ISSN: 2321-7782 (Online)

Volume 3, Issue 3, March 2015

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Energy Analysis of Algorithms in Public Key Cryptography of WSN

Kunal Narendra Bidkar
T.E Computer Engineering Department
Sinhgad Institute of Technology and Science (Pune University)
Pune, India

Abstract: Cryptography is a process of storing and transmitting the data in a secure manner such that only the person can read that information to which it was intended and process it. As wireless sensor networks have low speed, are less secure, we require cryptographic algorithms .The goal of this paper is to provide a statistical comparison of the two algorithms used in public key cryptography .This paper consists of two algorithms which are RSA and ECC. The main objective of this paper is to compare the various parameters such as Battery, Power consumption, Handshakes etc. in terms of energy of these algorithms. As the battery life in Wireless sensor networks is limited, we should make optimum energy usage algorithms.

Keywords: Cryptography, RSA, ECC, Wireless Sensor Networks, Public Key Cryptography.

I. INTRODUCTION

Cryptography is a technique of making the communication more secure from the adversaries. Today's cryptographic techniques are mainly focused on complex mathematical theory. Cryptography consists of Confidentiality and Authentication. For example, in a company, if the co-workers want to exchange the information, then the communication shouldn't be tampered, modified and compromised by any adversary, this is confidentiality. We all require online banking transactions regularly, but what if the website is not the real or actual one where we are submitting our data about debit cards etc. That's why to avoid this, we need authentication. In public key cryptography, the public key is widely spread but its paired private key must remain secret. We use public key for encryption and the private key for decryption. Due to the availability of inexpensive radio transceivers, new types of application were evolved which has created various challenges in front of researchers.

For many applications, energy is a limited resource. To estimate the energy demands of the public key cryptography, we quantify the energy cost of its algorithms in various graphs, charts etc. Furthermore, the impact of public-key cryptography on battery life and comparison public-key cryptography algorithms to factors including energy consumption, such as idle listening, data reception and transmission, symmetric cryptography, etc.

II. RELATED WORK

Earlier work by Gura showed public-key cryptography to be computationally feasible on 8-bit devices. However, the paper only considers processing time and memory requirements and does not analyze energy costs or the application of public-key operations in security protocols.

Carman estimated the energy usage of several public and private-key algorithms.

Goodman and Chandrakasan implemented a public-key coprocessor for RSA and ECC.

According to some researchers, ECC can yield a level of security with a 164-bit key that other systems require a 1,024-bit key to achieve. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications.

III. RSA ALGORITHM

RSA algorithm got its name from the initials of the surname of developers namely Ron Rivest, Adi Shamir, and Leonard Adleman who are three MIT researchers.

RSA is one of the first practicable public-key cryptosystems and is widely used for secure data transmission since 1977, which contains a public key and a private key.

RSA algorithm consists of three steps such as key generation, encryption and decryption.

The process speed of RSA algorithm is very fast. It has less security and it is best suited for multi-user environment. Digital signature for authentication is also provided by RSA algorithm.

RSA algorithm use different key for encryption and decryption.

RSA algorithm consists of following steps:

Step 1) Choose any two prime numbers ‘a’ and ‘b’ which should be distinct.

Step 2) Now, we take multiplication of these two numbers and store the value in a variable ‘n’.

Step 3) Choose a variable ‘z’ such that (a-1)(b-1) will be equated to z.

Step 4) Now, choose a variable ‘e’ such that 1<e<z, where ‘e’ is a prime number and is also co-prime with n.

Step 5) Next step is, choose a variable ‘d’ such that (d*e)%z=1.

Step 6) Let ku be the public key where ku=(e,n) and kr be the private key where kr=(d,n)

Note: We can also exchange positions of ‘e’ and ‘d’, there is no problem.

Step 7)Encryption : $c=m^e$ mod n where c is the encrypted message.

Step 8)Decryption : c^d mod n=m

For example, Consider 2 prime numbers such as 3 and 11.

Step 1 : a=3 , b=11

Step 2: n=a*b=3*11=33

Step 3: z=(a-1)(b-1)=(2)(10)=20

Step 4: Let e=7

Step 5: (d*e)%20=1

Now, we have e=7 so, if we take d=3 then 7*3=21 and 21%20 is 1.

