ISSN:0975-9646

Secured Data Transmission in Sensor Devices Using Minimal Keys

Satish Dekka,
Asst. Professor,
Sai Ganapathi Engineering College

Nandagiri R. G. K. Prasad,
Asst. Professor,
Sai Ganapathi Engineering College

Abstract: The Internet traffic is increasing day by day. Also, secure data communication is the need of the hour. But the standard compression techniques which are in use, are independent and do not consider the security issues. Hence, we present a general encoding technique for secure data communication over a language L with a finite alphabet set˘. The encoded message is a bi-tuple of which, the first is a vector of quotients denoted as Q and the second is a representation of remainders denoted as R with respect to a modulus M. The secrecy of the message is retained by communicating R over a secure channel using some standard encryption mechanism. The computation overhead is also reduced as the encryption is done only on one half of the encoded message. Further, this encoding mechanism provides a lossless compression are using TDEA algorithm for encryption.

Keywords: compression, encryption, decryption, encoding.

1. INTRODUCTION

The Internet traffic is increasing day by day. Also, secure data communication is the need of the hour. But the standard compression techniques which are in use, are independent and do not consider the security issues. Hence, we present a general encoding technique for secure data communication over a language L with a finite alphabet set ˘. The encoded message is a bi-tuple of which, the first is a vector of quotients denoted as Q and the second is a representation of remainders denoted as R with respect to a modulus M. The secrecy of the message is retained by communicating R over a secure channel using some standard encryption mechanism. The computation overhead is also reduced as the encryption is done only on one half of the encoded message. Further, this encoding mechanism provides a lossless compression.

- In present world, the Internet have become more popular for the resource sharing like h/w , s/w & data.
- The data transmission should be secure over the internet, because the confidential data may be open and vulnerable
- The volume of data transmitted over the Internet is also increasing. Presently, we have eBooks, multimedia, e-business etc. on the online.
- So, we should provide security for the data and also use the available bandwidth effectively.
- One such technique is "A Lossless Mod-Encoder Algorithm" which provides secure communication and also effective use of bandwidth.

Encryption and Decryption of data:

In cryptography, **Encryption** is the process of encoding messages or information in such a way that only authorized parties can read it. In an encryption scheme, the message or information, referred to as plain-text, is encrypted using an encryption algorithm [11], turning it into an unreadable cipher text [6].

Decryption: It is the process of decoding the data which has been encrypted into a secret format. An authorized user can only decrypt data because decryption requires a secret key or password. In simple terms it is the conversion of cipher text into plain text [6].

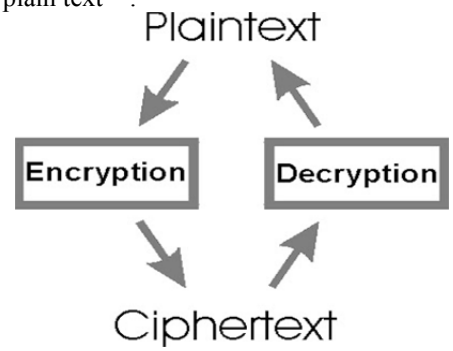

Figure 1: Conversion of plain text to cipher text and vice versa

Compression and Encoding algorithms are of major interest of the research since long. The following algorithms are a few of them are briefed below:

Triple DES

Triple DES (3DES) is the common name for the Triple Data Encryption Algorithm (TDEA) block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. Because of the availability of increasing computational power, the key size of the original DES cipher was becoming subject to brute force attacks; Triple DES was designed to provide a relatively simple method of increasing the key size of DES to protect against such attacks, without designing a completely new block cipher algorithm. ***This suffers from time complexity and long ciphers to be transmitted.***

AES

The Advanced Encryption Standard (AES) is a symmetric-key encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. ***This also suffers from time complexity and long ciphers to be transmitted.***

RC4

RC4 (also known as ARC4 or ARCFOUR meaning Alleged RC4, see below) is the most widely-used software stream cipher and is used in popular protocols such as Secure Sockets Layer (SSL) (to protect Internet traffic) and WEP (to secure wireless networks). **This is less computationally intensive compared to asynchronous algorithms.** Thus, an attempt to construct a simple, faster, and less heavy cipher encoder algorithm is being made in this project.

