextBox3.Text) - 1
    End Sub
End Class
```

If you do not want to repeat more and thus not to wait, repeat the same event more at the same time like following program. If you increase the number of the counters, probability is going to increase.

```
'Just use 18 text boxes, 7 timers and 1 button, and then copy
      the code below. It counts from 1000 down to 997 by the
      interval of 1 over Timer1. Textbox3.Text is set as 3 by the
      interval of 10. You can change them; but over these values,
      Textbox3.Text and the other textboxes has numbers 6,9,12,15
      and 18 become 0 many times but sometimes become 1 if you
      repeat the same operation by the same button. Also they do
      not get the same numbers together at the same time. You can
      use any frequency for the repeats over the same button.
      Especially use long interval clicks to be sure. You can
      also generate auto-repeat program. If you do not want to
      wait and want certain results, you can increase the number
      of textboxes by the same rule. You can download the
      following program called Large Counter.exe from
      https://zenodo.org/record/1450385#.W7iZanszaM8
Public Class Form1
    Private Sub Timer1_Tick(sender As Object, e As EventArgs)
        Handles Timer1.Tick
        TextBox1.Text = Val(TextBox1.Text) - 1
        TextBox4.Text = Val(TextBox4.Text) - 1
        TextBox7.Text = Val(TextBox7.Text) - 1
        TextBox10.Text = Val(TextBox10.Text) - 1
        TextBox13.Text = Val(TextBox13.Text) - 1
        TextBox16.Text = Val(TextBox16.Text) - 1
        If TextBox1.Text = Val(TextBox2.Text) Then
            Timer1.Enabled = False
            Timer2.Enabled = False
            Timer3.Enabled = False
            Timer4.Enabled = False
            Timer5.Enabled = False
            Timer6.Enabled = False
            Timer7.Enabled = False
        End If
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs)
        Handles Button1.Click
        TextBox3.Text = 3
        TextBox1.Text = 1000
        TextBox2.Text = 997
        TextBox6.Text = 3
        TextBox4.Text = 1000
```

*kavakmesut@outlook.com.tr

```
        TextBox5.Text = 997
        TextBox9.Text = 3
        TextBox7.Text = 1000
        TextBox8.Text = 997
        TextBox12.Text = 3
        TextBox10.Text = 1000
        TextBox11.Text = 997
        TextBox15.Text = 3
        TextBox13.Text = 1000
        TextBox14.Text = 997
        TextBox18.Text = 3
        TextBox16.Text = 1000
        TextBox17.Text = 997
        Timer1.Interval = 1
        Timer2.Interval = 10
        Timer3.Interval = 10
        Timer4.Interval = 10
        Timer5.Interval = 10
        Timer6.Interval = 10
        Timer7.Interval = 10
        Timer1.Start()
        Timer2.Start()
        Timer3.Start()
        Timer4.Start()
        Timer5.Start()
        Timer6.Start()
        Timer7.Start()
    End Sub
    Private Sub Timer2_Tick(sender As Object, e As EventArgs)
        Handles Timer2.Tick
        TextBox3.Text = Val(TextBox3.Text) - 1
    End Sub
    Private Sub Timer3_Tick(sender As Object, e As EventArgs)
        Handles Timer3.Tick
        TextBox6.Text = Val(TextBox6.Text) - 1
    End Sub
    Private Sub Timer4_Tick(sender As Object, e As EventArgs)
        Handles Timer4.Tick
        TextBox9.Text = Val(TextBox9.Text) - 1
    End Sub
    Private Sub Timer5_Tick(sender As Object, e As EventArgs)
        Handles Timer5.Tick
        TextBox12.Text = Val(TextBox12.Text) - 1
    End Sub
    Private Sub Timer6_Tick(sender As Object, e As EventArgs)
        Handles Timer6.Tick
        TextBox15.Text = Val(TextBox15.Text) - 1
    End Sub
    Private Sub Timer7_Tick(sender As Object, e As EventArgs)
        Handles Timer7.Tick
        TextBox18.Text = Val(TextBox18.Text) - 1
    End Sub
End Class
'This simple program shows us, that randomly changes the
    countdown time of each group. Sometimes one or more of them
    and some times the other ones become 1. You cannot guess.
