ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Key Management and Authentication in Cooperative Cache Using Wireless Networks

Latha .S [1,] Sridhar Raja .D[2]

Professor & Head, Department of Electronics & Instrumentation Engineering, Bharath University, Chennai, Tamil Nadu, India[1]

Assistant Professor, Department of Electronics & Instrumentation Engineering, Bharath University, Chennai, Tamil Nadu, India[2]

ABSTRACT: Cooperative caching, which allows the sharing and coordination of cached data among multiple nodes, can be used to improve the performance of data access in ad hoc networks. When caching is used, data from the server is replicated on the caching nodes. Since a mobile node may return the cached data, or modify the route and forward a request to a caching node, it is very important that the mobile nodes do not maliciously modify data , drop or forward the request to the wrong destination. In this paper, we identify possible security attacks on cache consistency and propose a randomized grouping based schemes for intrusion detection, damage recovery and intruder identification. We also address other security issues in cooperative cache based data access.

I. INTRODUCTION

Key management and authentication is important to security of Mobile Ad Hoc network (MANET). Based on the threshold cryptography, this paper introduces mobile agents to exchange private key and network topological information with nodes in the network. This method avoids a centralized certification authority to distribute the public keys and the certificates, thus enhances security. Carrying private key and some state variables, mobile agents navigate in the network according to the visit balance policy, i.e. the node with the least visits would be first visited by the mobile agent. Any node in the network can cooperate to perform an authentication upon a new node wanting to join the network. Experimental results show that the mobile agent performs very well for improving the success ratio of authentication and enhance security while reducing the communication overhead and resource consumption.

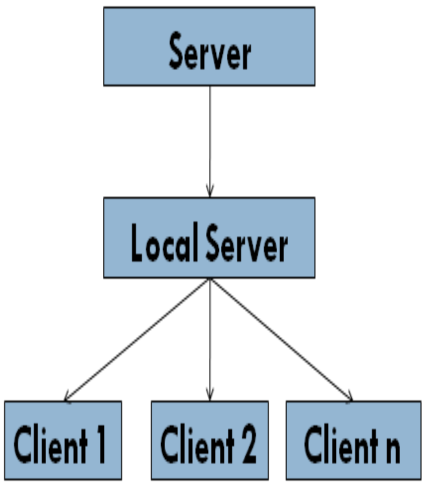

Fig 1.1 Client Server Model

1.1 Detailed Description

Cooperative caching, which allows the sharing and coordination of cached data among multiple nodes, can be used to improve the performance of data access in ad hoc networks [1]. When caching is used, data from the server is replicated on the caching nodes. Since a mobile node may return the cached data, or modify the route and forward a request to a caching node, it is very important that the mobile nodes do not maliciously modify data, drop or forward the request to the wrong destination. In this paper, we identify possible security attacks on cache consistency and propose a randomized grouping based schemes for intrusion detection, damage recovery and intruder identification [2]. We also address other security issues in cooperative cache based data access

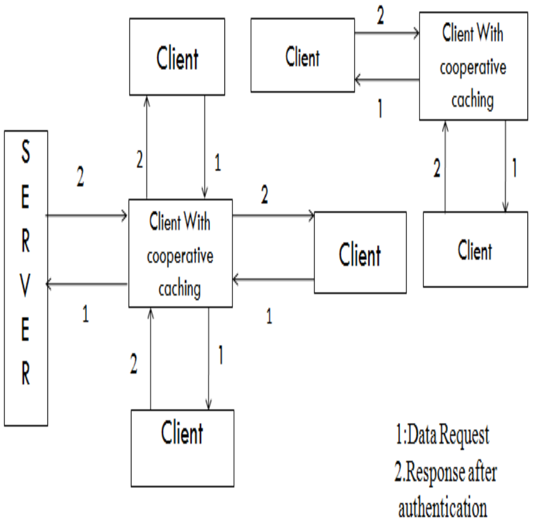
S E R V E R
Client
Client
Client With cooperative caching
Client With cooperative caching
Client
Client
Client
1
2
1:Data Request
2.Response after authentication

