ISSN 2313-1527 (PRINT)
ISSN 2313-1535 (ONLINE)

IJEIE

International Journal of Electronics
and Information Engineering

Vol. 1, No. 1 (Sept. 2014)

INTERNATIONAL JOURNAL OF ELECTRONICS & INFORMATION ENGINEERING

PUBLISHING OFFICE

Min-Shiang Hwang
Department of Computer Science & Information Engineering, Asia University, Taichung 41354, Taiwan, R.O.C.
Email: mshwang@asia.edu.tw

International Journal of Electronics and Information Engineering is published both in traditional paper form (ISSN 2313-1527) and in Internet (ISSN 2313-1535) at http://ijeie.jalaxy.com.tw

PUBLISHER: Candy C. H. Lin

© Jalaxy Technology Co., Ltd., Taiwan 2005
23-75, P.O. Box, Taichung, Taiwan 40199, R.O.C.

International Journal of Electronics and Information Engineering

Vol. 1, No. 1 (Sept. 1, 2014)

Cloud Computing Login Authentication Redesign

Eric Opoku Osei[1], James Benjamin Hayfron-Acquah[2],
(Corresponding author: Eric Opoku Osei)

Computer Science Department, Kwame Nkrumah University of Science and Technology[1]
Private Mail Bag, KNUST Kumasi

Computer Science Department, Kwame Nkrumah University of Science and Technology[2]
Private Mail Bag, KNUST Kumasi
(Email: eoosei@gmail.com)

(Received Dec. 23, 2013; revised and accepted Feb. 13, 2014)

Abstract

Trusted-security is a concern for cloud adoption by many enterprises. The work was to improve login security process at cloud's user-end using biometric to replace secret code generation as step-2 authentication. System development life cycle-revolutionary waterfall method was adapted for the design process and verified the redesign using Delphi technique with security experts. The work alerts authorised user to block domain intrusion on phone at real time as phone is permanently interlinked to access server. On experiment, authorised user must submit cloud username and password on computer workstation as step-1 key. On submission, a text message is generated to a predefined biometric-compatible phone for user to scan bio-feature. As backdoor, if predefined biometric phone is temporarily lost; open the alternative link sent to authorised user e-mail account to capture and resend the scanned bio-feature to server as step-2 key for access privilege. Results indicated rigidity to prevent hacking via user-workstation. We recommended multi-nodal bio-scanning to reduce authentication failure rate and argued conclusively that this work improves user-end login security for cloud clients.

Keywords: Access control; Biometric-phone; Cloud Computing; Security; 2-step Authentication;

1 Introduction

Cloud computing has come to stay. Cloud Security Alliance (2013) rated nine notorious cloud computing top threats. On account of severity; data breach came first on the rank [4]**.** Wentao Liu (2012) rated data privacy as the key security problem. The work further emphasized that a single security method cannot solve the cloud computing security problem and many traditional and new technologies and strategies must be used together for protecting the total cloud computing system [10]. Adam Maguire published a survey conducted amongst Chief Information Officers (CIO's) and I.T Directors during the 2010 pricewaterhouse Coopers forum (www.irishtimes.com, 2010) [11]. The publication provided feedback that clearly shows "Security" as the biggest concern for enterprises thinking about the cloud evolution. Simply put; Cloud Computing is the new evolution to expand ICT for development and virtual application subscription. The new idea is to leverage this virtual subscription concept into everyday enterprise business. In the late 1960's, the computer scientist John McCarthy once brought the concept of utility computing into the technology world, predicting that the life cycle of technology will not only stick as tangible products. Today, whatever the term, cloud computing, happens to do could be referenced to the primary research of the utility concept by John McCarthy [6].

Débora DG & Brunzel T, (2010) discussed in their thesis, the various services in the cloud. Service providers offer to corporate clients: *Infrastructure-as-a-Service (IaaS), Software-as-a-Service (SaaS), and Platform-as-a-Service (PaaS)*; all at pay-per-use and without initial Capital-intensive demand. Cloud offers better economic advantages over the traditional in-house-acquired IT infrastructure**.** However, the issue of security threatens small medium enterprises to leverage transactional and query traffics unto the cloud platform. Cloud computing is a flexible, cost-effective and proven delivery platform for providing business or Consumer IT services over the Internet [5]. Bill Claybrook in his contributory work in the SearchCloudComputing.com E-guide report explained that cloud resources can be rapidly deployed and easily scaled, with all processes, applications and services provisioned "on demand," regardless of user location. Improving its security has great potential for I.T deployment and integration in business operations. The very near concept of today's cloud computing was called grid computing. The intangible difference between cloud and grid is that cloud comes with service Level Agreement (SLA) between customer and the provider. The term grid computing also originated in the early 1990s as a metaphor for making computer power as easy to access as an electric power grid. Unlike Cloud; the Grid has disadvantages of relying heavily on dispersed data management and connectivity errors. There are different types of cloud computing and summary of differences are shown in Table 1 [3]: (www.mcrinc.com)

Table 1: Cloud topologies – benefits and risks

TOPOLOGY	BENEFITS & RISK
Public Cloud	-**Benefit***:* Low investment hurdle -**Risk**: Multi-tenancy security threat
Private Cloud	-**Benefit**: Sustains in-house security policy -**Risk:** High operation and maintenance cost
Hybrid Cloud (Public-Private)	-**Benefit***:* Constant network availability -**Risk:** vulnerable to all public network risk

Specific research question to improve on cloud security and robust access control at the user-end workstation pointed to the use of Two-step authentication method to help ensure that a hacker cannot access corporate data when in full possessions of authorised username, password, PIN code and the stolen mobile phone that is used to issue the second factor secret code. The need for 2-step authentication is critical for now. Mark Burnet [2] stated in their 182-page book on access control methods that user passwords are the keys to the network kingdom, yet most users choose overly simplistic passwords (like password) that anyone could guess, while system administrators demand impossible to remember passwords littered with obscure characters and random numerals. The need for more rigid access control to improve cloud trusted security has been the focus of this new work.

