Performance Analysis of Cryptographic Protocols to Enhance SMS and M-Commerce Security

Nikhil B. Khandare,
Department of Master of Computer Application, Veermata Jijabai Technological Institute
Mumbai - 400019, Maharashtra, India.

Abstract*—*This paper explores the issue of one time password which is sent in online electronic credit card transactions from payment gateway server to the customer. Confidentiality should be maintained in exchange of SMS between the parties. In section II, three solutions are proposed for secure transmission of this SMS, first is end to end encryption between the parties' i.e. symmetric and asymmetric encryption, their performance is also analyzed. In second solution Elliptic curve Diffie-Hellman key exchange is used to share key between the two parties. In third solution BB84 protocol of Quantum cryptography is used to share the SMS between two parties. In section III observations of research are discussed which suggest logically that Elliptic curve Diffie-Hellman is most secure for key exchange. Section IV concludes the paper and extension of this research is given in future scope.

Keywords*—*Symmetric cryptography, assymetric cryptography, Elliptic curve Diffie-Hellman, Elliptic curve cryptography, Quantum cryptography.

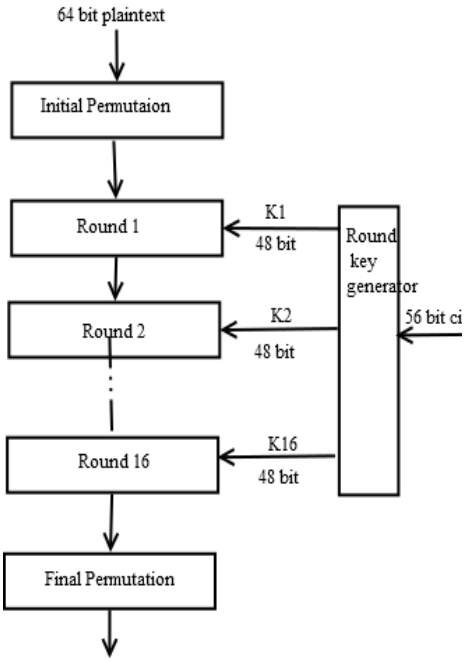
I. INTRODUCTION

A. Basic Cryptographic Paradigms

Privacy, data integrity, authentication and non-repudiation are the basic paradigms of security; every system should fulfill these requirements in order to be secure. Privacy or confidentiality is ensuring that the information can be accessed by authorized parties only, since unauthorized user can misuse the sensitive information. Data Integrity ensures information cannot be altered or tampered, integrity of the data should be maintained throughout the lifetime of data. Authentication is the process in which user credentials are compared to the credentials in the database (can be a local database or authentication server), if details match then user is given access to the system. Non-repudiation means parties cannot deny that the event has occurred i.e. sender cannot deny that the message was sent and receiver cannot deny that message was received. Various cryptographic techniques are used to achieve these goals.

B. Symmetric key Encryption Algorithms.

1. Data Encryption Standard: DES is symmetric key block cipher developed by NIST, block size and key size both are 64 bit. Effective key size is 56 bits since 8 out of 64 bits are not used. Structure of DES is given in Fig. 1.
2. Advance Encryption Standard is symmetric key block cipher and has 128 bit data size and 128/192/256 bit key size. Number of rounds in AES is variable, for 128 bit key size number of rounds is 10, for 192 bit key size number of rounds is 12 and for 256 bit key size number of rounds is 14. Structure of AES is given in Fig. 2.



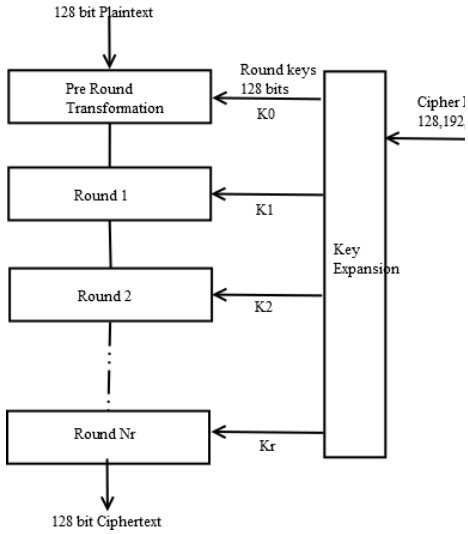


Fig. 2. AES Encryption Algorithm

Assymetric Key Encryption Algorithms.