Fig 1.2 Systems with Cooperative Cache

Mobile ad hoc networks have been the focus of recent research due to their potential applications in civilian and military environments such as battlefield, disaster recovery, group conference, and wireless office [3]. In ad hoc networks, mobile nodes communicate with each other using multi-hop wireless links. Due to lack of infrastructure support, each node acts as a router, forwarding data packets for other nodes. Although routing is an important issue in ad hoc networks, other issue such as information (data) access is also very important since the ultimate goal of using ad hoc networks is to provide information access to mobile nodes [4].

Caching frequently accessed data items at the client side is an effective technique to improve performance in mobile environments [5]. With caching, the data access delay is reduced since some data access requests can be served from the local cache, thereby obviating the need for data transmission over the scarce wireless links. However, caching techniques used in single-hop mobile environment (i.e., cellular networks) may not be applicable to multi-hop mobile environments since the data or request may need to go through multiple hops. As mobile nodes in ad hoc networks may have similar tasks and share common interests, cooperative caching, which allows the sharing and coordination of cached data among multiple nodes, can be used to reduce the bandwidth and power consumption. For example, in a battlefield, an ad hoc network may consist of several commanders and groups of soldiers [6]. The commander has the data center, and the solders need to access the data center to get information about the enemy, the battlefield, and the attack plans. After a soldier obtained the attack information from the data center, it is very likely that nearby soldiers also need the same information from the data center. Bandwidth and power can be saved if these data accesses are served by the soldier with the cached data instead of the data center, which may be far away. Certainly, this cache sharing requires return the cached data, or modify the route and forward a request to a caching node, it is very important that mobile nodes do not maliciously modify data, drop or forward the request to the wrong destination. The proposed research will study methods to avoid or detect such malicious nodes via authentication mechanisms [7]. One common approach for data authentication is based on digital signature. With this approach, the data source can sign the data with its private key, so that intermediate routers cannot modify the data. However, the digital signature approach

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

has high overhead, both in terms of time to sign and verify, and in terms of bandwidth. In this research, we design and evaluate techniques to reduce such overhead and balance system performance and security strength. Further, we identify possible security attacks on cache consistency and propose viable mechanisms to defend against such attacks. The paper is organized as follows. In the next section, we present our cooperative cache based data access schemes. In next section, we propose techniques to deal with cache consistency attacks and other security issues in cooperative cache based data access.

1.2. Wireless Networks

Wireless network is any type of computer network and is commonly associated with a communications network whose interconnection between nodes is implemented without the use of wires [9]. Wireless networks are generally implemented with some type of remote information transmission system that uses electromagnetic waves, such as radio waves. It has been the focus of recent research due to their potential applications in civilian and military environments such as battlefield, disaster recovery, group conference, and wireless office. In wireless network, the nodes communicate with each other using multi-hop wireless links. Due to lack of infrastructure support, each node acts as a router, forwarding data packets for other nodes [8].

Most of the previous research in wireless networks focuses on the development of dynamic routing protocols that can efficiently find routes between two communicating nodes. Although send data is an important issue in wireless networks, other issue such as information (data) access is also very important since the ultimate goal of using wireless networks is to provide information access to network nodes.

1.3 Wireless Security

Wireless networks are forcing to completely rethink how they secure their networks and devices to prevent attacks and misuse that expose critical assets and confidential data. By their very nature, wireless networks are difficult to roll out, secure and manage [10].
Wireless networks offer great potential for exploitation for two reasons; they use the airwaves for communication, and wireless-enabled laptops are ubiquitous. To make the most of their security planning, enterprises need to focus on threats that pose the greatest risk. Wireless networks are vulnerable in a many of ways, some of the most likely problems being malicious hacking attempts and denial-of-service (DoS) attacks are certainly possible as well.
In our project the security algorithm is going to be implemented in order to avoid the security related problems in the cooperative caching area. For example a non-trusted person can enter and access the data in a wireless enabled area. It causes the data hacking and unwanted data outlet. So a valid key generation scheme gives an acknowledgement of trusted user to access the data from Server/Cooperative caching.