The issue of Trusted-security remains peculiar, either we move enterprise data unto a private, public or hybrid cloud platforms. In the same manner, an enterprise decision to stay in-house with corporate data would not change security challenges. A company gains some amount of administrative control, whiles it remains within in-house than on cloud platform. This is so because corporate clients are compelled to work with unknown cloud system administrators managing such a multitenant data house. The most popular end-user authentication is the use of secret code generation. In an attempt to login, authorized user is required to generate a secret code and add to the password as second factor access pin to gain access. However the use of secret code generation has limitation. A hacker holder username and password can isolate the mobile phone from the login server since designers give option for mobile isolation option to users. A hacker can install password sniffer (keylogger) on the computer to track keys used by the authorised end user.

The discussion on unknown cloud administrator and the activities of hackers prompted a redesign of cloud login architecture to offer sufficient protection at user-end workstation. In this work the purpose was to redesign login architecture so that any attempted domain login or intrusion by unknown user is observed and blocked on real-time by the authorised user via permanently connected user mobile phone that is biometrically compatible. The next was to shift from the use of secret codes for authentication to biometric scanning on the mobile phone to complete the login process for authorised enterprise cloud user. This study primarily investigated existing 2-step authentication security methods and observed how to improve the method for cloud computing implementation. The main research question was to know what improvement on Two-step authentication method would ensure that a hacker in possession of authorised username, password, authentication PIN code and the stolen mobile phone cannot get access to corporate data sitting on cloud computing infrastructure using authorised end-user network route. The follow up question was to find a way an SME's subscribing to cloud computing platform could administer security control to block real time intrusion by a hacker criminally holding cloud credential to enter corporate domain. As a limitation we could only provide the descriptive results after applying Delphi technique. Some detailed challenges spelt out in material and method section in this paper.

Literature surveys revealed limitation in the existing two-step authentication methods used by giant cloud operators such as Google, Amazon, Dropbox, Barclay's internet banking, Fujitsu limited and others. Cloud operators using this method attempts to focus on improving either the secret code issuant process or avoid keylogger hacking attempts.

Within the limitations, the 2-step authentication method still provides some level of login security and confidentiality to enterprise data. The unique discovery was the fact that almost all the introductions in different cloud houses adopted similar or closely refined architecture as shown in Figure 1 and developed for use by Fujitsu limited.

S. Sotashi (2009) patented two-step authentication architecture at Fujitsu Limited to improve data security as shown in Figure 1 [12]**.** They discussed that an authentication system includes a user terminal to perform authentication based on a password corresponding to a seed number generated in accordance with a predefined rule. The system further includes a password issuance apparatus to issue the password in response to reception of a request message including the seed number as shown in Figure 1.

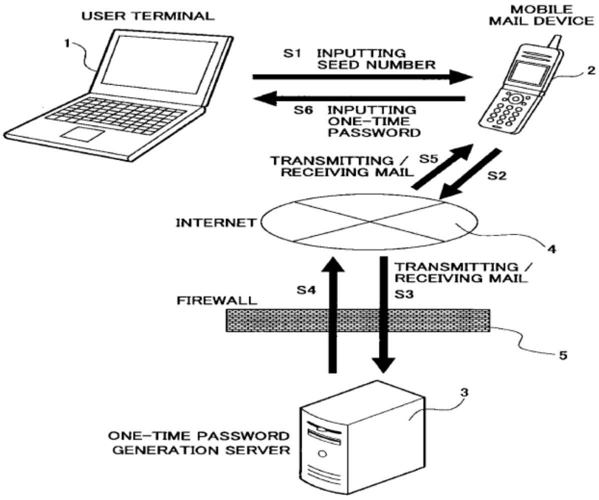

USER TERMINAL
MOBILE MAIL DEVICE
1
2
S1 INPUTTING SEED NUMBER
S6 INPUTTING ONE-TIME PASSWORD
TRANSMITTING / RECEIVING MAIL
S5
S2
INTERNET
4
TRANSMITTING / RECEIVING MAIL
S4
S3
FIREWALL
5
3
ONE-TIME PASSWORD GENERATION SERVER

Figure 1: The implementation of Two-Step Authentication as used by Fujitsu Limited

The method of design has three main device segments: User terminal, Mobile device, Server, and transmission medium.

Figure 1: The mobile device has software uploaded to communicate with the authentication server. Each user has a code called seed number; user enters that seed number on the mobile device and the seed number transmits to the authentication server. Authentication server compares SIM number sending request and Seed number assigned to that authorise user. If SIM and SEED numbers corresponds to pre-defined numbers on the server, then server grants the request to transmit one-time- password to the phone. Authorised user can now obtain password and use it to log-in through the computer domain to access corporate data on business operations server. *[Patent Document 1] Japanese Laid-open Patent Publication No. 2007-58469].*

The work of Google [9] and Dropbox Inc**.** [7] are highly connected to the architecture by Fujitsu. The same as Barclay's internet banking with additional soft feature [8]. Barclays introduced to the same architecture, soft screen keypad for entering the secret code sent via the user phone to avoid keylogger sniffing activities to track keys. Amazon [1] improved the method with the use of barcode scanning to compare with stored template. We will discuss the review further under results and discussion section in this paper since a common and fundamental architecture as execution process is used by aforementioned entities.

