An Enhancing Security Research of Tor Anonymous Communication to Against DDos Attacks

Tao FENG*, Ming-Tao ZHAO

College of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
zhaomingtao1226@163.com
**Corresponding author: fengt@lut.cn*

Abstract: Tor (The Second Onion Router) is modified by the first generation onion router and known as the most prevalent anonymous communication system. According to the advantage of low latency, high confidentiality of transmission content, high security of communication channels and et al., Tor is widely used in anonymous Web browsing, instant message and so on. However, the vulnerability and blemish of Tor affect system security. An identity and Signcryption-based concurrent signature scheme was used to prevent the behaviors of attackers from inserting controlled nodes and conspiring to make DDos attacks. The integrated security of Tor system was enhanced in our scheme. In addition we have proved the scheme.

1 Introduction

Anonymous communication can hide the telecommunication relationship in the business flow without changing the existing network protocol and complete the protection of the privacy information such as the user's identity. Consequently, the eavesdropper cannot directly or indirectly infer the telecommunication relationship and the identity of communications [1]. According to the advantage of simple configuration and excellent performance, the third system of Tor anonymous communication system has been developed quickly. Tor can resist the attack from listening and flow analysis, which is currently one of the most widely, used anonymous communication system on the Internet [2].

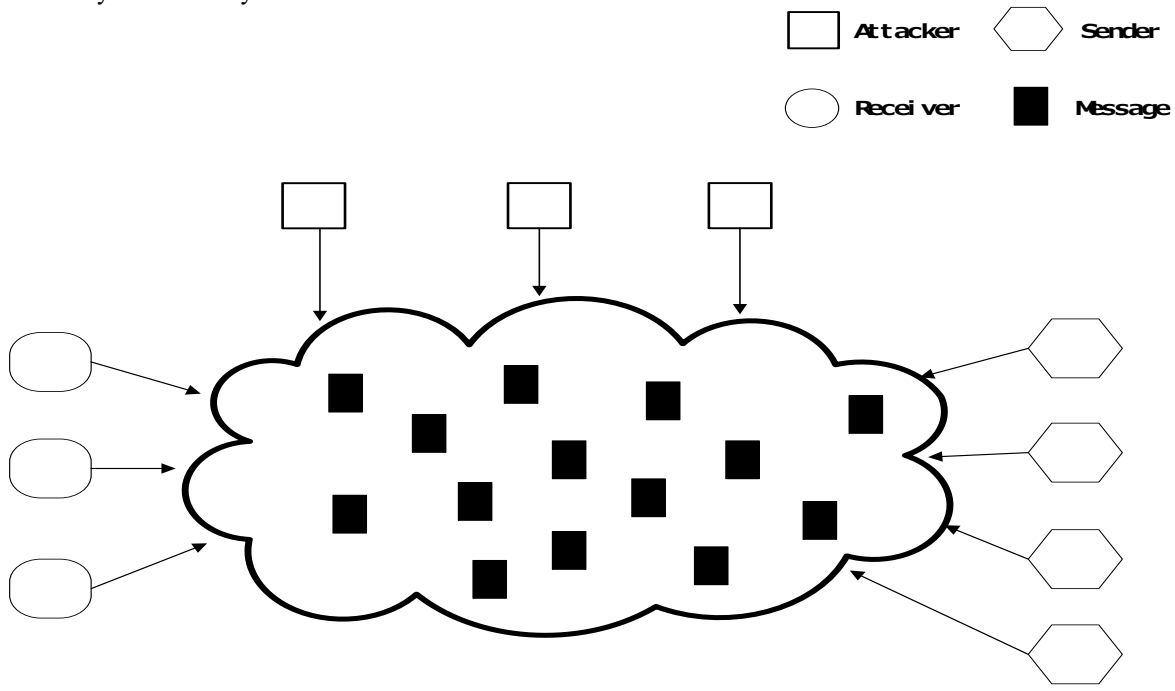

Figure 1. Traditional network model

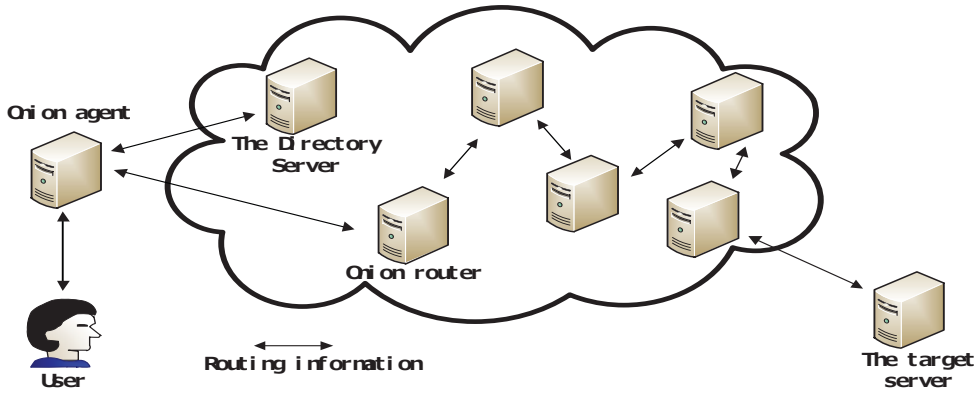

Figure 2. Tor network model

Compared with the traditional network communication, Tor can withstand strong flow analysis. So, Tor can guarantee the anonymity and security of the communication better. What's more, Tor can coordinate with existing many applications, so Tor has better flexibility and extensibility [3]. As one large-scale application anonymous communication system, in the design of Tor, performance, user demand and other factors have been considered. In the realization of the specific function, some unsafe strategy were adopted which exposed the vulnerability. In order to a low delay, Tor does not have completely anonymous, plus with lack of user access certification, which have has provided the favorable natural conditions for the attacker to DDos attacks[4].

In order to improve the security of Tor anonymous communication system, a new kind of anonymous communication scheme is proposed by Zhou et.al [1], the scheme make up the deficiency of the traditional Tor anonymous communication system in some extent. However, the scheme increases time. Based on the technology signcryption, Zheng et.al [4] proposes a node authentication mechanism, which is realizes a third party certification between TTP and node and among nodes by introducing the authority of the third party (TTP).

In the present study, a concurrent signature scheme based on identity authentication and signcryption[5] were adopted. The scheme can Verifies the reliability of accessed Tor nodes eliminate the false nodes. In the stage of middle volunteer node (OR) added to the Tor network, confirm the identity of the both sides through mutual authentication, and use signcryption to ensure the authentication information are unforgeable in the validation process. Through the authentication mechanism of the directory server and nodes, and between nodes, the scheme can effectively improve the reliability of system Tor nodes, prevent malicious nodes got into the system to improve security and effectiveness.

