Development and Analysis of High Data Rate Quality based Secured AODV-RC4 and AODV-RSA WSNs

Zaina Khan[1] Shish Ahmad[2] Shimaila[3]
[1,2,3]Department of Computer Science & Engineering
[1,2,3]Integral University, Lucknow, India

Abstract— Determination of the figuring strategy is vital on the grounds that the security concerns are secured by the cryptography. Sensor nodes have exceptionally restricted computational and memory abilities, so all the cryptographic procedures cannot be just applied to the WSNs. To fulfill the above security necessities requirements the current figuring systems are be altered alongside consideration of created novel directing methods. In this work security strategies are found in WSNs that can meet the transmission rate prerequisites of sensor hubs and the assessment are seen by code size, information size, handling time, and steering deferral utilization. In light of the RSA and RC4 figuring cryptographic strategies, we have thought about them as far as information size, number of system hubs and hub dissemination.

Key words: WSN, Ciphering, AODV, RC4, RSA, Network Security

I. INTRODUCTION

RC4 is the most broadly utilized programming based stream figure. The figure has been coordinated into TLS/SSL and WEP usage. The figure was planned by Ron Rivest in 1987 and kept as a competitive innovation until it was spilled out in 1994. RC4 is to a great degree quick and its outline is straightforward. The RC4 stream figure is taking into account a mystery inner condition of N = 256 bytes and two record pointers of size n = 8 bits. In this paper we display a predisposition in the first's dispersion two yield bytes. We watch that the initial two yield words are equivalent with likelihood that is altogether not exactly anticipated. Taking into account this inclination we build a distinguisher with non-immaterial point of interest that recognizes RC4 yields from irregular strings with just 224 sets of yield bytes when N = 256. All the more fundamentally, the inclination stays noticeable even subsequent to tossing the beginning N yield bytes. This helps us to make another handy distinguisher with just 232 sets of yield bytes that works 256 rounds far from the starting when N = 256.

This Ronald Rivest, Adi Shamir and Leonard Adleman in 1997 gave greatest security to the information over system by giving this RSA calculation. This security framework is made out of three stages in particular Key Generation, Encryption and Decryption. Likewise we can take note of that numerous security frameworks are fabricated utilizing this three stage plan. In this strategy there are two keys Private Key and Public Key. Open Key is utilized to scramble the message and can be seen by all, where as the private key additionally called as the mystery key is utilized to decode the messages.

Additionally there are techniques to break RSA security Public key cryptography is one of the framework which is not exceptionally secure in light of the fact that it is all that much inclined to insecurities while sending which is found in the web today. Be that as it may, there are numerous mathematical suppositions which we have considered as an imperative key in this issue. For instance, whole number considering issue and discovering prime numbers. To figure out n in RSA we need to discover p and q which are prime numbers. Likewise , modulo n is a NP hard issue and a considerable lot of the Public key cryptography are depended upon it yet it is not for all intents and purposes conceivable in light of the fact that the quadratic sifter is utilized for factorizing RSA-120 by Thomas, Bruce, Arjen and Mark[3] . Likewise, the RSA-140 is considered utilizing number field strainer by Cavallar, Dodson, Lenstra, Leyland, Lioen, Montgemery, Murphy and Zimmermann [4]. While RSA-155 is considered in 1999, additionally, the RSA-160 is figured in April 2003, and the RSA-576 is calculated in December 2003 by Eric [5]. The RSA-200 is considered in 2004; the RSA-640 is figured in November 2, 2005 by Bahr, Boehm, Franke and Kleinjung [6] and confirmed by RSA Laboratories. The connection in the middle of considering and the general population key encryption plans is one of the fundamental reasons that scientists are keen on figuring calculations. In 1976 Diffie-Hellmam [8] makes the first progressive examination out in the open key cryptography by means of exhibited another thought in cryptography and to test specialists to produce cryptography calculations that confronted the necessities for open key cryptosystems. On the other hand, the first response to the test is presented in 1978 by RSA [9].

