doi: 10.17221/12/2015-RAE

Using of tiny encryption algorithm in CAN-Bus communication

M. Jukl, J. Čupera

Department of Engineering and Automobile transport, Faculty of AgriSciences, Mendel University in Brno, Brno, Czech Republic

Abstract

Jukl M., Čupera J. (2016): **Using of tiny encryption algorithm in CAN-Bus communication**. Res. Agr. Eng., 62: 50–55.

The rising costs of agriculture machine operation force manufacturers to look for solutions that simplify the machine operation to its users and increase user comfort. However, this trend requires farm machinery to be equipped with electronic systems. Electronic control units do not receive only the information from its own sensors. Modern electronic systems communicate with each other via the data bus. The most common data bus in modern agricultural technology is the CAN-Bus (Controller Area Network). The most common standard used in modern machinery is SAE J1939 standard, which is commonly used for engine control systems. CAN-Bus in modern agricultural technology provides a considerable simplification of the wiring harness of the tractor. Standardized bus also opens the possibility of implementation of Plug & Play additional devices in agricultural tractor. This article is focused on the application of the encryption algorithm Tiny on the CAN-Bus, which is realistically applied to recognition of implement. This article aims to verify the suitability of encryption of Tiny algorithm for the CAN-Bus of 250 kbit/s. As the experiment demonstrated Tiny algorithm is suitable for data encrypting on the CAN-Bus.

Keywords: Tiny algorithm; encryption; CAN-Bus; implement; cipher

The main goal of any encryption system is to disguise the classified message to be unreadable to all unauthorized persons. Cryptographic systems most often come into practice by using of communication technologies (computer files sent over the internet). Encryption cannot protect sending files to get to the unauthorized person but prevents anyone to understand its contents. The information to be secured is usually known in specialized terminology as a plain text. The security process is called encryption. The rules for encryption of the plain text are called an encryption algorithm. The operations of this algorithm are derived from the encryption key. The encryption keys together with the text of the message are the input information to the algorithm. If the recipient of the cryptogram wants to get the original message, he has to use a decryption algorithm. The decryption algorithm uses the decryption key to convert encrypted text to the original. (Piper, Murphy 2006). The encryption algorithm is schematically shown in Fig. 1.

From cryptographic systems, that are known as a conventional or symmetrical, the decryption key may be easily derived from the encryption key. In

Supported by the Internal Grant Agency of the Mendel University in Brno, Faculty of Agronomy, Project No. IP 14/2014 "Diagnostic protocol OBD-2 based on SAE J1939".

doi: 10.17221/12/2015-RAE

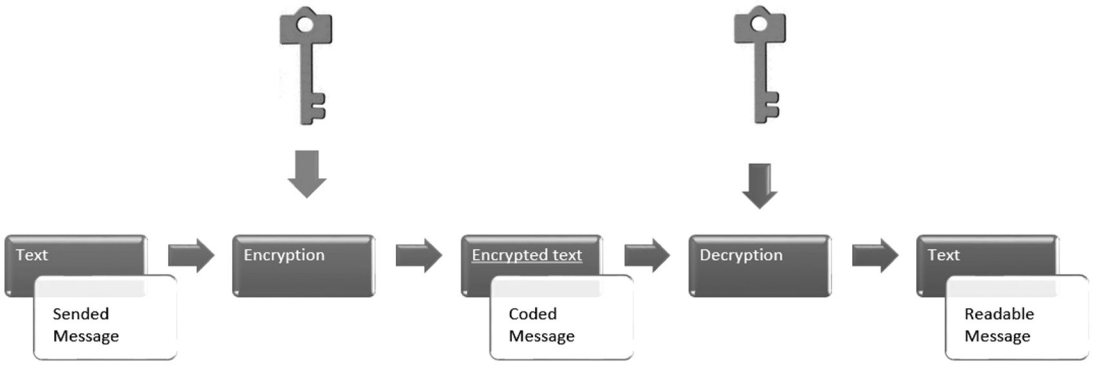

Fig. 1. General encryption algorithm

practice, both of these keys are identical in these systems. Due to such systems they are referred to as "secret-key systems" or "one-key systems". The second type of cryptographic systems is asymmetrical system. The derivation of the decryption key from this system is nearly impossible. Therefore asymmetric encryption system is often called as a "system with public-key". One of the possible variants of protection against unverified authorization is the application of encryption algorithm TEA. TEA means The Tiny Encryption Algorithm and was first published in 1994 by Roger Needham and David Wheeler from the Cambridge University. As it was mentioned in article by Grabowski and Keurian (2010), TEA was initially designed to be an extremely small algorithm when implemented in terms of the memory foot print required to store the algorithm. This was achieved by making the basic operations very simple and weak. By repeating these simple operations many times better security is reached. As the basic operations are very simple, TEA is also regarded as a very high speed encryption algorithm (Grabowski 2010).

