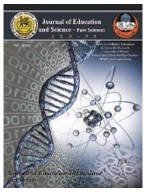
Journal of Education and Science (ISSN 1812-125X)

www.edusj.uomosul.edu.iq



Performance Evaluation of RSA Variants For Network Security Using SIP

M. Y. Al-Fathi[1], **A. A. Khudher** [2]

[1, 2]Department of Computer Science, College of Education for Pure Science, University of Mosul, Mosul, Iraq

Article information

Article history:
Received: March 04, 2025
Revised: April 20, 2025
Accepted: April, 29, 2025
Available online: July 01, 2025

Keywords:
Network
Network Security
SIP server

Correspondence:
Maan Y, Al-Fathi
dr.maan.y@uomosul.edu.iq

Abstract

Networks and especially network security, have recently become an area of essential concern in data communication due to the ever-present need for data security. Especially that data is to be exposed to the public and traveled from one network to another. Recently, network security has had to be carefully deployed in data communication, especially real-time communication, such as voice over the Internet protocol. SIP is a real-time protocol that functions as a call initiator and controller, ensuring its communication is secure through various means. A cryptographic algorithm is a promising approach to secure SIP end-to-end communication. However, that does not come without cost and penalty, especially with performance. In this paper, we looked at how the SIP server performs using different RSA algorithms, including RSA-Standard, RSA Chinese Remainder Theorem, and Multi-prime RSA, by measuring CPU usage and how quickly calls are answered. The experiment reveals that when applying 1000 calls, the RSA standard consumes the highest CPU percentage. The decryption computations during the inbound calls scenario result in extra CPU usage for the three algorithms. In addition, call response time in all cases is still within an acceptable call response time level to spend 50 ms in higher call load 1000 calls. In sum, from these results, SIP service providers will gain a clear comprehension when it comes to deploying RSA algorithms in their services, which in turn facilitates smooth implementation in the voice communication industry and provides benchmarked material for research communities.

DOI: 10.33899/edusj.2025.157805.1538, ©Authors, 2025, College of Education for Pure Science, University of Mosul.

1. Introduction

Network security is a concept to protect data transmitted between end-to-end users, to provide a shelled, secure communication system with a high level of user confidentiality [1]. Basically, a network security system depends on multiple protective components, including network monitoring, security software, security hardware, and cryptography [2]. In general, all means of security participate together to ensure the overall security of computer networks [3-4]. However, cryptography, which encrypts and decrypts the traveled data, is considered the most important part of data security [5-6]. In fact, cryptography in all computer science fields is an emerging technology that is essential for network security. Various protocols, including Transport Layer Security (TLS), deploy cryptography algorithms [7].TLS is an IETF standardization initiative which is found to secure end-to-end connections. In general, TLS aims to provide robust security services between end-to-end TCP connections. Session Initiation Protocol (SIP] like other Internet applications depends on TLS to secure its connection and capable of transmitting secure data through TLS over a TCP connection. SIP over TLS utilizes symmetric encryption algorithms to provide encryption, authentication, and data integrity during data transmission. Technically, TLS provides confidentiality through various types of cryptographic algorithms such as RSA, AES, DES, Diffie–Hellman, and Elliptic Curves [1] [8-9].

Rivest, Shamir, and Adleman founded RSA in 1978 [10] as a method to implement the cryptosystem public key. The RSA algorithm is considered the most popular security technique for different Internet technologies, such as the web, email, voice-over IP, IoT, and other related fields in computer science. RSA utilizes a large exponent and modulus with exponentiation to provide security. RSA can be set up with different key sizes and prime numbers based on what the application needs, like using a smaller public key to make encryption faster for low-resource devices (IoT). This has led the research society and the industrial field to develop variant RSA types, each offering varying levels of security and efficiency. Unfortunately, increasing the key sizes will affect the performance, especially in real-time communication, such as voice-over IP, which leads VoIP service providers to remain uncertain about which RSA name to implement and for which circumstances. The just-mentioned drawbacks caused the need for a wide performance study on RSA [1][5][11]. The paper is organized by providing a related works section followed by a background on RSA in Section 2. Details are given about RSA with SIP section 3. Sections 4,5 and 6 overview of RSA and show the Testing methodology. Results and discussion in section 7. Last, the conclusion is presented in section 8.

2. Literature review and problem statement

RSA is well-known and deployed in the research community and industrial fields due to its security level and acceptable efficiency. Recently, the RSA algorithm has been tested and evaluated through several systems, especially network systems, to secure the end-to-end connection. In [12] the authors have studied the impact of RSA on IoT resource-constrained nodes the study was conducted based on System-on-Chip. The study has evaluated based on energy consumption devices and data throughput. RSA in the IoT has also been evaluated [13] using fog and mist computing architectures and compared with other cryptography in terms of energy consumption devices and data throughput. In [9], RSA and RSA-CRT are both evaluated in various factors, including key-length value, encryption ratio, computational speed, and overhead in memory using the Java programming language to perform the decryption process, and the authors found that the CRT decryption outperforms the basic RSA in a faster process. The authors in [14] proposed a new variant of RSA containing 4 prime numbers along with 2 public-key used in RSA-CRT to enhance the performance and increase security. The paper also found that RSA performs slower after it reaches the bit size threshold of 1024 bit. [6] conducted an analytical performance of RSA over Wireless Transport Layer Security (WTLS) using 2 handshake protocols, the work evaluated parameters related to RSA over WTLS including latency, data throughput, and processing time. In [2] a Secured Hybrid RSA has been proposed with 4 layers authentication stack to achieve 4% lower CPU and 3% memory compared to RSA-CRT. The results also presented the relevancies of the SHRSA scheme, which is implemented in blockchain architecture and the Internet of Everything. The author in [5] presented a comparative analysis of RSA with other cryptography to find that RSA is categorized based on integer factorization problem (IFP) which takes sub-exponential time which impact directly to the limited memory devices. All the aforementioned literature focused on RSA and its variants with different systems, however, there is no focus on RSA with SIP proxy.