2 Materials and Methods

We investigated the design using survey with some IT security experts in Ghana and only provided descriptive results concluded with the help of the experts that were engaged. The design, which supposed to provide experimental results, has to adopt the alternate approach of using the Delphi technique with security experts, local software developers and Telecom data switching engineers due to lack of biometric laboratory in the region. The cost of international lab assistance was another challenge. One typical challenge was the need for biometric compatible mobile phone that can that has multimodal feature to accept more than one biometric templates of the human anatomy. The largest area to acquire very related secondary materials on 2-step authentication method were online libraries of the companies using the approach as second factor security for their clients' data privacy. Most academic and journal papers reviewed on two-step authentication have to deal with network paths and address protocols using 2-step methods; not for the purpose of user-end workstation as in our work. These papers were later avoided as part of secondary data collection in this paper to reduce scope deviation. Primary data collection was captured through unstructured interviews with I.T security experts, cloud administrators and mobile switching engineers to improve on the descriptive work. The System Development Life Cycle-revolutionary waterfall approach was adopted to make decision on the software development process.

Prototyping the design has been the major choice for initiating these codes. The advantage was to give chance to both the designers and potential users suggest improvement to the project completion. It was important to determine the testing standard suitable for this design to pre-inform the implementation stage of the system development Life cycle-SDLC. There are three basic testing methods (Thomas Vian, 2002) .Top-down, Bottom-up and Ends-in and we selected the Top-down approach. Top-down is a method in which a programmer joins the overall skeletal structure of the program before performing test. Next, the programmer would 'load in' the reviewed codes in some sections and then test it again and so on until the final program is completed and tested**.** The advantage of this method is that the programmer is always looking at the whole problem in one go, as all the parts of the program are related to each stage of execution [13].

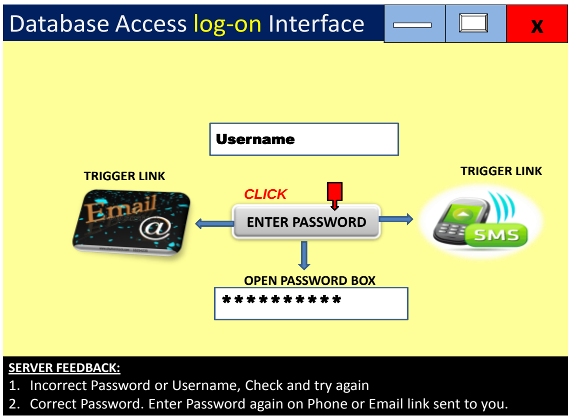
3 Results and Discussions

A new login architecture design was developed using the 2-step authentication concept as shown in Figure 2. The design has 4-minutes validation or expiration period within which an authorised user has to provide biometric authentication input to gain access to data.

Figure 2: The use of biometric for 2-step authentication process

Unique feature introduced to this concept was the use of logic "AND" gate mathematical model for interlocking user computer (workstation) to user mobile phone permanently. Cloud password "AND" biometric scan input must be provided at any log-in attempt to gain access to data. This is quite different form the existing works. The existing works make use of logic "OR" function so that users can have the option to disconnect and re-connect their mobile device the use to accept authentication sms link. User self-service privilege to connect and disconnect authentication phone from the login security server is a limitation and a threat to data privacy or protection in the cloud.

On experiment as shown in Figure 3, an authority user must login with cloud username and password on computer as step-1 key to do access request. The user has to complete login using biometric input for authentication as step-2 key. The authentication server picks the request by step-1 password key "AND" embeds login link to a pre-defined user mobile phone to allow biometric scan and resend to the server. The scan is finally sent back to the authentication server to compare with stored biometric template. There is a backdoor alternative to login when phone is lost: Figure 3 shows the backdoor routing link to user corporate e-mail account.



SERVER FEEDBACK:
1. Incorrect Password or Username, Check and try again
2. Correct Password. Enter Password again on Phone or Email link sent to you.

Figure 3: Backdoor Authentication routing through user E-mail account

Imagine you lose your biometric compatible phone in a car; what alternative can ensure business continuity in the cloud by such user? Backdoor entry is important in any access design and programming. There is backdoor login process as shown in Figure 3; link is sent through E-mail account so that any enterprise computer with embedded biometric scanner is used to authenticate user when the pre-defined mobile phone stolen is pending for replacement. The implementation is geared towards enterprise interest so as to provide level of trust to operate both transactional and query-based traffics on the cloud platform with hope for data confidentiality and integrity assurance.

Discussing the literature review further; it was observed that the Two-step authentication in general is a customer-felt security method that can convince and attract more clients to trust cloud adoption as well as improve domain access control to users of the cloud. If the issuance of existing secret code or OTP by some cloud operators through mobile device is powerful to some extent, then the use of biometric scan on mobile device to complete second factor authentication would be a great potential to improve loud security. Again, it was noted that a hacker who steals the configured mobile phone in addition to username and password could fully access every data on the cloud datacenter because of the use of OTP codes. We can resolve this problem with the use of biometric introductions. Therefore, for small and medium enterprises to hook unto cloud business solutions, each corporate user requires specialised multi-modal biometric compatible mobile phones, which is now coincidentally made available for different purpose by the U.S department of defense research sponsorship to AOptix development limited (2013). Using such same biometric iPhone, a hacker holding username, password and stolen iPhone cannot access data as the life body of the authorised user may still be required to complete the second factor authentication. The only source of breakthrough for a hacker is when the user biometric is decoded into the logic digit representation. As a way forward to combat any transmission interception and decoding, the biometric data requires encryption within the transmission path to ensure data privacy.