II. COOPERATIVE CACHE BASED ON DATA ACCESS

The idea of cooperative caching can be explained by Figure 2.1, which shows part of an ad hoc network. Some nodes in the ad hoc network may have wireless interfaces to connect to the wireless infrastructure such as wireless LAN or cellular networks. Suppose node N11 is a data source (center), which

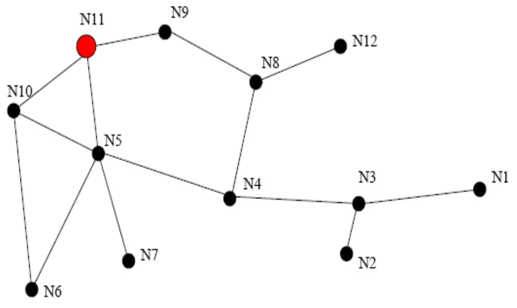

Fig. 1. An ad hoc network

Fig 2.1 An Ad hoc Network

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

networks. Suppose node N11 is a data source (center), which contains a database of n items d1; d2; :::; dn. Note that N11 may be a connecting node to the wired network which has the database. In ad hoc networks, a data request is forwarded hop-by- hop until it reaches the data center and then the data center sends the requested data back. Various routing algorithms have been designed to route messages in ad hoc networks. To reduce the bandwidth consumption and the query delay, the number of hops between the data center and the requester should be as small as possible.

2.1. Cache on data path (Cache Path)
The idea of Cache Path can be explained using Figure 2.1. Suppose node N1 has requested a data item di from N11. When N3 forwards the data $_{di}$ back to N1, N3 knows that N1 has a copy of $_{di}$. Later, if N2 requests $_{di}$, N3 knows that the data source N11 is three hops away whereas N1 is only one hop away. Thus, N3 forwards the request to N1 instead of N4. Note that many routing algorithms provide the hop count information between the source and destination. By caching the data path for each data item, bandwidth and power can be reduced since the data can be obtained through less number of hops. However, recording the map between data items and caching nodes increases routing overhead. In the following, we propose various optimization techniques to improve the performance of Cache Path.

In Cache Path, a node does not need to record the path information of all passing data. For example, when $_{di}$ is forwarded from N11 to the destination node N1 along the path N5□N4□N3, N4 and N5 will not cache the path information of di since N4 and N5 are closer to the data source than the caching node N1. Thus, a router node only records the data path when it is closer to the caching node than the data source. Due to mobility, the node which caches the data may move. The cached data may be replaced due to cache size limitation. As a result, the node which modified the route should reroute the request to the original data source after it finds out the problem. Thus, the cached path may not be reliable and using it may adversely increase the overhead [11].

2.2. Cache the data (Cache Data)
In the Cache Data approach, the router node caches the data instead of the path when it finds that the data is frequently accessed. For example, in Figure 2.1, if both N6 and N7 request $_{di}$ through N5, N5 may think that $_{di}$ is a popular data and cache it locally. Future requests by N4 can be served by N5 directly. Since the Cache Data approach needs extra space to save the data, it should be used prudently. Suppose the data source receives several requests for forwarded by N3. The nodes along the path N_3 □ N4 □ N5 may found that di is a popular item and should be cached. However, it will waste a large amount of cache space if three of them all cache d_i. To avoid this to happen, another rule is enforced: a node does not cache the data if all requests for the data are from the same node. In the previous example, all requests received by N5 are from N4, which in turn are from N3. With the new rule, N4
and N5 will not cache . If the requests received by N3 are from different nodes such as N1 and N2, N3 will cache the data. If the requests all come from N1, N3 will not cache the data, but N1 will cache it [12]. Certainly, if N5 receives requests for di from N6 and N7 later, it may also cache $_{di}$.