1. RSA Algorithm

RSA Algorithm is the most popular and widely used algorithm in asymmetric key cryptography. Steps in algorithm are

a. Choose two large primes p and q and calculate N= p*q.
b. Then we calculate Ø (N)=(p-1)(q-1).
c. Private key 'd' and Public key 'e' are chosen such that they are multiplicative inverse of each other i.e. (e*d)= 1 mod Ø (N).
d. Keys 'p' and 'q' are not public. If one wants to discover them, one has to factorize large prime 'N'.
e. Security of RSA depends on the difficulty of integer factorization problem, If one can factorize 'N', he can break the algorithm.
f. Encryption of message in RSA

Ciphertext $C = M^e \pmod N$

g. Decryption of Ciphertext in RSA

Message $M = C^d \pmod N$.

2. Elliptic Curve Cryptography

ECC is another public key cryptography which attains same level of security as RSA with very small key size. The equation of an elliptic curve is given as, $y^2=x^3+ax+b$. Implementation of an Elliptic Curve Cryptosystem from the MasseyOmura system is discussed [1].

a. Let E be a fixed elliptic curve over a finite field Fp^k ,any point P ∈ E, P added to itself N times produces the identity (the point at infinity); NP = id the notation nQ for integer n and point Q ∈ E means the n-fold addition of Q with itself .

b. To transmit message m to Bob, Alice first represents m as a point X_0 on the elliptic curve E. Alice now chooses a random integer $0 < c < N$ with gcd(c, N) = 1 and sends Bob cX_0

c. Bob chooses another random integer $0 < d < N$ with gcd(d, N) = 1 and adds the point he received to itself d times, yielding $d(cX_0)$, and sends the result to Alice

d. Since gcd(c, N) was 1, Alice may compute $c^{-1} \pmod N$; Alice then sends Bob $c^{-1}\, d(c(X_0)))$.

e. Finally, Bob similarly calculates d^{-1} and computes $d^{-1}\,(c^{-1}\,(d(cX_0)))$

f. Clearly for this cryptosystem to work $d^{-1}\,(c^{-1}\,(d(cX_0)))$ must equal X_0.

g. That is message M, which was represented as point on elliptic curve.

C. Diffie Hellman Key Exchange Algorithm.

Diffie-Hellman key exchange is used when two parties want to share a secret key but communication channel is insecure. Steps in key exchange protocol are

1. Both the parties agree on prime p and nonzero integer g modulo p.
2. Alice picks secret random integer a and bob picks a secret random integer b. Alice calculate $A=g^a \pmod p$ and bob calculates $B=g^b \pmod p$.
3. Alice sends 'A' to bob and Bob sends 'B' to alice.
4. Alice calculates $A'=B^a \pmod p$ and bob calculates $B'=A^b \pmod p$.
5. Thus shared secret key A' and B' are same i.e. $g^{ab} \pmod p$.

D. Onsite Elecrtonic Transactions [3].

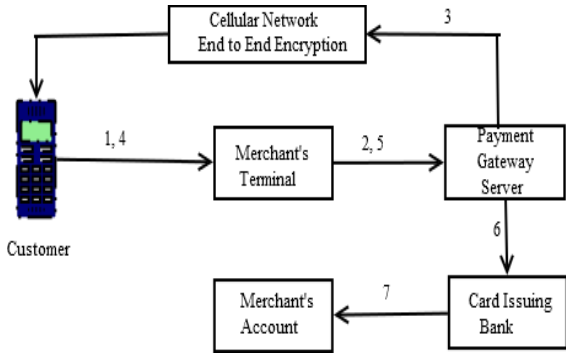


Fig. 3. Onsite Electronic Transaction

Steps in onsite transaction processing

1. Cardholder gives his smartcard at merchant terminal, payment details are fed and card is swapped at merchant terminal
2. Credit card details are encrypted and then forwarded to payment gateway server

3. Payment Gateway server then sends the one time password to customer via cellular network where end to end encryption is applied.
4. Customer enters received one time password at merchant terminal
5. Merchant terminal forwards this OTP to payment gateway server and verifies the authenticity of customer
6. After authentication payment gateway server forward payment details to card issuing bank.
7. Card issuing bank transfer funds to customer account.