2. METHODOLOGY

The proposed MOD-ENCODER algorithm uses any standard encryption technique which incorporates lossless compression in order to cater to the needs of low bandwidth and data security. As mentioned above, let L be a defined language over the alphabet set. For example, L be English and be $\{A,B, \ldots Z,a, b, \ldots , z\}$. Let the letters of be indexed by a bi-junction function I that maps a letter to an integer i, where $1 \leq i \leq ||$. Δ is a constant, called modulus constant. Let the data string M be $\{m_1,m_2,m_3, \ldots , m_n\} \in$. Performing modulus operation on every $I(mi)$ by Δ, sequentially yields remainder set R as $\{r_1, r_2, r_3, \ldots , r_n\}$ and quotient set Q as $\{q_1, q_2, q_3, \ldots , q_n\}$. The elements in R have the values between $[0,\Delta - 1]$. So, we consider the elements in R to be a vector of numbers in base R. Each r_i takes log2 R bits for binary representation. If the message is of n characters, the number of bits required to represent the vector R is $n \times$ log2 R. The quotient set is represented in a different way. Let B $=[\Delta] + 1$ be another parameter called *Base-value*. The elements of Q will have values within $[0,B-1]$. Consider Q as a number in *base B*, i.e. $(q_1, q_2, \ldots , q_n)B$. Convert the number to a higher base. It is obvious that a higher base representation would reduce the digits in the number. If B is less than 10, then we convert Q_B to a Q_{10} number.

The MOD-ENCODER Encoding Algorithm can be stated as:

1) Input: M
2) $n = |M|$, i.e. length of M
3) $Z = n \times$ *bit size* i.e. *bit size* is the number of bits require to represent each char.
4) for $i = 1$ to n
4.1) Read mi the i^{th} character from M.
4.2) **Find** R*:* $R[i] = I(mi)\%\Delta$
4.3) **Find** Q*:* $Q[i] = I(mi)/\Delta$
5) **Representation of** R*:* for $i = 1$ to n
5.1) Represent $R[i]$ in *Base* Δ.
6) **Representation of** Q*:* Interpret Q as *Base B* number and convert it to *Base 10* The vector R is communicated through open channel, whereas Q is encrypted to a cipher Qc using any standard cryptographic technique and communicated to the receiver to ensure the confidentiality of the message M. By doing so, the overhead of encryption is reduced as we encrypt only tuple Q, rather than the whole message M. The receiver on receiving R and Qc, decrypts Qc to Q and decodes the message from the bi-tuple $< R,Q >$.

The MOD-ENCODER Decoding Algorithm as follows: 1) Input: Bi-tuple $< R,Q >$
2) **Convert** Q **from** ***Base 10*** **to** ***Base B:*** Let $QB = (q_1, q_2, \ldots , q_n)$ be the representation in *Base B*
3) **Interpret** R **as a vector of** ***Base*** Δ **number:** for $1 \leq i \leq n$
3.1) $i = q_i \times\Delta+r_i$ where q_i the i^{th} digit of QB r_i the i^{th} element of R.
3.2) $mi = I-1(i)$
4) $M = (m_1,m_2, \ldots , m_n)$

2.2 Compression mechanism

As mentioned above, the encrypted message M is a bi-tuple $< Q,R >$ of quotient and remainder. So, the size of the encrypted message can be obtained by calculating the number of bits required to represent Q and R. Let X be the total number of bits required to represent R and is given by $X = n*$ log2 Δ, where n is the length of the message and log2Δ is the number of bits required to represent each remainder. The quotient Q is looked upon as a *Base B* number. Each qi needs log2 B bits for its representation. As we know, a number in *Base 10* requires lesser number of bits than its equivalent in another *Base B* for representation, considering $B < 10$. Therefore, to lessen the number of bits required to represent Q, we first convert it into a *Base 10* number, say T. So, the number of bits required to represent Q is given as $Y = log2\ T$. Hence, the total number of bits needed for representation of the encrypted message M can be given by $X + Y$. Considering, a 7-bit representation for every character as in ASCII (or 8-bit in Unicode), $Z = n* 7$ is the total number of bits required for plain text message. We can observe that Z >(X+Y). So this reduction in bits provides us with the desired compression and the Compression Ratio *C.R.* is given by $C.R. = (X+Y) /Z$