3. The aim and objectives of the study

This paper aims to analyze the performance of variant RSA algorithms on SIP servers in terms of efficiency in order to come up with recommended RSA names for SIP security. The paper conducts an experimental performance study for three RSA names, including Based RSA, RSA with Chinese Remainder Theorem, and Multi-prime RSA applied on SIP server. Each RSA name has different key/prime sizes, which can lead to varying performances in the SIP server. We evaluated the experiments using different parameters like time and CPU usage.

4. RSA variants

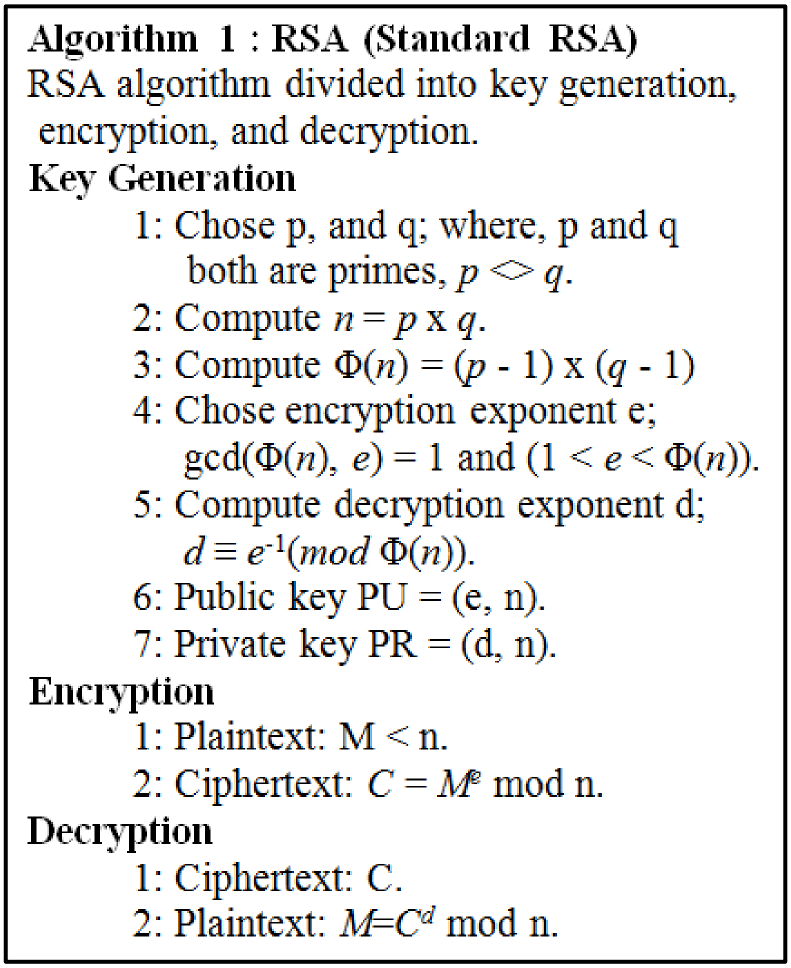
RSA is a cryptographic system known as a secured algorithm that is widely deployed to secure data during transmission over networks. Technically, RSA utilizes an encryption key asymmetry, which means it generates two different keys; public and private. RSA utilizes a public key for both encryption and decryption operations. In general, the length of the key is always variable and does not have a specific defined length. For instance, we recommend using a long key for systems that require high security. On the other hand, cost-effective systems use shortcuts. Generally speaking, 512 bits is the maximum key length that is available across all ranges. The power of RSA comes from the consequence of one of the hardest problems in mathematics, which is prime factorization. Mathematically, it is considered algorithmically infeasible to factorize large numbers back to their primes using RSA. That leads RSA to provide a high level of optimum security. RSA also employs a 1024-bit composite number as its key, which is large enough for today's technology to factorize into primes. Technically, that large number requires high processing to solve such primes, which in turn affects the network device's efficiency. Recently, researchers have found several RSA variants that are suitable for the required systems. The following subsections present the three most usable algorithms recommended by the research community.

4.1 RSA (Standard)

RSA (Standard) is a default RSA algorithm without any modification or improvement. The main concept of this algorithm relies on an integer factorization problem. At this stage, far-end devices generate a pre-agreed key. After completing the key generation, the end-to-end devices can now communicate securely. From Figure 1, during the encryption phase, RSA starts to choose an exponent e; gcd(Φ(n), e) is equal to 1. On the other side, RSA will decrypt an exponent; in this case, d is generated by invoking the process to inverse the e mod Φ(n). In sum, during the encryption process, a public key is used to encrypt the plain message. While at the far end side, the receiver will decrypt the received cipher text using its own private key i.e., d and n.