II. Proposed Solutions

In onsite electronic transactions [3], when SMS is sent from payment gateway server to customer it should secure. More secure cryptographic techniques should be used for exchange of data between PGS and customer. In our paper we propose three solutions for secure transmission of this SMS.

A. End to end encryption using between PGS and customer using
   a. By symmetric key algorithm like AES and DES.
   b. By asymmetric key algorithm like RSA and ECC.

   These algorithms are applied and their performance is analyzed for 160 characters/ 1 text message/ 1K data.

   Also data size is increased to 2K, 3K, 4K and corresponding encryption and decryption times are measured.

B. Use of EC-Diffie-Hellman Key Exchange to transmit OTP from PGS to Customer over insecure channel.

C. Key exchange between Payment Gateway Server and Customer using Quantum cryptography.

A. Encrypted communication between Payment Gateway Server and Customer

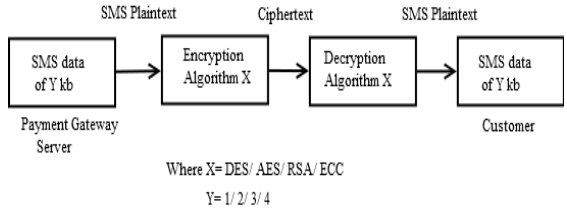


Fig. 4. Encrypted communication

A text message consisting one time password can be considered as alphanumerical data and can be encrypted or decrypted using any cryptographic algorithms, message is encrypted using both symmetric (AES and DES) and asymmetric encryption (ECC and RSA) algorithm. Message is encrypted at Payment Gateway Server side and decrypted at customer side.

Experimental setup and Technologies used

Java development kit 8 is used and all the cryptographic algorithms were implemented in java. All the algorithms were executed on machine with core-2-duo processor and 3 GB RAM and 64 bit windows operating system.

Message to be encrypted was same for all algorithms which was "OTP is 709889 please don't share" and time for signature and verification were calculated by adding timer in code of each RSA, ECC, AES and DES. Each message of 1K size was assumed to be 160 characters long and results were extended for 2K, 3K, 4K data and performance of each algorithm were observed. All times were calculated in seconds.

In RSA algorithm encryption key e was taken 28199 which is 16 bit, corresponding decryption key d was 8639 and product of two primes (N) was 29999. Java code for RSA on the above specified machine was executed.

In ECC messy-omura cryptosystem was designed in java and also hybrid encryption ECC-AES was done on same text message in Cryptool and time for encryption and decryption was measured.

In AES input vector is initialized as a raw data and plaintext is same as above and encryption key is randomly chosen.

Similarly DES was implemented in java and corresponding encryption, decryption time was calculated. Experimental result obtained is shown in tables.

Experimental results were also represented in bar graph forms plotting size of data versus time in seconds, same results were represented with line graph with size of data on one axis and time in second on another axis. For encryption of 1K data, time required is infinitesimally small as compared to encryption of 2K data. First reading is for data less than or equal to 160 characters or 1 KB thus first reading was sometimes found deviating from results as compared to 2, 3, 4KB of data. Also there are various factors affecting the encryption and decryption time which may be number of processes executing, temperature etc

Size	AES Algorithm (time in seconds)		DES Algorithm (time in seconds)	
	Encryption	Decryption	Encryption	Decryption
1K	0.013727406	0.000444497	0.016442981	0.000436969
2K	0.014450442	0.000419518	0.017747045	0.000692238
3K	0.018437231	0.000486244	0.027137515	0.000925267
4K	0.016389942	0.000718245	0.022639938	0.001154868

TABLE I. Encryption and Decryption Times using Symmetric Key Cryptographic Algorithm

