Tiny Encryption Algorithm on Various Platforms

Sasmita Rani Rout[1], Amit Prakash Divekar[2]
School of Electronics Engineering
VIT University, Vellore

***Abstract* - This paper describes a cryptographic algorithm design called Tiny Encryption Algorithm i.e. TEA. TEA is a very simple algorithm. It is perfect for embedded systems as it requires little time and space. The main objective of TEA is minimization of memory footstep and maximization of speed, mainly embattled for mobile and embedded systems which aim more on speed and space. In TEA encryption and decryption is performed. These are performed using the operations from mixed (orthogonal) algebraic groups and an enormous number of rounds to achieve security and simplicity. To acclimatize with many real time constraints such as low cost, memory and data loss TEA can be implemented in a microcontroller. TEA can also be implemented in various platforms like LabVIEW, Matlab, FPGA. This paper compares and contrasts execution of TEA on two platforms: 1) LabVIEW and 2) ARM Cortex M0 based NUC140VE3CN microcontroller board. The report is generated by comparing these 2 implementations on the basis parameters like memory requirement, time, speed, user friendliness, etc.**

***Keywords* - TEA, LabVIEW, ARM Cortex M0, NUC140VE3CN**

I. LITERATURE REVIEW

Many papers were studied of which 3 main papers were used as base for this comparison report. The first paper shows design of a short encryption algorithm which gives security with simplicity using a large number of rounds based upon feistel iterations. This algorithm is called Tiny Encryption Algorithm or TEA. The second paper Microcontroller Based Cryptosystem With Key Generation Unit, describes use of a microcontroller to analyze performance of cryptographic algorithm for embedded security. The third paper discusses about LabVIEW and its advantages over microcontroller (and their relevant compilers) based embedded system and how and why it should be used in embedded system design.

II. INTRODUCTION AND IMPLEMENTATION

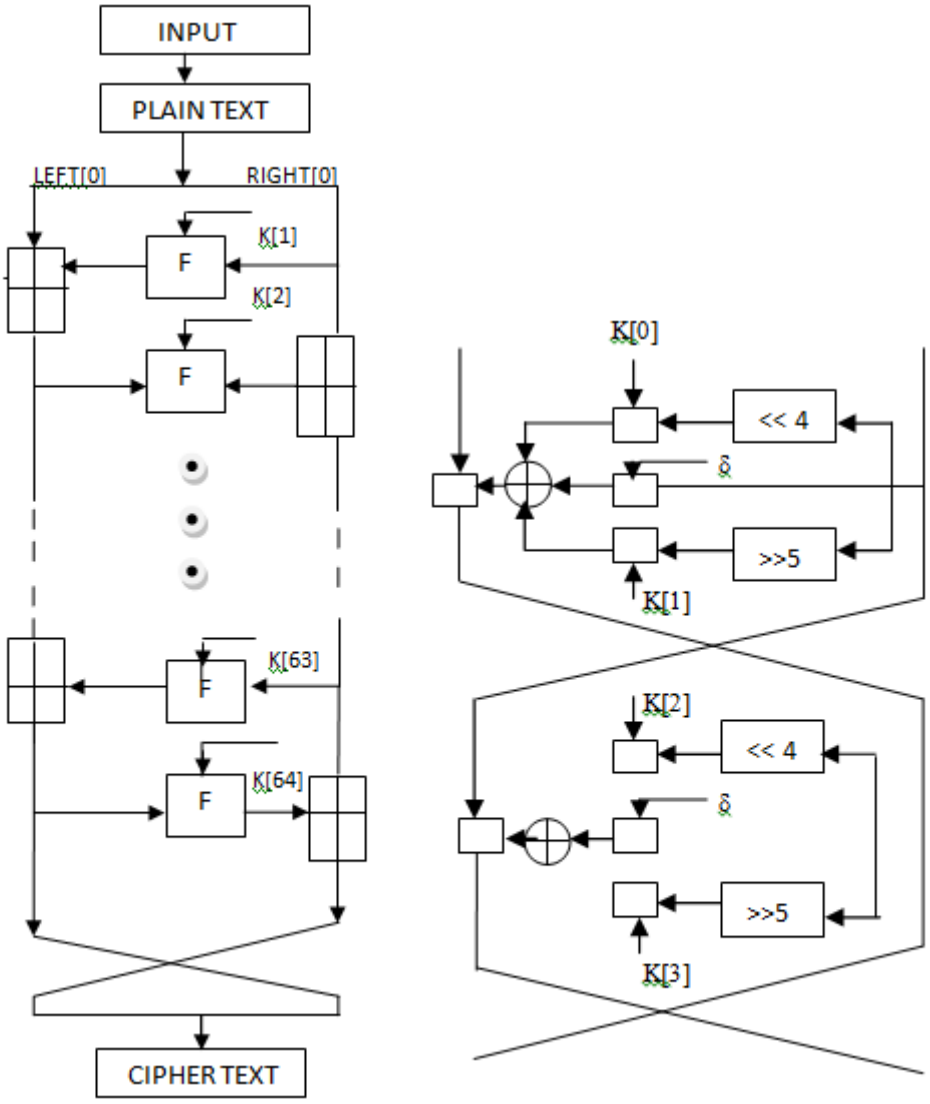
Data security and integrity has become an essential requirement in today's world where data communication has gained much more importance then voice or any other communication service. Many cryptographic algorithms are available to fulfill this requirement along with availability of various implementation platforms for these algorithms. One of these is tiny encryption Algorithm.