The Algorithm Format

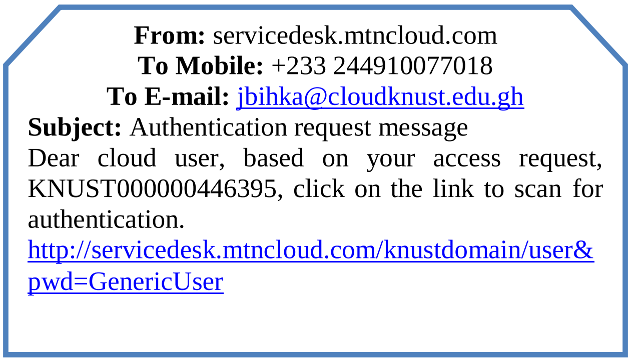
Algorithm 1: Authentication Execution through mobile phone

```
Begin
Initialize user terminal side: Enter correct cloud username and password.
Send service request to the dedicated login server.
while Not end of user session do
  Accept transaction data on login server and compare with stored user credential.
  Belief sent credential.
  Send authentication request to predefined user mobile phone.
  Capture biometric feature on user phone and send back to authentication server
  if biometric captured on phone matches template on authentication server
  then
     Open user domain when username, password "AND" biometric feature matches for that user domain.
  end if
  Periodically block access in ideal situation to allow fresh login
end while
End
```

Algorithm 2: Backdoor Authentication on Laptop/E-mail

```
Begin
Initialize user terminal side: Enter correct cloud username and password.
Send service request to the dedicated login server.
while Not end of user session do
  Accept transaction data on login server and compare with stored user credential.
  Belief sent credential.
  Send authentication request to both predefined user mobile phone and Email account.
  Capture biometric feature on biometric compatible laptop and send back to authentication server
  if biometric captured on laptop matches template on authentication server
  then
    Open user domain only.
  end if
  Periodically block access in ideal situation to allow fresh login
end while
End
```

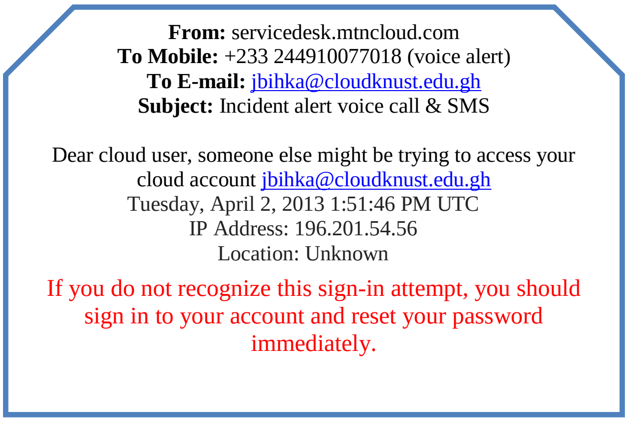
Sample Alert SMS shown in Figure 4.



From: servicedesk.mtncloud.com
To Mobile: +233 244910077018
To E-mail: jbihka@cloudknust.edu.gh
Subject: Authentication request message
Dear cloud user, based on your access request, KNUST000000446395, click on the link to scan for authentication.
http://servicedesk.mtncloud.com/knustdomain/user&pwd=GenericUser

Figure 4: Authentication SMS link to user.

There is no way to assume perpetual motion machine in the world of knowledge advancement. Although the new design predicts a robust access control to data, the programming involved alert message after third time password attempt as shown in Figure 5. This is to create real time awareness to user to change cloud password suspected to be compromised.



From: servicedesk.mtncloud.com
To Mobile: +233 244910077018 (voice alert)
To E-mail: jbihka@cloudknust.edu.gh
Subject: Incident alert voice call & SMS

Dear cloud user, someone else might be trying to access your cloud account jbihka@cloudknust.edu.gh
Tuesday, April 2, 2013 1:51:46 PM UTC
IP Address: 196.201.54.56
Location: Unknown

If you do not recognize this sign-in attempt, you should sign in to your account and reset your password immediately.

Figure 5: Alert message after three times password entry

4 Conclusions and Recommendations

Unlike the existing works, a hacker holding user-password, user-phone or any biometric computer, cannot access enterprise cloud data without authorised fingerprint or other biometric feature scans provided as second factor input by authorised cloud user. In conclusion, the experience obtained from observing the trailing of existing software packages and discussions with experts have enhanced the development of the idea to implement two-step authentication method using multi-modal biometric to enhance cloud security trust for clients. Similarly, the use of real time biometric authentication, instead of secret codes for authentication, offers rigid security and trust on cloud data access. We recommended eye and fingerprint scans (multi-nodal scanning) to reduce authentication failure rate and argued conclusively that this re-design work can improve cloud trusted-security.

1). Recommendation for further work is to investigate encryption and decryption algorithms to secure biometric data transmission from the mobile phone to the authentication server so that when such biometric data is intercepted on the gateway it will be useless information when decoded.

2). E-government datacenters in developing countries stand a chance to benefit from this login security authentication. Many developing countries are integrating ICT or yet to use ICT in day-to-day government sector transaction. With this development, many civil and public servants are likely to depend on their family relatives to support them process government electronic data at home and office. With such biometric authentication, supporting relatives would find it difficult to login without consent of authorised user. This may enhance data privacy on the e-government platform.

Acknowledgement

I am grateful to my interviewees: Wilfred Bruku (SMS transmission engineer), Edgar Zormelo (Cloud system administrator) and Eric Afanu (I.T security consultant) for immense contribution gathered from their experiences. I also thank all the various IT users recruited to comment and criticise the design for possible limitation for security breech.