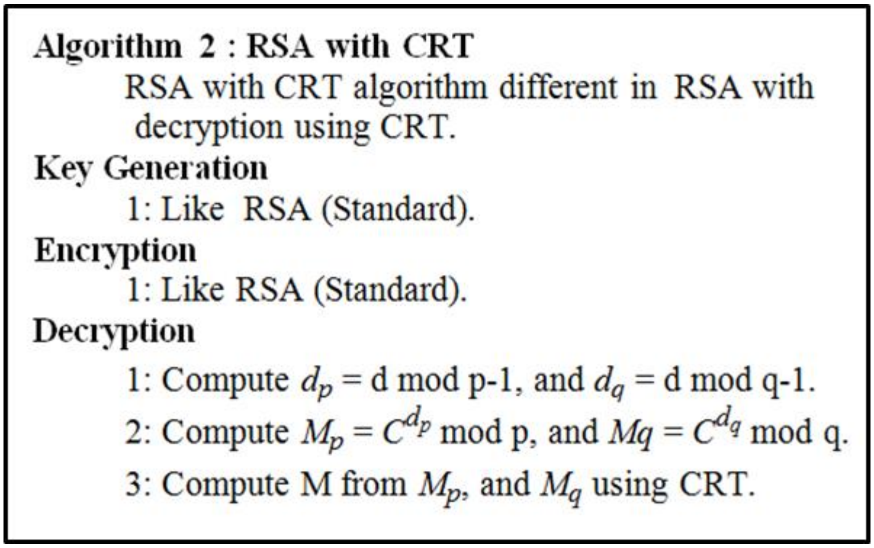
4.2 RSA CRT

Chinese Remainder Theorem (CRT) is a variant of the RSA standard, found to enhance the decryption process in terms of both security and efficiency. The idea beyond RSA-CRT is presented by a method to divide the decryption exponent d into two parts (dp, dq). The change aims to reduce the decryption time of RSA, which in turn fastens the decryption process by a factor of 4 times faster than RSA (Standard). Figure 2 illustrates the amendment that occurred solely on the decryption side.



```
Algorithm 1 : RSA (Standard RSA)
RSA algorithm divided into key generation,
 encryption, and decryption.
Key Generation
        1: Chose p, and q; where, p and q
           both are primes, $p <> q$.
        2: Compute $n = p$ x $q$.
        3: Compute $\Phi(n) = (p - 1)$ x $(q - 1)$
        4: Chose encryption exponent e;
           gcd($\Phi(n)$, $e$) = 1 and ($1 < e < \Phi(n)$).
        5: Compute decryption exponent d;
           $d \equiv e^{-1}(mod\ \Phi(n))$.
        6: Public key PU = (e, n).
        7: Private key PR = (d, n).
Encryption
        1: Plaintext: M < n.
        2: Ciphertext: $C = M^e$ mod n.
Decryption
        1: Ciphertext: C.
        2: Plaintext: $M=C^d$ mod n.
```

Figure 1. RSA (Standard)

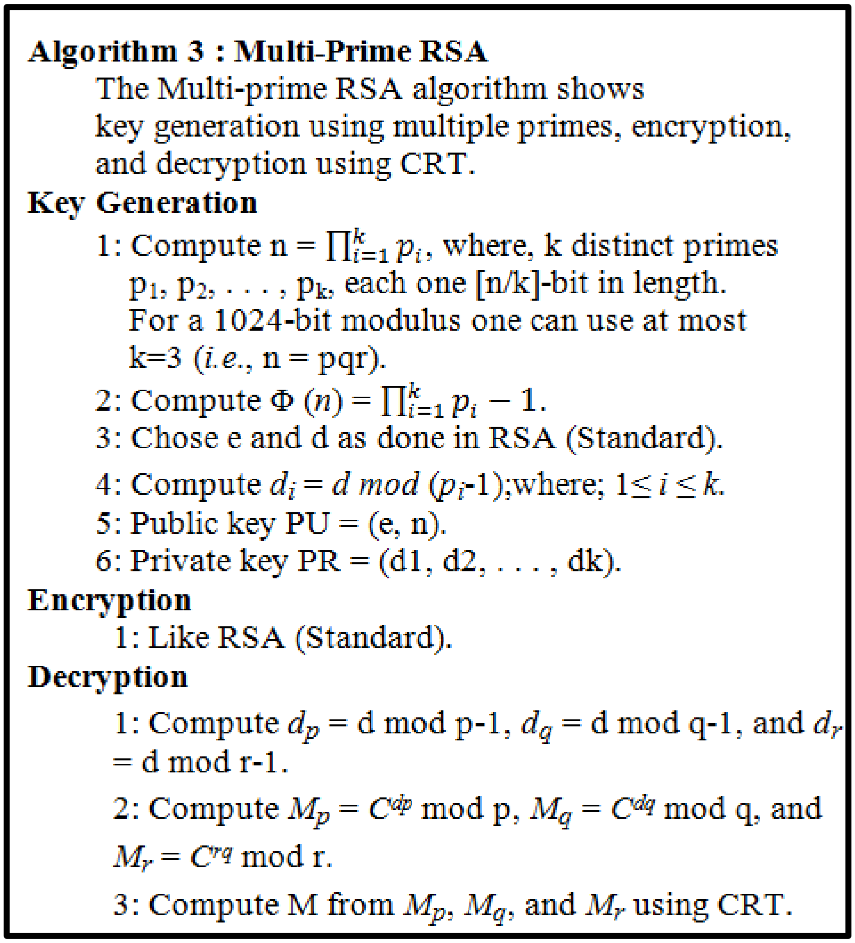


```
Algorithm 2 : RSA with CRT
    RSA with CRT algorithm different in RSA with
    decryption using CRT.
Key Generation
    1: Like RSA (Standard).
Encryption
    1: Like RSA (Standard).
Decryption
    1: Compute $d_p$ = d mod p-1, and $d_q$ = d mod q-1.
    2: Compute $M_p = C^{d_p}$ mod p, and $Mq = C^{d_q}$ mod q.
    3: Compute M from $M_p$, and $M_q$ using CRT.
```

