International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 20 (2017) pp. 10095-10100


# An Investigation of Optimal Design of Source Anonymous Message Authentication Scheme using Analytical and Computational Approach

**Uma Meena [1] and Dr. Anand Sharma[2]**

*[1, 2] College of Engineering & Technology, Mody University of Science & Technology, Lakshmangarh-332311, Sikar (Rajasthan), India.*

*[1]Orcid: 0000-0002-7691-973X.*

**Abstract**

One of the most effective techniques to prevent the tainted and adversary messages from being forwarded in wireless sensor network is message authentication. Message authentication is a short piece of information used to offer integrity, validate a message and legitimacy declarations on the message. In this paper, we have used the Source Anonymous Message Authentication scheme (SAMA) that is on the basis of elliptic curve to provide better security. Here we are using a hashing function (SHA-256) for signature generation and verification at the transmitter and receiver end which is on the basis of acceptance or rejection of the messages. SAMA technique allows nodes to authenticate messages for the purpose of saving sensor power provided source node privacy, resilience to node attacks and eliminated the problem of threshold of number of messages at the sending and the receiving end. We have evaluated SAMA and polynomial techniques in MATLAB (R2014a) instead of ns2 simulator that gives us better results in less time as compared with the previous technique and also analyzed SAMA and polynomial techniques at the transmitter and the receiver end differently.

**Keywords:** Wireless Sensor Network, SAMA Elliptic curve cryptography, SHA-256 hash algorithm, MATLAB.

## INTRODUCTION

WSN commonly comprised of a large number of resource constriction sensor nodes that are normally deployed in the unattended/hostile environment and provided that desired data security is a major concern, such as confidentiality, authenticity, and availability from which it exposed too many types of severe insider occurrences due to node conciliation.

Message authentication is a system in which the modified message will almost detect by the receiver and this technique allows the sender to send a message to the receiver [1]. Public-key cryptosystems or symmetric-key cryptosystems are the arrangements in which most of them have various limitations like communication overhead, node compromise attacks, high computational and lack of scalability. Secret polynomial and public-key based approach are the two categories which were employed in the process to achieve the qualities as mentioned in the abstract [2, 3].

The elliptic curve cryptographic progress has proved to be a boom for the public key approach. It not only saves memory usage and reduces message complexity, but also provides security. Based on elliptic curve cryptography, Li *et al.* [4] proposed a scalable authentication technique that permits any node to convey an unlimited number of messages without suffering the threshold problem while allowing intermediary node authentication. In terms of computational and communication overhead, results were evaluated using MATLAB which is more efficient than the polynomial based method. For bivariate polynomial scheme the computational overhead is determined by $\ell*(d_y +1)$ and for SAMA is determined by $4*\ell*(n+1) +n [\log_2\alpha]$, where $\alpha$ is the total number of nodes in the system and n is the number of nodes in the ambiguity set (AS).

In this paper, we have used hashing function (SHA-256) algorithm for key generation and verification [6]. This technique required key synchronization which caused a delay in the process. The main logic behind the SAMA technique is that each node generates a message authenticator for the rest of the group members in the ambiguity set [7]. This secures the privacy of the actual sender and also gives liberty to all nodes to check the originality of the message. In case an intruder takes over a node and forwards a futile message, its integrity shall not be hampered and the sink node shall be able to identify the compromised node. Issues like computational overhead, message and sender's authenticity at each intermediate node level have been trying to be dealt with in this paper. The aim of this paper is to show efficiency at the transmitter as well as the receiver's end.

## SOURCE ANONYMOUS MESSAGE AUTHENTICATION SCHEME FOR ELLIPTIC CURVE CRYPTOGRAPHY [4]

SAMA is a technique that gives originality to the message with respect to the actual and fake senders that makes it popularly known as sender's anonymity. SAMA of a message

m, S (m) is generated in a group of n members, AS or S= $\{Q_1,Q_2.....,Q_n\}$ where $Q_i$ represents the public key of each member and $Q_t$ represents the public key of the sender & where $1\leq t\leq n$ (from within the group) [4].

**Implementation of MES Scheme [5] in MATLAB with the help of Elliptic curve cryptography**

Elliptic curve equation is:

$$E: y^2 = x^3 + x \bmod w \qquad (1)$$

where w is a prime number (w>3), a,b € $F_p$ and $4a^3+27b^2\neq 0$.And a random private key, $d_A$ generated by the user, is used to compute the public key, $Q_A$. All points on the curve along with the point at infinity, O are elements of $F_P$ $G(x_G,y_G)$ is a base point with large N. Selects a random private key $d_A$ where $1\leq d_A\leq N-1$. The public key is thus computed as $Q_A=d_A* G$.