Hence, d=3

Step 6: k_u=(7,33) k_r=(3,33)

Consider, if we have a simple message, m=2 , We can encrypt it as 2^7mod33 which is 29

On the decrypting end, 29^3mod33 we get the answer as 2 which is the original message.

Encryption:

The public key is transmitted globally and the private key is kept secret.

If suppose, the message to be send is ‘M’ then he first turns the ‘M’ into ‘m’ which is an integer value . He then computes the cipher text with,

$C=m^{e}$ (mod n)

Decryption:

The person then receives this cipher text and can decrypt the messing using the private key let’s say ‘d’ by calculating,

$m=c^{d}$ (mod n),

where d=private key, c=cipher text, m=encrypted message. With this the person can get the original message M.

IV. ECC Algorithm

ECC stands for Elliptic curve cryptography.

It provides same level of security, with small key sizes .Reduces storage as size is small.

It is very complex as there is lot of math involved, it’s very detailed. It may use real numbers, fractions, integers etc.

An elliptic curve is a curve which consists of points satisfying the equation $y^{2}=x^{3}+ax+b$.

Elliptic curve systems as applied to cryptography were first proposed in 1985 independently by Neal Koblitz and Victor Miller.

For example, consider $y^{2}=x^{3}+2x+3$

If we put x=0, then $y^{2}=3$ which has no solution (mod 5)

If we put x=1, then $y^{2}=6$ mod 5 =1 which yields y=1, 4 since 6%5=1, and $4^{2}=16$, and 16%5=1.

If we put x=2, then $y^{2}=15$ mod 5 =0 which yields y=0.

If we put x=3, then $y^{2}=36$ mod 5 =1 which yields y=1, 4 since 6%5=1, and $4^{2}=16$, and 16%5=1.

If we put x=4, then $y^{2}=75$ mod 5 =0 which yields y=0.

Hence when we plot (x,y) we get the following points:

(1,1),(1,4),(2,0),(3,1),(3,4),(4,0).

An elliptic curve over a field K is nonsingular cublic curve into two variables f(x,y)=0 with a rational point.

The field K is usually taken to be the complex numbers, reals, rationals, algebraic, extensions of rationals, p-adic numbers or a finite field.Their characteristic make them ideal for use in smart cards and other environments where features such as storage, time and energy are limited. The majority of public key systems in use today use 1024-bit parameters for RSA and Diffie-Hellman.

Following is the table showing the Key sizes of RSA and ECC algorithms along with Symmetric key sizes.

Encryption: A plain text ‘L’ is encrypted to cipher text ‘C’ using public exponent ‘e’ and modulus ‘M’ as $C=L^{e}$ mod M.

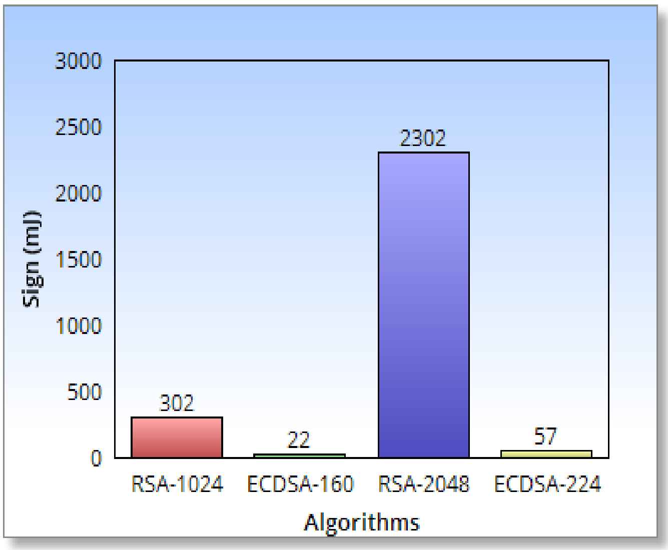
Decryption: A cipher text ‘C’ for a given plain text is decrypted using private exponent d and modulus M as $L=C^{d}$ mod M.