Figure 2.RSA with CRT

4.3 Multi-prime RSA

Multi-prime is another common variant of RSA, it is found to reduce decryption processing time by forming modulus 'n' utilizing multiple primes rather than two primes. From the name indication, this algorithm generates k primes: p1, p2, . . . , pk. Figure 3 shows the extra processing in the key generation phase.



```
Algorithm 3 : Multi-Prime RSA
    The Multi-prime RSA algorithm shows
    key generation using multiple primes, encryption,
    and decryption using CRT.
Key Generation
    1: Compute n = $\prod_{i=1}^{k} p_i$, where, k distinct primes
       $p_1$, $p_2$, . . . , $p_k$, each one [n/k]-bit in length.
       For a 1024-bit modulus one can use at most
       k=3 (i.e., n = pqr).
    2: Compute $\Phi$ $(n) = \prod_{i=1}^{k} p_i - 1$.
    3: Chose e and d as done in RSA (Standard).
    4: Compute $d_i = d \ mod \ (p_i$-1);where; $1 \leq i \leq k$.
    5: Public key PU = (e, n).
    6: Private key PR = (d1, d2, . . . , dk).
Encryption
    1: Like RSA (Standard).
Decryption
    1: Compute $d_p$ = d mod p-1, $d_q$ = d mod q-1, and $d_r$
       = d mod r-1.
    2: Compute $M_p = C^{dp}$ mod p, $M_q = C^{dq}$ mod q, and
       $M_r = C^{rq}$ mod r.
    3: Compute M from $M_p$, $M_q$, and $M_r$ using CRT.
```

Figure 3. Multi Prime RSA

5. RSA for VoIP communication

Signaling path security is the key factor in securing VoIP communication, which is accomplished by several combined protocols. SIP and TLS protocols are the main components for securing calls in VoIP communication; they work jointly to provide confidentiality, integrity, and authentication between end-to-end devices. It is worth mentioning that SIP is only responsible for handling signaling sessions and no media. This means that security is solely focused on the signaling session. Technically, transport layers like UDP or TCP carry signaling messages. Despite attempts, we do not recommend using UDP with TLS in practice. However, TLS is very well implemented and deployed over TCP connections. That made TLS in TCP well implemented in SIP connectivity, which in turn led SIP technology to make full use of TLS. In particular, OpenSSL is an open-source server that provides SSL in the old version and TLS in the current new version.

TLS provides confidentiality and integrity for the transmission of data via end-to-end devices. TLS makes hop-by-hop devices in communication links to authenticate each other in real-time communication. However, real-time communication is challenging due to problems with the quality of voice. Thus, several researchers [17] have proven that using TLS with mutual authentication can reduce performance by up to a factor of 17 compared to no TLS connections. The mentioned drawbacks stem from the fact that the cryptographic algorithm has heavy processing in both encryption and decryption. The RSA algorithm is one of the heaviest cryptographic algorithms but with a high-security history.

The main impact in TLS is primarily driven by the cipher suite string, which is constructed as TLS e s WITH r, e, indicating a key encapsulation mechanism. S for the signature scheme in the handshake phase, and r will hold the authenticated encryption scheme in the record layer phase. One commonly used cipher suite string in an OpenSSL server is RSA_WITH_AES_256_CBC_SHA: the client exchanges a new premaster secret encrypted by the OpenSSL public key. Later, either UAC or UAS utilizes it to extract a master secret to be used as a seed of a SHA1-based to derive four keys in SHA1-based MACs and AES encryption in CBC mode.

RSA is very well implemented in OpenSSL with supported documentation in [RFC 8017]. The RSA standard is implemented by default in OpenSSL v.3. However, the SIP server must manually set RSA-CRT and RSA-Multi-prime in OpenSSL before it can invoke them using the openssl req –newkey command. OpenSSL will be able to implement the library of that RSA variant using RSA_generate_multi_prime_key to get multi-prime, namely, RSA_get0_multi_prime_crt_params and RSA_set0_multi_prime_params for CRT and multi-prime, respectively.

6. Testing Methodology

This research concentrates on the performance analysis of SIP servers when they consider security in terms of cryptographic algorithms. RSA and its two variants are to be tested on the SIP server and analyzed for performance in real-time communication. The heavy processing and computations of RSA directly impact SIP technology, just like they do other Internet technologies. This section illustrates the testing environment and the performance, as shown in Figure 4.