Tiny Encryption Algorithm

TEA is a simple, fast feistel block cipher developed by David J Wheeler and Roger M. Needham from Cambridge University with a simple design strategy which has less time and space requirement. The code is not issued to any patent.

The Algorithm:

- A symmetric block cipher notable for its simplicity.
- Uses non-destructive operations, XOR and ADD in the TEA's case.
- 64 bit block, 128 bit key.



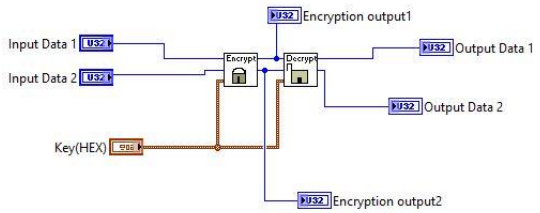
- 32-bit arithmetic operations done here.
- Number of rounds is variable (32 is considered to be secure).
- Uses + and - instead of ⊕ (XOR); since it uses "weak" round function, large number of rounds are required.
- Simple, easy to implement, fast, low memory requirement, etc.

Its encryption and decryption techniques follow same procedure only. The only difference here is that the keys are provided in reverse order for decryption. Figure 1 here summarizes this algorithm.

Each cycle is made up of 2 rounds here as it can be seen in figure 2.To offer nonlinearity TEA operations depend upon XOR and ADD using alternatively. Instead of XOR addition and subtraction are used for encryption and decryption as reversible operators. All bits of the key and data are mixed repetitively by a dual shift. The number of rounds may be 16 cycles (32-iterations) or 32 cycles (64-iterations).To avert simple search techniques from exploding the key, the key is fixed at 128 bits. A constant number called delta is used which guarantees that sub keys are different and their accurate values has no cryptographic importance. Delta is derived from the golden number ratio:

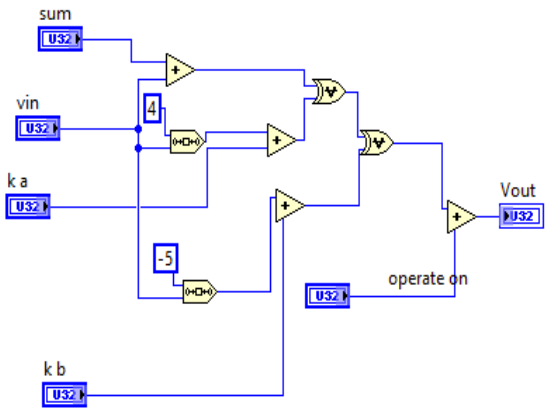
$$\text{delta} = \frac{(\sqrt{5}-1)*2^{31}}{} = 9E3779B9h$$