II. RSA ALGORITHM

RSA has been widely used for many years on the internet for security and authentication in many applications including credit card payments, email and remote login sessions. After seeing several examples of “classical" cryptography, where the encoding procedure has to be kept secret (because otherwise it would be easy to design the decryption procedure), we turn to more modern methods, in which one can make the encryption procedure public, without sacrifice of security: knowing how to encrypt does not enable you to decrypt for these public key systems. To understand how the algorithm was designed, and why it works, we shall need several mathematical ingredients drawn from a branch of mathematics known as Number Theory, the study of whole numbers. In recent times it has been found very useful, as we shall see. Here are the ingredients we will draw from number theory:

- Modular arithmetic
- Fermat's “little" theorem
- The Euclidean Algorithm

A. Public Key Encryption

This idea omits the need of a carrier to deliver keys to recipients over another secure channel before transmitting the originally intended message. In RSA encryption keys are public, while the decryption keys are not, so only the person

with the correct decryption keys can decipher an encrypted message. Everyone has their own encryption and decryption keys. The keys must be made in such a way that the decryption key cannot be easily deduced from the public encryption key.

III. RC4 ALGORITHM

RC4 runs in two phases. The first part is the key scheduling algorithm KSA which takes an array S or S-box to derive a permutation of (0; 1; 2;.....;N-1) using a variable size key K. The second part is the output generation part PRGA which produces pseudo-random bytes using the permutation derived from KSA. Each iteration or loop or `round' produces one output value. Plaintext bytes are bit-wise XORed with the output bytes to produce ciphertext. In most of the applications RC4 is used with word length n = 8 bits and N = 256. The symbol *l* denotes the byte-length of the secret key.

The concept of security is generally interpreted as the idea of confidentiality of data being transmitted, particularly the digital information transmitted over the wireless network. Most commonly security is provided using cryptographic primitives. As shown in Fig. 1 the cryptographic primitives are classified into three main categories; not using key, symmetric key and asymmetric key [1]. Although Fig. 1 is not presenting an exhaustive list of these primitives but is highlighting the important and relevant areas. In this paper we have focused on symmetric key ciphers which are also known as secret key or single key ciphers. Secret key ciphers are further classified as block ciphers and stream ciphers. In block ciphers, a block of bits/bytes is processed at a time. DES, IDEA, RC5, AES, BLOWFISH, TWOFISH are the different available block ciphers. Whereas in stream ciphers one bit or a byte of data is processed at a time. Stream ciphers are further classified as synchronous and self-synchronous stream ciphers. Synchronous stream ciphers (SSC) are prominently discussed in literature. However, generally due to the design problems, self-synchronizing stream cipher (SSSC) are not much explored in literature and are less used in practice [2]. Different synchronous stream ciphers available in the literature are RC4, E0 (a stream cipher used in Bluetooth), A5/1 and A5/2 (stream ciphers used in GSM), SNOW 3G, ZUC (4G stream ciphers), Rabbit, FISH, and HC-256 etc.