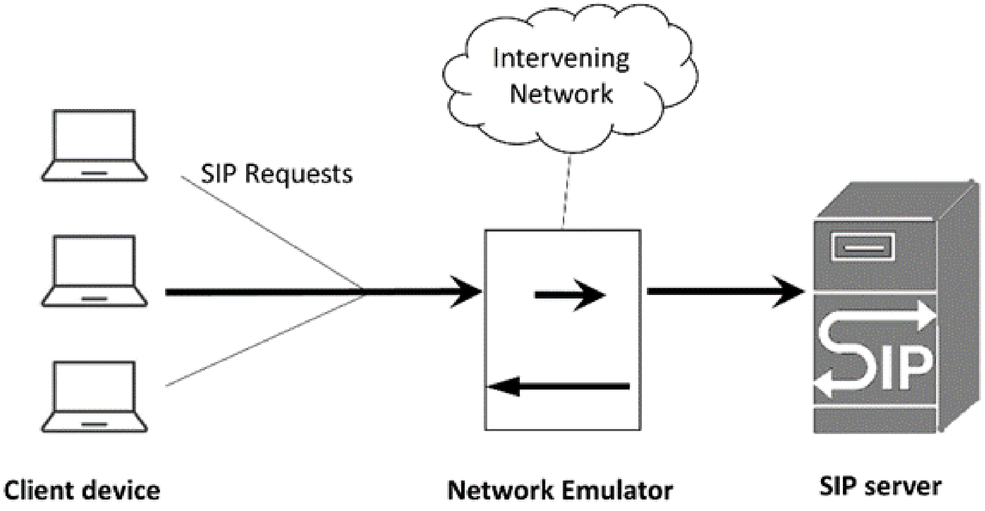


Figure 4: Representing methodology for SIP server the performance test setup

6.1 Testing Platform

The SIP server has been selected to test the impact of the RSA algorithm on the networking services. Specifically, we use the OpenSIPs [15-16] server, currently running at version 3.5.3, as a testing platform. We install and run OpenSIPs as a proxy, registrar, and location server on a single machine (Ubuntu 18.4 server). In this research, the OpenSIPs proxy cooperates with the OpenSSL server version (1.0.2g), which invokes all the security libraries. The standard RSA is deployed on the server by default; however, the CRT and multi-prime RSA algorithms are generated manually in the OpenSSL server using RSA_generate_multi_prime_key.

6.2 Hardware and connectivity

Testing environments have been conducted in a local network with SIP services available. All the network elements involved in this testing are connected through the isolated network using 1000 Mbps to avoid side factors. The SIP proxy server runs on an Ubuntu 18.4 server machine with 3.00 GHz (deactivating other core processors) and 2 gigabytes of RAM. The UAC machine runs on an i3-2310 M @ 2.10 GHz and 4 gigabytes of RAM. In contrast, the UAS operates on an i3-2310 M processor running at 2.20 GHz with 2 gigabytes of RAM. Despite not benchmarking Elliptic Curve Cryptography (ECC) and other SIP security cryptographic techniques, CPU utilization and reaction time are examined. Comparisons would show security-performance trade-offs. For instance, ECC provides equivalent security with smaller key sizes and less computational effort than RSA [18]. Comparative studies like [19] emphasize SIP security performance evaluation and the practical effects of ECC in low-resource contexts.

6.3 Performance Scenarios

The OpenSIPs server was tested using SIP traffic generator software called SIPp, which is a de facto standard used to generate SIP traffic toward the SIP server with a high load of messages. We chose SIPp because of its high-load SIP traffic generator and TLS support parameters. SIPp is configured to support TLS by including "--with-openssl" during the compiling setup.

UAC in SIPp generates a workload with 1000 calls per second toward OpenSIPs. Call rate, server capacity, and resource use all affect a 1000-call load stress test result. Though traffic patterns, network conditions, and server configurations may vary, this test approximates high usage. The server deploys one RSA algorithm each time. It is worth mentioning here that all the RSA algorithms in this test have a higher key size, which is 1024 bits, to get maximum use of evaluation. This test evaluates the measured parameters of CPU usage and time. The time is calculated from the call response time related to SIP messages travelling from source to destination. The test is conducted through two scenarios, which are an outbound call to handle the encryption side and an inbound call for decryption.

7. Results And Discussion

This section analyzes the research findings on how the RSA and its variant algorithms are impacting the SIP server performance in different scenarios. The evaluation was conducted through two measured parameters, which are CPU usage and time. CPU is measured by measuring the percentage of one core of the machine (disabling the other core), whereas time is calculated using the call response time.

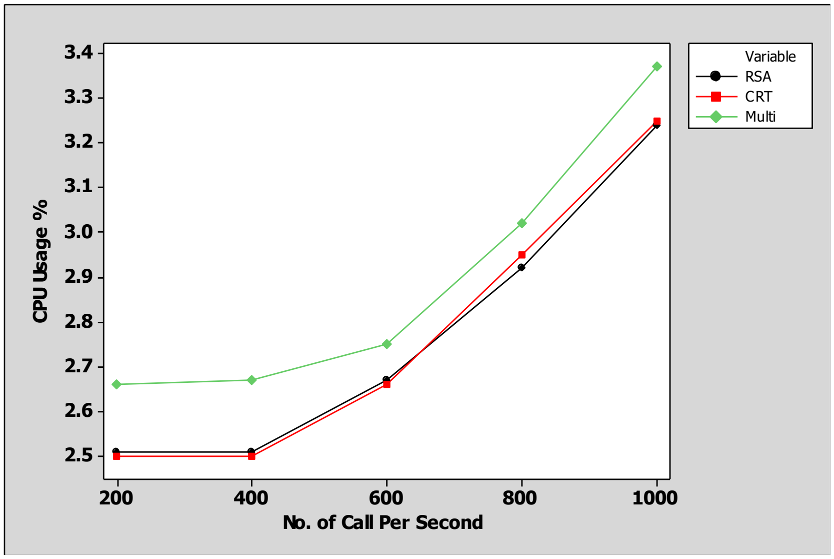


Figure 5. CPU usage vs. number of call handling encryption process (Outbound call)

Figure 6 shows that at the early testing, when 200 and 400 calls are applied, the RSA Multi-prime algorithm consumes higher CPU resources compared to standard and CRT algorithms. That is due to the extra computation process spent for the key generation phase. As the load increased, the standard and CRT RSA started to increase their CPU usage. That is because the

OpenSIPS server gets the overhead of handling 800 and 1000 calls per second. Obviously, the two algorithms, the RSA standard and CRT, use almost the same resource of CPU across all the tests; that is because the mathematical operations in the encryption phase are the same. The OpenSIPs server is impacted directly by the heavy operations of the multi-prime algorithm when it needs to call the library of OpenSSL and reuse it in the call route inside the OpenSIPs routing script. From other perspectives, OpenSIPS is still with its full ability to handle 1000 calls per second and its maximum CPU usage is only 3.35%..

