Scripting Security Based On RSA and RC6 with Header Identification Alert (HIA)

Kailash Patidar

Department of Computer Science Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore HOD CSE/IT

Received 2017-11-28; Accepted 2017-12-25
DOI: http://currentsciences.info/10.15520/ctst.v8i01.235.pdf

Abstract-
Web security is a critical part of correspondence now days. As the zone is perplexing because of the idea of its scripting and its document arrange. So there is a wide extension here. Cryptography can be connected for the information security yet if there should arise an occurrence of various document organizes all cryptography calculations are not appropriate as the encoding plan is diverse for various record positions. In this regard an efficient scripting security mechanism based on RSA and RC6 with header identification alert (HIA) has been presented in this paper. The key processing is completely random to showing the unbiased property. Our procedure enhance the security system because of this hybridization and irregular process, the arbitrary procedure can effective in producing runtime passwords inevitably. The fundamental advantage of RSA calculation is the square size and key length estimate expanded by 512 bits. The fundamental favorable position of RC6 is its variable nature of key size and information piece estimate. The most extreme upper bound is 2040 bits. Past research demonstrates RSA calculation is moderate so we have settled the last irregular key process by RC6 as it is quick. By the use of header identification alert it is capable in identification unauthorized access of data.

Keywords: JSP, HTML, CSS, Eavesdrop Time, HIA

1. Introduction:
Presently the web applications are utilized by many people groups in various ways. It is depended on the applications to complete trades, be it business, individual or something different [1]. It is intensely included with web applications when it dealing with a record trades, visit long range casual correspondence areas, et cetera [1-2]. This dynamic nature of the web applications empowers customers to data information that will choose how a webpage responds to the customer [3]. In many locales, these customer information sources are not authentically affirmed along these lines making such a site unprotected against cross-site scripting (XSS). XSS are a security issue that occurs in web applications. Assorted customers with different intentions can realize SQL Injection ambush in the particular course in the web world. The disobedient and most skillfully threatening strike is SQL Injection alteration [4]. In this Modify the hawkish supporting completions the affirmation, by sincere enroll with parts, for the game plan for of permit in-cause and to execute self-confident code [5]. With
respect to four frameworks and computation are proposed in [6-12], yet there is need of change in the said field. In [13] essayist prescribed bend in a flash a gaudy is really sure; the state interface uncovered by an intrigue transforms into the fundamental wellspring of Feign. SQL Hastily Attack derriere be second-hand by relations who truancy to wish relate wherever enter to the database and usurp, course of action or delete data for which they don't endeavor on approval. In [14] surrogate methodologies was propositional to adjust fulfill for SQL Injection Attacks, however striking of these courses of action have obstacles that impact their feasibility and practicability.

In [15] makers suggested that the passage structure can be used for SQL Injection system which can be contacted cryptography structure. Cryptography is old to settlement the dynamic clear serenity to encode or request exonerates incongruous shape from content [16]. The brisk information are precarious on the sender accessory in perform to endeavor them end and protected wean away wean a long way from out of line get to and now sent by methods for the framework. About the bits of knowledge are regular then the in opposition to development quality be low down for unscrambling depending upon estimation. Unscrambling is the lead of regular figures from by stealth pull in up to their one of a kind association [17-19]. Cryptography helps in securing the data in the correspondence channel. Encryption and Decryption technique are moreover proposed like DES, RSA, RC4 and RC5 computations [20]. Go based supposition truly be quick close to subset superset mining or dividing [21, 22]. It is except for profitable in the occasion wheel the trade discernments and the condition assurance be decision in this way go wrong tumult assurance be augmentations and the stay in the tolerant gathering up resolve be more constrained. In cryptography we do encryption on the noteworthy cheerful to dispatch the system euphoria and interpreting is only an unpropitious intermediation to shape the plaintext. In steganography we trap the new plaintext viscera change off, pleasant, PDF, pictures et cetera. The medium of symptom the creative substance will be to the opposite side sent to the tranny for data scrutinizing.

2. Literature Review:

In 2007, Garcia-Alfaro et al. [23] concentrate on the particular issue of averting cross-webpage scripting attacks against web applications. They have displayed an examination on kind of ambushes, and diagram current systems for their shirking. The central focuses what's more, limitations of each suggestion are inspected, and a choice game plan is displayed. Our proposal relies upon the usage of X.509 validations, and XACML for the surge of endorsement courses of action. By using our answer, creators or conceivably chiefs of a given web application can especially express its security essentials from the server side, and require the right approval of such essentials on a predictable client. This strategy is impeccably planned in nonexclusive web applications by moving in the SSL and secure occupy calls.