References

[1] Amazon web services library, Multifactor authentication system; (2009) Available from: <http://aws.amazon.com/mfa> Accessed on 28/02/2013.

[2] Mark Burnett and Dave Kleiman, "Perfect Password: Selection, Protection Authentication Illustrated", Syngress Publications, 2005, Chapters 1, 2, 13 pp. 182.

[3] Bill Claybrook, "Differences explained: Private vs. Public vs. hybrid cloud computing. Available from: http://www.mcrinc.com/Documents/Newsletters/201207/, Accessed on 6/03/2013.

[4] Cloud Security Alliance, "The notorious nine cloud computing top threats in 2013", cloud security Alliance top threat working group report, 2013.

[5] D. G. Débora & T. Brunzel, " Cloud computing Evalation-How it differs to Traditional IT outsourcing", Jönköping University academic archive online, Master student Thesis, pp. 1-29, 2010.

[6] Definitions of theories associated with cloud computing. Available from: http://en.wikipedia.org/wiki/Cloud_computing/, Accessed on 6/03/2013.

[7] Dropbox support-2-step authentication system; Available from: https://www.dropbox.com/help/363/en/, Accessed on 21/03/2013.

[8] Barclays Bank Ghana, "Internet banking library-step authentication system", Available on http://barclays.ghana@barclays.com/internetbanking/OTP, Accessed on 15/03/2013.

[9] Google support- 2-step verification system; Available from: http://support.google.com/accounts/bin/answer.py?hl, Accessed on: 04/02/2013.

[10] Wentao Liu, "Research on cloud computing security problem and strategy", 2nd International Conference on Consumer Electronics, Communicatons and Network, Three Gorges, China, April 2012.

[11] Adam Maguire, "PricewaterhouseCoopers forum on cloud", Available http://www.irishtimes.com, Accesseed on 5/03/2013.

[12] Sotashi Samba (2009) Fujitsu limited 2-step authentication system and method http://www.google.com/patents/EP2131302A2?cl, Accessed on 11/02/2013.

[13] Thomas Vian, "Design an interactive website to help teach maths, to year two, key stage one children", Academic thesis, Pages 82, Accessed on 2/04/2013.

Eric Opoku is a graduate of Kwame Nkrumah University of Science and Technology-Ghana. He holds an MPhil in Information Technology and a telecom network engineer at Mobile Telecommunication Network and Cloud computing

operator (MTN-Ghana). His research interest is in cloud computing –Security & IT applications for business solutions.

James B. Hayford-Acquah is a senior lecturer in Kwame Nkrumah University of Science and Technology-Ghana. His main research interests are image processing and securities. He was the former Head of department and currently the Exams officer of the department of computer science. He has extensive teaching experience at both postgraduate and undergraduate levels.

1-2 Skip List Approach for Efficient Security Checks in Wireless Mesh Networks

Hemraj Saini

Department of Computer Science & Engineering,
Jaypee University of Infromation Technology, Wakanaghat-173234 (INDIA)
(Email: hemraj1977@yahoo.co.in; hemraj.saini@juit.ac.in)

(Received Jan. 16, 2014; revised and accepted Mar. 27, 2014)

Abstract

In the fast growing era of the online business, Wireless Mesh Networks (WMNs) are playing a significance role to grow the economy of different countries. Therefore, it is essential to implement the efficient security measures or methods at the Wireless Mesh Gateway (GW) as it is only the place to enter the incoming traffic to the particular WMN. In the traditional solutions, the monotonically increased traffic at GW is handled by the help of "linear queue" data structure, which is not the efficient way for security analysis in the present criteria. Therefore, in the manuscript, it is replaced by another efficient data structure named "1-2 Skip List". In addition, it has shown that detection of flooding or DoS like attacks are also easily possible by using 1-2 Skip List approach. A sufficient analysis is also provided for the proposed solution with performance analysis in the manuscript.

Keywords: Wireless Mesh Network, WLAN, Network Security, 1-2 Skip List

1 Introduction

In case of sparsely populated areas it is difficult and costly to use traditional communication networks. In such types of situation Wireless Mesh Network (WMN) plays an important role for communication and helps to grow the business there. In WMN a number of radio nodes are organized under mesh topology. WMN generally contains mesh client, mesh routers and mesh gateways. Laptops, cell phones, vehicles and other devices having wireless capability can be the part of a WMN. In WMN mesh routers are used to forward traffic to and from the mesh gateway which may, but need not, connect to the Internet. Every radio node creates its own mesh cloud having the rage up to its coverage area [1, 2, 3] and likely to be a single network. Mesh cloud can be accessed by the permission of the radio node which is working in coordination with other radio nodes to create radio network. A WMN preserves the properties of reliability that offers redundancy. Non operational mode of a mesh node doesn't mean that rest of WMN is not in operation, the rest of the nodes can still communicate to each other. They can communicate directly or through other one or more intermediate mesh nodes. Figure 1 depicts a sample of WMN. A WMN can be implemented with various technologies such as 802.11, 802.15, 802.16, Cellular technology or combination of more than one type [4, 5, 6, 7, 8].