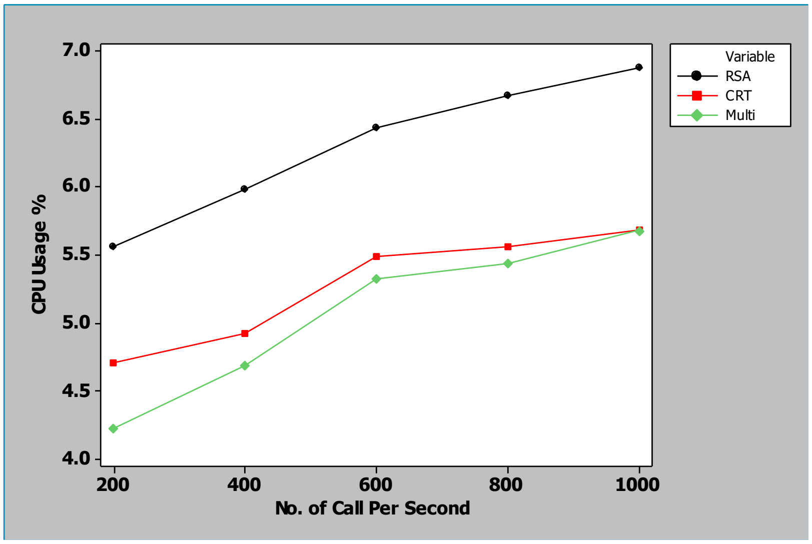


Figure 6. CPU usage vs. number of call handling decryption process (Inbound call)

The encryption process directly impacts the OpenSIPs server when it manages inbound calls from UAC. Figure 5 depicts the extra mathematical operations of the decryption process, and one can notice the high CPU usage of all three algorithms compared to the encryption performance in Figure 4. Initially, the standard RSA consumed 5.51 % of the CPU on the OpeSIPs server when 200 calls were applied, which clearly shows the difference between inbound and outbound call processing. Approximately 60% is the difference in the RSA standard in both scenarios; OpenSIPs handle the inbound calls with the penalty of decryption overhead performance. The CRT algorithm improves the decryption process compared to standard RSA by using its two primes, P and q. That method in CRT can reduce CPU usage by around 1.7% when compared with standard RSA. However, multi-prime algorithms clearly improve CPU consumption and that is because of the advantage of multi-prime that it used to decrypt the message. Also, one can notice that the CPU usage at the point of heaviest load of the OpenSIPs server when it handles 1000 calls per second, both CRT and multi-prime algorithms are in very near cost; they consume around 5.5% of CPU..

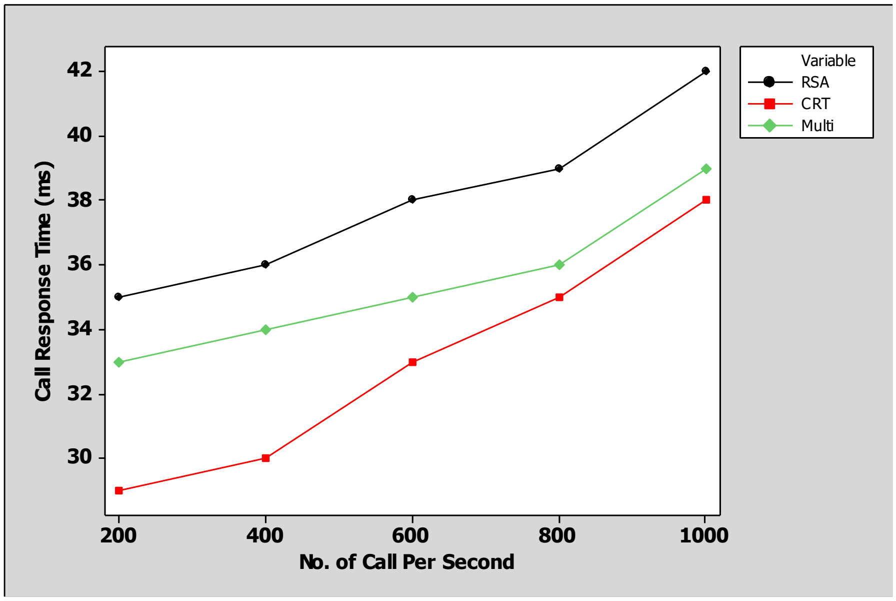


Figure 7. Call response time vs. number of call handling encryption process (Outbound call)

In Figure 7, call response time is calculated by measuring the round trip of the successful call when the total time is spent from source to destination and back to the source again. This section analyzes the time factor to demonstrate how long each algorithm takes to handle an outbound call. The OpenSIPs server handles the calls coming from the upstream with 35 ms when the RSA Standard is used under 200 calls. However, at the same load of 200 calls, the multi-prime algorithm is still near to the standard one due to the extra calculation being used to generate keys for outbound calls; it hits 33 ms. The maximum time spent in this experiment is when the RSA standard algorithm is used with the full call load (1000 calls) to spend around 42 ms. During outbound calls, the OpenSIPs server operates as a proxy with the job of forwarding calls just like they are; no new route is required in this mode. It is worth mentioning here that the tradeoff between security and efficiency is quite equal [20].