2 Prepared Knowledge

2.1 Bilinear Maps

Let G_0 and G_1 be two multiplicative cyclic groups of prime order p. Let g be a generator of G_0 and e be a bilinear map, $e:G_0 \times G_0 \to G_1$. The bilinear map e has the following properties:

(1) Bilinearity: for all $u,v \in G_0$ and $a,b \in Z_p$, we have $e(u^a,v^b)=e(u,v)^{ab}$.

(2) Non-degeneracy: $e(g,g) \neq 1$.

We say that G_0 is a bilinear group if the group operation in G_0 and the bilinear map $e:G_0 \times G_0 \to G_1$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a,g^b)=e(g,g)^{ab}=e(g^b,g^a)$.

2.2 Tor System Basic Model

The Tor system is a distributed global network, using a centralized directory management structure. The entire Tor network consists of seven parts [6]:

(1) User Clients(User): A computer that communicates anonymously with the outside world through onion routing networks, usually the user needs to connect to the Tor network with an onion proxy.

(2) Onion Proxy: Responsible for the establishment of data transmission channel, receive TCP data stream, and sends the data stream to the established transmission channel.

(3) Onion Router: To establish an anonymous communication link and to forward data.

(4) Entry Node: The first onion router on the path, which communicates directly with the user.

(5) Exit Node: Refers to the last onion router on the path that communicates directly with the target server.

(6) Middle Node: Means an onion router on a path other than the ingress and egress nodes.

(7) The Directory Server: Mainly responsible for the collection, management, storage of all OR information and node exit and join the network-related operations.

Onion routing system can be roughly divided into three stages which are establishing connection, transferring data and closing connection [7].

2.3 Establishment of Anonymous Communication Link

First, Alice's onion proxy sends a "Create" command packet to the first onion router OR1 which is chosen. This "Create" packet, let C_{AB} represent the CircID between Alice and Bob, and also includes the first half of the Diffie-Hellman handshake information (g^{x1}), then use Bob's public key encryption. Bob returns a Created packet, which contains the latter half of Diffie-Hellman handshake information (g^{y1}) and the hash value $H(\mathrm{K}_1)$ of the session key K_1 (g^{x1y1})[8]. Once the link is established, Alice and Bob can send forward packets. The forwarded packets are encrypted with their negotiated session key K_1.

If you want to expand this virtual circuit, Alice sends an Relay Extend packet with K_1 encrypted to Bob, which contains the encrypted information g^{x2} and the next-hop onion router address (Known as Carol).Bob builds a "Create" packet. Use C_{BC} to represent the CircID between Bob and Carol, also contains the half of the handshake information g^{x2}, then encrypted with Carol's public key and sent to Carol. Carol returns a "Created" packet, which contains the latter half of Diffie-Hellman handshake information (g^{y2}) and the hash value $H(\mathrm{K}_2)$ of the session key K_2 (g^{x2y2}). When Bob receives the Carol response message, Bob encrypts a Relay Extended packet with the session key K_1 between it and Alice (contains g^{y2} and $H(\mathrm{K}_2)$), then send it back to Alice. Now, the link to Carol, Alice, and Carol's session key is K_2 [9].