The monitoring system, which was used in this experiment, exchanges data via CAN-Bus. The CAN-Bus (Controller Area Network) is the most common data bus in modern agricultural technology. The data bus serves to the electronic units to exchange the data from sensors and other control systems (Túró 2010). As it was mentioned in Túró (2011), the data bus is one of key components of modern vehicle systems. The experiment is expected to verify the functionality of Tiny algorithm applied in real conditions of such monitoring system. The specific applications will be targeted towards keeping the confidentiality of the implement identifiers.

MATERIAL AND METHODS

To investigate the possibility of using Tiny data encryption algorithm on the CAN-Bus, the experiment with a Stellaris microcontroller was performed. For problem-free application of encryption mechanism it was important to determine how long the process will take. This finding is especially important due to other parameters such as e.g. transmission rate of the message.

Tiny Encryption Algorithm. The Tiny Encryption Algorithm is a Feistel type cipher (Feistel 1973) that uses operations from mixed algebraic groups. A dual shift causes all bits of the data and key to be mixed repeatedly. The specification for TEA states a 128-bit key is to be divided into four 32-bit key words K = (K[0], K[1], K[2], K[3]) and the block size of each encryption is 64 bits, which is to be divided into two 32-bit words. At each iteration, the bits are further substituted and permuted via a system of logical operators (Yee-Tak Ma et al. 2011). Different multiples of a magic constant are used to prevent simple attack based on the symmetry of the rounds. The magic constant 2654435769 or $9E3779B9_{16}$ is chosen to be $2^{32}/\phi$, where ϕ is the golden ratio.

Encryption and decryption routine. The following figure (Fig. 2a) shows the structure of the TEA encryption. The inputs to the encryption algorithm are a plaintext block and a key K. The plaintext block is split into two halves (Left [0], Right [0]). Each half is used to encrypt the other half over 64 rounds of processing and then combine to produce the cipher text block.

In general, the decryption (Fig. 2b) is the same process as the encryption process. In the decode routine the cipher text is used as input to the al-