A talented purpose of WMN is the low cost extension of wireless local area network (WLAN) in a sparsely populated area. A WLAN links two or more devices using some wireless distribution method (typically spread-spectrum or OFDM radio), and usually providing a connection through an access point to the wider Internet. This gives users the mobility to move around within a local coverage area and still be connected to the network. Most modern WLANs are based on IEEE 802.11 standards, marketed under the Wi-Fi brand name. WLANs were once called LAWNs (for local area wireless network) by the Department of Defense. An application of WMN can be a pilot program to provide city workers in all different municipality of the city with wireless Internet. The traffic of all the access clients passes through the Mesh Gateway (GW) which causes a monotonically increase in the traffic [9, 10, 11]. Additionally, there is a great possibility of the malicious attack from the incoming traffic from the Internet. Therefore, it is heavily required to keep an eye over the incoming traffic. Another thing, the incoming traffic from the internet can be at high rate and hence an efficient data structure is to be used to synchronize it with the delivery speed at their respective destinations. In traditional implementation this data structure is either a linear array or a linear linked list [12, 13, 14]. These two data structures are having the scope of enhancement in the performance of the WMN as well as better security check process. In the manuscript, 1-2 skip list has been used for the purpose which leads the better results.

The paper is further organized into six more sections. Section-II is devoted to understand the concepts of skip list. Section-III discussed the usage of 1-2 skip list for temporary storage of incoming traffic. Section-IV provides the implementation of 1-2 skip list to detect DoS like attacks. Section-V explains about the performance analysis of proposed method and the section-VI provides the conclusion about the whole work.

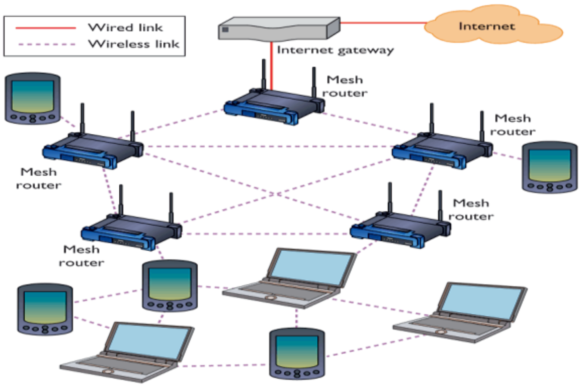


Figure 1: Wireless mesh network

2 Skip List

A skip list is a data structure for storing a sorted list of items using a hierarchy of linked lists that connect increasingly sparse subsequences of the items. These auxiliary lists allow item lookup with efficiency comparable to balanced binary search trees (that is, with number of probes proportional to log n instead of n).

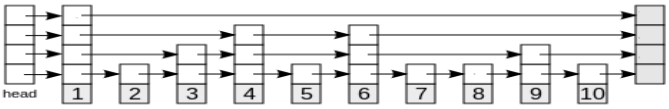


Figure 2: Skip list

Each link of the sparser lists skips over many items of the full list in one step, hence the structure's name as shown in Figure 2. These forward links may be added in a randomized way with a geometric / negative binomial distribution. Insert, search and delete operations are performed in logarithmic expected time. The links may also be added in a non-probabilistic way so as to guarantee amortized (rather than merely expected) logarithmic cost [15, 16].

A simple version of a skip list is called an 1-2 skip list, that has either 1 or 2 nodes of height h-1 between any two nodes of height h or more. This is depicted in Figure 3.

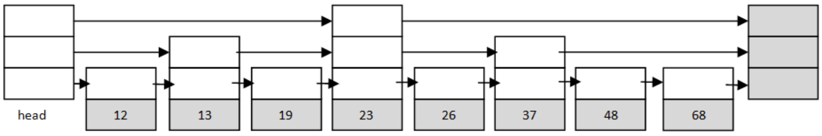


Figure 3: 1-2 Skip list

3 1-2 Skip List For Temporary Storage of Incoming Traffic

Incoming traffic at the mesh gateway (GW) node in Wireless Mesh Network (WMN) from the internet is continuously available. There are many number of access client nodes (ACNs) are existing in WMN and want to interact with the external points existing in the outer Internet. Therefore, the traffic can be increased monotonically at the GW node. To handle this incoming traffic it is required to temporarily store at the GW node by the use of some data structure. This data structure is generally storage queue which is less efficient to analyze the traffic for security check as the incoming traffic may contains malicious information. Therefore, we are proposing the usage of 1-2 skip list for temporary storage of incoming traffic. Incoming traffic stored in 1-2 skip list can be analyzed in a better way for security checks and increases the efficiency of the network. Let us consider a criteria where more than on DoS attacks are available in the incoming traffic from different source IPs. If the traffic is stored in a queue than it is difficult to analyze each packet for a particular time duration but 1-2 skip list make it easier. Assuming that there is incoming traffic from different source IPs and is recorded for a fixed time interval Δt shown as below in Table 1.

Table 1: Incoming packets from various source IPs in Δt time

Sr. No.	Source IP	No. Of Packets
1	IP1	10
2	IP1	20
3	IP2	12
4	IP3	13
5	IP1	23
6	IP2	34
7	IP3	27
8	IP4	24
9	IP3	15
10	IP2	31
11	IP1	15

1-2 skip list can be designed for Table 1 depicted in Figure 4. In Figure 3 every value is build by two atomic values i.e. source IP and number of packets in Δt time.

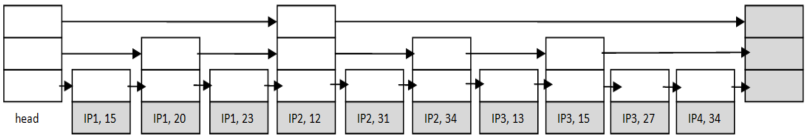


Figure 4: Corresponding 1-2 skip list for Table 1

4 Using 1-2 Skip List To Detect DoS Like Attack

Incoming traffic from the outer side will entered into the WMN from WG only therefore, either a queue of a 1-2 skip list is required to handle the incoming traffic. In the queue it is difficult to identify an IP which is continuously sending the packets beyond a threshold value during Δt time but 1-2 skip list can by tool to solve this problem.