Size	RSA Algorithm (time in seconds)		ECC-AES Algorithm (time in seconds)	
	Encryption	Decryption	Encryption	Decryption
1K	0.003815683	0.004228015	0.075	0.031
2K	0.018390631	0.029998508	0.0266	0.016
3K	0.022525571	0.037964873	0.047	0.031
4K	0.033078156	0.03236881	0.047	0.014

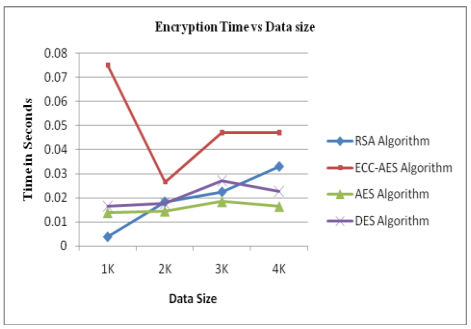


Fig. 5. Line graph asymmetric encryption

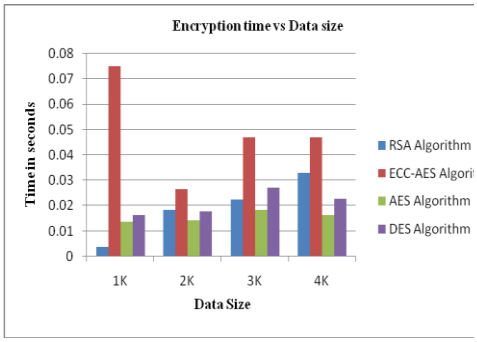


Fig. 6. Bar graph asymmetric encryption

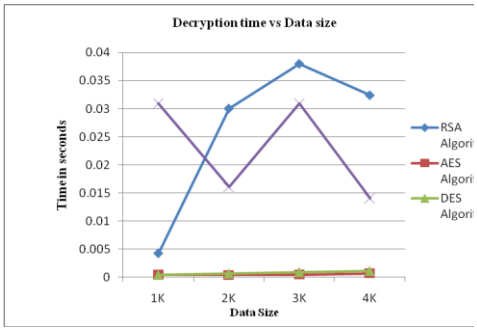


Fig. 7. Line graph asymmetric decryption

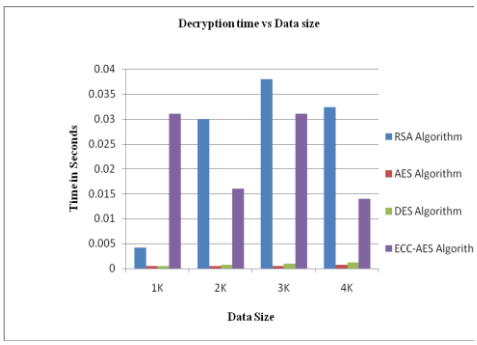


Fig. 8. Bar graph asymmetric decryption

.

B. *Exchange to transmit One Time Password from Payment Gateway Server to Customer over insecure channel.*

When payment gateway server wants to send the OTP to customer, both the parties can agree on an Elliptic curve over a finite field $E(F_q)$ and point P on Elliptic Curve.

1. PGS chooses a secret key 'a' and calculates P_a= a*P and sends it to customer.
2. Customer chooses secret key 'b' and calculates P_b=b*P and sends it to PGS.
3. PGS has his own secret key 'a' and has 'P_b' sent by customer and hence calculate a*P_b, which is a*b*P.
4. Customer has his own secret key 'b' and 'P_a' sent by PGS and hence calculate b*P_a, which is b*a*P.
5. Thus shared secret key is a*b*P, but this is multiple of a point P on elliptical curve, now PGS and Customer agree on x-coordinate or y-coordinate of point as OTP or last eight bits of x-coordinate as key.
6. Diagrammatic representation of solution is shown in Fig. 8.