doi: 10.17221/12/2015-RAE

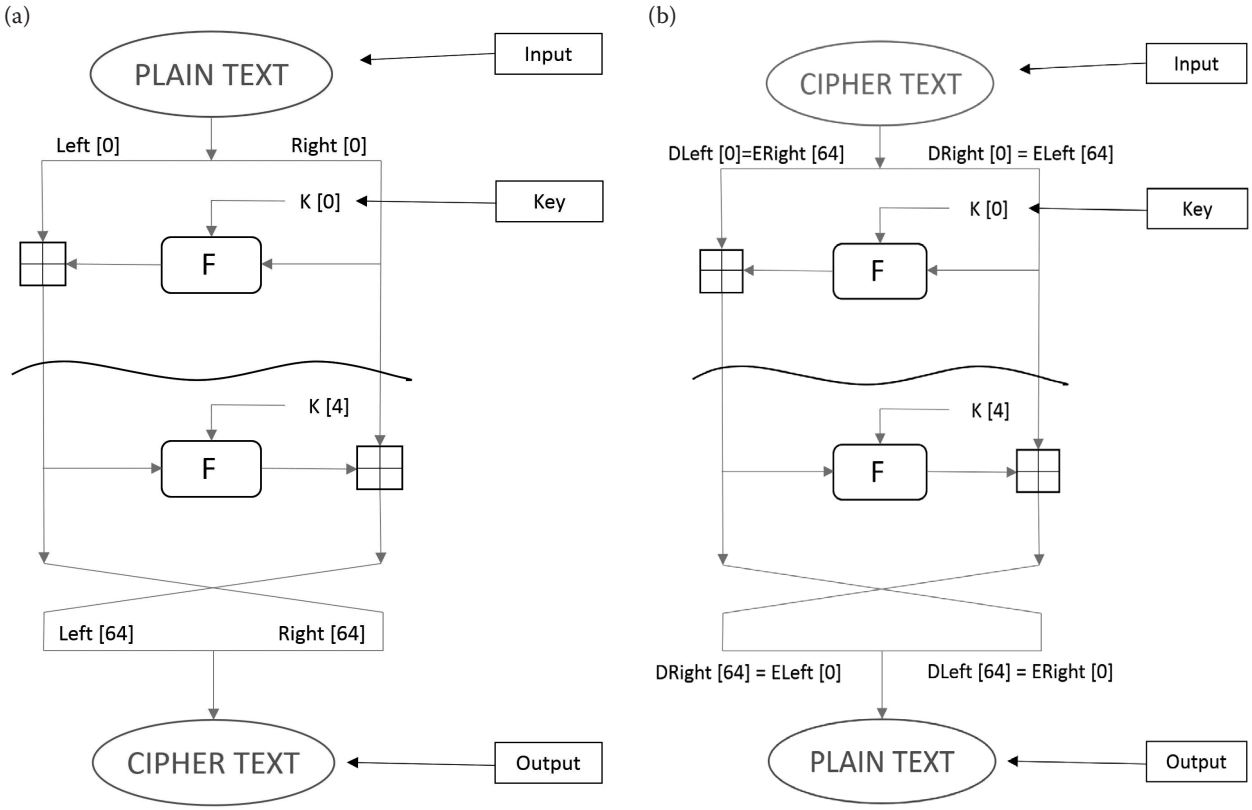

Fig. 2. Encryption (a) and decryption routine (b)

Encryption Routine

```
void code(long* v, long* k)  {
unsigned long y=v[0],z=v[1], sum=0,   /* set up */
 delta=0x9e3779b9, n=32 ;             /* a key schedule constant */
while (n-->0) {                       /* basic cycle start */
  sum += delta ;
    y += (z<<4)+k[0] ^ z+sum ^ (z>>5)+k[1] ;
    z += (y<<4)+k[2] ^ y+sum ^ (y>>5)+k[3] ;   /* end cycle */
              }
v[0]=y ; v[1]=z ; }
```

Decryption Routine

```
void decode(long* v,long* k)  {
 unsigned long n=32, sum, y=v[0], z=v[1],
 delta=0x9e3779b9 ;
sum=delta<<5 ;
                      /* start cycle */
while (n-->0) {
    z-= (y<<4)+k[2] ^ y+sum ^ (y>>5)+k[3] ;
    y-= (z<<4)+k[0] ^ z+sum ^ (z>>5)+k[1] ;
   sum-=delta ;  }
                      /* end cycle */
v[0]=y ; v[1]=z ;  }
```

Fig. 3. Encryption and decryption routine in C language (Wheeler et al. 1994)

doi: 10.17221/12/2015-RAE

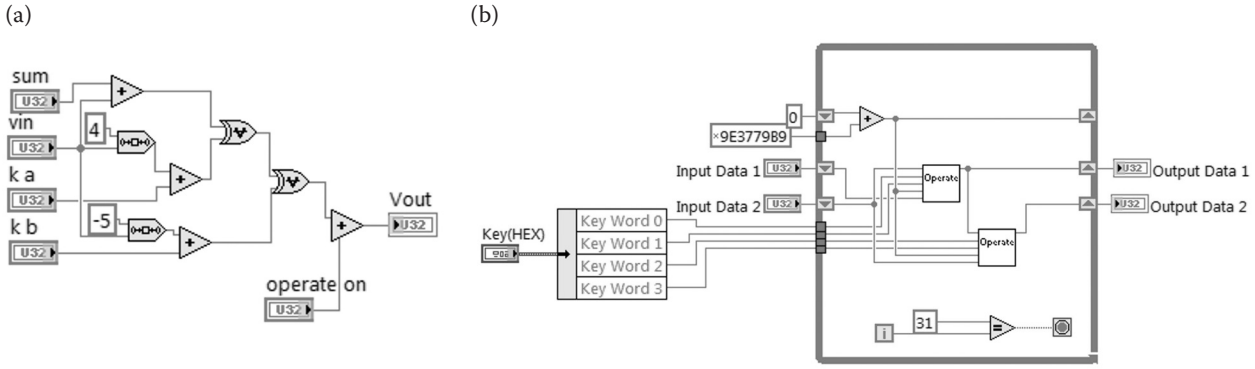

Fig. 4. Principle of TEA in graphical form (a) and data encryption mechanism (b) (both LabView)

gorithm and the keys K are used in the opposite sequence.

Mathematical description of the TEA. In this paragraph the encryption and decryption routines (Fig. 3) of the Tiny Encryption Algorithm in C language will be presented (Wheeler et al. 1994).

Instrumentation. The application of the encryption algorithm was performed on Stellaris LM3S8962 microcontroller by Texas Instruments (Dallas, USA). This device belongs into a group of devices that use ARM Cortex-M3 processors. Controller core is able to operate up to 50 MHz. This device is targeted for industrial applications, such as test and measurement equipment, network appliances and switches, etc. (Texas Instruments 2011).