$$\text{delta}[i] = (i+1) * \text{delta},\ i = 0,1,2,\ldots,31$$

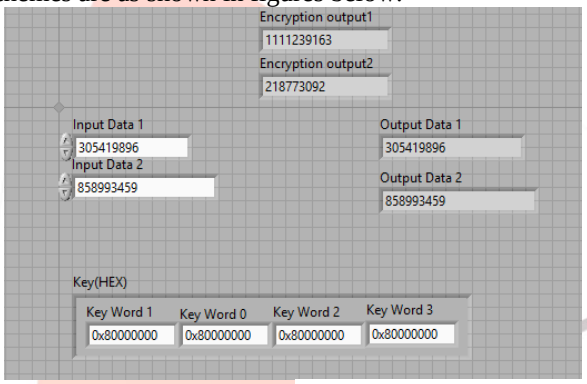


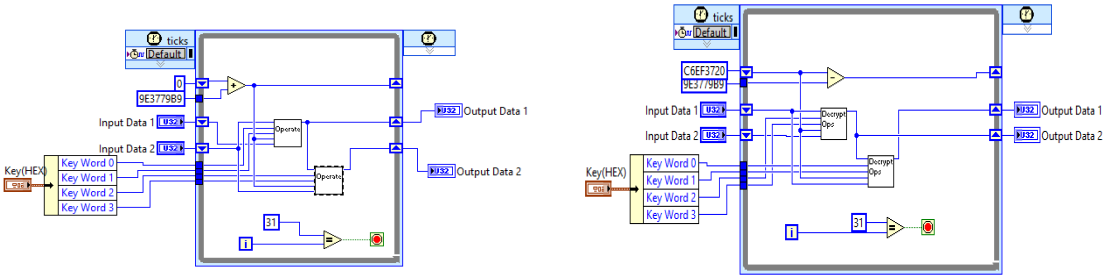
LabVIEW

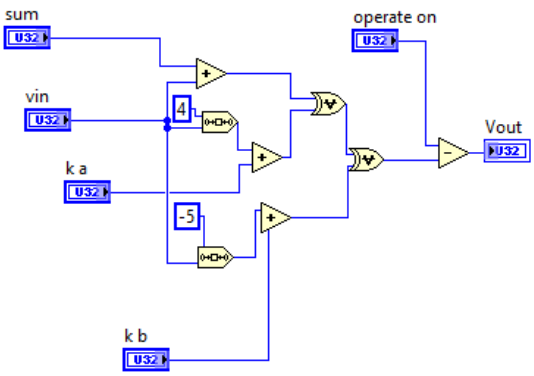
LabVIEW stands for Laboratory Virtual Instrumentation Engineering Workbench. NI LabVIEW is a platform designed for engineers and scientists which gives a graphical development. Like C, JAVA, the LabVIEW software is known as 'G' language. Its interfacing is GUI (Graphical User Interfacing) i.e. the complete program is represented in block diagrams instead of having syntaxes. This makes the programming language more easily understandable. This paper will be focusing on making use of LabVIEW in cryptographic application.



Here 2 separate subVIs are developed for encryption and decryption. Within encryption subVI 2 more subVIs are developed signifying 2 rounds (or 1 cycle) of TEA encryption scheme. Similarly for decryption 2 subVIs within decryption subVI is developed to perform 2 rounds (or 1 cycle) of TEA decryption scheme. Snapshots of front panel and block diagram windows of TEA and its encryption+decryption schemes are as shown in figures below.



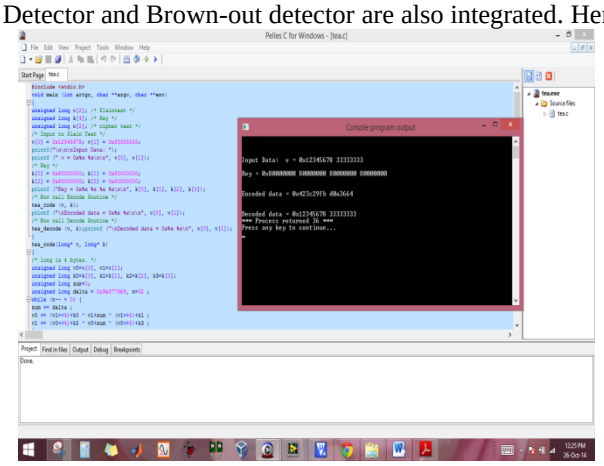


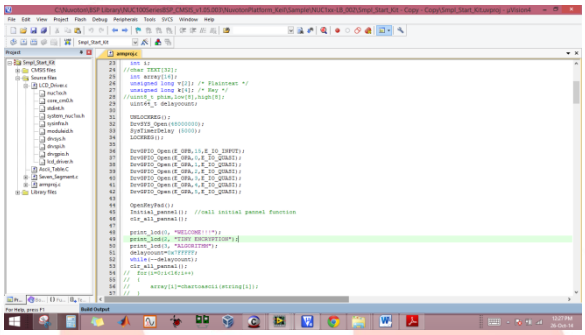


III. ARM CORTEX M0 BASED NUC140VE3CN

For industrial manage and the applications which require affluent communication functions need NUC1XX series which are ARM® Cortex™-M0 core embedded microcontroller. The Cortex™-M0 is the latest ARM embedded processor having 32-bit

presentation and the cost is correspondent to that of a conventional 8-bit microcontroller. The NUC1XX series with Cortex™-M0 core runs up to 50MHz, having 128K-byte embedded flash, and 16K-byte embedded SRAM. Timers, Watchdog Timer, UART, PDMA, RTC,GPIO SPI/SSP, I2C, PWM Timer, LIN, 12-bit ADC, CAN, Analog Comparator, USB 2.0 FS Device, Low Voltage Detector and Brown-out detector are also integrated. Here this microcontroller board is used for cryptographic function.





