Web Crawler based on Secure Mobile Agent

Qasim Rafiq, Sandhya Pundhir

Related papers

[PDF] WEB CRAWLER BASED ON SECURE MOBILE AGENT
Sandhya Pundhir, Qasim Rafiq

ACICE-039
Lavanya Damodaram

A Novel approach towards the Performance Optimization of the Embedded Systems
Sastry Kodanda Rama Jammalamadaka
INTRODUCTION

The World Wide Web (WWW) is a big dynamic network containing a huge amount of information. The Web is a collection of interconnected documents and other resources, linked by hyperlinks and URLs. Web crawlers are used to recursively traverse and download web pages for search engines to create and index later. The concept of mobile agent in web crawler has increased the crawling speed. It also minimizes the network overhead. But it suffers with the problem of security. Our proposed method provides better result than traditional web crawler and more reliable and secured than SMABC (Security on Mobile Agent based Crawler). It protects the mobile agent and the data it carries. No harmful malicious hosts, eavesdroppers and attacker can access the mobile agent code.

Keywords: Web crawler, Mobile Agent, Web Pages, Security, Probabilistic encryption

1. Mobile Agent

Mobile agent is a software program functions on behalf of a user in a distributed environment, and is able to migrate independently from one host to other to accomplish the assigned work.

Features of mobile agent that provide a better architecture for web crawler[8][9][10]:-
1- They reduce the network load: Mobile agents allow dispatching the information to the destination host at a time so that interaction can take place locally rather than transferred over the network.
2- Overcome network latency: A mobile agent migrates to the destination host, disconnect from the owner and then dispatch the information there. After completing the whole task it reconnects with the owner and returns back.
3- They encapsulate protocols: Mobile agent encapsulates the data as well as code.
4- They execute asynchronously and autonomously: Mobile agents executes task independently and autonomously on the destination host on behalf of their owner.
5- They adapt dynamically: Mobile agent can creates their clones and distribute them to the network for solving a critical problem optimally.
6- They are naturally heterogeneous:. Mobile agent mainly depends on the executing environment and is independent of both hardware and software.
7- They are robust and fault tolerant: Mobile agent can adapt dynamically while solving a critical problem. This feature provides robust and fault tolerant capability.

**Drawbacks of mobile agent based web crawler :-**

1- Unauthorized Access: Attackers or malicious agents tries to access the information and resources of platform without having authority to access them. Some security policy is required so that only the authorized agents can access the resources with proper permission.

2- Masquerading: A malicious agent gets the identity of another authorized agent and bluffs the host by using the information. These attackers after accessing, can damage the information and resources of the platform.

3- Eavesdropping: Malicious platform monitors the behavior of the visiting agent and gets the agent’s information.

4- Alteration: Sometime a malicious platform tries to introduce some other data or code in mobile agents information thus altering the actual information of the mobile agent.

5- Confidentiality: The information carried by mobile agent or stored on the platform must be confidential so that they can be accessed by authorized agents only.

**2. RELATED WORK**

Lots of previous work has focused on the web crawler using mobile agents so far. Work has been done to provide security of mobile agent. In [5] S. Mudumbai, A. Essiari, W. Johnsion presented a mobile agent system (Anchor Toolkit) that provides for the secure transmission and management of mobile agents The toolkit protects the agents being dispatched between hosts through encrypted channels. In [12] W. Farmer, J. Guttman, V. Swarup presented a security mechanism (State State Appraisal) which defines a security mechanism for protection of mobile agents Appraisal functions are used to determine what privileges to grant to an agent based both on conditional factors and whether the identified state invariants hold. An agent platform uses the functions to verify the correct state of an incoming agent and to determine what privileges the agent can possess during execution. Privileges are issued by a platform based on the results of the appraisal function and the platform’s security policy. In [7] F. Gasparetti, A. Micarelli presented an adaptive web search system based on a multi-agent reactive architecture, which comes from biological researches on the ant searching behavior. As an alternative here an algorithm is proposed for the security of mobile agent based on probabilistic encryption technique. Major advantage is that the information carried by mobile agent will be more correct and secured [11]. Because there is no need to consider a trusted computer and pre-shared information.

**3. ALGORITHM IMPLEMENTED**

Web crawler is implemented using mobile agent technology and security of mobile agent is implemented using probabilistic encryption technique to achieve better result. Work is mainly categorized as Client side work and Server side work.

