Trends in Applied Sciences Research

ISSN 1819-3579

Academic
Journals Inc.

www.academicjournals.com

Trends in Applied Sciences Research 1 (4): 416-425, 2006
ISSN 1819-3579

A Power Efficient Encryption Algorithm for Multimedia Data in Mobile Ad hoc Network

[1]Sanjeev Sharma, [2]R.C. Jain and [3]Sarita Bhadauria
[1]School of Information Technology,
Rajiv Gandhi Technological University,
Bhopal (MP) 462036, India
[2]Department of Computer Application, Samrat Ashok Technology,
Institute Vidisha (MP), India
[3]Department of Electronic Engineering,
Madhav Institute of Technology and Sc. Gwalior, (M.P.) India

Abstract: The name mobile ad hoc network (MANET) is often used to describe infrastructure less mobile networks. Since there is a lack of infrastructure and the node mobility is way larger than in wired network and power limitations of a mobile node, new protocols are proposed to handle these new challenges. With the anticipated growth of connected devices the usage of ad hoc networks will probably increase. As more and more uses for these devices pop up it is certain that the need for securing the communications will be a high priority in the future. There are solutions proposed but all suffers from some drawbacks. Multimedia streaming is quite resource demanding. For encryption of such kind of data require high processing and consume more battery power. To secure end-to-end multimedia traffic the recommendation falls on hybrid algorithms. Many different approaches have been proposed over the years. The ones that have gotten the most common use are the Data Encryption Standard (DES) and the Rivest-Shami r- Adleman (RSA) Cryptosystem. The DES system is getting older and the new standard, the Advanced Encryption Standard (AES), is getting deployed in more and more solutions. We have developed a new application layer protocol PE^2 for encryption for both multimedia and text data on mobile Ad hoc network. This protocol attempts to adapt the Rijndael 256 bit block and key for mobile Ad hoc Network Environment encryption through the introduction of compression algorithms, Haar Transform for image data and Run Length encoding for text data to reduce input bits keeping the amount of computation less conforming to limitations of power of the devices used in MANET as limited battery resources are available at node.

Key words: Inter protocol (IP), MANET, PE^2 algorithm, Haar transform

Introduction

With recent performance advancements in computer and wireless communications technologies, advanced mobile wireless computing is expected to see increasingly widespread use and application, much of which will involve the use of the Internet Protocol (IP) suite. The vision of mobile ad hoc networking is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. Such networks are envisioned to have dynamic, sometimes rapidly-changing, random, multi-hop topologies which are likely composed of relatively bandwidth-constrained wireless links.

Corresponding Author: Sanjeev Sharma, School of Information Technology, Rajiv Gandhi Technological University, Bhopal (M.P) 462036, India Tel: 91 755 2678825 Fax: 91 755 2678834

Within the Internet community, routing support for mobile hosts is presently being formulated as mobile IP technology. This is a technology to support nomadic host roaming, where a roaming host may be connected through various means to the Internet other than its well known fixed-address domain space. The host may be directly physically connected to the fixed network on a foreign subnet, or be connected via a wireless link, dial-up line, etc. Supporting this form of host mobility (or nomadcity) requires address management, protocol interoperability enhancements and the like, but core network functions such as hop-by-hop routing still presently rely upon pre-existing routing protocols operating within the fixed network. In contrast, the goal of mobile ad hoc networking is to extend mobility into the realm of autonomous, mobile, wireless domains, where a set of nodes-which may be combined routers and hosts-form the network routing infrastructure in an ad hoc fashion.

MANET's Have Several Salient Characteristics

Dynamic Topologies

Nodes are free to move arbitrarily; thus, the network topology-which is typically multi-hop-may change randomly and rapidly at unpredictable times and may consist of both bidirectional and unidirectional links.

Bandwidth-constrained, Variable Capacity Links

Wireless links will continue to have significantly lower capacity than their hardwired counterparts. In addition, the realized throughput of wireless communications-after accounting for the effects of multiple access, fading, noise and interference conditions, etc.-is often much less than a radio's maximum transmission rate.

One effect of the relatively low to moderate link capacities is that congestion is typically the norm rather than the exception, i.e., aggregate application demand will likely exceed network capacity frequently. As the mobile network is often simply an extension of the fixed network infrastructure, mobile ad hoc users will demand similar services. These demands will continue to increase as multimedia computing and collaborative networking applications rise.

Energy-constrained Operation

Some or all of the nodes in a MANET may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation.