- **Generation:** With the help of this public key, $Q_A$ and ℓ leftmost bits of the cryptographic hash function (SHA-256), $h_A$, signature (r, s) is computed by the user while sending a message. After selecting k from [1, N-1] and calculating $(x_A,y_A)=k_AG$, receiver's part of the signature is computed as:

  $$r = x_A \bmod N \text{ until } r \neq 0 \qquad (2)$$

  Then after finding $h_A$ calculate sender's part of the signature

  $$s = rd_Ah_A + k_A \bmod N \text{ until } s \neq 0 \qquad (3)$$

  where $1\leq k_A\leq N-1$, $(x_A,y_A) = k_AG$.

- **Verification:** The signature (r, s) thus generated can be verified using simple calculations i.e. $(x_1, x_2)$ = sG - $rh_AQ_A$ mod N provided r and s € [1, N-1] with an initial condition that $Q_A$ lies on the curve but not on the point at infinity &$nQ_A$=O. Now if $r = x_1 \bmod N$, then the user has successfully verified the signature [4].

**Implementation of SAMA in MATLAB with the help of Elliptic curve cryptography**

- **Generation:** Receiver's key for all the members except the sender ($i\neq t$) is $(r_i,y_i) = k_iG$ where $1\leq k_i\leq n$. Then choose $k_t$ from $F_p$ and calculate $(r_t,y_t) = k_tG-\Sigma(i\neq t)r_ih_iQ_i$ such that $r_t\neq 0$ or $r_i$ (for the sender) and the sender's key for each member is thus calculated as:

  $$s = k_t + \sum(i\neq t)k_i + r_th_td_t \bmod N \qquad (4)$$

- **Verification:** Any receiver in the group can thus verify the authenticity of the SAMA using $(x_0,y_0)$ = sG–

  $\Sigma^n_{i-1}r_ih_iQ_i$. If all $r_i,y_i$ belongs to [1, N-1] and checking if $x_0=\Sigma_ir_i$ with an initial condition that $Q_i$ lies on the curve but not equal to Q and $nQ_i$=O. In fact, SAMA is equivalent to MES signature for signature for n=1 [4].

**THEORETICAL ANALYSIS**

Most of the development schemes till now provide authentication at the end nodes with the secret key shared by the sender to the end node. So the corrupted message may travel through many nodes before it is verified and rejected which means wastage of power. It increases the collisions in the network and decreases the message delivery ratio. After trying different methods to solve the issue, it was found they had flaws and it thus led to the development of SAMA. Larger values of $d_x$ and $d_y$ may cause resilience of node compromise attacks, adding small noise to the polynomial but it was removable by error correcting techniques and encrypting public key but it had high computational overhead.

SAMA gives security similar to bivariate polynomial in n-1, when n>1 the security level increased. It lets the other messages to flow with security even if a corrupted message is encountered. This technique enables all the other nodes to act as intermediate nodes. This makes the network easily attackable and causes traffic in packet delivery of the message to the sink node from all other sensor nodes. But this technique is better in terms of memory, computational overhead, and security, at the transmitter as well as the receiver's end [8].

Under bivariate polynomial based technique only one node or base station generates the message and the rest of the nodes act as intermediate nodes that take part in the transmission and receiving end where the authenticity is verified [9]. This increases the risk of attacks. The difference between this technique and the SAMA has been experimented and analyzed. Taking five security levels into consideration that is key size ℓ=24, 32, 40, 64 & 80 bits for ECC and $d_x$ and $d_y$ are used to determine the degree of the polynomial and have equal that is 80, 100 & 150 for bivariate. The number of nodes (AS size) is chosen 1, 10, 15 & 20. We have evaluated our simulation in MATLAB (R2014a) as compare to the previous techniques that are simulated in ns2 on Red Hat Linux system. The resultant tables from 1 to 5 and figures 1(a) Message delay, 1(b) Delivery ratio, 1(c) Energy consumption shows that the results of SAMA and polynomial techniques has improved by around 96% at the transmitter end at ℓ=24 and at the receiver end has improved approximately 67% to 96%.