**Client side Algorithm steps are as follows:**

1- Input a seed url.
2- Verify it is correct.
3- Search the url from the database containing uncrawled pages.
4- Extract the urls from the seed url and store in a queue.
5- Repeat 3 till the list is not empty.
6- encrypt the url.
7- Encrypted url is given to mobile agent.

**Server side Algorithm steps are as follows:**

1- Read the url received from mobile agent.
2- Decrypt the url.
3- Redirect the crawled url to the database.
4- Derive the bit stream $y$ as follows.

**Encryption Algorithm steps are as follows:**

1- Convert the input url string into binary message string $m$.
2- Compute $A(i,j)$.
3- Derive the bit stream $y$ as follows.

$$b_i^j = 0 \text{ if } a_{(i,j)} \in J_t^+ , 0 \leq j \leq t - 1$$

$$b_i^j = 1 \text{ if } a_{(i,j)} \equiv x_{(i,j)} \mod n , 0 \leq j \leq t - 1$$

4- Compute $[c] \oplus [y]$.
5- Encrypted message to be send to mobile agent is $([c], A(i,j))$.

**Decryption Algorithm steps are as follows:**

1- Server receives $([c]A(i,j))$. 
Web Crawler using mobile agent is implemented in Java on Windows-XP platform and experiments are run on Intel core2 duo n series CPU with 3GB RAM.

### 3.1 Algorithm Explanation

Every url is associated with a web page or site. Whole content of the web page is downloaded by the web crawler. Other relevant urls associated with the seed url are extracted from the web page. Web crawler’s working starts with a seed url. User enters the url to be searched at the client side. The seed url is stored in a queue. Seed url is an integer denoting a relevant web document and then redirects it to the crawler manager. At the client side, crawler manager receives the encrypted url from the server. Mobile agent carrying the encrypted url is received by the server. Server first decrypts the url. Mobile agent search for relevant web document and then redirects it to the crawler manager. At the client side, crawler manager maintains a queue for storing the new linked urls extracted from downloaded web document.

Other urls downloaded by web crawler are first checked with the queue to avoid duplicate urls and then stored in a queue maintained by the crawler manager.

### Mathematical background

An integer \( a \in \mathbb{Z}_n^* \) is a quadratic residue modulo \( n \), if there exists some \( x \in \mathbb{Z}_n^* \) such that \( a \equiv x^2 \mod n \). Then it is denoted as \( a \in \mathbb{Q}_n \). Otherwise \( a \) is quadratic non-residue modulo \( n \) and is denoted as \( a \in \overline{\mathbb{Q}}_n \).

Let \( n \geq 3 \) be odd number, Jacobi symbol \( \left( \frac{a}{n} \right) \) is defined as:

\[
\left( \frac{a}{n} \right) = \begin{cases} 
-1 & a \in \overline{\mathbb{Q}}_n \\
1 & a \in \mathbb{Q}_n \\
0 & \gcd(a, n) > 1
\end{cases}
\]

For \( n \) being a product of two primes, given an element \( a \in \mathbb{Z}_n^* \), if \( \left( \frac{a}{p} \right) = 1 \), it is hard to decide whether \( a \in \mathbb{Q}_n \). Whereas, if \( \left( \frac{a}{p} \right) = -1 \), then it is sure that \( a \in \overline{\mathbb{Q}}_n \).

If \( n = p \times q \) and the two prime factors are known then, given any \( a \in \mathbb{Z}_n^* \), if \( \left( \frac{a}{p} \right) = 1 \), it is easy to determine whether \( a \in \mathbb{Q}_n \). In that case \( a \in \mathbb{Q}_n \), if both \( \left( \frac{a}{p} \right) = 1 \) and \( \left( \frac{a}{q} \right) = 1 \). On the other hand if both \( \left( \frac{a}{p} \right) = -1 \) and \( \left( \frac{a}{q} \right) = -1 \), then \( a \in \overline{\mathbb{Q}}_n \).

Let \( n \geq 3 \) be an odd composite number, \( J_n^+ \) is the set of all pseudosquares and defined as \( J_n^+ = \{ a \in \mathbb{Z}_n^* \mid ab=1 \} \). \( J_n^+ \) is the set of all quadratic nonresidues and defined as \( J_n^- = \{ a \in \mathbb{Z}_n^* \mid ab=-1 \} \).