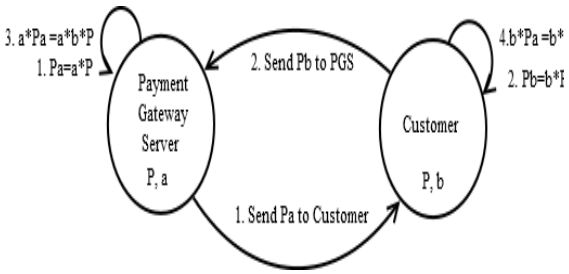


Fig. 8. Key exchange between PGS and Customer

Security Analysis

1. Method to send OTP from PGS to Customer over insecure channel is secured by Elliptic curve Deffie-Hellman problem. If an eavesdropper is able to access the channel and he gets P, a*P, b*P on the elliptic curve $E(F_p)$ but calculating a*b*P is impossible thus SMS is secured and can be accessed by customer only.
2. If intruder gets P, a*P, b*P on the elliptic curve $E(F_p)$ and also gets one point P'on $E(F_p)$ which is claimed to be product of a, b and P still attacker is unable to verify correctness of solution i.e. if P'=a*b*P. This is called Elliptic Curve Decision Deffie Hellman Problem.
3. Above two problems ECDHP and ECDDHP does not have any solution hence it is impossible to get the SMS sent from PGS to customer and this verifies the security.

C. Key exchange between Payment Gateway Server and Customer using Quantum cryptography

SMS/ OTP/ Key to be sent from PGS to customer can be sent by using quantum key distribution (QKD).

Quantum cryptography has quantum bits whose value is either zero or one. When a wave of polarized photon is passed through birefringent calcite crystal, it decomposes ray of light in two rays ordinary ray (horizontal polarization) and extraordinary ray (vertical polarization). Using calcite crystal photons which are either vertically or horizontally polarized can be detected, other photons are lost. If crystal is rotated by 45^0, diagonally polarized photons can be detected other (horizontally and vertically polarized) are lost[17].

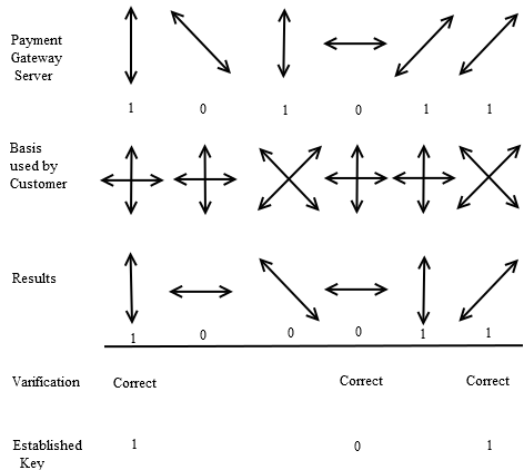


Fig. 9. Key distribution by PGS to customer by Quantum key distribution

Steps in Protocol

1. PGS sends customer randomly chosen string of bits on either rectilinear or vertical basis.
2. Customer detects them using randomly chosen basis (either rectilinear or vertical).
3. Customer informs PGS which basis he has used.
4. PGS inform customer when he has used correct basis
5. Customer discloses information about basis used and result of operation are kept secret.
6. New key is result when PGS and customer have used same basis.

Eavesdropping is not possible because it will change quantum state of photon and eve can't even clone photon without changing quantum state of photon. Thus quantum cryptography ensures highest level of security. This key distribution system assumes that customer has photon analyzer and it can choose basis for accepting a photon, but till date very few experimentation are done on Quantum cryptography but theoretically it is unbreakable till date.

III. Observations

When SMS as data is encrypted using symmetric key cryptographic algorithms like DES and AES, encryption time increases as size of SMS increases from 1K to 4K. Similarly decryption time also increases with the increase in data size. From results it can be observed that AES is faster than DES and from security point of view AES is more secure.

When SMS as data is encrypted using asymmetric key cryptographic algorithms like RSA and ECC-AES, encryption time increases with increase in data size. Similarly Decryption time also increases with increase in data size except for the first data of 1K. From security point of view ECC is more secure than RSA, ECC provide same level of security as RSA with smaller key size and also many attacks are possible on symmetric and asymmetric key cryptography[18].

Elliptic curve Diffie-Hellman key exchange should be used when communication channel is insecure. Hiffie-Hellman is secure for key exchange but Elliptic curve adds to security of Diffie-Hellman key exchange. Proposed solution of ECDH key exchange is secure by ECDDH problem and ECDHP.