```

2.1 Encoding

Let us use the second method. Now assume, that English Alphabet is used for the key. 13 of the letters are going to be selected for using instead of 1 and the other 13 of them are going to be used for 0 that these are 0 and 1 of the bits. First of all for example let us choose the letters for 1. For this I wrote a simple program over MS Visual Studio Community as the above. You can write your own timer by using own language and own software or hardware architecture or you can use another language. These would be very sensitive and thus you can get better results. This is just for example.

Now assume that there are 26 pieces of the timers for each letter. Click on the button or repeat the action by an automated program until at least 1 of them becomes 1 over the Large Timer. If more than 1 of them become 1, then you are going to repeat only for the different group until only 1 of them becomes 1. Finally you selected 1 letter for encoding bit 1. Repeat the same for the other 25 letters until there exist only 16 pieces. After that the rest namely the existent other 16 letters are going to be used for encoding bit 0.

2.2 Offline Calibration

While determining the key for bit 1 and bit 0 over the letters, two computers which are aimed to be connected to each other for communication, the key is recorded for both of them. Even the system architects which are built the machine cannot know the key. Now assume that each communication parts are performed by some certain bit lengths. Namely two computer always send 10 KB data parts even if the data is shorter then this. After each 10 KB packet, the key changes. The changer computer sends the new key; but this new key is also determined for the last time over the previous key. This previous key was already random namely even the system architects do not know. This repeats itself continual manner in this way. Namely nobody knows what the two computers do. We made them confidant. Even attacks can disrupt communication and packets but also they cannot steal from us even if we cannot perform communication. Nobody is going to know what the key is going to be next time approximately. It is not the solution recording only the calibration. You must save any act always. If you are a theft or a wicked engineer, then if you lost only one 10 KB packet for the life time of the communication during recording from the beginning, you cannot do anything.

2.3 Remote Calibration

This may be dangerous if you think that any signal is recorded by a third unknown computer. For the worst possibility, think that a user is recording any act and making a simulation of the communication by some algorithms over any possibility even by using new hardware architecture to catch the copy of the computers and the software. Namely making them sensible.

Actually the solution is making the data packets smaller and thus making the key change at a high frequency at last only during remote calibration.

3 Conclusion

As it can bee seen, Any information and the information searched for become the same after a point relatively to us. If you have only one bit for encoding like 1 or 0, it can get any letter. Namely there are 26 possibilities for 1 or 0. If you have a bit group like 10, 11, 01 or 00 each one of them can get the same letters. You cannot decide which one of them is the main aim if you do not know the random key. the condition is the same for any length bit group.

The key is also determined randomly. The builder system architect people also cannot know what is going to be the key. An approximately guess is also not going to be possible. It can suddenly be the worst and unpredictable possibility. Namely does not give benefit to kidnap and threaten the engineers for the key.

To work of this build in the best manner, actually a new hardware support is required. Especially for some specific communication instruments, you should build a new hardware architecture, and then the hardware will be doing the selection over unknown new hardware characters or frequencies. Even so, it can block most of the disturbing attacks by some simple programs. For example a simple windows emulator can block this. All windows is going to work over a simple encoding program first. Even computers are going to answer to a key-logger software or email by meaningless characters. Namely actually they can steal from us but cannot know what that information mean. It can be any information.

Acknowledgement

Thanks to Microsoft for VS Community.

I shall not demand a patent right. Anyone who wants to use the above stated things can use freely without asking.

References

1. Kavak M. 2018, Complementary Inferences on Theoretical Physics and Mathematics, OSF Preprints, Available online: https://osf.io/tw52w/
2. Kavak M. 2016, On the Uncertainty Principle, American Journal of Physics and Applications, Vol. 4, No. 4, 2016, pp. 90-123. Available online: https://osf.io/t8zqw/
3. Heisenberg W. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift für Physik*, 1927, v. 43 (3), 172–198.