Examples and discussion

Let us consider L to be the alphabet English with 26 characters. Let as take a sample message string M as 'I love my country'. The following illustrates the functioning of the above proposed algorithm. Length of M, i.e. $n = 14$. Number of bits required for original message = 98 bits ***Quotient and remainder for different values of δ***

Case 1: Take $\Delta = 4$. Above table shows the corresponding quotients and remainders. X=Total number of bits needed for $R = 28$. Now, we convert the *Base 7* Quotient to a *Base 10* number T. *Base B* = (26/4)+1 = 7 T = 491220602 Y = Total number of bits required for Q = 29 Total number of bits required to represent the encrypted message = X + Y = 57 Compression Ratio CR = 0.58

Case 2: Take $\Delta = 5$. The value of Δ is now varied and above table gives the corresponding quotients and remainders. X = Total number of bits needed for $R = 42$. Now, we convert the *Base 6* Quotient to a *Base 10* number T. *Base B* = (26/5)+1 = 6 T = 5789913061 Y = Total number of bits required for Q = 33 Total number of bits required to represent the encrypted message = X + Y = 75 Compression Ratio CR = 1.307 This shown that, for any fixed Δ, if the input message size increases the compression ratio remains almost constant. This implies that the compression ratio is independent to the size of

the message. The comparative results of the degree of compression with different *Base-values* Δ. Here the *Q* tuple is converted to the *Base 10* number for all the cases. That is, for any quotient $Q = (q_1, q_2, \ldots, q_n)$ *B* the number is converted to a *Base 10* number as $Q_{10} = (\overline{q_1}, \overline{q_2}, \ldots, \overline{q_n})$. It can be realized that, as *B* is much smaller than 10, the degree of compression is better. Whereas, when *B* is close to 10 the degree of compression reduces. For values of *B* greater than 10 there is no compression as the size of encoded message becomes bigger than M.

3. TRIPLE DATA ENCRYPTION ALGORITHM (TDEA):

DES (the Data Encryption Standard) is a symmetric block cipher developed by IBM. The algorithm uses a 56-bit key to encipher/decipher a 64-bit block of data. The key is always presented as a 64-bit block, every 8th bit of which is ignored. However, it is usual to set each 8th bit so that each group of 8 bits has an odd number of bits set to 1.

The algorithm is best suited to implementation in hardware, probably to discourage implementations in software, which tend to be slow by comparison. However, modern computers are so fast that satisfactory software implementations are readily available.

DES is the most widely used symmetric algorithm in the world, despite claims that the key length is too short. Ever since DES was first announced, controversy has raged about whether 56 bits is long enough to guarantee security.

The key length argument goes like this. Assuming that the only feasible attack on DES is to try each key in turn until the right one is found, then 1,000,000 machines each capable of testing 1,000,000 keys per second would find (on average) one key every 12 hours. Most reasonable people might find this rather comforting and a good measure of the strength of the algorithm.

Those who consider the exhaustive key-search attack to be a real possibility (and to be fair the technology to do such a search is becoming a reality) can overcome the problem by using double or triple length keys. In fact, double length keys have been recommended for the financial industry for many years.

Use of multiple length keys leads us to the Triple-DES algorithm, in which DES is applied three times. If we consider a triple length key to consist of three 56-bit keys K1, K2, K3 then encryption is as follows:

- Encrypt with K1
- Decrypt with K2
- Encrypt with K3

Decryption is the reverse process:

- Decrypt with K3
- Encrypt with K2
- Decrypt with K1

Setting K3 equal to K1 in these processes gives us a double length key K1, K2. Setting K1, K2 and K3 all equal to K has the same effect as using a single-length (56-bit key). Thus it is possible for a system using triple-DES to be compatible with a system using single-DES.