Application of TEA in LabView. For the implementation of the encryption algorithm to the Stellaris microcontroller, proprietary software using a graphical programming environment LabView (G language) from the National Instruments (Austin, USA) was created. The diagrams in Fig. 4 show the basic structures of the encryption algorithm TEA in graphical form in LabView. To translate the software from G to C language, application ARM toolkit in LabView was used. The translation was carried out according to standard ANSI C.

RESULTS AND DISCUSSION

To verify if the Tiny encryption algorithm is suitable for messages encrypting on the CAN-Bus, the experiment was performed with help of microcontroller Stellaris (Texas Instruments, Dallas, USA). As it was mentioned above a limiting factor is the time during which the message is being encrypted. Input variable was a message of data length 8 byte (64-bit). For an objective determination of the duration of the encryption loop, a digital output of microcontroller Stellaris was observed with an oscilloscope (Texas Instruments, Dallas, USA)(Fig. 5).

After each finished loop the digital output changed the voltage level from 3.3 V to 0 V and *vice versa*. When the cycle was measured, the encryption loop was carried out two times. The final message processing time is therefore one half of the measured period.

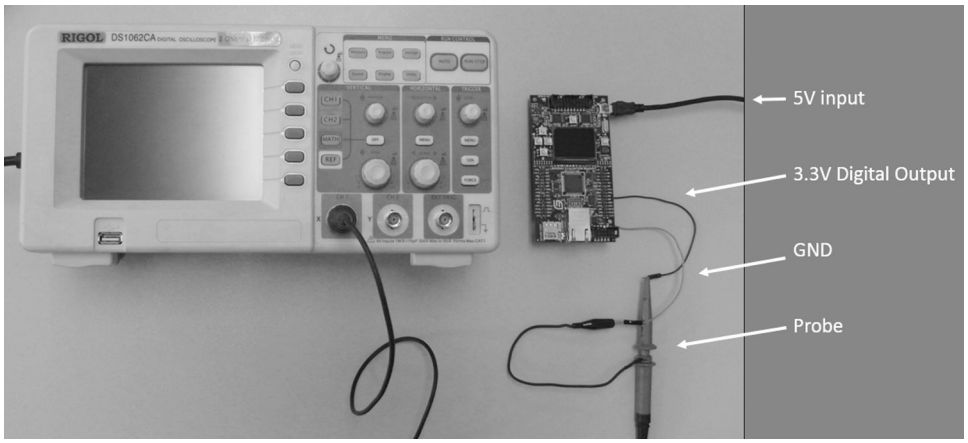

Fig. 5. Measuring of digital output with oscilloscope

doi: 10.17221/12/2015-RAE

(a) (b)

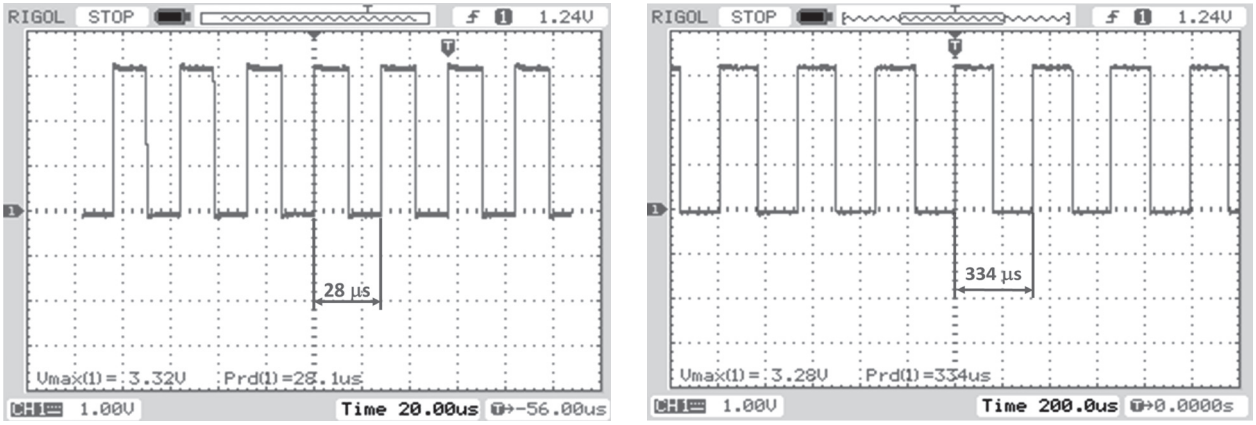

Fig. 6. Time of two sent messages (a) without encryption and (b) with encryption

Table 1. Calculated processing time (T)