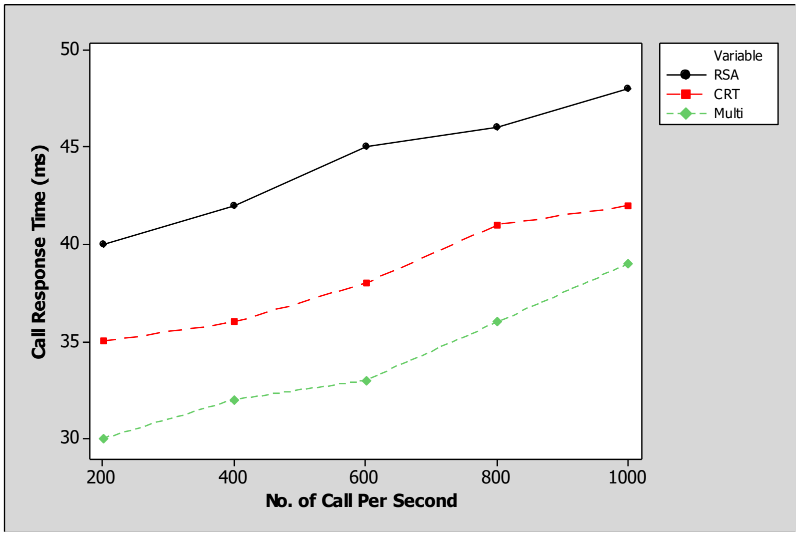


Figure 8. Call response time vs. number of call handling decryption process (Inbound call)

From Figure 8, the reader can clearly notice that all the calls received from the upstream were performed with long call duration compared to the outbound proxy. For example, when loading 200 calls, RSA basic adds an extra 5 ms to the outbound call cost. The CRT algorithm performs better than the standard one, hitting 37 ms in 600 calls due to the two prime keys that have been generated for its decryption. In contrast to CRT, the multi-prime algorithm performs faster in handling calls, spending 34 ms on 800 calls, which is 5 ms less than CRT. In all cases, the OpenSIPs server has to add an extra penalty when calls are received from the upstream side by adding some header in SIP messages and routing for the best route. To conclude, 47 ms is the longest time spent in this scenario when the RSA standard is used with 1000 calls; however, this value is still within the acceptable amount of call response time recommended by RFC 3261 [21].

8. Conclusion

Network security is still an open issue when user privacy comes into consideration. That is true, since all our data is traveling from network to network. Networking has applied and developed several cryptographic algorithms to achieve the desired security, particularly in real-time communication. SIP as a real-time protocol utilizes RSA algorithms to secure its connection with an acceptable level of security. Performance is another issue that impacts the SIP server performance, especially when heavy cryptographic algorithms are used, such as RSA. This paper analyzed three variants of RSA algorithms in SIP servers with different scenarios. The paper analyzes the performance based on two measurement parameters, which are CPU usage and time. The experiment shows that in early testing, when 200 and 400 calls are applied, the RSA Multi-prime algorithm consumes higher CPU resources compared to standard and CRT algorithms. Furthermore, the standard RSA consumed 5.51 % of CPU in the OpeSIPs server when 200 calls to highlight the variant between inbound and outbound calls. Finally, all the call response times that measure the call duration pour into an acceptable recommendation of SIP documentation.

9. Acknowledgements

The authors are very grateful to the University of Mosul/Collage of Education for Pure Science for their provided facilities, which helped to improve the quality of this work.

10. References

[1] A. Gambhir and R. Arya, “Performance Analysis of DES Algorithm and RSA Algorithm with Audio Steganography, ” *International Conference on Communication and Signal Processing 2016 (ICCASP 2016). Atlantis Press*, 2016, doi: 10.2991/iccasp-16.2017.52.

[2] B.Aniruddha, X. Zhong, and X. Li, “A lightweight and efficient secure hybrid RSA (SHRSA) messaging scheme with four-layered authentication stack, ” in *IEEE Access*, vol. 7, pp. 30487-30506, 2019, doi: 10.1109/ACCESS.2019.2900300.

[3] O. Yeremenko, “ Development of the dynamic tensor model for traffic management in a telecommunication network with the support of different classes of service, ” *Eastern-European Journal Of Enterprise Technologies*, vol. 6 , No. 9 ,pp 12-19, 2016, doi: 10.15587/1729-4061.2016.85602.

[4] L. Tokar, E. Belousova, A. Kolyadenko, and I. Lukinov, “Development of the model for a backhaul network based on the long term evolution technology ,” *Eastern-European Journal Of Enterprise Technologies*, vol. 2, no. 9 (86) ,pp 38–44 , 2017, doi.10.15587/1729-4061.2017.96040.

[5] D. Mahto and D. K. Yadav, “RSA and ECC: A Comparative Analysis, ” *International Journal of Applied Engineering Research*, vol. 12, no. 19, pp. 9053-9061, 2017, doi: 10.2991/iccasp-16.2017.52.

[6] A. Levi and E. Savas, “Performance evaluation of public-key cryptosystem operations in WTLS protocol, ” *Proceedings of the Eighth IEEE Symposium on Computers and Communications. ISCC 2003*, Kemer-Antalya, Turkey, vol.2, pp.1245-1250, 2003, doi: 10.1109/ISCC.2003.1214285.

[7] C. Shen, E. Nahum, H. Schulzrinne and C. P. Wright, “The Impact of TLS on SIP Server Performance: Measurement and Modeling, ” in *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1217-1230, 2012, doi: 10.1109/TNET.2011.2180922.

[8] R. Abdullah, “A New Algorithm to Encryption and Compression Image Data File. ” *Journal of Education and Science* , ” vol. 24, no. 1 , 2011, doi:10.33899/edusj.2011.51410.

[9] A. Mantri, et al. “Analytical comparison of rsa and rsa with chinese remainder theorem, ” Journal of Independent Studies and Research Computing , vol. 14, no.1, pp. 16-21, 2016, doi: /10.31645/jisrc/(2016).14.1.0003

[10] R. L. Rivest, A. Shamir, L. , “AdlemanAuthors Info & Claims, ” *Communications of the ACM* , vol 21, no 2 , pp 120–126, 1978, doi:10.1145/359340.359342.