Alice just need repeats the process, it can gradually expand the length of the virtual circuit, and finally to establish a connection with the destination node (Figure 3 Tor link establishment process).

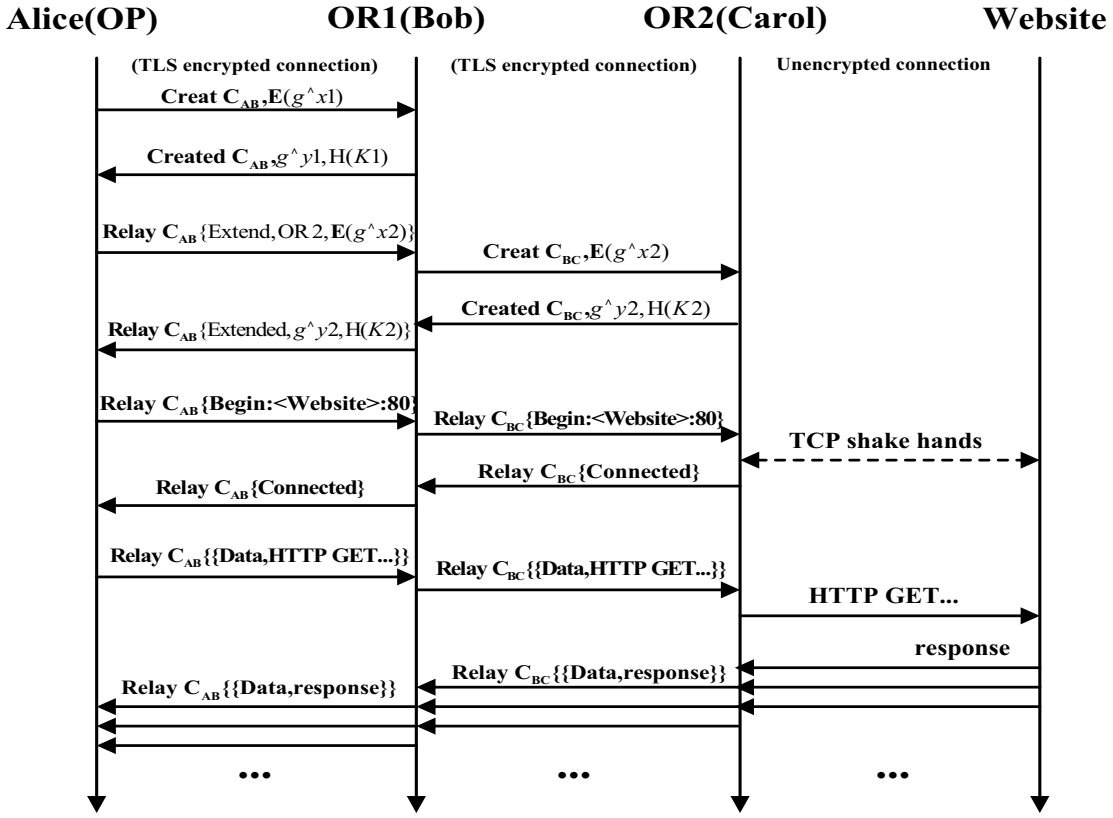

Figure 3 . Tor link establishment process

2.4 The Data Transfer Process of Tor

OP sends the encrypted packet to the OR node of next-hop. Theoretically, each node on the Tor rerouting path is randomly selected by the user. But in practical application, the OR nodes responsible for forwarding are randomly selected by Tor. In Figure 4, we choose a client OP, three OR nodes to form an anonymous communication link. Shows the transfer of data from the OP to the server E [10].

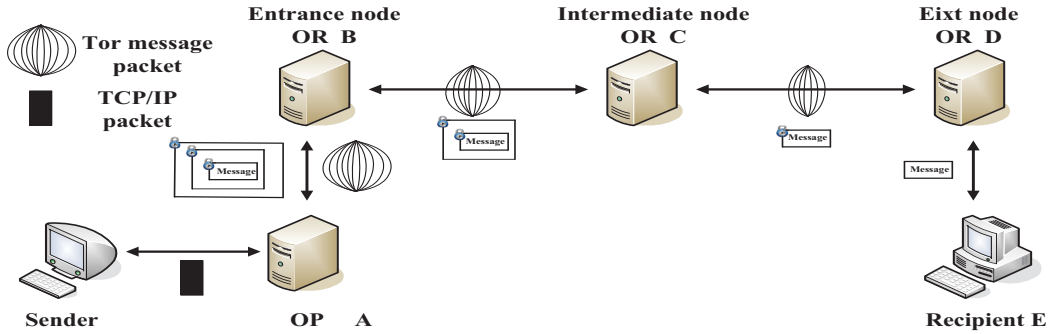

Figure 4. Tor data transfer process

As can be seen from the figure, the link establishment procedure from A to E, need to go through B, C, D three forwarding nodes (Relay). Corresponding to each forwarding node, before the transmission, communication data was also encrypted from the back to front by onion agent according to its public key, which passed Tor nodes, and formed a recursive hierarchical data structure called the onion package [11]. Thus, the communication usually uses Onion to replace the usual TCP / IP packets. This ensures that each onion router only knows the onion router that passes it data and the next-hop onion router it wants to pass data to, without knowing all the onion routers that are going to pass between the sender and the receiver. So as to achieve the purpose of hiding routing information and resist traffic analysis. In the sending process, the message packets are decrypted one layer while passed an onion routing node. Until after routing node D export of Tor network, the sending information is restored to the plain text message. In the whole link, the forwarding node only needs to know the nodes of the next hop and the next hop. Thus ensuring the security of the identity information of the two sides of communication [12].