**Table 1.** Generation delay & Verification delay at the transmitting end using SAMA scheme

| ℓ | no. of user:1 | | no. of user: 10 | | no. of user: 15 | | no. of user: 20 | |
|---|---|---|---|---|---|---|---|---|
| | G | V | G | V | G | V | G | V |
| 24 | 0.2332 | Inf | 4.1643 | Inf | 6.3356 | Inf | 8.5158 | Inf |
| 32 | 0.3423 | Inf | 11.9751 | Inf | 8.9016 | Inf | 12.1563 | Inf |
| 40 | 0.4588 | Inf | 7.906 | Inf | 11.794 | Inf | 15.755 | Inf |
| 64 | 1.1827 | Inf | 20.6237 | Inf | 30.3152 | Inf | 40.4902 | Inf |
| 80 | 1.4088 | Inf | 26.3506 | Inf | 37.5421 | Inf | 50.7315 | Inf |

**Table 2.** Generation delay & Verification delay at the receiver end using SAMA scheme

| ℓ | no. of user: 1 | | no. of user: 10 | | no. of user: 15 | | no. of user: 20 | |
|---|---|---|---|---|---|---|---|---|
| | G | V | G | V | G | V | G | V |
| 24 | NaN | 0.2226 | NaN | 4.1623 | NaN | 6.3433 | NaN | 8.5177 |
| 32 | NaN | 0.3373 | NaN | 5.9751 | NaN | 8.9014 | NaN | 12.1489 |
| 40 | NaN | 0.452 | NaN | 7.9047 | NaN | 11.7791 | NaN | 15.7706 |
| 64 | NaN | 1.1741 | NaN | 20.6162 | NaN | 30.2949 | NaN | 40.4982 |
| 80 | NaN | 1.4201 | NaN | 26.3482 | NaN | 37.5365 | NaN | 50.7279 |

**Table 3.** Memory table for polynomial and SAMA at the transmitter end

Polynomial

| ℓ | Degree of polynomial of x & y= 80 | | Degree of polynomial of x & y= 100 | | Degree of polynomial of x & y= 150 | |
|---|---|---|---|---|---|---|
| | ROM | RAM | ROM | RAM | ROM | RAM |
| 24 | 20 | 1 | 20 | 1 | 20 | 1 |
| 32 | 20 | 1 | 20 | 1 | 20 | 1 |
| 40 | 20 | 1 | 20 | 1 | 20 | 1 |
| 64 | 20 | 1 | 20 | 1 | 20 | 1 |
| 80 | 20 | 1 | 20 | 1 | 20 | 1 |

SAMA

| ℓ | no. of user: 1 | | no. of user: 10 | | no. of user :15 | | no. of user: 20 | |
|---|---|---|---|---|---|---|---|---|
| | ROM | RAM | ROM | RAM | ROM | RAM | ROM | RAM |
| 24 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |
| 32 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |
| 40 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |
| 64 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |
| 80 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |

**Table 4.** ROM and RAM at the transmitter &receiver end using polynomial

| ℓ | Degree of polynomial of x & y = 80 | | Degree of polynomial of x & y = 100 | | Degree of polynomial of x & y = 150 | | ℓ | Degree of polynomial of x & y= 80 | | Degree of polynomial of x & y= 100 | | Degree of polynomial of x & y= 150 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ROM | RAM | ROM | RAM | ROM | RAM | | ROM | RAM | ROM | RAM | ROM | RAM |
| 24 | 7.5496 | NaN | 12.8273 | NaN | 26.5444 | NaN | 24 | NaN | 0.2666 | NaN | 0.327 | NaN | 0.4725 |
| 32 | 9.575 | NaN | 18.7974 | NaN | 39.4892 | NaN | 32 | NaN | 0.3571 | NaN | 0.4328 | NaN | 0.6381 |
| 40 | 10.5752 | NaN | 19.2824 | NaN | 40.527 | NaN | 40 | NaN | 0.3689 | NaN | 0.4617 | NaN | 0.6707 |
| 64 | 18.6503 | NaN | 31.8433 | NaN | 68.5779 | NaN | 64 | NaN | 0.5938 | NaN | 0.7307 | NaN | 1.0817 |
| 80 | 23.6127 | NaN | 38.8563 | NaN | 84.6256 | NaN | 80 | NaN | 0.7477 | NaN | 0.8739 | NaN | 1.3668 |

**Table 5.** Memory table for polynomial and SAMA at the receiver end

Polynomial

| ℓ | Degree of polynomial of x & y= 80 | | Degree of polynomial of x & y= 100 | | Degree of polynomial of x & y= 150 | |
|---|---|---|---|---|---|---|
| | ROM | RAM | ROM | RAM | ROM | RAM |
| 24 | 20 | 1 | 20 | 1 | 20 | 1 |
| 32 | 20 | 1 | 20 | 1 | 20 | 1 |
| 40 | 20 | 1 | 20 | 1 | 20 | 1 |
| 64 | 20 | 1 | 20 | 1 | 20 | 1 |
| 80 | 20 | 1 | 20 | 1 | 20 | 1 |