Limited Physical Security

Mobile wireless networks are generally more prone to physical security threats than are fixed-cable nets. The increased possibility of eavesdropping, spoofing and denial-of-service attacks should be carefully considered. Existing link security techniques are often applied within wireless networks to reduce security threats. As a benefit, the decentralized nature of network control in MANETs provides additional robustness against the single points of failure of more centralized approaches.

MANETs offer a promising new wireless communications paradigm, but researchers must develop efficient routing algorithms and address security concerns before such networks can be extensively deployed. Wireless technologies such as General Packet Radio Service, Wi-Fi, Home-RF and Bluetooth make it possible to access the Web from mobile phones, print documents from PDAs and synchronize data among various office devices. However, such applications rely at some point on mobility support routers or base stations and it is often necessary to establish communication when

the wired infrastructure is inaccessible, overloaded, damaged, or destroyed. Mobile ad hoc networks remove this dependence on a fixed network infrastructure by treating every available mobile node as an intermediate switch, thereby extending the range of mobile nodes well beyond that of their base transceivers. Other advantages of MANETs include easy installation and upgrade, low cost and maintenance, more flexibility and the ability to employ new and efficient protocols for wireless communication.

Security in MANETS

The use of wireless links makes MANETs susceptible to attack. Eavesdroppers can access secret information, violating network confidentiality. Hackers can directly attack the network to delete messages, inject erroneous messages, or impersonate a node, which violates availability, integrity, authentication and non-repudiation. Compromised nodes also can launch attacks from within a network (Hubaux *et al*., 2001). Ad hoc On-demand distance Vector Routing (AODV) (Perkins *et al*., 2003) and Destination Sequence routing algorithms (Johnson and Maltz, 1996) are employed in MANET. Bandwidth efficient routing protocol in which route selection is based on delay variance (Sharma *et al*., 2006) does not specify a scheme to protect data or sensitive routing information. Because any centralized entity could lead to significant vulnerability in MANETs, a security solution must be based on the principle of distributed trust. This is similar to the dilemma posed by the classic Byzantine generals problem (Lamport *et al*., 1982), in which a general commands each division of the army and some of the generals, who communicate via messenger, are traitors. All loyal generals must decide upon the same plan of action that is, a small number of traitors cannot cause the loyal generals to adopt a bad plan. The same holds for MANETs: A number of compromised nodes cannot cause the network to fail. Although no single node in a MANET is trustworthy, cryptography can distribute trust to an aggregation of nodes. Traditional key distribution schemes either do not apply to the ad hoc scenario or are not efficient for resource-constrained devices. Combining identity-based techniques with cryptography can achieve flexible and efficient key distribution.

To secure end-to-end traffic the recommendation falls on hybrid algorithms in which asymmetric cryptography is used to set up the session and the bulk traffic is encrypted using more efficient symmetric primitives. This way all kinds of hosts can be used and a handshake will make sure the best available solution is used in each case (Nichols and Lekkas, 2002). This is how SSL and TLS are working today and they have gained very much popularity. As to what specific algorithms to use it is best to propose the use of symmetric encryption for its higher efficiency in relation to other public key systems. RAC Algorithm (Dressler, 2005) provides reliable and semi- reliable communication service in Ad hoc network based on the typical properties of reliable communication protocols, in addition, the same methodology allows to include security services such as data integrity check, message authentication and address verification. These two services are based on shared secrets known to both communication ends. SPREAD Algorithm (Wenjing *et al*., 2004) provides data security over wireless channel by transforming a secret message into multiple shares by secret sharing scheme and deliver the shares via multiple independent paths to the destination. A protocol for setting initial secret key in Mobile Adhoc Network and extending the circle of authenticated nodes as new nodes arrive to join the network is developed (Cihan and Koc, 2005) which is based on Resurrecting Duckling Protocol.

These all systems are complex and require more processing and hence more power is consumed. Since the modulus can be kept smaller the computations will be smaller and thus faster. In the case

of symmetric systems the standard AES will be the most suitable in addition with Harr wavelet transform. This study introduces new scheme to add data security with optimum power consumption.

AES -Rijndael (Daemen and Rijmen, 2002)

AES can also be referred to as the American Encryption Standard. The NIST (American National Institute of Standards and Technology) began an international competition in 1997 with the goal of setting a new symmetric encryption standard to surpass DES (Hevia and Kiwi, 1999). Finally Rijndael of Deamon Rijmen (1999) from Belgium actually won the competition. Rijndael was a refinement of an earlier design by Daemen and Rijmen (1999), Square; Square was a development from Shark. Unlike its predecessor DES, Rijndael is a substitution-permutation network, not a Feistel network. AES is fast in both software and hardware, is relatively easy to implement and requires little memory. Rijndael was chosen as the standard because it had the best combination of security, efficiency, performance, flexibility and is easy to implement. As a new encryption standard, it is currently being deployed on a large scale.