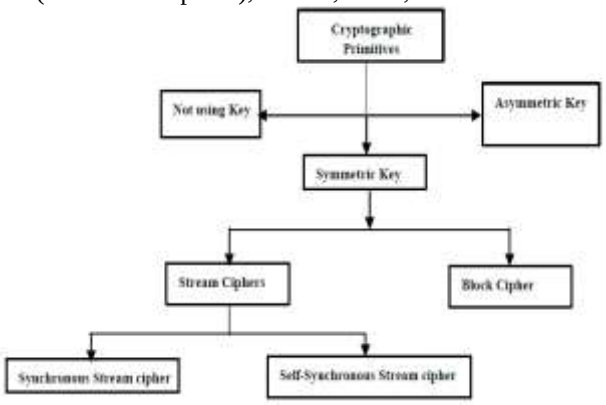


Fig. 1: Cryptographic Primitives

IV. AODV PROTOCOL

The AODV [7] routing protocol is a reactive routing protocol; therefore, routes are determined only when needed. Figure 1 shows the message exchanges of the AODV protocol. Hello messages may be used to detect and monitor links to neighbors. If Hello messages are used, each active node periodically broadcasts a Hello message that all its neighbors receive. Because nodes periodically send Hello messages, if a node fails to receive several Hello messages from a neighbor, a link break is detected. When a source has data to transmit to an unknown destination, it broadcasts a Route Request (RREQ) for that destination. At each intermediate node, when a RREQ is received a route to the source is created. If the receiving node has not received this RREQ before, is not the destination and does not have a current route to the destination, it rebroadcasts the RREQ. If the receiving node is the destination or has a current route to the destination, it generates a Route Reply (RREP). The RREP is unicast in a hop-byhop fashion to the source. As the RREP propagates, each intermediate node creates a route to the destination. When the source receives the RREP, it records the route to the destination and can begin sending data. If multiple RREPs are received by the source, the route with the shortest hop count is chosen [10].

As data follows from the source to the destination, each node along the route updates the timers associated with the routes to the source and destination, maintaining the routes in the routing table. If a route is not used for some period of time, a node cannot be sure whether the route is still valid; consequently, the node removes the route from its routing table.

If data is following and a link break is detected, a Route Error (RERR) is sent to the source of the data in a hop-byhop fashion. As the RERR propagates towards the source, each intermediate node invalidates routes to any unreachable destinations. When the source of the data receives the RERR, it invalidates the route and reinitiates route discovery if necessary.

V. RESULT AND DISCUSSION

We have developed algorithm in MATLAB that generates a WSN network of MxM field size with N number of nodes. The size of WSN and the nodes i.e. M, N parameters are given by user and as per the user choice this algorithm develops a WSN network with all the nodes are distributed randomly. For N nodes we have consider any one of the node as the source and another as destination node. After generation and distribution of WSN nodes the AODV routing algorithm is applied to make the path for data transmission in between the source and destination node. This multi-hop routing is as established the algorithm applies ciphering of data packets and

Data	**RC4**					**RSA**				
	Na40	**Nb40**	**Nc40**	**Nd40**	**Ne40**	**Na40**	**Nb40**	**Nc40**	**Nd40**	**Ne40**

500	8.592546	7.978491	7.937985	7.896030	8.165513	42.901571	45.079291	47.009096	43.150135	42.129754
1000	7.997469	8.003174	7.998916	7.975573	7.974537	75.953118	87.984796	91.920606	86.826139	83.686129
1500	8.087748	8.101132	8.026273	8.169917	8.158472	122.934412	128.801114	123.339601	123.644320	119.945194

Table 1: Time consumed in RC4 and RSA ciphering by AODV route generation for 40 nodes.

Then the ciphered packets are transferred from source to destination through the route developed by AODV protocol.

We have compared the RC4 and RSA ciphering techniques performance for different types of networks having 40, 60, 80 and 100 nodes and at the transmission rate of 500.1000 and 1500 data packets. The time taken by both ciphering techniques is observed for different configuration of network named as Na, Nb, Nc, Nd and Ne and results are tabulated in next paragraphs.

Table 1 shows the results for network Na, Nb, Nc, Nd and Ne at 40 nodes for both RC4 and RSA ciphering based transmission over the AODV generated route The time consumed in each network is given in sec.

Similarly table 2, 3 and 4 are for the 60, 80 and 100 node networks. The analysis is performed in terms of time consumed in WSN generation, AODV routing, ciphering and deciphering of the numeric data.

Data	**RC4**					**RSA**				
	Na60	**Nb60**	**Nc60**	**Nd60**	**Ne60**	**Na60**	**Nb60**	**Nc60**	**Nd60**	**Ne60**
500	7.897509	7.948273	7.882802	7.924918	7.947482	49.244712	39.241486	44.817470	42.310711	42.944932
1000	8.038558	8.032278	8.260771	8.208563	8.027609	83.222015	81.250568	83.017903	80.794498	78.830936
1500	8.027609	8.227012	8.259446	8.251064	8.202301	133.870420	123.819234	130.615140	122.972483	123.061509