2.3. The hybrid cache approach

Cache Path and Cache Data can significantly improve the system performance. Our analysis showed that Cache Path performs better in some situations such as small cache size or low data update rate, while Cache Data performs better in other situations. To further improve the performance, we propose a hybrid scheme called Hybrid Cache to take advantage of Cache Data and Cache Path while avoiding their weakness.
Specifically, when a mobile node forwards a data item, it caches the data or path based on some criteria. These criteria include the data item size and the number of hops that can be saved due to using Cache Path. Detailed information about setting these parameters can be found. Next, we present solutions to defend against attacks on cache consistency, and then discuss some other security issues.

III. DEFENDING AGAINST ATTACKS ON CACHE

In this section, we identify cache consistency attacks and propose solutions to deal with such attacks:
3.1. Cache consistency
When cache is used, cache consistency issues must be addressed to ensure that clients see only valid states of the data or at least do not unknowingly access data that is stale according to the rules of the consistency model. Problems related

to cache consistency have been studied in many other systems such as multi-processor architectures, distributed file systems, distributed shared memory, and client-server database systems. Two widely used cache consistency models are the weak consistency model and the strong consistency model. In the weak consistency model, a stale data might be returned to the client [1,3]. In the strong consistency model, after a write completes, no stale copy of the modified data will be returned to the client. The exact definition of the completion of a write varies by the consistency approaches. The commonly used weak consistency mechanism is TTL-based (Time-To- Live), in which a client considers a cached copy up-to-date if its TTL has not expired. For strong cache consistency, invalidation-based and polling-based approaches are used. In the invalidation-based approach, the server keeps track of the clients that cache the data item, and sends invalidation messages to the clients when the data is changed. In the polling based approach, every time the user requests a data item and there is a cached copy, the cache first contacts the server to validate the cached copy, and then returns the copy to the user. Since Polling-based approach may generate significant network traffic, the TTL-based approach is widely used for the weak cache consistency model and the invalidation-based approach is used for the strong cache consistency model.

3.2. Defend against attacks on cache consistency

There are two kinds of attacks on the invalidation-based approach. First, the malicious node may stop propagating the invalidation message, and mobile nodes far away from the data source may not be able to receive the invalidation message and may use the stale cache without realizing it. Second, a malicious node may send invalidation to some nodes and invalidate their caches which are still valid. If the data source signs the invalidation messages with a digital signature, the receiver can authenticate the message and avoid the second kind of attack [6]. Note that a digital signature cannot be used to defend against the first kind of attack, since the mobile node may never receive the signed invalidation message. As for TTL-based approach, since the TTL and the cached data can be protected by digital signature, it does not suffer from this kind of attack. However, TTL-based approach can only provide weak consistency, and it can only be used for data which are rarely updated. In some strategic scenarios such as in the battlefield, accessing stale data (e.g., outdated enemy information) may be life threatening, and hence the strong consistency model should be adopted. In the following, we focus on dealing with attacks to the invalidation-based approach [8].

To prevent malicious nodes from dropping the invalidation messages, we borrow ideas from the IR-based cache invalidation. In this approach, the server periodically broadcasts an invalidation report (IR) in which the changed data items are indicated. The IR consists of the current timestamp Ti and a list of tuples (dx; tx) such that tx > (Ti w _ L), where dx is the data item id, tx is the most recent update timestamp of dx, and w is the invalidation broadcast window size. In other words, IR contains the update history of the past w broadcast intervals. Based on the value of w, clients can still validate their local cache even after missing w 1 IRs. Similar to the invalidation-based approach, clients use the IR to invalidate their local cache. Different from invalidation based approach, the IR is sent out regularly, and the clients expect the IR at regular time interval. Therefore, if a client maliciously drops an IR, the nodes that are expecting the IR can detect it, and some measures can be taken to address the intrusion. Although flooding can be used to distribute IR to the mobile nodes, due to the high overhead, we assume that a multicast tree, with the server as the root, is used to distribute the IRs to the mobile nodes.