AES is not precisely Rijndael (although in practice they are used interchangeably) as Rijndael supports a larger range of block and key sizes; AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits, whereas Rijndael can be specified with key and block sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits (Bl¨omer *et al*., 2004).

The block cipher Rijndael is designed to use only simple whole-byte operations. AES is primarily focussed on 128 bit block size. Rijndael is a relatively simple cipher in many aspects. Rijndael has a variable number of rounds. Not counting an extra round performed at the end of encipherment with one step omitted. The number of rounds in Rijndael is

- 9 if the block and the key are 128 bits long.
- 11 if either the block or the key is 192 bits long, or neither of them is longer than that.
- 13 if either the block or the key is 256 bits long.

AES operates on a 4×4 array of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). For encryption, each round of AES (except the last round) consists of four stages:

SubBytes

A non-linear substitution step where each byte is replaced with another according to a lookup table.

ShiftRows

A transposition step where each row of the state is shifted cyclically a certain number of steps.

MixColumns

A mixing operation which operates on the columns of the state, combining the four bytes in each column using a linear transformation.

AddRoundKey

Each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.

The final round omits the MixColumns stage.

PE^2 attempts to apply the Rijndael 256 bit block and key for Mobile Ad-Hoc Network Environment through the introduction of Haar Transform to reduce input bits keeping the amount of computation less conforming to the limitations of power of the devices used.

The Haar Transform

The Haar Transform is basically a wavelet can be thought of as a function that oscillates, similar to a wave, but as it tends toward plus or minus infinity, it approaches zero. Wavelets are used for many reasons. Wavelets allow signals to be decomposed into various parts, some of which can simply be thrown away. This is called quantization. The general idea behind a wavelet-based transform is to use a high-pass and a low-pass filter (Steve, 1991).

The high-pass (difference) filter is defined as:

$$D = a - b$$

where, a and b are the two values to transform and D is the difference between a and b.

The low-pass (average) filter is defined as:

$$A = (a + b)/2$$

where, a and b are again the two values to transform and A is the average between a and b. By taking values of every pair of vertically adjacent pixels in the image and transforming them into their average and difference, the entire image can be transformed.

Implementation of Haar Transform (Gary, 1991)

Three types of errors can occur in the transform, keeping the algorithm from being successfully reversed. On a computer, when the average of two values is calculated using integer arithmetic, some precision errors may occur. This is due to the decimal portion of the number being truncated. In fact, precision errors will always occur if the sum or difference of the two numbers is odd. This is a very simple problem to deal with. When the Haar transform is applied to an image, two numbers, the average and difference of the original two numbers, are stored for every pair of numbers in an image. The precision errors occur in the average. The difference can be used to determine if a precision error occurred; if this value is odd an error has occurred.

Another type of error also occurs, due in part to the way in which rounding errors are dealt with. The range of numbers is between -128 and 127, for a given value in the images. If the sum of the two numbers is odd, then a precision error will occur. However, if the sum of the two numbers is also negative, then it is important to realize how the numbers will be truncated. With computers, numbers are always truncated by simply chopping off the decimal portion of the number. In effect, all numbers are rounded toward zero using this method. This means that negative numbers are always rounded up, instead of down, like positive numbers. In order to use the fix described previously for dealing with precision errors, a statement must be added to the program to allow negative numbers to be rounded down instead of up. A "floor" function (greatest integer less than or equal to) must be implemented instead of truncation.

Finally, if the difference between two numbers is larger than the set limits of the averaging function, then the average of the two numbers will be off in the eighth bit. This is an extremely tough problem to diagnose. However, once the problem was recognized, the fix was easy. If the difference between the two numbers is less than -128 or greater than 127, then the eighth bit of the average must be set to represent the overflow.

The PE^2

Considering Secure Key distribution in the Mobile Ad-Hoc Network and assuming both sender and receiver are provided with the same key, we can now use Rijndael 256 block and key algorithm to encrypt secret data (Daemen and Rijmen, 1999)

The data that must be converted into cipher text in PE^2 256 bit and key operates on a 4x4 array of bytes, termed the State.

Before the encryption begins the state is applied with the Haar Transformation in the case of image data or to Run Length Encoding in the case of Text data. We now consider only image data as input data.