3 An Identity and Signcryption-based Concurrent Signature Scheme

User A represents the initial signer, and user B represents the matching signer. The signcryption algorithm in the scheme is mutual. A and B each run this algorithm once. In this scheme, i is the signer and j is the matching signer, $i, j \in \{\mathrm{A}, \mathrm{B}\}$ the scheme includes the following algorithm:

(1) System establishment:

G_1 is a q -order cyclic additive group, q is a prime number, p is the generator of G_1; G_2 is a g -order cyclic multiplicative group. PKG(Private Key Generator)randomly selects the master key $s \in Z_q^*$, calculate $P_{pub} = s^P$ as the public key of PKG. Select two secure Hash functions: $H_0 : \{0,1\}^* -> G_1, H_1 : \{0,1\}^l \times G_1 -> \{0,1\}^* \quad G_1 \quad Z_q^*$, l is the length of the plaintext to be encrypted. PKG public system parameters $\pi = \{\mathrm{q}, \mathrm{G}_1, \mathrm{G}_2, \mathrm{e}, \mathrm{l}, \mathrm{P}, \mathrm{P}_{pub}, \mathrm{H}_0, \mathrm{H}_1\}$.

(2) Key extraction: using A,BID: ID_A, ID_B, PKG calculate $S_A = sQ_A$, $S_b \approx sQ_B$, and $Q_A = H_0(\mathrm{ID}_A)$, $Q_B = H_0(\mathrm{ID}_B)$. The public key of A and B is: Q_A, Q_b, and private keys is $\mathrm{S}_A, \mathrm{S}_B$.

(3) keystone mapping(Footprint):The input parameter of this algorithm is (β, k_i), and $\beta \in G_1$, $k_i \in Z_q^*$, k_i is the keystone selected by the signer. Calculate $f_i = \beta + H_0(\mathrm{k}_i)$, output f_i.

(4) Signcryption: the input parameter for this algorithm is $(\mathrm{ID}_i, \mathrm{ID}_j, \mathrm{Q}_i, \mathrm{Q}_j, \mathrm{S}_i, \mathrm{f}_i, \mathrm{m}_i)$, the algorithm runs as follows:

①selected randomly $a_i \in Z_q^*$, Calculate $U_i = a_i P$, $w_i = e(\mathrm{P}_{pub}, \mathrm{Q}_j)^{ai}$.

②let $R_j = f_i$, Calculate $h_i = H_1(\mathrm{m} \| \mathrm{R}_j \| \mathrm{ID}_A \oplus ID_B)$.

③selected randomly $\mathrm{r} \in rZ_q^*$,calculate

$$\begin{cases} R_i = rQ_i - R_j - h_j Q_j \\ h_i = H_1(\mathrm{m} \| \mathrm{R}_i \| \mathrm{ID}_i \oplus ID_j) \\ V = (\mathrm{h}_i + \mathrm{r})\mathrm{S}_i \end{cases}$$

④Calculate: $c_i = m \oplus w_i$, is the ciphertext after encrypts, output signcryption $\sigma_i = \{c_i, V, \mathrm{R}_A, \mathrm{R}_B\}$.

(5) Designcrypt: This algorithm is matching signers restore the ciphertext to plaintext. The input parameter is $(\mathrm{c}_i, \mathrm{U}_i, \mathrm{S}_j)$, Calculate: $\mathrm{m} = c_i + e(\mathrm{U}_i, \mathrm{S}_j)$ output m.

(6) Mutual authentication (Averify): This algorithm is mutual authentication between A and B. It is used to verify the validity of the received signature and the plaintext obtained in the previous algorithm. The input for this algorithm is $\sigma_i = \{c_i, \mathrm{Q}_A, \mathrm{Q}_B, \mathrm{m}\}$. First calculate:

$$h_A = H_1(\mathrm{m} \| \mathrm{R}_A \| \mathrm{ID}_A \oplus ID_B)$$
$$h_B = H_1(\mathrm{m} \| \mathrm{R}_B \| \mathrm{ID}_A \oplus ID_B)$$

Then, verify if $e(\mathrm{P}_{pub}, \mathrm{R}_A + \mathrm{h}_A Q_A + \mathrm{R}_B + h_B \mathrm{Q}_B) \stackrel{?}{=} e(\mathrm{P}, \mathrm{V})$. If it is true, output accept, else output reject.