BB84 protocol for quantum key distribution between PGS and customer has highest level of security theoretically but no experimentation is done till date on quantum cryptography and has assumptions.

IV. Conclusion and Future Scope

Elliptic curve cryptography should be applied for encryption and decryption if size of data is large and level of needed security is highest. RSA also achieves good security but key size increases, 1024 bit key size is current benchmark in security of RSA. Attacks are possible on symmetric key cryptographic algorithm like AES and DES and thus they are not used for higher security needs. Elliptic curve Diffie-Hellman is best solution to exchange shared secret key on insecure channel.

This work can be extended by using Quantum cryptography which depends on Heisenberg uncertainty principle and calculating encryption and decryption times of messages using quantum cryptography. These results can be compared with Elliptical curve cryptography which is believed to be highest level of security. Using ECDHP security of mobile banking can be enhanced.

References

[1] Washington, Lawrence C. Elliptic curves: number theory and cryptography. CRC press, 2008.

[2] Saxena, Neetesh, and Narendra S. Chaudhari. "A secure approach for SMS in GSM network." Proceedings of the CUBE International Information Technology Conference. ACM, 2012.

[3] Kaushik, Sona, and Shalini Puri. "Online transaction processing using enhanced sensitive data transfer security model." Engineering and Systems (SCES), 2012 Students Conference on. IEEE, 2012.

[4] Saxena, Neetesh, and Narendra S. Chaudhari. "An Approach for SMS Security using Authentication Functions." Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on (0975–8887), Singapore, Digital Object Identifier. Vol. 10. 2012.

[5] Saxena, Neetesh, and Narendra S. Chaudhari. "SecureSMS: A secure SMS protocol for VAS and other applications." Journal of Systems and Software90 (2014): 138-150.

[6] Toorani, Mohsen, and A. Beheshti. "SSMS-A secure SMS messaging protocol for the m-payment systems." Computers and Communications, 2008. ISCC 2008. IEEE Symposium on. IEEE, 2008.

[7] Toorani, Mohsen, and A. Beheshti. "Solutions to the GSM security weaknesses." Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST'08. The Second International Conference on. IEEE, 2008.

[8] Saxena, Navrati, Narendra S. Chaudhari, and Julian Thomas. "Solution to an attack on digital signature in SMS security." Modeling, Simulation and Applied Optimization (ICMSAO), 2013 5th International Conference on. IEEE, 2013.

[9] Saxena, Neetesh, and Narendra S. Chaudhari. "An Approach for SMS Security using Authentication Functions." Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on (0975–8887), Singapore, Digital Object Identifier. Vol. 10. 2012.

[10] Saxena, Neetesh, and Narendra S. Chaudhari. "Prevention of SMS against Repudiation Attack over the GSM Network." Journal of Information Assurance & Security 8.3 (2013): 156-166.

[11] Narendiran, C., S. Albert Rabara, and N. Rajendran. "Performance evaluation on end-to-end security architecture for mobile banking system." Wireless Days, 2008. WD'08. 1st IFIP. IEEE, 2008.

[12] Ma, Kun, Han Liang, and Kaijie Wu. "Homomorphic property-based concurrent error detection of RSA: a countermeasure to fault attack."Computers, IEEE Transactions on 61.7 (2012): 1040-1049.

[13] Agoyi, Mary, and Devrim Seral. "SMS security: an asymmetric encryption approach." Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on. IEEE, 2010.

[14] Bella, Giampaolo, Fabio Massacci, and Lawrence C. Paulson. "Verifying the SET registration protocols." Selected Areas in Communications, IEEE Journal on 21.1 (2003): 77-87.

[15] Sun, Hung-Min, et al. "Dual RSA and its security analysis." Information Theory, IEEE Transactions on 53.8 (2007): 2922-2933.

[16] Sim, Kwang Mong, and Raymond Chan. "A brokering protocol for agent-based e-commerce." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 30.4 (2000): 474-484.

[17] Niemiec, Marcin, and Andrzej R. Pach. "Management of security in quantum cryptography." Communications Magazine, IEEE 51.8 (2013): 36-41.

[18] Stuttard, Dafydd, and Marcus Pinto. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. John Wiley & Sons, 2011.