Pseudocode for Haar transform

Encoding

```
A = (a + b)/ 2;                 /* Low-pass (average) filter */
D = a - b;                      /* High-pass (difference) filter */
if((a + b < 0) and (D and 1)    /* Use floor function if A was */
A = A - 1;                      /* Rounded off */
if((a - b < -128) or (a - b > 127)
A = A + 128;                    /* See if the difference is out of  range */
```

In case of Run Length Encoding, Similar Strings are looked up for repetition and are replaced with pointers thereby reducing the requirement of space and thus compressing data (Gladman, 1992).

The obtained state is now compressed by haar compression or run length encoding. The encoded state is now ready for encryption.

For encryption, each round of PE^2 (except the last round) consists of four stages:

The SubBytes Step

In the SubBytes step, each byte in the array is updated using an 8-bit S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the inverse function over GF(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement) and also any opposite fixed points.

The ShiftRowsStep

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. The first row is left unchanged. Each byte of t he second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state.

The MixColumnsStep

A mixing operation which operates on the columns of the state by combining four bytes in each column using a linear transformation.

The AddRoundKeystep

In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using the key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

These operations are performed on the state for 13 rounds. In the last round the mix columns step is omitted and last round is performed.

The result after all the rounds are performed is that we get cipher text. Cipher text is decrypted at the receivers end. The received state is then subjected to inverse HAAR Transform in the case of image data or Run Length decoding in the case of Text data.

Decoding

```
a = A + (D / 2);            /*Calculate the first value */
b = A - (D / 2);            /* Calculate the second value */
if (D and 1)                /* Test to see if D is odd */
if (D < 0) b++ else a++;    /* Increment the larger value */
```

which will resurrect plain text and image from the state obtained by Rijndael.

This implementation has been done in School of Information Technology, Rajiv Gandhi Technological University, Bhopal, India by Mobile Adhoc Network Research Group (MANRG).

Implementation Aspects

Figure 1 shows block diagram of PE^2 implementation. PE^2 can be implemented as software based implementations and hardware based implementations. The software based implementations are designed and coded in programming languages, such as C, C++, Java and assembly. These implementations are executed, e.g., on general-purpose microprocessors, Digital Signal Processors (DSP) and micro-controllers (such as smart cards). The hardware based implementations are designed and coded in hardware description languages.

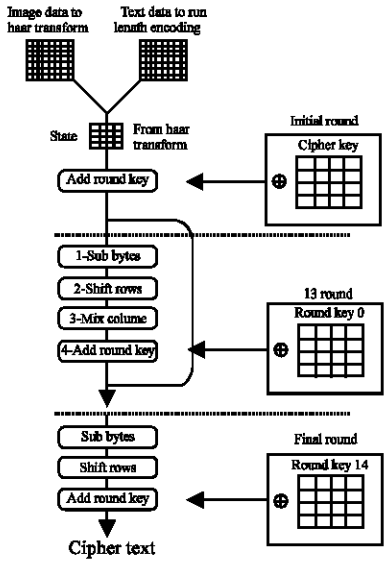

Fig. 1: Procedure of Power Efficient Encryption Algorithm (PE^2) for MANET

While hardware implementations of PE^2 can treat the input, output and cipher key inputs as bit sequences, software implementations, like Mobile Ad-Hoc networks will almost always to treat these entities as arrays of 8-bit bytes. Equally, while a hardware implementation will have to include a description of how PE^2 inputs and outputs are interfaced, a software implementation will often operate in an environment where PE^2's two key enumerations -the enumeration of bits within 8-bit bytes and the enumeration of bytes within arrays -are already defined.

Mobile Ad-Hoc environment in which PE^2 is implemented provides both for 8-bit bytes as addressable entities and for the enumeration of bits within bytes, it is reasonable to assume that PE^2 inputs and outputs will comply with any other Software implementation conventions. In consequence PE^2 implementations should either indicate that this assumption is correct or alternatively undertake one of the following:

- convert inputs and outputs to (or from) these standard formats to those being used internally;
- Document the interface to ensure that users of the implementation know that the inputs and outputs are in non-standard formats.

Conclusions

Since PE^2 is so simple that it can even be implemented on a 8 bit computer, yet almost impossible to crack, it has huge scope in Mobile Ad-Hoc Networks for multimedia streaming. The Haar Transform is the simplest in its kind and is easy to implement. The benefit of this algorithm in the area of MANETs is that the data that must be transmitted by the mobile node in to the network are reduced for sender with little overhead in computation as the algorithm has used Haar Transform, the computation is kept to a minimum, which results in less data that has to be received by the mobile node at the receivers end, thereby conforming to the power saving requirement of the nodes in a mobile ad hoc network.