3. Reducing the authentication overhead

To authenticate the data source, digital signatures can be used. However, this approach has high overhead both in terms of computation and bandwidth. As shown in , asymmetric cryptographic operations (such as digital signatures) are three or four orders of magnitude slower than symmetric cryptographic operations (such as hash functions), and a 1024-bit RSA digital signature is roughly equivalent to the use of a 72- bit key in a symmetric encryption algorithm. To address this issue, researchers apply symmetric cryptographic techniques such as TESLA to secure broadcast multicast, and routing in ad hoc networks. TESLA provides source authentication with message authentication codes (MACs) using only symmetric cryptography, based on delayed disclosure of keys by the sender. Suppose the receiver can loosely synchronize with the sender. Each packet is attached with a MAC computed using a key known only to the sender at that time. A short while later, the sender discloses the key and the receiver is able to authenticate the packet. If the packet arrives at the receiver after the key has been disclosed, it is discarded [10].

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

When applying TESLA to a large ad hoc network, different nodes may experience different amount of delay. For example, nodes closer to the data source receive the IR earlier than nodes far away from the data source. As a result, how long should the data source wait before disclosing the key? Waiting until the furthest node receiving the IR may take too long. The authors of TESLA also proposed enhanced approaches to allow immediate authentication at the client side, but this approach requires the server to delay sending the IR at the sender, which also has large overhead [11]. Further, TESLA has synchronization requirements for all nodes in the network. Next, we present a novel solution to reduce the authentication overhead by randomized grouping.

3.3.1. The basic idea

To prevent intruders from modifying the invalidation messages, we enhance the IR-based approach with techniques for damage recovery, intrusion detection and isolation, based on the idea of randomized grouping as follows. The nodes (IR receivers) are randomly distributed into multiple groups, and the data center (server) shares a unique group key with the receivers of each group. The IR is protected by several MACs each with one group key. Since each receiver only knows one of the keys, if an intruder modifies the IR and the MAC using the group key it knows, the modification can be detected by a descendant in a different receiver group [12].

For example, in the IR multicast tree, N0 is the server and two group keys K0 and K1 are used. N0 appends two MACs to the IR < IR;MAC0;MAC1 >, 4 and sends it to N1, which forwards to N2 and N3. Suppose N1 and N3 know K1 while N2 knows K0. Then, malicious modifications can be easily identified. For example, if N2 is a malicious node and it modifies the IR, N3 will be able to detect this modification by verifying MAC1. After N3 reports this modification to the server, the server can easily find that N2 is the malicious node.

In the ideal case where neighbor nodes use different keys, the malicious node can be easily isolated. However, if two neighbor nodes use the same group key, further isolation may be difficult. For example, if both N1 and N2 both use K0, it will be difficult for the server to find out who is the malicious node. When this happens, a tree reconfiguration may be necessary.

3.3.2. Intruder, Identification, Isolation and Recovery

When a node detects an intrusion, the detector should report the received IR (attached with the MACs) and the id of its parent, say $_{Ni}$, directly to the server. It is possible that this message has to go through the malicious node, which may modify the report, and hence it should be signed by its group key. Certainly, the malicious node may still drop the report. In this case, the detector has to find other routes. The detector should still try until it receives the correct IR from the server. If going through the malicious node is the only way to reach the server, it will be similar to a network partition due to the malicious node, and the node must be aware that the cached data may be stale.

Receiving such a report, the server should ask $_{Ni}$ to report the received IR and its parent id. This recursive process continues until the server has enough information to find out the compromised group keys, identify and isolate the intruder. A group key Kk is compromised if there exists a compromised IR in which MACk matches the compromised IR, where MACk is the MAC of IR using Kk. Based on the knowledge of the compromised group keys, the server checks the reports collected through a serious of nodes along a path, to identify the intruder or suspected intruders among these nodes. Suppose Kj is compromised and $_{Ni}$ knows Kj . If its direct child uses different key and detects the modification, $_{Ni}$ is the malicious node unless its direct parent has the same group key. In which case, both will be suspects.