(7) Third-party authentication (Verify): This algorithm is implemented after the respective keystone is released. The role is that any third party can confirm who is the signer by this algorithm except the both parties involved in this scheme，so that binding signature to signer's identity is realized. The input of this algorithm is $(\sigma, \mathrm{Q}_A, \mathrm{Q}_B, \mathrm{m})$, and σ may be the signcryption of A or the signcryption of B. If $R_B = H(\mathrm{k}_A)$, and $\mathrm{Averify}(\sigma, \mathrm{Q}_A, \mathrm{Q}_B, \mathrm{m}) = \mathrm{accept}$, then σ is the signcryption of A; If $R_A = H(\mathrm{k}_A) + \mathrm{H}(\mathrm{k}_B)$, and Averify $(\sigma, \mathrm{Q}_A, \mathrm{Q}_B, \mathrm{m}) = \mathrm{accept}$, then σ is the signcryption of B.

4 The Authentication Process of Tor Node

The directory server in the Tor network is used as a third-party authentication authority. If OR node $R_i = (1, 2, 3 \ldots \mathrm{n})$ wants to join the Tor network, which must register with the directory server. The directory server then checks the performance of the OR, including uptime, bandwidth, and so on. OR needs to be authenticated to confirm its reliability when it joins the Tor network. In order to prevent the information from being falsified and tampered with during the verification process, we combine identity-based signcryption. Increased node reliability through directory server and node, node-to-node authentication mechanisms. The authentication procedure is shown in Figure 5 and Figure 6:

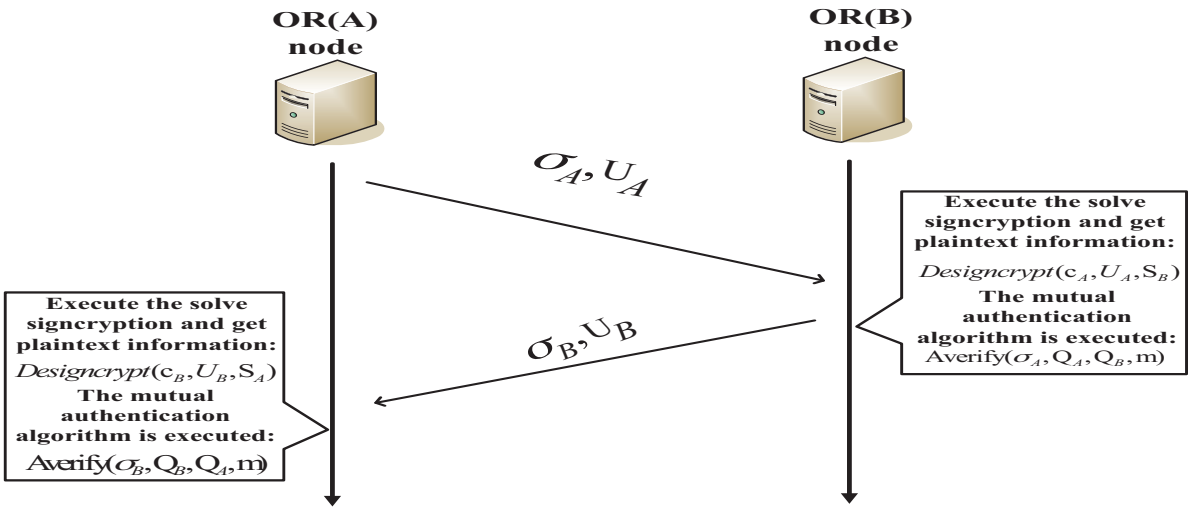

Figure 5. Mutual authentications between the Tor nodes

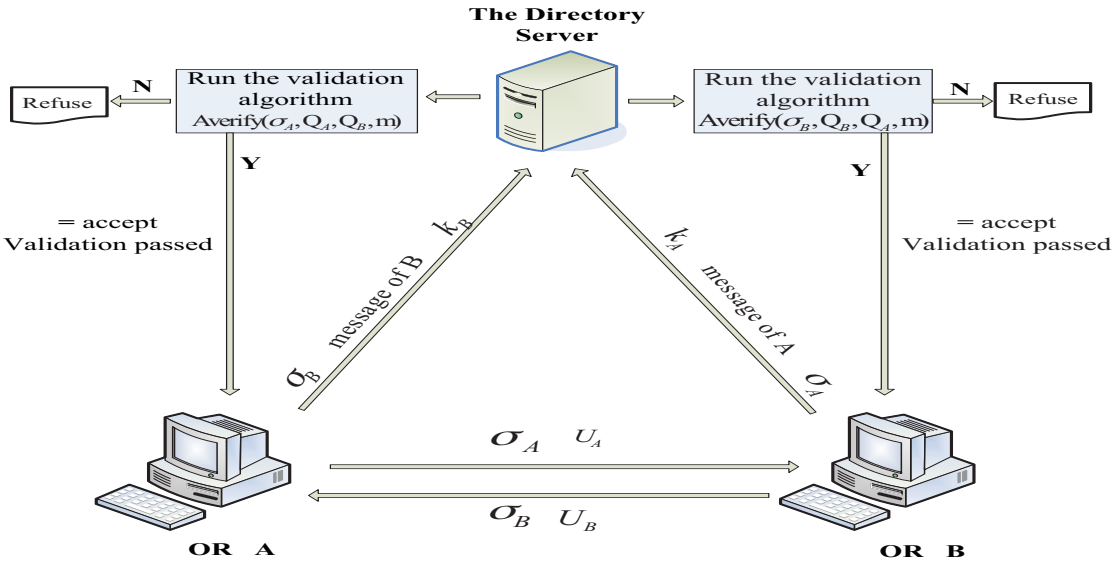

Figure 6. Third-party authentications for the Tor directory server

Specific operations are as follows:

The parameters in Fig. 6 are explained as follows: σ_A is the signcryption generated by the OR (A) node; σ_B is the signcryption generated by the OR (B) node; U_A is the secret key to decode the signcryption from node of OR (A); similarly, U_B with OR (B) node; k_A is the keystone of OR (A); k_B is the keystone of OR (B).

(1) In the Tor network node A wants to communicate with node B, node A can generates a set of information m containing time parameters and partial identity information. Node A will randomly select $k_A \in Z_q^*$, k_A will be used as its keystone.

if node A send message to node B
then
do $f_A = Footpr\text{int}(0, k_A)$
$\sigma_A = signcrypt(ID_A, ID_B, Q_A, Q_B, S_A, f_A, m)$
node A send σ_A, U_A to node B
else
$\sigma_A = signcrypt(ID_A, ID_B, Q_A, Q_B, S_A, f_A, m)$ Will not be performed
node A don't send any message to node B
end if

(2) After node A sends message to node B:
if node B received node A message
then
do $m = Designcrypt(c_A, U_A, S_B)$, and do $Averify(\sigma_A, Q_A, Q_B, m)$
if $Averify(\sigma_A, Q_A, Q_B, m) = reject$
then
The identity of the node A will be questioned by node B
node B not to provide any services to node A
else
if $Averify(\sigma_A, Q_A, Q_B, m) = accept$
then
continue with step 5
end if
else
node B continue to wait node A message

(3) In the Tor network node B wants to communicate with node A, node B can generates a set of information m containing time parameters and partial identity information. Node B will randomly select $k_B \in Z_q^*$, k_B will be used as its keystone.

if node B send message to node A
then
do $f_B = Footpr\text{int}(f_A, k_B)$
$\sigma_B = signcrypt(ID_B, ID_A, Q_B, Q_A, S_B, f_B, m)$
node B send σ_B, U_B to node A
else
$\sigma_B = signcrypt(ID_B, ID_A, Q_B, Q_A, S_B, f_B, m)$ Will not be performed
node B don't send any message to node A
end if

(4)After node B sends message to node A:
if node A received node B message
then
do $m = Designcrypt(c_B, U_B, S_A)$, and do $Averify(\sigma_B, Q_B, Q_A, m)$
if $Averify(\sigma_B, Q_B, Q_A, m) = reject$
then

The identity of the node B will be questioned by node A
node A not to provide any services to node B
else
if $Averify(\sigma_B, Q_B, Q_A, m) = accept$
then
continue with step 6
end if
else
node A continue to wait node B message

If it has already been validated that what B print is indeed the signcryption of A, then the verification from B to A will pass.

(5) After node B verifies the node A:

if $Averify(\sigma_A, Q_A, Q_B, m) = accept$
then
node B send node A's message m、signcrypt σ_A and node A's keystone k_A to The DirecTory Server
do $Verify(\sigma_A, Q_A, Q_B, m)$
if $R_B = H(k_A)$
then
node B to node A's verify will be passed
else
The identity of the node A will be questioned by node B
else
The identity of the node A will be questioned by node B

(6) Similarly, after node A verifies the node B:

if $Averify(\sigma_B, Q_B, Q_A, m) = accept$
then
node A send node B's message m、signcrypt σ_B and node A's keystone k_B to The DirecTory Server
do $Verify(\sigma_B, Q_B, Q_A, m)$
if $R_A = H(k_B)$
then
The verification from node A to node B will be passed.
else
The identity of the node B will be questioned by node A
else
The identity of the node B will be questioned by node A

5 Proof of Signcryption-based Concurrent Signature Scheme

5.1 Proof of Correctness

(1) Proof of Decryption:

$$c_i \oplus e(U_i, S_j) = c_i \oplus e(a_i P, sQ_j) = c_i \oplus e(sP, Q_j)^{a_i} = c_i \oplus e(P_{pub}, Q_j)^{a_i} = m$$

As shown in the above equation, as long as the encryption algorithm is executed correctly, the decryption algorithm solves the encrypted plaintext in the encryption algorithm.