Symmetric key Size in Bits	RSA key size in bits	ECC size in bits
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

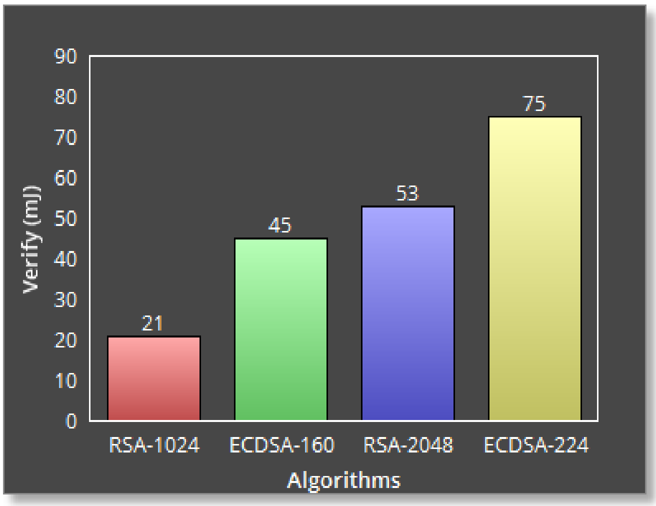
Table 4.1

Table 4.1 shows the key sizes of respective algorithms in bits. With an observation we can clearly see that as the symmetric key size increases, the key sizes for RSA key increase faster than the ECC key size. Therefore, we can state that EC systems can offer more security per bit increase in key sizes. In the elliptic curve case, there is actually one additional bit that needs to be transmitted in each direction which allows the recovery of both the x and y coordinates of an elliptic curve point. Faster, smaller, and more efficient cryptographic keys are created. ECC generated key through the elliptic key equation rather than the traditional way of prime numbers. Manufactures such as Motorola, 3COM, Siemens, TRW have included support for their products in ECC.

ENERGY COMPARISON OF BOTH THE ALGORITHMS:

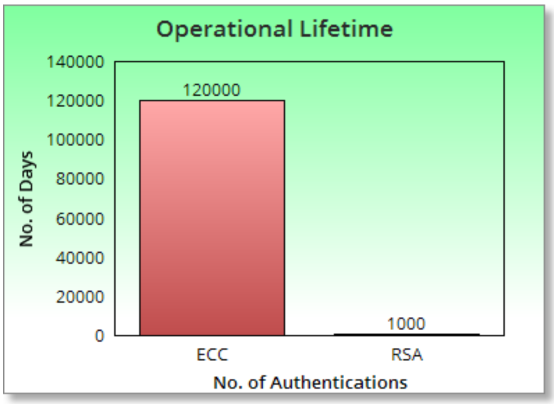


A digital signature is an electronic form of a signature that can be used to authenticate the identity of the sender, and also ensure that the content is original and document that has been sent is unchanged. We can clearly cite from the graph that Energy required generating the Digital Signature in RSA algorithm is more than Energy required for ECC algorithm.



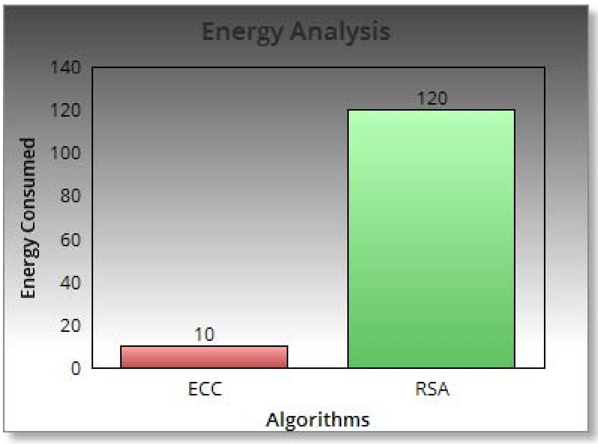
As we are generating digital signatures, it's also important to verify it. As energy is required to generate, it's also needed to verify it. We can clearly cite from the graph that Energy required verifying the Digital Signature in RSA algorithm is less than Energy required for ECC algorithm.

Operational Lifetime of both the Algorithms:



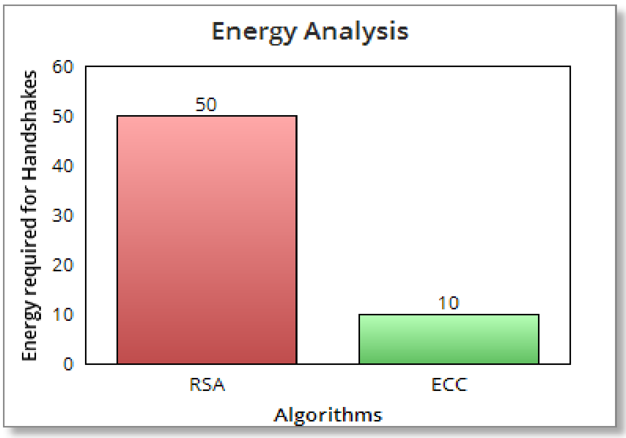
Following is the graph plotted of No. of Authentications VS No. of Days. It shows us that the operational time for ECC algorithm is almost 12 times more than that of RSA algorithm. Hence, as the lifetime is increased by 12 times, the battery power would be less consumed. As a result, the sensor nodes will be profited as the less battery would be required, resulting high lifetime of the sensor node.