After finding the intruder, the server broadcasts this information to all nodes, which may reconfigure their multicast tree to remove this intruder from the tree. If several nodes are possible suspects and there is no way to isolate them, the server should remember their ids and inform them to reconfigure their multicast tree or assign new group keys to some of them so that two neighbor nodes can use different keys. Then, the server reconstructs and retransmits the IR so that no node will use the modified IR. Certainly, this new IR should not be sent to nodes that already received it correctly; i.e., the nodes that are parents of the malicious node.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

IV. CONCLUSION

Other security issues

When caching is used, data from the server is replicated in the caching nodes. One of the major concerns is the duplication of sensitive data, which the owner might want to restrict access. One solution is to define different levels of security for the data, with regard to duplication and storage in node caches. The data server can specify the level of security for each data item. Based on the security level, some data may not be cached, or only cached by a limited number of nodes. In the case of most sensitive data, the data server only sends the encrypted version to a number of nodes. Those trusted nodes will be able to get a shared key from the data server to decrypt the data. Future research should focus on designing mechanisms that would provide the owner a handle to control the scope of caching, and yet would not undermine the flexibility of caching. Note that there is a balance between security strength and system performance. By encrypting or limiting the distribution of the most sensitive data, some benefits of the cooperative caching will be lost. With cooperative caching, the mobile nodes may return the cached data or modify the route and forward the request to the caching node, and hence, it is very important that the mobile nodes do not maliciously modify the data. One commonly used solution is data authentication which allows a receiver to ensure that the received data is authentic (i.e., it originates from the source and was not modified on the way), even when none of the other receivers of the data is trusted. To authenticate the data source, appending each packet with a message authentication code that is calculated using a shared key does not work since any receiver that has the shared key can forge the data

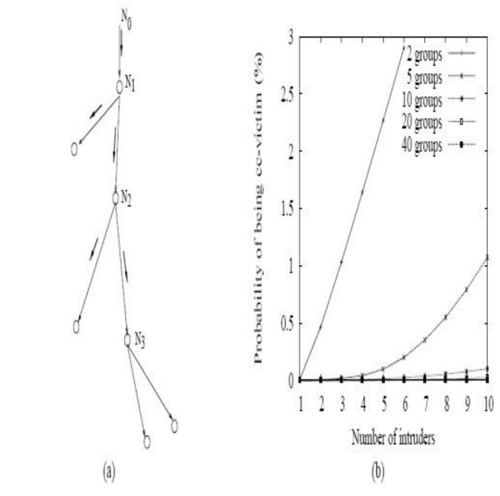

Fig. 2. Reducing the authentication overhead with randomized grouping

Fig 4.1 Reducing the authentication overhead with randomized grouping

Consequently, it is natural to look for solutions based on asymmetric cryptography, namely digital signature schemes. After the data source signs the data with its private key, mobile nodes can verify the integrity of the data by using the public key of the data source. However, the digital signature approach has high overhead, both in terms of time to sign and verify,

and in terms of bandwidth. Thus, we focus on reducing such authentication overhead. For example, if the data has gone through nodes with good reputation, the receiver may not need to verify the signature. These will tradeoff some security strength for system performance. Periodically, the receiver may want to verify the signature, and change the credit rating of other nodes based on the verification results. A mobile node has the option of verifying the signature if the data is very important or if it has enough computation power. This provides the user an option to choose the proper tradeoffs between security and performance; i.e., a mobile node can tradeoff security strength for performance if the data is not very important and it has less computation power or tradeoff performance for security strength otherwise.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