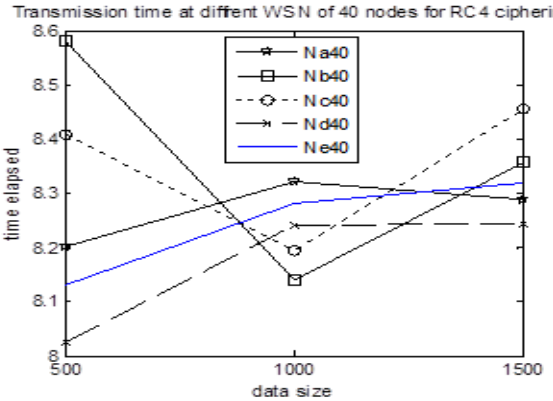
Table 2: Time consumed in RC4 and RSA ciphering by AODV route generation for 60 nodes

Data	**RC4**					**RSA**				
	Na80	**Nb80**	**Nc80**	**Nd80**	**Ne80**	**Na80**	**Nb80**	**Nc80**	**Nd80**	**Ne80**
500	8.555675	8.335917	8.399778	8.130358	8.536544	46.123596	43.660926	41.027254	41.590748	45.299263
1000	8.116473	8.081653	8.087842	8.189053	8.088512	89.818120	85.193248	83.357719	86.969831	87.353243
1500	8.299645	8.218739	8.116158	8.150069	8.254012	125.319694	123.990089	129.458053	125.166333	125.137945

Table 3: Time consumed in RC4 and RSA ciphering by AODV route generation for 80 nodes

Data	**RC4**					**RSA**				
	Na100	**Nb100**	**Nc100**	**Nd100**	**Ne100**	**Na100**	**Nb100**	**Nc100**	**Nd100**	**Ne100**
500	8.200726	8.579709	8.408091	8.025483	8.130306	43.872449	43.168072	38.398452	37.352918	43.605436
1000	8.321029	8.140178	8.193989	8.241613	8.281248	85.907631	82.249823	90.144863	85.820330	79.675313
1500	8.288994	8.355415	8.455887	8.244264	8.318857	117.090843	129.496905	125.559064	117.515919	124.664330

Table 4: Time consumed in RC4 and RSA ciphering by AODV route generation for 100 nodes



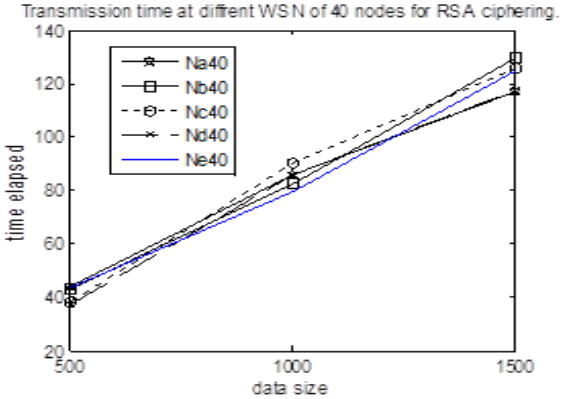


Fig. 1: Transmission time required at WSN of 40 nodes at different packet size of 500,1000 &1500 For RC4(left) and RSA ciphering(right) using table 1.