½ of digital output period (µs)	
T – without encryption	14
T – with encryption	167
ΔT	153

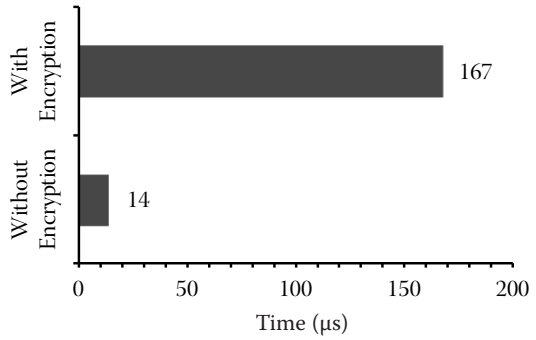

Fig. 7. Digital output response time

The measured signal from the oscilloscope is shown in the Fig 6. The first of the captured signals (Fig. 6a) is related to the time of two loops without encryption. The signal period was 28 µs. Since the loop was carried out two times, the processing time is 14 µs.

In the following oscillogram (Fig. 6b) there is captured digital output signal while two messages were encrypted. The length of the period takes 334 µs. The processing time of one message before sending to the bus takes one half of the period which equals to 167 µs.

The time values are plotted in the graph in Fig. 7 and are also presented in Table 1. It is obvious that the time difference ΔT will be equal to the time needed to encrypt a message of 64 bits.

As it is stated in publication Bauer et al. (2013), the bus with baud rate of 250 kbit/s is able to transmit 1894 frames per second (frame length of 135 bits – a standard message with identifier of 11 bits). The message used in experiment had the length of 128 bits. As a control mechanism in this frame 12 stuff bits were added. The result of CRC will be 4 bits. It is also necessary to respect the 3 bits as an intermission (gap between messages). There are 147 bits in total. With these parameters one message takes approximately 580 µs. According to the results it can be assumed that the message encryption (data of 64 bits) takes about 150 µs. This shows that the processing of message including encryption routines do not exceed the maximum time limit of 580 µs. The type of such encryption can be used for this specific case (baud rate = 250 kbit/s).

CONCLUSION

As it was demonstrated according to the experiment, encryption of data in CAN-Bus messages by using the Tiny encryption algorithm is possible. The duration of encryption loop do not exceed the critical limit 580 µs at 250 kbit/s. The transfer of important data between tractor and implement is also used on machines where the operator must set up parameters such as a dose of fertilizer or seed, working width or the other exact parameters for adjustment of the machine. This bus is called ISO-Bus and allows using of data from the bus of

doi: 10.17221/12/2015-RAE

tractor. These mentioned systems still do not use any encryption. The coding of transmitted data should be used in cases when there is an effort to hide information such as the vehicle identification number or other identifications. The mentioned encryption mechanism has been tested in a real application with microcontroller to detect the implement. The aim was to protect confidential data on the bus against eavesdropping. Application was designed for the specific processor with the defined parameters.

References

Bauer F., Sedlák P., Čupera J., Polcar A., Fajman M., Šmerda T., Katrenčík J. (2013): Traktory a jejich využití. Prague, ProfiPress, s.r.o.

Feistel H. (1973): Cryptography and Computer Privacy. Scientific American, 228: 15–23.

Grabowski J., Keurian J. (2010): Tiny Encryption Algorithm. Available at http://people.rit.edu

Piper F., Murphy S. (2006): Kryptografie – průvodce pro každého. Prague, Dokořán: 157.

Texas Instruments (2011): Stellaris® LM3S8962 Microcontroller Data Sheet, 1-806. Available at http://www.ti.com/stellaris

Túró T. (2010): Telemetry and diagnostics of military vehicles. In: 14th International Conference Transport Means 2010, October 21, 2010. Kaunas, Lithuania: 96–99.

Túró T. (2011): Issues of telemetry information in vehicle network. In: 15th International Conference Transport Means 2011, October 20, 2011. Kaunas, Lithuania: 189–192.

Wheeler D., Needham R. (1994): TEA, a Tiny Encryption Algorithm. Available at www.movable-type.co.uk/scripts/tea.pdf

Yee-Tak Ma E., Obimbo Ch. (2011): An evolutionary computation attack on one-round TEA. Procedia Computer Science, 6: 171–176.

Received for publication February 12, 2015
Accepted after corrections June 19, 2016

Corresponding author:

Ing. Michal Jukl, Mendel University in Brno, Faculty of AgriSciences, Department of Engineering and Automobile transport, Zemědělská 1, 613 00 Brno, Czech Republic; e-mail: michal.jukl@mendelu.cz