In 2010, Athanasopoulos et al. [24] displayed xHunter, an instrument that takes as data a web take after and inspects it for recognizing possible XSS abuses. xHunter does not give any obstructions against attacks in web applications and projects. The gadget is expected for setting up a colossal number of URLs and segregating XSS abuses. Using xHunter one can see how real XSS mishandle take after, what is the geographical transport of web programs that trigger XSS manhandle, and other gainful properties, which if joined can draw an unrivaled photograph of the XSS scene today.

In 2013, Sadeghian et al. [25] prompts leave a tremendous SQL marksman wonder opening Secretiveness, Integrity and openness of information in the database. In perspective of the quantifiable investigates this maker of grasp very influenced business. Choice the normal connect with nitpick a be dependable to catch or defame the SQL implantation is vital. To chat with this inquiry settle experts influence variety strategies to stand get codes, turn away SQL imburement attacks and recognize them. They solid a foul research of surrogate sorts of SQL imburement area and neutralizing activity systems. They decode subvention and deficiencies of each strategy. Such a characteristic help in the significant stop substitute experts to pick the right technique for the further audits.

In 2013, Sadeghian et al. [26] advise SQL nip is twosome of the commanding troubles for the web application security. In perspective of the audits by OWASP, SQL spot has the fundamental exorbitant in the electronic vulnerabilities. Makers acknowledged the significant nature of SQL markswoman put on, fiery they explored affirmed SQL drink exposure wheeze strategies and putting they base give the go by the revealing channels, a
while later they proposed a mix of plans which mitigates the risk of SQL mixture attack. In 2013, Amirmohammad Sadeghian et al. [27] fundamental they gave establishment information on this frailty. Develop they comprehended a construed revile of substitute sorts of SQL implantation bother. For each ambush they supply and containerize go off shows how the attack dispatches. Unequivocally they get the control take at arise date to beat SQL mixture and conclusion.

In 2015, Nithya et. al. [28] proposed that the general in hacking procedure to assault the web application is XSS. XSS vulnerabilities are being manhandled by the attackers to take web program's benefits, for instance, treats, capabilities et cetera by implanting the poisonous JavaScript code on the setback's web applications. Since web programs support the execution of charges introduced in Web pages to enable dynamic site pages aggressors can influence use of this component to approve the execution of noxious code in a customer's web to program. The examination of recognizable proof and reckoning of XSS help to keep up a key separation from this kind of strike. We depict a framework to recognize and keep this kind of control and from now on discard cross-site scripting assault.

In 2015, Gupta et al. [29] XSS attacks happen when an application takes precarious data and sends it to the program without true blue endorsement or escaping. This can achieve catching of customer sessions, harming locales and occupying the customers to malignant goals. They have shown new XSS defend approach which relies upon the OWASP rules available for balancing activity of XSS strikes. In this approach for XSS protection there is a XSS checker that will check for the unapproved characters in each parameter in the information and square them on both client side and server side of a web application. Client side courses of action decreases the run time overhead and server side game plans are more trustworthy as any attack happening when request is going from client to server will be recognized by server side game plan just anyway it procures runtime overhead. So a mix of both will be more energetic as it can check by far most of the ambushes and regulate runtime overhead reasonably. This approach is attempted on a model. It is discovered that this approach covers genuine classes of XSS ambushes i.e. reflected and set away and will require no additional frameworks.

In 2015, Maheswari et. al. [30] recommended that the cross website scripting assault broadly known as XSS attack abuses the web program rather than the application itself. This stances certified hazard to the architects who found the opportunity to ensure the security of the web organizations. Web intrusion acknowledgment systems are security programs that assistance the planners and furthermore the customers to evaluate whether events and activities occurring in a Web Application are certified. Their objective of Web IDS is to perceive interferences with high false alerts and low acknowledgment rate while eating up minor properties. Their proposed work demonstrates an interference area structure that separates web requests hunting down verification of noxious lead and gives a refined request examination. The implantation of vulnerabilities and strikes is without a doubt an effective way to deal with survey security frameworks and to point out their inadequacies and in addition courses for their change. This work endeavored a response for the vulnerable web strikes through ceaseless mechanical assemblies, for instance, DVWA and XSS Me.

In 2016, Rohilla et al. [31] proposed that most of the web applications have security vulnerabilities like XSS (Cross Site Scripting) attacks, phishing attacks which are abused by the aggressors to hack the accreditation and individual data from the web application for noxious reason. They have recommended XSS attacks about with their requests. Decision of setback web application which is helpless for XSS ambush and some lack of protection scanners are similarly discussed. A segment of the XSS worms are discussed in detail with honest to goodness relevant examinations and principles to thwart.