REFERENCES

1. Huges and M.A Austin, "AUTHENTICATION IN ADHOC NETWORKS", Communication of the ACM VOL-42, 1999, pp.41-46.
2. Das M.P., Jeyanthi Rebecca L., Sharmila S., "Evaluation of antibacterial and antifungal efficacy of Wedelia chinensis leaf extracts", Journal of Chemical and Pharmaceutical Research, ISSN : 0975 – 7384, 5(2) (2013) pp.265-269.
3. R.N KEEPLER,"RECOGNITION MEMORY FOR WORDS, SENTENCES AND PICTURES", journal of verbal learning and verbal behavior, VOL-6, 1967, pp 156-163.
4. Subhashini V., Ponnusamy S., Muthamizhchelvan C., "Growth and characterization of novel organic optical crystal: Anilinium d-tartrate (ADT)", Spectrochimica Acta - Part A: Molecular and Biomolecular Spectroscopy, ISSN : 1386-1425, 87() (2012) pp.265-272.
5. F.Hartung, Begird "WIRELESS NETWORKS AND KEY MANAGEMENT". Signal processing and access control for multimedia services VOL-66(3), 1998, pp 283-301.
6. Thomas J., Ragavi B.S., Raneesha P.K., Ahmed N.A., Cynthia S., Manoharan D., Manoharan R., "Hallermann-Streiff syndrome", Indian Journal of Dermatology, ISSN : 0019-5154, 58(5) (2013) pp.383-384.
7. G.C Langelaar and R.L Lagendijk "", VOL 10(1), 2001,pp 148-158.
8. Subha Palaneeswari M., Ganesh M., Karthikeyan T., Manjula Devi A.J., Mythili S.V., "Hepcidin-minireview", Journal of Clinical and Diagnostic Research, ISSN : 0973 - 709X, 7(8) (2013) pp.1767-1771.
9. Johnson, Neil. F, Sushiljajodia "Encryption using wireless networks" IEEE computer, feb.1998, pp 26-34.
10. Laljee R.P., Muddaiah S., Salagundi B., Cariappa P.M., Indra A.S., Sanjay V., Ramanathan A., "Interferon stimulated gene - ISG15 is a potential diagnostic biomarker in oral squamous cell carcinomas", Asian Pacific Journal of Cancer Prevention, ISSN : 1513-7368, 14(2) (2013) pp.1147-1150.
11. Danning Dorothy E-information warfare and security boston, MA:ACM press,1999,pp 310-313.
12. Pandimadevi V., Kujambal A., Sudhakar S., Ramaiah B., Gupta S., Srinivasan K.S.V., "Upgrading low-grade grain leather using high performance polymeric dispersions by a transfer coating system", Journal of the American Leather Chemists Association, ISSN : 0002-9726, 96(5) (2001) pp.157-161.
13. T.Nalini ,A.Gayathri,HVS Based Enhanced Medical Image Fusion ,International Journal of Innovative Research in Computer and Communication Engineering,ISSN (Print) : 2320 – 9798 , pp 170-173, Vol. 1, Issue 2, April 2013.
14. S.Thirunavukkarasu, r.K.P.Kaliyamurthie ,EFFICIENT ALLOCATION OF DYNAMICRESOURCES IN A CLOUD,International Journal of Innovative Research in Computer and Communication Engineering, ISSN: 2249-2651, pp 24-29,Volume1 Issue3 Number2–Dec2011.
15. K.G.S. VENKATESAN,Planning in FARS by dynamic multipath Reconfiguration system failure recovery in Wireless Mesh Network,International Journal of Innovative Research in Computer and Communication Engineering,ISSN(Online): 2320-9801,pp 5304-5312,Vol. 2, Issue 8, August 2014.
16. G.Michael, An Empirical Approach – Distributed Mobility Management for Target Tracking in MANETs ,International Journal of Innovative Research in Computer and Communication Engineering , ISSN (Print) : 2320 – 9798 , pp 789-794 , Vol. 1, Issue 4, June 2013.
17. G.AYYAPAN , Malicious Packet Loss during Routing Misbehavior - Identification, International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801, pp 4610-4613 ,Vol. 2, Issue 6, June 2014.