1-2 skip list sorts all the incoming traffic with IP and number of packets shown in Figure 3 and hence it can be utilized to improve the performance. Identifying which IP is flowing maximum number of packets during Δt time, it can be checked with respect to the ***threshold value*** (threshold value can be different for different scenarios of the network and traffic management in network and can be adjusted by an adaptive way [17]). If it is beyond the threshold value, related IP can be categorized as the suspected IP and kept under supervision. If same property of this IP is repeated for 5 times (or fixed according to the application), it can be marked as attack carrying IP and security measures are to be taken over it. In this way Distributed Denial of Service Attacks (DoS) of flooding attacks can be easily detected and sorted out which is depicted in Figure 5 and Figure 6 in the form of flow chart and algorithm respectively.

5 PERFORMANCE ANALYSIS OF PROPOSED METHOD

Most common operations used over 1-2 skip list at GW node are search, insert and delete. Incoming traffic can be handled at GW node by using any one of linear queue, linear linked list and 1-2 skip list. Incoming traffic at GW node handled by linear queue allows search operation at cost O(logn) and insertion and deletion at the cost O(n). But in case of using linear linked list search is at the cost O(n) and insertion and deletion at the cost O(1).

In case of 1-2 skip list all the operations are optimized and performance increased. Search operation for 1-2 skip list is at the cost O(n) and updating operation have the cost O(logn). An experiment has been carried out in a constant computer network of seven (07) computer nodes. Results of the network statistics when queue is used and 1-2 skip list is used are shown in Table 2 and Table 3 respectively.

It is depicted in Figure 7 that there is no packet drop in case of 1-2 skip list implementations as it is dynamic but in case of queue implementation there are packet drops. Figure 8 depicts that there is no false positive detected in case of 1-2 skip list implementation as it will detect the attack only after a sufficient time supervision but in case of queue there are false positive detected as it will every time detect the attack if the incoming traffic goes beyond the threshold value.

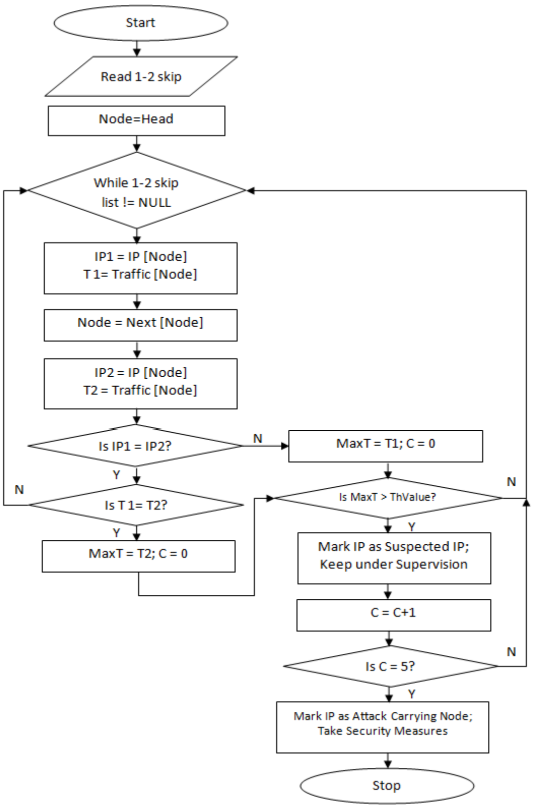
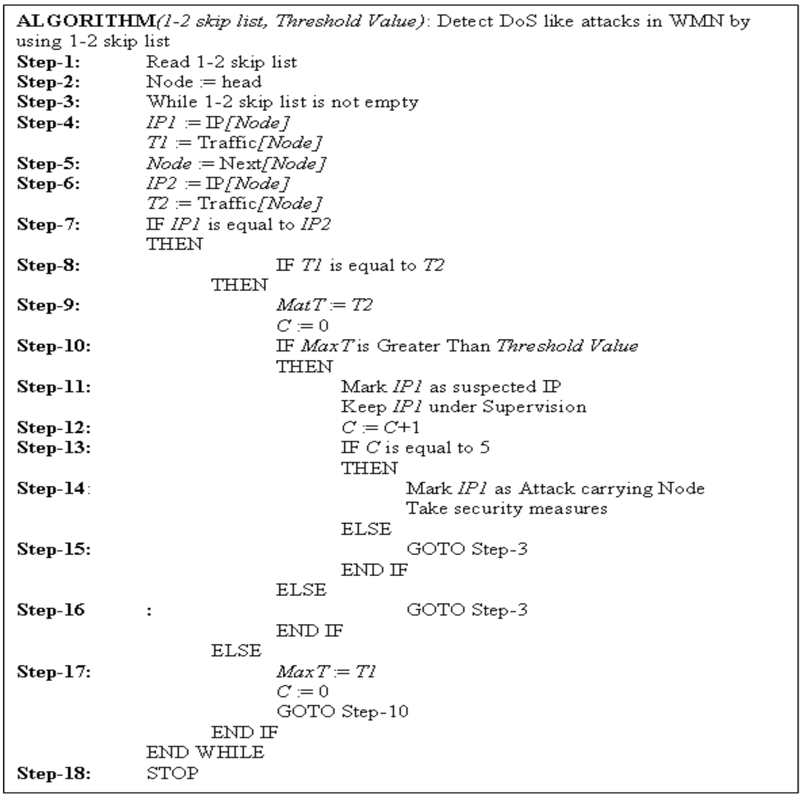