4. DATA ANALYSIS

Delta and base value calculation:

For mod encoder we are placing delta value is 44. By using delta value we are calculating base value as shown in figure 1. In our experiments we are designed two windows one for client and second one is for server. Both server and client must have same delta value for communication to perform encoding and decoding mechanisms.

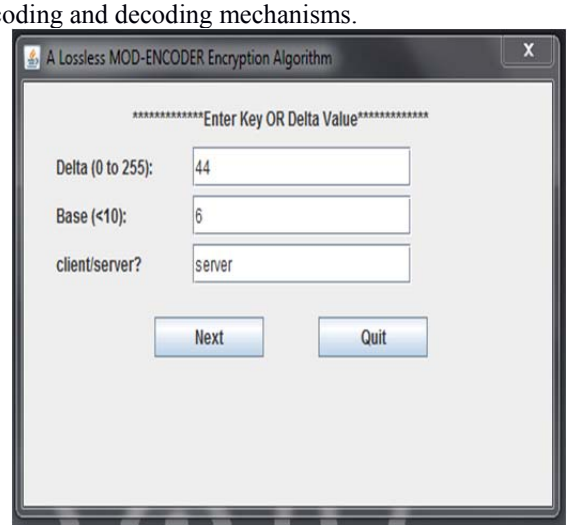

Figure 1: delta and base values in server/client.

Encoding process:

Figure 2 shows the encoding process for the given text. It shows quotient vector, reminder vector values

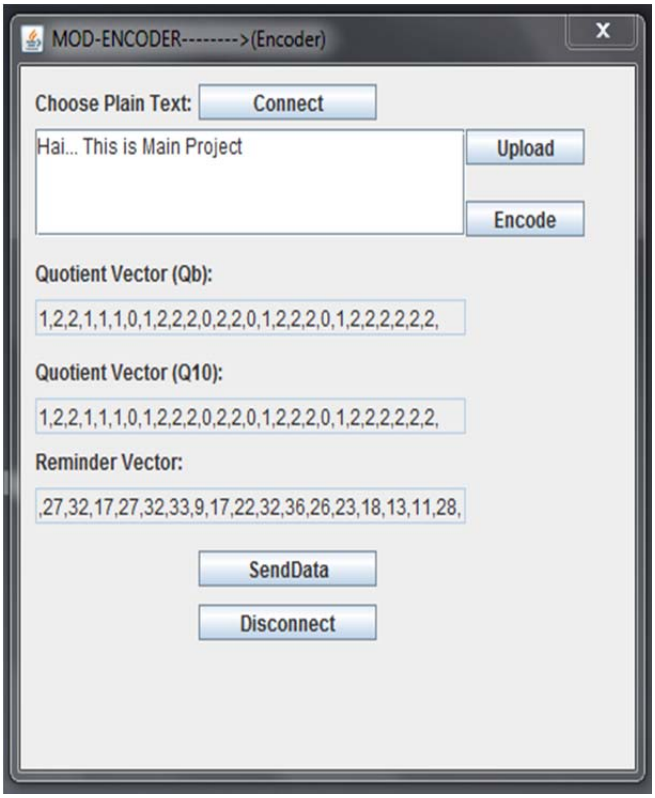

Figure 2: Encoding

Decoding process:

Figure 3 shows the decoding process for the given text. It shows quotient vector, reminder vector values and original message after decoding.