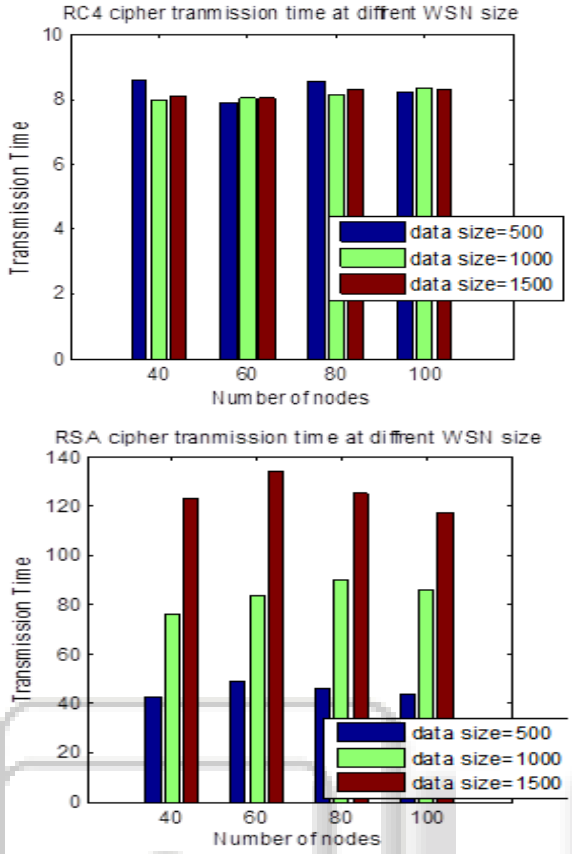


Fig. 2: Transmission time at network size of 40,60,80 and 100 nodes where each bar represent time elapsed at particular data packet size of 500,1000 and 1500 for RC4(left) and RSA(right) using table 1 to 4.

VI. Conclusion

This article demonstrates one of the aspects of WSN network called as data security. We have focused on ciphering techniques performance over the transmission delay. For this purpose different WSN networks with variety of sensor node distribution at different nodes are observed in terms of time consumed in transmission mode with AODV routing time prior to RC4 and RSA ciphering. It has been observed that for all the cases RC4 ciphering consumes less time as compared to RSA ciphering. Hence it proves that for WSN networks RC4 ciphering provides higher transmission rate due to small time consumption in ciphering deciphering. In future we can also check performance for routing technique other than AODV. We may also consider composite routing mechanism that involves artificial intelligence tools for determining the shortest possible route in minimum time. It can also helps in minimizing time delay in data transmission in WSN network with high security concerns.

References

[1] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Hand- book of Applied Cryptography. CRC Press, August 2011 edition, 1996. Fifth Printing.

[2] Douglas R. Stinson. Cryptography: Theory and Practice. CRC Press, third November 2005) edition, 1995.

[3] Thomsan D. Bruce D. Arjen L. and Mark M.,"On the Factoring of RSA-120", (169), pp.166-174, 1994

[4] Cavallar S, Dodson B, Lenstra A, Leyland P,Lioen W, Montgemery P, Murphy B, and Zimmermann P, "Factoring of RSA-140 using the number field sieve", 1999

[5] Eric W "Prime Factorization Algorithm", Mathworld.woiframe.com/news/ 2003

[6] Bahr F, Boehm M, Franke J and Kleinjung T, "For the Successful Factorization of RSA-200" www.rsasecurity.com

[7] C. E. Perkins, E. M. Belding-Royer, and S. Das. Ad hoc On- Demand Distance Vector (AODV) Routing. RFC 3561, July 2003.

[8] Diffie W and Hellman M, "New Direction in Cryptography, IEEE Transaction on Information Theory, IT-22(6): 644-654, 1976

[9] Rivest R, Shamir A and Adelman L, "A Method for Obtaining Digital Signature and Public Key Cryptosystems", Communications of the ACM, 21, pp. 120-126, 1978

[10] C. E. Perkins and E. M. Royer. The Ad hoc On-Demand Distance Vector Protocol. In C. E. Perkins, editor, Ad hoc Networking, pages 173–219. Addison-Wesley, 2000.

[11] Priteshkumar Prajapati et. al., "Comparative Analysis of DES, AES, RSA Encryption Algorithms", International Journal of Engineering and Management Research, Volume-4, Issue-1, February-2014.

[12] Sourav Sen Gupta and Subhamoy Maitra, "(Non-) Random Sequences from (Non-) Random Permutations—Analysis of RC4 Stream Cipher" J. Cryptol. (2014) 27: 67–108.

[13] Avala Ramesh et. al., "Analysis On Biometric Encryption using RSA Algorithm", International Journal Of Multidisciplinary Educational Research, Volume 2, Issue 11(2), October 2013.

[14] Ayesha Khan, "Geo Location Based RSA Encryption Technique", International Journal on Advanced Computer Theory and Engineering (IJACTE), Volume-2, Issue-2, 2013.

[15] Sourav Sen Gupta et. al., "Dependence in IV-related bytes of RC4 key enhances vulnerabilities in WP" IACR 2014.