A binary stream is generated by the crawler manager. Two methods used to compute the matrix A(i,j) are as follows:

**Method I:** If \( i^{th} \) bit of the binary stream is 1 (i.e. \( b^i = 1 \))

- Crawler manager selects \( t \) number of elements \( x_j \in \mathbb{Z}_n^* \) for \( 0 \leq j \leq t - 1 \) and computes \( a_j \equiv x_j^2 \mod n \)

**Method II:** If \( i^{th} \) bit of the binary stream is 0 (i.e. \( b^i = 0 \))

- Crawler manager selects \( t \) number of elements such that \( a_j \in J_n^+ \), \( for \ 0 \leq j \leq t - 1 \)

Encryption algorithm used is as follows:

- Let \( m \) is k bit url message that is to be send. The bit representation of \( m \) is \([b_{k-1}, \ldots, b_1, b_0]_m \). To encrypt the message, a bit stream of \( k \) bit is constructed at the client side. Bit wise xor with \( b_1^m \) and \( b_0^m \) is used as encryption operation. Bit stream \( y \) is constructed using Method I and Method II.

Crawler manager encrypt the url message \( m \) to \( c \) as:

\[
C = \begin{bmatrix}
\vdots & \vdots & \vdots & \vdots \\
\vdots & b_k^m & \vdots & \vdots \\
\vdots & b_1^m & \vdots & \vdots \\
\vdots & b_0^m & \vdots & \vdots \\
\end{bmatrix}
\oplus \begin{bmatrix}
\vdots & \vdots & \vdots & \vdots \\
\vdots & b_{k-1}^i & \vdots & \vdots \\
\vdots & b_{i-1}^j & \vdots & \vdots \\
\vdots & b_0^i & \vdots & \vdots \\
\end{bmatrix}
\]

This encrypted message \( c \) and \( A(i,j) \) matrix is given to the mobile agent. Decryption algorithm used is as follows: Server receives the encrypted information from the mobile agent. Reconstructs the String \( y \) from \( A(i,j) \) and then decrypts the encrypted url message \( [m] = [c] \oplus [b] \).
4. RESULTS

This algorithm considers the problem of security on mobile agent. First is web crawling speed increases and bandwidth utilization decreases using mobile agent. Secondly, using probabilistic encryption for mobile agent helps in protecting information from attackers without any pre-shared information.

Table 4.1: Parameters Value comparison for Traditional and My_web_crawler.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Traditional crawler</th>
<th>My-web crawler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Crawl Time(ms)</td>
<td>40944</td>
<td>37089</td>
</tr>
<tr>
<td>Avg. data downloaded(KB)</td>
<td>30.476</td>
<td>33.68</td>
</tr>
</tbody>
</table>

Table 4.2: Parameters Value comparison for security on mobile agent using RSA encryption and Probabilistic encryption

<table>
<thead>
<tr>
<th>Parameters</th>
<th>RSA encryption</th>
<th>Probabilistic encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Message size(bit)</td>
<td>168</td>
<td>168</td>
</tr>
<tr>
<td>Encryption Time (ms)</td>
<td>49.88</td>
<td>48.67</td>
</tr>
<tr>
<td>Decryption Time (ms)</td>
<td>32.77</td>
<td>30.96</td>
</tr>
</tbody>
</table>

This algorithm considers the problem of security on mobile agent. First is web crawling speed increases and bandwidth utilization decreases using mobile agent. Secondly, using probabilistic encryption for mobile agent helps in protecting information from attackers without any pre-shared information.
Benefit in crawling time = 100 - (37089 / 40944) * 100 = 9.415 %
Percentage security improvement in encryption = (100 - (48.67/49.88)) * 100 = 2.426%
Percentage security improvement in decryption = (100 - (30.96/32.77)) * 100 = 5.523%

5. CONCLUSION AND FUTURE WORK
In this paper an algorithm is proposed based on probabilistic Encryption technique. It performs comparatively better than traditional web crawler in terms of crawling speed, robustness and more secured than SMABC. In this scheme no pre-shared secret information is required as in case of other encryption schemes. It has achieved security improved by 3.97 % and crawling time improvement 9.415 %. Finally, web crawler based on mobile agent using probabilistic encryption provides complete secrecy and security for smaller message space and smaller key size. Dishonest opening of fake information to the attackers is under implementation.

REFERENCES