SAMA

| ℓ | no. of user: 1 | | no. of user: 10 | | no. of user: 15 | | no. of user: 20 | |
|---|---|---|---|---|---|---|---|---|
| | ROM | RAM | ROM | RAM | ROM | RAM | ROM | RAM |
| 24 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |
| 32 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |
| 40 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |
| 64 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |
| 80 | 18 | 1 | 18 | 1 | 18 | 1 | 18 | 2 |

The resultant tables from 1 to 5 shows that the results of SAMA and polynomial techniques has improved by around 96% at the transmitter end at ℓ=24 and at the receiver end has improved approximately 67% to 96%.(simulated in MATLAB).

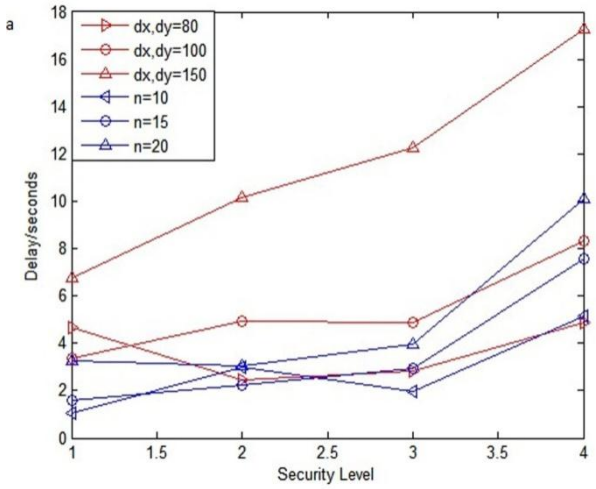


**Figure 1.** (a) Message delay;

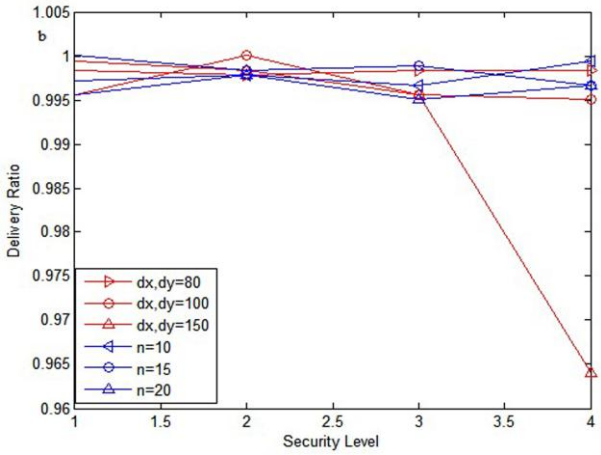


**Figure 1. (b)** Delivery ratio;

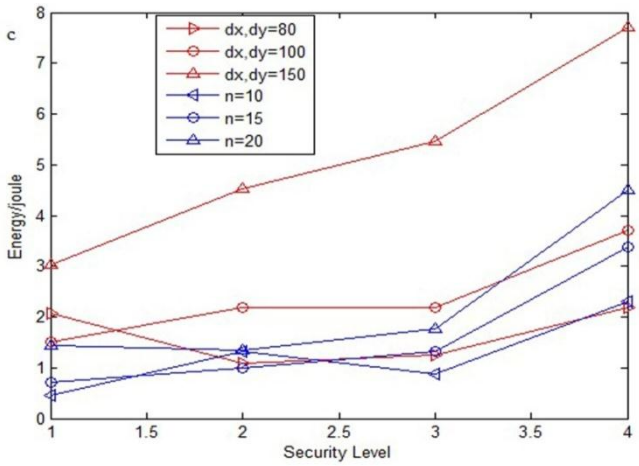


**Figure 1.(c)** Energy consumption

Figures 1(a) Message delay, 1(b) Delivery ratio, 1(c) Energy consumption shows that the results of SAMA and polynomial techniques has improved by around 96% at the transmitter end at ℓ=24 and at the receiver end has improved approximately 67% to 96% (simulated in MATLAB).

## EXPERIMENTAL RESULTS

The SAMA technique shows improvement in many of its calculated parameters. On analyzing the results of the transmitter table and considering ℓ=24 as the hashing function length, then the time taken for transmission by SAMA is approximately between 0.233 seconds to 8.5158 seconds. However, if we look into the table of transmitter of polynomial, then the values of transmitter will vary between 7.5496 seconds to 26.5444 seconds.