(2) OR nodes are mutually authenticated

The OR (B) node verifies the signature of the OR (A) node:

$$e(P_{pub}R_A, h_A Q_A + R_B + h_B Q_B) = e(P_{pub}, rQ_A + h_A Q_A)$$
$$= e(sP, (r + h_A)Q_A) = e(P, (r + h_A)S_A) = e(P, V)$$

The OR (A) node verifies the signature of the OR (B) node:

$$e(P_{pub}R_A, h_A Q_A + R_B + h_B Q_B) = e(P_{pub}, rQ_B + h_B Q_B)$$
$$= e(sP, (r + h_B)Q_B) = e(P, (r + h_B)S_B) = e(P, V)$$

As shown in the above equation, If the OR (A), OR (B) node's signcryption algorithm is executed correctly, then the Averify algorithm will return the accept result.

5.2 Proof of Safety

Through the Tor anonymous network mechanism combined with the identity-based concurrent signature technology, the middle volunteer node join the Tor network must to authenticate the identity. The OR (A) node sends the attribute information to the OR (B) node, include σ_A, U_A. Between the OR (A) and OR (B) nodes, if there is an attacker Bob, Bob must be informed $e(\mathrm{P}_{pub}, Q_B)^{a_A} = e(\mathrm{s}\, a_A \mathrm{P}, Q_B)$ about identity information. Since Bob cannot know the private key S_B of OR (B), Bob must have the contents of $\mathrm{s}\, a_A \mathrm{P}$. Because $\mathrm{s}, a \in Z_q^*$, given $(\mathrm{P}, \mathrm{sP}, \mathrm{a}_A\mathrm{P}) \in G_1^3$, so to calculate A is obviously B's CDH problem. The middle attacker Bob can't decrypt the content (σ_A and U_A) transferred from node OR(A) to node OR(B), even though he intercepts it. So the mutual authentication process between OR nodes is confidential.

5.3 Proof of Anti-collusion

In the entire Tor network, the directory server managers all OR nodes added and exit. The entire Tor network is also composed of many relay nodes. As shown in Figure 7.

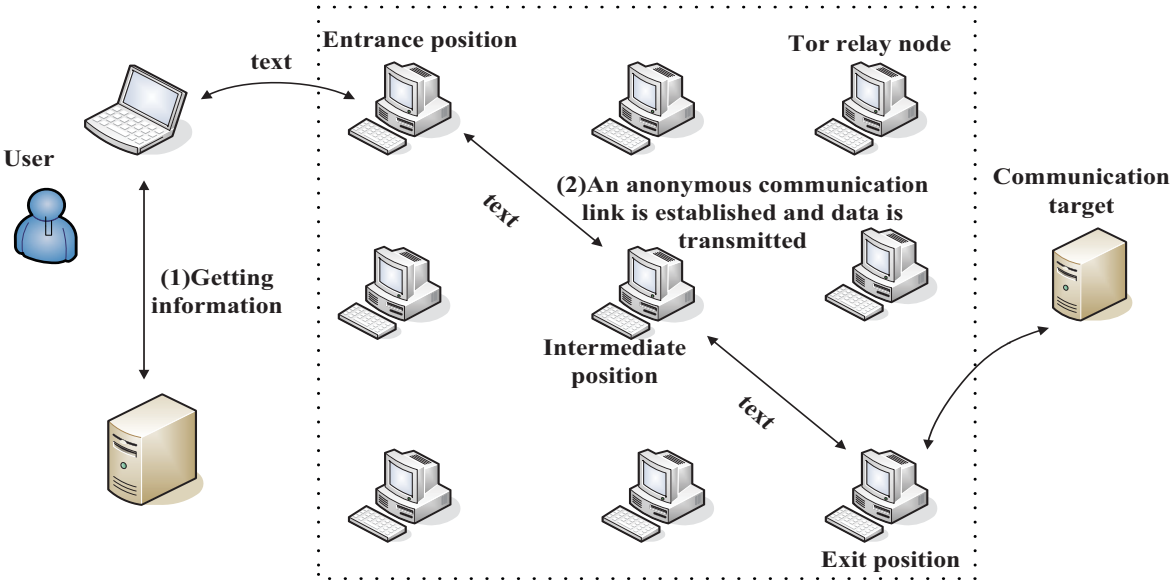

Figure 7. Tor network structure diagram

As shown in Figure 7above, in the first phase of the entire Tor network (The joining phase of the volunteer node). Through this program can guarantee that each legitimate relay node to join Tor to form the entire Tor network system，When the user wants to communicate with the target node，users are required to apply to the Tor directory server. The Tor directory server randomly selects the Tor relay node among the valid nodes that pass the audit and thereby establishing a communication link. The number of relay nodes in a Tor network is very large and the relay nodes before and after the link are randomly selected by the directory server. There is no interaction between the relay nodes during the link establishment phase. When the link is established, at the time of information transmission, the transmission of the message packets takes place in one direction. The message packet passes through each onion routing node, it will decrypt a layer to get to the next hop address, and the onion packet is then passed to the next node. Do not know the entire onion router between the senders to the receiver to go through. When this transmission is complete the link will be destroyed, so as to avoid the DDos attack initiated by OR nodes in the process of information transmission.