Programming this board can be done in higher as well as in assembly language. Programming of this application has been done in C language here.C code for TEA has been developed (using Pelles C software tool) and implemented on nuvoton microcontroller board. Like in LabVIEW, here also different functions for encryption and decryption were developed and implemented. Use of GPIO pins, LCD screen and Keypad of this microcontroller board has been done. For further development, data can be received and encrypted/decrypted from any of the available different ports on the board and can be again transmitted from any of the available ports on microcontroller board.

IV. COMPARING AND CONTRASTING

This comparison is done here for:
64 bit input data:
0x1234567833333333
128 bit key: 0x80000000800000008000000080000000

TIMING REQUIREMENTS

LABVIEW:

Using Profile Performance and Memory tool, Total Time taken by the LabVIEW based application to run is 15.6ms. It took this time to run encrypt and decrypt subVIs 1 time each and both of their operation subVIs were made to run for 64 times each.

NUC140VE3CN BOARD:

Using execution profiler of Keil-MDK tool, total time taken to execute the application on NUC140VE3CN board at the frequency of 48MHz was 7.937ms.

MEMORY REQUIREMENTS

LABVIEW:

Using Profile Performance and Memory tool, memory requirements of various parts of this application are:
Example VI: 3.70kB
Encrypt subVI: 8.08kB
Decrypt subVI: 8.09kB
Encryption Operation subVI: 3.65kB
Decryption Operation subVI: 3.65kB
LabVIEW Software Tool: 5.7GB (including all the default drivers from NI) for Windows 8.1

NUC140VE3CN BOARD:

Main µvision project file: 7.20MB
Main program C file: 8kB

Other supporting C files: 12kB
Other supporting and startup files: 5180kB
Keil MDK tool: 28.3MB

USER FRIENDLINESS

LABVIEW:

In LabVIEW programming is done using G language which makes logic implementation easy and provides more intuitive design notations. Graphical representation makes code easily understandable and doing any modification in the code is also stress-free. 2 separate windows showing the front panel and the logic behind it in block diagram window distinctly reduces confusion and misperception.

NUC140VE3CN BOARD:

Developing and implementing text based code for encryption algorithm takes time and is not as user friendly as developing code in G language.

V. CONCLUSION

The paper presents a comparison report between 2 different platforms for implementation and execution of tiny encryption algorithm (TEA). It can be concluded from the obtained results that the LabVIEW is a user friendly platform but requires higher memory usage. Whereas the implementation of cryptographic algorithm on ARM Cortex M0 based microcontroller NUC140VE3CN requires less memory and time. Anyhow this application is comparatively less user friendly than the LabVIEW based application.

VI. REFERENCES

[1] TEA, a Tiny Encryption Algorithm By David J. Wheeler & Roger M. Needham-Computer Laboratory, Cambridge University, England,1995
[2] The Implementation of Tiny Encryption Algorithm(TEA) on PIC18F4550 microcontroller by Edi Permadi, President University, 2005
[3] Microcontroller Based Cryptosystem With Key Generation Unit, By SenthilKumar.S and Manjupriya.M, IEEE 2011
[4] Enhanced Tiny Encryption Algorithm with Embedding (ETEA) by Dr. Deepali Virmani, Nidhi Beniwal, Gargi Mandal, Saloni Talwar , BPIT New Delhi, 2011
[5] The Development of Tiny Encryption Algorithm (TEA) Crypto-Core for Mobile Systems by Stephanie Ang Yee Hunn1, SitiZarinabinti Md. Naziri1,* Norinabinti Idris1, IEEE 2012
[6] Advantages of LabVIEW over Embedded System in Home Automations by B V Sumangala& K Bhargava Ram, International Conference on Advancement in Engineering Studies & Technology, 2012
[7] Nu_LB-002 Rev 2.0 -User's Manual
[8] NuMicro NUC100-Seies Driver Reference Guide
[9] NuMicro Family -NUC140 Product Brief
[10] NuMicro Family -NUC140 Data Sheet
[11] NuMicro NUC100 Series NUC130/NUC140 Technical Reference Manual
[12] LabVIEW Graphical Programming Course< http://cnx.org/content/col10241/1.4/ >
[13] http://www.ni.com/academic/students/learn-labview/
[14] http://www.ni.com/getting-started/labview-basics/
[15] http://forums.ni.com/