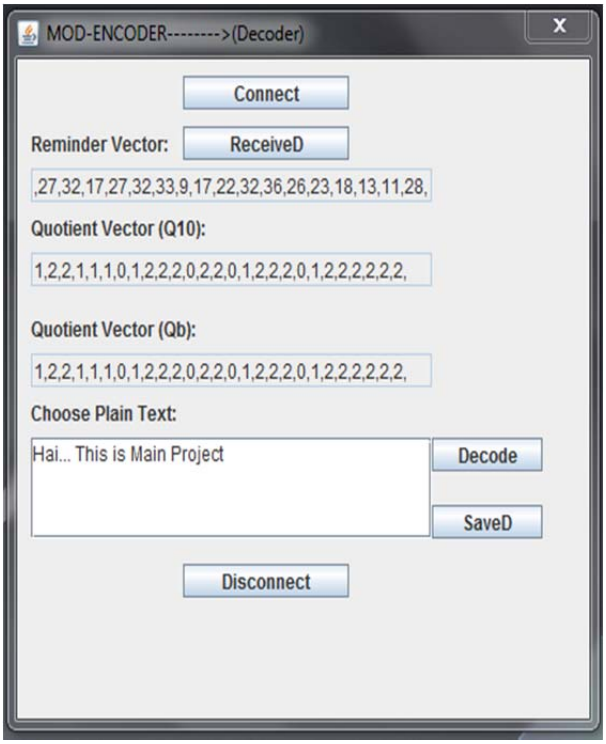

Figure 3: decoding.

5. CONCLUSION

The transmission of tuple Q to the receiver via a secure channel ensures the confidentiality of the message. However, tuple R can be communicated through open channel. So, R becomes easily accessible. Let some intruder capture R as (r1, r2, . . . , rn) from the open channel. In that case, the probability that an element ri is interpreted correctly by the intruder is Δ . For example, if is the set of 26 English characters and Δ is 5, any remainder ri = 3 has 5 possible outcomes as {d, i, n, s, x}. The message can be correctly decoded if the corresponding quotient qi is known. As Q is communicated securely, the intruder cannot decode the message. In general, if the message is of n characters, the unconditional probability for correctly decoding a message is Δ n.

The proposed MOD-ENCODER technique represents the message as a bi-tuple <R,Q >. The individual tuples can be communicated independently to the receiver. The secrecy of the message is ensured by encoding one half, generally Q, of the bi-tuple. The error probability of decoding is considerably high when either Q or R is unknown. The proposed technique also provides a lossless compression that facilitates 0 0.5 1 1.5 2 2.5 3 0 1000 2000 3000 4000 5000 Compression Ratio (CR) Size of Plain Text (n) DELTA = 4 DELTA = 5 DELTA = 7 Fig. 1. C.R. vs n 0 0.5 1 1.5 2 2.5 3 0 5 10 15 20 Compression Ratio (CR) Modulus Constant (Delta) Fig. 2. C.R. vs Δ better bandwidth utilization. Moreover, as the encryption is applied to one half of the encoded message, it reduces the computational complexity.

REFERENCES

[1] Debra A. Lelewer, Daniel S. Hirschberg ”Data Compression”, ACM Computing Surveys (CSUR), vol 19, Issue 3, pp. 261 - 296, Sep. 1987.

[2] William C. Barker, ”Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher”, National Institute of Standards and Technology, NIST Special Publication 800-67, 2008.

[3] ”Announcing the ADVANCED ENCRYPTION STANDARD (AES)”,Federal Information Processing Standards Publication 197, Nov. 2001

[4] R.L. Rivest, ”The RC5 encryption algorithm”, Proceedings of the 1994 Leuven Workshop on Fast Software Encryption, pp. 86-96, Springer- Verlag, 1995.

[5] Ron Rivest, Adi Shamir and Len Adleman, ”A method for obtaining Digital Signatures and Public Key Cryptosystems”, Communications of the ACM, pp 120-126, Feb. 1978.

[6] Taher ElGamal, ”A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”, IEEE Transactions on Information Theory, v. IT-31, n. 4, 1985, pp.469472 or CRYPTO 84, pp.1018, Springer-Verlag.

[7] Elliptic Curve Cryptography, Certicom Research, 2000

[8] Huffman’s original article: D.A. Huffman, ”A Method for the Construction of Minimum-Redundancy Codes”, Proceedings of the I.R.E., Sep.1952, pp.10981102

[9] Amit Jain, Ravindra Patel, ”An Efficient Compression Algorithm (ECA) for Text Data”, icsps, pp.762-765, 2009 International Conference on Signal Processing Systems, 2009

[10] Farina, A.; Navarro, G.; Parama, J.R., ”Word-Based Statistical Compressors as Natural Language Compression Boosters”, Data Compression Conference 2008, pp. 162 - 171, Mar. 2008 332.