In the Tor network，in the volunteer node to join the stage if not legitimate certification，most illegitimate volunteer nodes can be disguised as legitimate nodes to join the Tor network. If the number of these illegal relay nodes occupies more than half of the legitimate relay nodes, when the link is established, the false probability of a randomly selected relay node of the directory server is extremely large. Multiple false nodes in the current link collusion before data communication get the private key of other false nodes that Decrypt the layer of their own. At the time of data transmission, a fake node obtains the onion packet because it owns the private key of a number of other false nodes; this onion packet security will have a great threat. The scheme adopted in this paper can authenticate the identities of the relay nodes that join the Tor network. Greatly reducing the possibility of false nodes joining the Tor network, which greatly reduces the DDos behavior that launched by the false nodes in the link.

5.4 Proof of High Efficiency

The same signcryption technology is used to prevents the insertion of malicious managed nodes into the Tor network，The literature [4] is aimed at the user in the establishment of the anonymous link stage. The reliability, stability and reliability of the intermediate nodes of the Tor network are evaluated by a trusted third-party authority. Issue a reliable certificate for the Verifies OR node which has passed and generates a public-private pair for the OR node. By verifying each other's trusted certificate to determine each other's identity and a signcryption technique is used to ensure the unforgeability of the trusted certificate. The concurrent signature scheme in this paper does not require the third authority to issue trusted certificate, therefore part of the time overhead of issuing trusted certificate can be saved.

In this paper, the signature scheme based on signcryption technique is less expensive than the scheme in [4]. The specific operation situation is shown in Table 1.

Table 1. The scheme of this paper is compared with the literature

Operation	The program of literature		The program of this paper	
	Signcryption	Decryption	Signcryption	Decryption
Exponent	0	0	1	0
Bilinear pairings	1	1	1	1
Hash	2	2	2	0
multiplication	3	1	3	0

From Table 1 can be drawn the scheme adopted in this paper has different degree of reduction in the decryption link Hash, multiplication operation. In addition, the scheme adopted in this paper is carried out in the OR node adding to the Tor network phase, without involving other time cost. The scheme adopted in [4] is carried out in the link establishment phase of the Tor network, which involves the fusion of original keys with OR nodes. So the scheme of this paper is more efficient than the literature [4].

6 Conclusion

The present concurrent signature scheme does not need a third authority party to issue a trusted certificate, save the overhead. Compared with [4], the present program has different degrees of reduction in decryption link Hash and multiplication operation. Proved from security, unforgeable and collusion-resistance, the present concurrent signature scheme has the advantage of safe, unforgeable and collusion-resistance. So the scheme can effectively resist the collusion DDos attack behavior of attacker from the very great degree in the Tor network insert controlled node.

Acknowledgement

This work is supported by the National Nature Science Foundation of China (No.61462060).

References

1. Yanwei Zhou,Qiliang Yang,Bo Yang,et.al. A Tor Anonymous Communication System with Security Enhancements[J]. Journal of Computer Research and Development, 2014,51(7):1538-1546(in Chinese).
2. Bo Wang. How to use Tor network safely [J]. Computer and Network, 2015, 41(14):37-37 (in Chinese).
3. Yong Zhou. Research on Anonymous Communication Based on TOR[D].Xidian University, 2013 (in Chinese).
4. Guang Zheng. Security analysis and research of Tor anonymous communication system.Shanghai Jiao Tong University, 2011 (in Chinese).
5. Kui Liu, Xiangqian Liang, Xiaolin Li. Concurrent signature scheme constructed by identity-based ring signcryption[J]. Journal of Computer Applications, 2013, 33(5):1386-1390 (in Chinese).
6. Chengqiang Huang. Research on Tor-based Construction Technique for Backward Anonymous Channels[D]. Xidian University, 2014 (in Chinese).
7. Xin Liu, Neng Wang. Research on Anonymous Communication Based on Tor Network[D], Shanghai: East China Normal University, 2011 (in Chinese).
8. Yue Han, Tianbo Lu. Tor Hidden Service Scaling[J]. software, 2016(2) (in Chinese).
9. Qi Liang, Shiti Li. Design Principles and Implementation Analysis of Anonymous network Tor[J]. Information & Communications, 2016(3):130-131 (in Chinese).
10. Ling Z, Luo J, Yu W, et al. 1 Extensive Analysis and Large-Scale Empirical Evaluation of Tor Bridge Discovery[J]. Proceedings-IEEE INFOCOM, 2015, 26(7):2381-2389.
11. Eltschinger S, Loewith R. TOR Complexes and the Maintenance of Cellular Homeostasis.[J]. Trends in Cell Biology, 2016, 26(2):148-159.
12. Sankar S, Chopra R, Xu W, et al. Identification of Lkb1 Mutation as a Predictive Biomarker for Sensitivity to Tor Kinase Inhibitors: US, EP2992878[P]. 2016.