In 2014, Nugraha et al. [32] suggested several future internet (FI) architectures to address the problems of the Internet including flexibility, host-based addressing, and security. They have suggested that eXpressive Internet Architecture (XIA) as it is the most secure and open-source content-centric network (CCN). A few existing arrangements have been dissected to determine the necessities for the proposed arrangement. By executing the proposed convention, XIA is presently ready to alleviate the greater part of the evaluated arranges attacks. The assessment demonstrates that the proposed arrangement is more secure and less unpredictable over the current arrangements.

3. Methodology:
In this paper we have proposed an efficient Hybrid Web Data Security mechanism based on RSA and RC6 with header identification alert (HIA). Our work is categorized by the following five steps. It can also be better understood from the figure 1.

**Design:**

Our structure is comprised of java server pages (JSP) and Hyper Text Markup Language (HTML) which is principally utilized for server side programming. This system is made in Netbeans7.2 condition. The plan is produced by Cascading Style Sheets (CSS). Tomcat server 7 is utilized for conjuring of server customer condition. It is an electronic bundle to give summon of Java Server Pages (JSP) based web applications. It keeps running on the of course port of 8080 or it can be designed by the server require or utilize.

**Data Preprocessing:**

We have considered five distinctive sort of information for information security content, website pages, content file, pdf without pictures and picture records. In this procedure the picture information is changed over in the encoded upheld document that is a twofold record. This information is the plaintext for the following handling.

**Data Encryption:**

At the point when the server gets the demand from the customer. At that point it initially set up the information. At that point RSA calculation is connected for information encryption process. This procedure is connected by the calculation 1 and 2. In this procedure the plain information is connected to the mix of RSA and RC6 calculation. The fundamental advantage of RSA calculation is the piece size and key length measure expanded by 512 bits. The primary preferred standpoint of RC 6 is its variable nature of key size and information piece estimate. The most extreme upper bound is 2040 bits. Past research indicates RSA calculation is moderate so we have finished the last arbitrary key process by RC6 as it is quick.

![Figure 1: Flowchart](image)

Rivest-Shamir-Adleman (RSA) makes utilization of an expression with exponentials. Plaintext is scrambled in blocks, with every piece having a paired esteem not exactly some whole number n. That is, the square size must be less or equivalent to \( \log_2 n \) in practice, the block size is k bits, where \( 2^k < n \leq 2^{k+1} \). Encryption and decoding are of the accompanying structure, for some plaintext block M and ciphertext block C:

\[
C = M^e \mod n
\]

\[
M = C^d \mod n = (M^e)^d \mod n = M^{ed} \mod n
\]

Both sender and beneficiary must know the estimation of n. The sender knows the estimation of e, and just

Kailash Patidar / Scripting security based on RSA and RC6 with header identification alert (HIA)


beneficiary knows the estimation of d. Hence, this is an open key encryption calculation with an open key of \( KU = \{e,n\} \), and a private key of \( KR = \{d,n\} \).

First a key of length k bytes and the 128-piece plaintext square is stacked into words A; B; C; and D beginning with the low-arrange byte of A. These four w-bit words contain the yield figure content toward the end. Here we portray encryption and unscrambling. RC6 works with four w-bit enlists A; B; C; D which contain the beginning info plaintext and in addition the yield figure content toward the end of encryption. The main byte of plaintext or figure content is put at all critical byte of A; the last byte of plaintext or figure content is set into the most huge byte of D. We utilize \((A; B; C; D) = (B; C; D; A)\) to mean the parallel task of qualities on the privilege to enrolls on the left. The randomization process is applied according to algorithm 3.

Algorithm 1 [33]:
The encryption key \((e, n)\), is calculated in the following way:

Step 1: The public/private key pair is generated by the following steps:
- Choose two large primes at random – a, b
Step 2: Calculate system modulus \( N = a \cdot b \)
- \( \phi(N) = (a-1)(b-1) \)
- Step 3: Encryption key e is now chosen in this manner that the \( e \) lies in \( 1 < e < \phi(N) \), \( \gcd(e, \phi(N)) = 1 \)
- Step 4: Decryption key d is calculated then \( e \cdot d \equiv 1 \pmod{\phi(N)} \) and \( 0 \leq d \leq N \)
- Step 5: public encryption key: \( KU = \{e, N\} \)
- Step 6: private decryption key: \( KR = \{d, a, b\} \)
- Step 7: For encrypting the message M first receive the public key of the receiver: \( KU = \{e, N\} \)
- Step 8: For decrypting it use the private key \( KR = \{d, a, b\} \) \( M = C^d \mod N \)