Table 1: Performance of software based PE2 implementation

Platform	Compiler	Throughput/Latency
Pentium Pro 200 MHZ. 64 MB RAM, Windows 95	Borland C++ 5.01	718 cycles
Pentium Pro 200 MHZ. 64 MB RAM, Windows 95	Borland C++ 5.01	718 cycles
Pentium Pro 200 MHZ. 64 MB RAM, Windows 95	Visual C 6.0	1193 cycles
Pentium III 600 MHZ. 128 MB RAM, Windows 98	Borland C++ 5.01	764 cycles
Pentium III 600 MHZ. 64 MB RAM, Linux	Visual C 6.0	891 cycle
Pentium Pro 200 MHZ. 64 MB RAM, Windows 95	GCC 2.8.1	43.4 Mbits s^{-1}
Sun 300 MHZ. Ultra SPARC-II 2 MB cache, 128 MB RAM	GCC 2.95	48.2 Mbits s^{-1}
Pentium IV 3.2 GHz.	Assembly	261 Cycles 1561.3 Mbits s^{-1}

This algorithm is implemented in ANSI C and tested on various platforms. Easily deployment on the MANET, as MANET has limited battery and processing power, it is desirable to evaluate PE^2 algorithm on various kind of mobile nodes having different technical specifications, Operating systems.

Table 1 shows the performance of PE^2 over different mobile nodes of different technical specifications and operating systems obtained throughput / latency of encryption process.

References

Bl¨omer, J., J.G. Merchan and V. Krummel, 2004. Provably Secure Masking of AES. Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2004/101.

Cihan, M. and C.K. Koc, 2005. Setting initial secret keys in a mobile adhoc network. International Symposium on Information Technologies-ISIT 2005, Girne, North Cyprus.

Daemen, J. and V. Rijmen, 1999. AES Proposal: Rijndael. http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf.

Daemen, J. and V. Rijmen, 2002. The Design of Rijndael: AES-the Advanced Encryption Standard. Springer-Verlag.

Dressler, F., 2005. Reliable and semi-reliable communication with authentication in mobile ad hoc networks. Proceedings of 2nd IEEE International Conference on Mobile Ad hoc and Sensor Systems (IEEE MASS 2005,Washington, DC, USA., pp: 781-786.

Gary, S., 1991. Encoding the Neatness of Ones and Zeroes. Published in the issue of Scientific American Magazine,pp: 54-58.

Gladman, B., 1992. Implementations of AES (Rijndael) in C/C++. (http://fp.gladman.plus.com/cryptography technology/rijndael/index.htm).

Hevia, A. and M. Kiwi, 1999. Strength of two data encryption standerd implementations under timing attacks. ACM Transaction on Information and System Security.

Hubaux, J.P., L. Buttyán and S. Capkun, 2001. The quest for security in mobile ad hoc networks. In Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing, http://citeseer.ist.psu.edu/hubaux01quest.html.

Johnson, D.B. and D.A. Maltz, 1996. Dynamic Source Routing in Ad Hoc Wireless Networks. In Mobile Computing (Eds.) Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, pp: 153-181.

Lamport, L., R. Shostak and M. Pease, 1982. The byzantine generals problem. ACM Trans. Programming Languages and Systems, 4: 382-401.

Lou, W., W. Liu and F. Yuguang, 2004. SPREAD: Enhancing data confidentiality in mobile ad hoc networks. IEEE INFOCOM, Hong Kong, China.

Nichols, R.K. and P.C. Lekkas, 2002. Wireless Security Models. Threats and Solutions. McGraw-Hill.

Perkins, C.E., E.M. Belding-Royer and S.R. Das, 2003. Ad Hoc On-Demand Distance Vector (AODV) Routing. IETF Manets Working Group Internet Draft, www.ietf.org/internet-drafts/draftietf-manet-aodv-13.txt.

Schiller, J. Mobile, 2000. Communications. Addison-Wesley Publishing Company.

Steve, A., 1991. Lossless Data Compression. Byte Mar Publication , pp: 309 -387.

Sharma, S., R.C. Jain and S. Bhadauria, 2006. Bandwidth efficient variance adaptive routing protocol for mobile adhoc network. ADIT. J. Engin., 2: 15-19.

Wenjing, L., W. Liu and Y. Fang, 2004. SPREAD: Enhancing data confidentiality in mobile and Ad hoc networks. IEEE INFOCOM 2004, Hong Kong, China.