ENERGY CONSUMED BY BOTH THE ALGORITHMS:



Following is the graph plotted of Algorithms VS Amount of Energy consumed. It shows us that the energy consumed by ECC algorithm is almost 12 times less than that of RSA algorithm. As energy is a limited resource, it would be much profitable if we make use of ECC algorithm which provides the same level of security as the RSA algorithm.

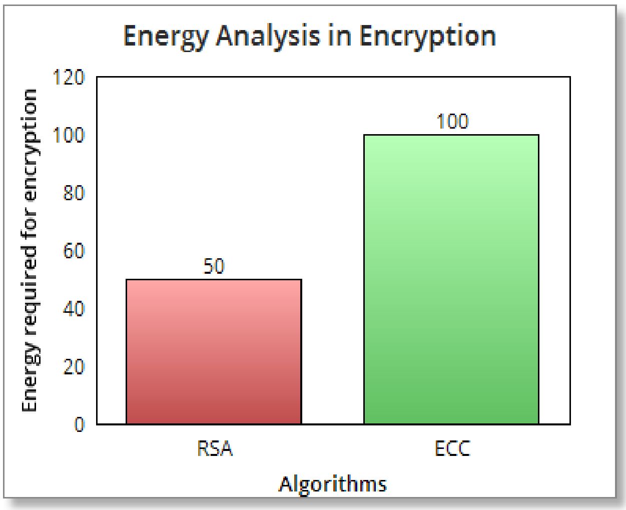
ENERGY REQUIRED FOR HANDSHAKES:



Symmetric-key encryption is faster than public-key encryption, but public-key encryption provides more effective authentication techniques. An SSL(Secure Software Layer) session always begins with an exchange of messages called an SSL handshake. The SSL handshake allows the server to authenticate itself to the client by using public-key techniques. It then allows the client and server to cooperate in creating symmetric keys that are used for encryption, decryption, and tamper detection during the SSL session that follows. The SSL handshake can also allow the client to authenticate itself to the server. So, during this exchange, lot of energy is required. Hence, if energy requirement can be reduced, it would lead to efficiency.

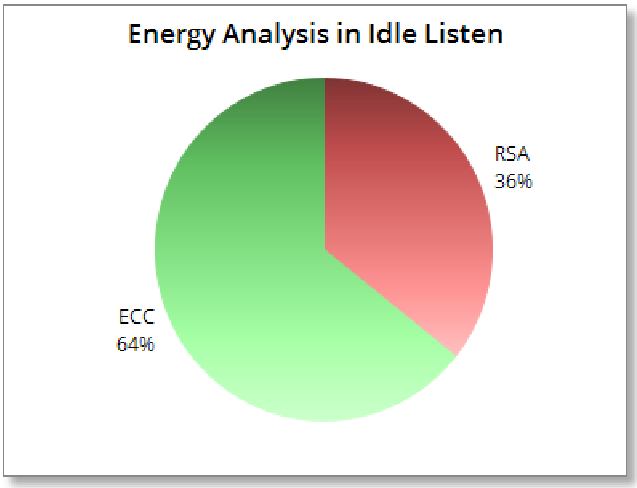
From the above graph, we can observe that ECC requires less energy for handshakes.

ENERGY ANALYSIS IN ENCRYPTION:



Encryption, in very simple terms can be stated as, the process of converting the data to be sent into cipher text so that only authorized user can read the data. Data is converted into cipher text using pseudo random generator. So, again energy requirement of key generation comes into picture. This energy required for key generation should be as small as possible. From the graph, we can see that energy required for RSA is less than ECC. Hence, until now, we can state that under what requirements and circumstances which algorithm is suitable. The energy plotted on y-axis is in mJ.