Algorithm 2 [34]:

Step 1: b byte preloaded into the c-word
Step 2: \( L[0, \ldots, c - 1] \)
Step 3: Number r of rounds [all the initialization are random]
- \( x = \) First initialization
- \( y = \) Second initialization
Yield:
Step 4: w-bit round keys \( S[0, \ldots, 2r + 3] \)
Strategy:
\( S[0] = x \)
Step 5: for \( i = 1 \) to \((2r + 3)\) do
\( S[i] = S[i - 1] + y \)
Step 6: Each block key is generated until reached to the end of file vector.
Step 7: Shifting process
- \( A = S[i] = (S[i] + A + B) \ll 3 \)
Step 8: Then again shifting is performed with the 3 bit java shifting to make the substitution matrix.
Step 9: The whole process is applied to the whole block division achieved.
Step 10: The final key is generated according to the r rounds.
Step 7: End;

Algorithm 3: Random Algorithm

INPUT: (character, Number)
OUTPUT: random_data
random_data = F(character, number)
Key’ = Math.random(character, number)
Return Key

Data Decryption:
The reversible procedure is connected for recovering the information after encryption for translating part with the four diverse keys. The keys are open, private, modulus and the normal RC6 key. It is secured by the encryption mystery key. In the wake of applying the correct mystery word we will fulfill the last data.

Header identification alert (HIA):
In this procedure the information will be send to the customer with the character header which is automatically recreated when the other data is recovered according to the match customer. Along these lines the assault/unauthorized access will be recognized.

4. Results:
In this scripting approach the server readies a few document for the information storage and additionally the key coordinating. The fundamental records which are set up at the season of information sending are the appeared in table 1. It demonstrates the when the information is send the put away HIA which will be changed for some document in table 2 as it is assaulted before the record is open from the customer side and it is advised legitimately to the server. The server sending and accepting time likewise plainly portrayed by table 2. Our character imitation based assault recognition system as appeared in table 3 unmistakably demonstrates that the time taken in milliseconds is less to know the progressions or updation performed. It is favorable position as the
security instrument is extremely productive and in addition we have the adaptability of location. Table 4, figure 2 and figure 3 show the comparison from the previous methods and it shows the efficiency of our approach used total 4 messages / key and 3 keys are random. The key length of our mechanism is also variable due to RC6 mechanism.

**Table 1: Data Preparation Stage**

<table>
<thead>
<tr>
<th>File Name</th>
<th>File Size</th>
<th>RC6 Key</th>
<th>Replica Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>image4.jpg</td>
<td>12600</td>
<td>aB7Sq1B7</td>
<td>b</td>
</tr>
<tr>
<td>a3.doc</td>
<td>2932</td>
<td>dJ8K6G1</td>
<td>b</td>
</tr>
<tr>
<td>a1.doc</td>
<td>761</td>
<td>cZ5Bm0A6</td>
<td>b</td>
</tr>
<tr>
<td>w2.html</td>
<td>53</td>
<td>fN2M6G9</td>
<td></td>
</tr>
<tr>
<td>pd2.pdf</td>
<td>20859</td>
<td>hG0Po5F1</td>
<td>b</td>
</tr>
<tr>
<td>n1.txt</td>
<td>678</td>
<td>hHPm07H1</td>
<td>b</td>
</tr>
<tr>
<td>image1.jpg</td>
<td>7731</td>
<td>iZ8NmTH9</td>
<td>b</td>
</tr>
<tr>
<td>pd3.pdf</td>
<td>71212</td>
<td>kSUu8I4</td>
<td>b</td>
</tr>
<tr>
<td>MAIN.CPP</td>
<td>4194</td>
<td>lD0Rc0A8</td>
<td>b</td>
</tr>
<tr>
<td>do1.doc</td>
<td>23552</td>
<td>h55r0A5</td>
<td></td>
</tr>
<tr>
<td>n2.txt</td>
<td>20586</td>
<td>mA1D8b14</td>
<td></td>
</tr>
<tr>
<td>w1.html</td>
<td>1249</td>
<td>nW8Nm0A5</td>
<td></td>
</tr>
<tr>
<td>image2.jpg</td>
<td>2138</td>
<td>xJ6G0A2</td>
<td></td>
</tr>
<tr>
<td>pd1.pdf</td>
<td>405044</td>
<td>yIRj8I4</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2: Data Finally Send:**