[11] D. Mahto, DK. Yadav, “Performance analysis of RSA and elliptic curve cryptography, ” *International Journal of Network Security*, vol. 20 , no.4 ,pp. 625-635 , 2018, doi:10.6633/IJNS.201807_20(4).04.

[12] Q. Jing, A.V. Vasilakos, J. Wan, J. Lu, D. Qiu., “ Security of the Internet of Things: perspectives and challenges, ” *Wireless networks,* vol.20 no. 8, pp.2481-2501 , 2014, doi:10.1007/s11276-014-0761-7.

[13] M. Suárez-Albela, P. Fraga-Lamas, TM. Fernández-Caramés , “ A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices, ” Sensors, vol. 18, no. 11 pp. 3868 , 2018 doi:10.3390/s18113868

[14] RS. Abdeldaym, HM. Abd Elkader, R. Hussein , “ Modified RSA algorithm using two public key and Chinese remainder theorem. ” J. of Electronics and Information Engineering, vol.10, no.1, pp.51-64, 2019, doi:10.6636/IJEIE.201903 10(1).

[15] FE. Goncalves, BA. Iancu, “ Building telephony systems with OpenSIPS ,” *Packt Publishing Ltd*, 2016. https://doi.org/10.25124/ijait.v3i02.2503

[16] G.Satrya, M.Nicovandia., “ A Security Analysis on Open SIPS*,* ” *IJAIT (International Journal of Applied Information Technology)* vol.3, no. 02. pp.77 ,2020, doi:10.25124/ijait.v3i02.2503.

[17] LY. Beng, AA. Khudher, S. Manickam, S. Al-Salem - J. Comput, “ An Adaptive Assessment and Prediction Mechanism in Network Security Situation Awareness. ” *Journal of Computer Science* ,vol.13 no. 5 ,pp.114-129, 2017 , doi:10.3844/jcssp.2017.114.129

[18] D. Johnson, A. Menezes, and S. Vanstone, “ The Elliptic Curve Digital Signature Algorithm (ECDSA). ” International Journal of Information Security, Volume 1, Issue 1,Pages 36 – 63,2001,doi: 10.1007/s102070100002.

[19] M. Azrour, M Ouanan, Y Farhaoui, “ SIP authentication protocols based on elliptic curve cryptography: survey and comparison. ” Indonesian Journal of Electrical Engineering and Computer Science vol 4 ,no. 1, pp 231-239.,2016 , doi: 10.11591/ijeecs.v4.i1.

[20]. A. Al-Allawee, M. Mihoubi, P. Lorenz and K. S. Abakar, "Efficient Dispatcher Mechanism for SIP Cluster Based on Memory Utilization," *ICC 2023 - IEEE International Conference on Communications*, Rome, Italy, pp. 3370-3375, 2023, doi: 10.1109/ICC45041.2023.10278652.

[21] A. Al-Allawee, P. Lorenz, A. Munther. "Efficient Collaborative Edge Computing for Vehicular Network Using Clustering Service." *Network* ,vol. 4 no. 3 ,pp. 390-403 ,2024, doi: 10.3390/network4030018

تقييم أداء متغيرات RSA لأمن الشبكة باستخدامSIP

معن يونس الفتحي [1] ، علي عبدالرزاق خضر[2]

[2,1] قسم علوم الحاسوب، كلية التربية للعلوم الصرفة، جامعة الموصل، الموصل، العراق

الخلاصة:

أصبحت الشبكات وخصوصًا أمان الشبكات مؤخرًا مجالًا ذا أهمية أساسية في اتصالات البيانات نظرًا للحاجة الدائمة إلى أمان البيانات. وهذا صحيح لأن البيانات المنقولة يجب أن تُعرض على الجمهور؛ شبكة تلو الأخرى. في الآونة الأخيرة، يجب نشر أمان الشبكة بعناية في اتصالات البيانات؛ وخاصة الاتصالات في الوقت الفعلي، مثل الصوت عبر بروتوكول الإنترنت. يعمل بروتوكول SIP باعتباره بروتوكولًا في الوقت الفعلي كمبادر ومراقب للاتصالات على تأمين اتصالاته بعدة وسائل. تعد الخوارزمية التشفيرية نهجًا واعدًا لتأمين اتصالات SIP من البداية إلى النهاية. ومع ذلك، فإن هذا لا يأتي بدون تكلفة وعقوبة، وخاصة فيما يتعلق بالأداء. في هذه الورقة، تم تقييم خادم SIP وتحليله باستخدام خوارزميات RSA مختلفة؛ وهي RSA-Standard وRSA Chinese Remainder Theorem و Multi-prime RSA من حيث استخدام وحدة المعالجة المركزية ووقت استجابة المكالمة. تُظهر التجربة أن معيار RSA يستهلك أعلى نسبة وحدة معالجة مركزية عند تطبيق 1000 مكالمة. أثناء سيناريو المكالمات الواردة، يتم تنفيذ الخوارزميات الثلاث باستخدام وحدة معالجة مركزية إضافية بسبب حسابات فك التشفير. بالإضافة إلى ذلك، لا يزال وقت الاستجابة للمكالمة في جميع الحالات بمستوى وقت استجابة للمكالمة مقبولاً لقضاء 50 مللي ثانية في حمل مكالمات أعلى يبلغ 1000 مكالمة. باختصار، من هذه النتائج، سيكتسب مزودو خدمة SIP فهمًا شاملاً واضحًا عندما يتعلق الأمر بنشر خوارزميات RSA في خدماتهم، مما يدفع صناعة الاتصالات الصوتية نحو التنفيذ السلس من ناحية، ويوفر مادة معيارية لمجتمعات البحث.