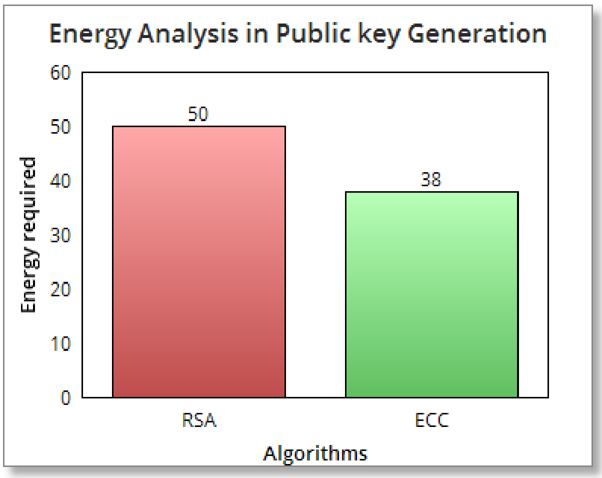
ENERGY ANALYSIS IN IDLE LISTEN:



Idle listen is the state in which transceiver is active but has no data to transmit or to receive.

As the graph shows that most of the time in ECC algorithm, the transceiver is in idle state. Hence it can be utilized for some other work. RSA has less idle listen time because, most of the energy gets consumed in handshakes, so very less amount of energy is left out for idle listening.

ENERGY ANALYSIS IN PUBLIC KEY GENERATION:



From the above graph we can state that if RSA algorithm requires 50% of energy for public key generation, then ECC algorithm requires only 38% energy for key generation. Symmetric key algorithms require only a single key to be generated, but in public key cryptography two keys are generated public and private as discussed earlier. Keys are usually generated randomly, using a pseudo random generator. The key size should be as long as possible and it should contain a complex mathematical function which is very difficult to decode. Hence, some energy is required during this process. From the above graph, energy required for ECC algorithm is again less as compared to RSA algorithm.

The size of the key in ECC algorithm is logarithm of number of points on the chosen prime sub group of points on the elliptic curve. The key size in ECC is very small, which provides a lot of security.

V. Conclusion

The United States, the UK, Canada and certain other NATO nations have all adopted some form of elliptic curve cryptography. Still in India, we are using RSA algorithm (Survey : Feb 2014) hence, ECC algorithm should be replaced with RSA under beneficial circumstances. For faster cryptographic operations and reliability , ECC can be implemented in hardware chips also.

RSA algorithm can be used where fast and easy implementation is required.

Analysis shows that ECC algorithm is much more energy efficient is several aspects than the RSA algorithm.

In addition to the computational benefits of ECC, its smaller keys and certificates lead to significant savings in public-key communication costs. With a given energy, we can perform 4.2 times key exchange operations with ECC as compared to RSA.

ECC algorithm can save 12 times energy than RSA algorithm. Since, ECC offers better security features and withstand attacks when compared to other cryptosystems it is feasible to use ECC in Distributed sensor networks with an additional consumption of very few units of energy.

References

1. A. Perrig, J. Stankovic and D. Wagner, “Security in Wireless Sensor Networks”, Communications of the ACM, Vol. 47 No. 6, June 2004.
2. N. Gura, A. Patel, A. Wander, H. Eberle, S. Chang Shantz, “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs”, CHES, August 2004.
3. D. W. Carman, P. S. Kruus, and B. J. Matt, “Constraints and Approaches for Distributed Sensor Security”, Network Associates Labs Tech. Rep. 2000.
4. L. Yuan and G. Qu, “Design Space Exploration for Energy-Efficient Secure Sensor Network”, ASAP 2002.
5. N. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, “Analyzing the Energy Consumption of Security Protocols”, ISLPED 2003.
6. C. K. Koc, "High-speed RSA implementation", Tech. Rep. TR 201, RSA Laboratories, November 1994.

AUTHOR(S) PROFILE

Kunal Bidkar, pursuing the B.E degree in Computer Engineering at Sinhgad Institute of Technology and Science, Pune. He has achieved certifications from Stanford Online courses (MOOC), University of New Mexico, University of Maryland, Microsoft (pursuing). He has done projects on web development using PHP, HTML and Javascript. His interest include, Php , Java, c++ programming , security , databases. He is an active member of an NGO in Pune.

He possesses good leadership skills as he was an active volunteer at blood donation camp. Furthermore, is a Swimmer having a consistency of 11 years. Finally, he has a deep interest in doing research in the field of Network security, public key cryptography.