<table>
<thead>
<tr>
<th>File Name</th>
<th>RC 6 Key</th>
<th>hidden</th>
<th>Sending time</th>
<th>Rec time</th>
<th>File Size</th>
<th>Client</th>
<th>Public Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>n2.txt</td>
<td>mA1Db8I4</td>
<td>a</td>
<td>11:53:4:360</td>
<td>11:53:4:391</td>
<td>20586</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>MAIN.CPP</td>
<td>lD0Rc0A8</td>
<td>b</td>
<td>11:51:5:200</td>
<td>11:51:5:206</td>
<td>4709</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>w1.html</td>
<td>nW8Nm0A5</td>
<td>b</td>
<td>11:54:1:456</td>
<td>11:54:1:485</td>
<td>7731</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>pd3.pdf</td>
<td>kSUu8I4</td>
<td>b</td>
<td>2:17:25:327</td>
<td>2:17:25:327</td>
<td>71212</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>image1.jpg</td>
<td>iZ8NmTH9</td>
<td>b</td>
<td>11:52:24:829</td>
<td>11:52:24:845</td>
<td>7731</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>image2.jpg</td>
<td>lD0Rc0A8</td>
<td>b</td>
<td>11:52:4:612</td>
<td>11:52:4:617</td>
<td>7731</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>a1.doc</td>
<td>cZ5Bm0A6</td>
<td>b</td>
<td>11:51:5:320</td>
<td>11:51:5:320</td>
<td>761</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>do1.doc</td>
<td>h55r0A5</td>
<td>b</td>
<td>11:51:5:944</td>
<td>11:51:5:944</td>
<td>761</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>pd1.pdf</td>
<td>yIRj8I4</td>
<td>b</td>
<td>11:51:5:919</td>
<td>11:51:5:935</td>
<td>405044</td>
<td>abc</td>
<td>65537</td>
</tr>
<tr>
<td>n1.txt</td>
<td>hHPm07H1</td>
<td>b</td>
<td>11:52:55:499</td>
<td>11:52:55:530</td>
<td>678</td>
<td>abc</td>
<td>65537</td>
</tr>
</tbody>
</table>

**Table 3: Data after attack:**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Attack time</th>
<th>Server response time</th>
<th>difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN.CPP</td>
<td>11:14:36:765</td>
<td>11:14:36:796</td>
<td>31</td>
</tr>
<tr>
<td>n1.txt</td>
<td>4:14:15:765</td>
<td>4:14:15:812</td>
<td>47</td>
</tr>
<tr>
<td>w2.html</td>
<td>4:14:29:711</td>
<td>4:14:29:758</td>
<td>47</td>
</tr>
</tbody>
</table>
Table 4: Comparison:

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Number of Messages</th>
<th>Amount of Random Numbers</th>
<th>Number of Data Encryption</th>
<th>Number of Data Decryption</th>
<th>Key Length (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffie-Hellman</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Lamport’s Password</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>S Key one time password</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>Message Binding</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Timestamp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Luo-Shieh-Shen Protocol</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Yoon-Jeon Protocol</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Tseng-Jou Protocol</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Nugraha et al. [32]</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>280</td>
</tr>
<tr>
<td>Proposed Work</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>512-2040</td>
</tr>
</tbody>
</table>

Figure 2: Keys Comparison

Figure 3: Random Keys Comparison
5. Conclusion:
In this paper an efficient script security mechanism based on RSA and RC6 algorithm has been developed. In this approach the web information is secured with the assistance of half breed calculation in light of RSA and RC6 calculation. In this process the data will be sending to the client with the header identification alert (HIA) which is automatically replicated when the other information is retrieved as per the match client. This is an efficient method for unauthorized alteration detection in the scripts. The primary advantages of our approach are the irregular keys, key length, information security and opportune assault location by character replication. The engaging outcomes will diagram the impact of our system.

References:
Kailash Patidar / Scripting security based on RSA and RC6 with header identification alert (HIA)

2011 (pp. 381-390). Springer Berlin Heidelberg.


20. Dubey AK, Dubey AK, Namdev M, Shrivastava SS. Cloud-user security based on RSA and MD5 algorithm for resource attestation and sharing in java environment. In Software Engineering (CONSEG), 2012 CSI Sixth International Conference on 2012 Sep 5 (pp. 1-8). IEEE.

21. Dubey AK, Dubey AK, Agarwal V, Khandagre Y. Knowledge discovery with a subset-superset approach for Mining Heterogeneous Data with dynamic support. In Software Engineering (CONSEG), 2012 CSI Sixth International Conference on 2012 Sep 5 (pp. 1-6). IEEE.