Similarly the variation in minimum time for SAMA and polynomial at the transmitting end can be calculated as the percentage ratio for a set of time then the variation can be calculated as:

Improvement at the transmitter end = ((7.5496-0.233)/7.5496)*100 = 96.91%

Thus, there is an improvement of 96% in the calculation of time evaluated between, polynomial and SAMA at the transmitter end. Similarly the variation in maximum time for transmitting end can be calculated as:

Improvement at the transmitter end = ((26.5444-8.5158)/26.5444)*100 = 67.91%

This shows that receiving end code 67% more efficient than the transmitting end code. Thus, the overall percentage improvement in the table of the transmitter at ℓ=24 is varying approximately 67% to 96% approximately.

Similarly, if the table of the receiver end is seen and its

comparison is done on behalf of delay time in the verification the improvement in the code can be seen. Minimum time in the SAMA and polynomial is 0.2226 seconds and 0.2666 seconds respectively. So the improvement in the SAMA receiver verification is calculated on the same basis when ℓ=24 is taken as:

Improvement at the receiver end = ((0.2666-0.2226)/0.2666)*100 = 16.50%

Similarly, the Maximum time in SAMA and the polynomial is 0.4725 seconds and 8.5177 seconds respectively. So the improvements at the receiving end for maximum time when ℓ=24 is taken as:

Improvement at the receiver end = ((8.5177-0.4725)/8.5177)*100 = 94.45%.

## CONCLUSION

In this paper, we have evaluated SAMA and bivariate polynomial techniques in MATLAB (R2014 a) instead of ns2 simulator that gives us better results in less time as compared with previous technique and also analyzed SAMA and bivariate polynomial techniques at the transmitter and at the receiver end differently. For bivariate polynomial scheme the computational overhead is determined by $\ell*(d_y +1)$ and for SAMA is determined by $4*\ell*(n+1)+n\ [\log_2\alpha]$. In analyzing the results we have taken five different values of key size: ℓ=24, 32, 40, 64 & 80 bits for ECC and $d_x$ & $d_y$ are used to determine the degree of the polynomial and have equal values: 80, 100 & 150 for bivariate, and the number of nodes (users)is chosen as 1, 10, 15 & 20. Both theoretical and simulation results show that results of SAMA and polynomial techniques has improved by around 96% at the transmitter end at ℓ=24 and at the receiver end has improved approximately 67% to 96%.

## REFERENCES

[1] M. Franklin, R. Gelles, R. Ostrovsky, L. J. Schulman, "Optimal coding for streaming authentication and interactive communication," IEEE Transactions Information Theory, Vol. 61, no. 1, pp. 133-45, 2015.

[2] A. Rasheed, RN. Mahapatra, "Key predistribution schemes for establishing pairwise keys with a mobile sink in sensor networks," *Paralleland Distributed Systems, IEEE Transactions on*. Vol. 22, no. 1, pp. 176-84, 2011 Jan.

[3] L. Harn, J. Ren, "Generalized digital certificate for user authentication and key establishment for secure communications," IEEE Transactions Wireless Communications Vol. 10, no. 7, pp. 2372-9, Jul 2011.

[4] J. Li, Y. Li, J. Ren, J. Wu, "Hop-by-Hop Message Authentication and Source Privacy in Wireless Sensor Networks," Parallel and Distributed Systems, IEEE Transactions. Vol. 25, no. 5, pp. 1223-1232, 2014.

[5] D. Pointcheval and J. Stern, "Security Arguments for Digital Signatures and Blind Signatures," J. Cryptology, vol. 13, no. 3, pp. 361- 396, 2000.

[6] Harris E. Michail, Athanasios P. Kakarountas, Athanasios S. Milidonis, and Costas E. Goutis, "A Top-Down Design Methodology for Ultrahigh-Performance Hashing Cores," IEEE Transactions on Dependable and Secure computing, Vol.6, no.4, pp.255-268, 2009.

[7] W. Zhang, N. Subramanian, and G. Wang, "Lightweight and Compromise-Resilient Message Authentication in Sensor Networks," Proc. IEEE INFOCOM, Apr. 2008.

[8] A.Rasheed, RN. Mahapatra, "The three-tier security scheme in wireless sensor networks with mobile sinks," Parallel and Distributed Systems, IEEE Transactions on. Vol. 23, no. 5, pp. 958-65, May 2012.

[9] L. Harn, J. Ren, "Generalized digital certificate for user authentication and key establishment for secure communications," Wireless Communications, IEEE Transactions on. Vol. 10, no. 7, pp. 2372-9, Jul 2011.