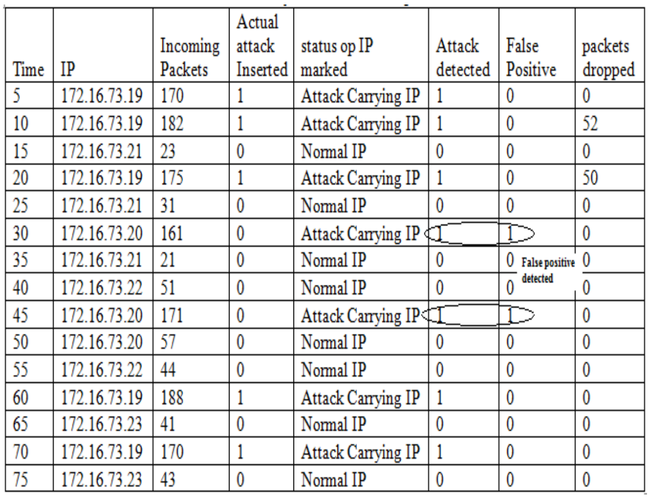
Figure 5: Flow chart to detect DoS like attacks in WMN by using 1-2 skip list



```
ALGORITHM(1-2 skip list, Threshold Value): Detect DoS like attacks in WMN by
using 1-2 skip list
Step-1:     Read 1-2 skip list
Step-2:     Node := head
Step-3:     While 1-2 skip list is not empty
Step-4:     IP1 := IP[Node]
            T1 := Traffic[Node]
Step-5:     Node := Next[Node]
Step-6:     IP2 := IP[Node]
            T2 := Traffic[Node]
Step-7:     IF IP1 is equal to IP2
            THEN
Step-8:                 IF T1 is equal to T2
                  THEN
Step-9:                 MatT := T2
                        C := 0
Step-10:                IF MaxT is Greater Than Threshold Value
                        THEN
Step-11:                      Mark IP1 as suspected IP
                              Keep IP1 under Supervision
Step-12:                      C := C+1
Step-13:                      IF C is equal to 5
                              THEN
Step-14:                            Mark IP1 as Attack carrying Node
                                    Take security measures
                              ELSE
Step-15:                            GOTO Step-3
                              END IF
                        ELSE
Step-16     :                       GOTO Step-3
                        END IF
                  ELSE
Step-17:                MaxT := T1
                        C := 0
                        GOTO Step-10
                  END IF
            END WHILE
Step-18:    STOP
```

Figure 6: Algorithm to detect DoS like attacks in WMN by using 1-2 skip list

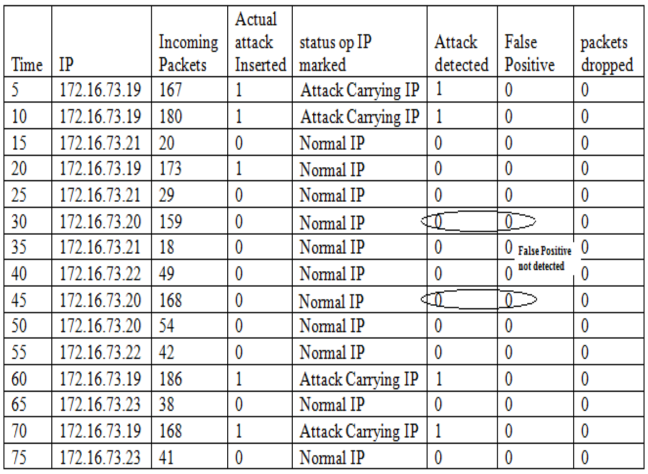
Table 2: Network Statistics in case of Queue and constant parameters of Network



Time	IP	Incoming Packets	Actual attack Inserted	status op IP marked	Attack detected	False Positive	packets dropped
5	172.16.73.19	170	1	Attack Carrying IP	1	0	0
10	172.16.73.19	182	1	Attack Carrying IP	1	0	52
15	172.16.73.21	23	0	Normal IP	0	0	0
20	172.16.73.19	175	1	Attack Carrying IP	1	0	50
25	172.16.73.21	31	0	Normal IP	0	0	0
30	172.16.73.20	161	0	Attack Carrying IP	1	1	0
35	172.16.73.21	21	0	Normal IP	0	0	0
40	172.16.73.22	51	0	Normal IP	0	0	0
45	172.16.73.20	171	0	Attack Carrying IP	1	1	0
50	172.16.73.20	57	0	Normal IP	0	0	0
55	172.16.73.22	44	0	Normal IP	0	0	0
60	172.16.73.19	188	1	Attack Carrying IP	1	0	0
65	172.16.73.23	41	0	Normal IP	0	0	0
70	172.16.73.19	170	1	Attack Carrying IP	1	0	0
75	172.16.73.23	43	0	Normal IP	0	0	0

False positive detected

Table 3: Network Statistics in case of 1-2 skip list and constant parameters of Network



Time	IP	Incoming Packets	Actual attack Inserted	status op IP marked	Attack detected	False Positive	packets dropped
5	172.16.73.19	167	1	Attack Carrying IP	1	0	0
10	172.16.73.19	180	1	Attack Carrying IP	1	0	0
15	172.16.73.21	20	0	Normal IP	0	0	0
20	172.16.73.19	173	1	Normal IP	0	0	0
25	172.16.73.21	29	0	Normal IP	0	0	0
30	172.16.73.20	159	0	Normal IP	0	0	0
35	172.16.73.21	18	0	Normal IP	0	0	0
40	172.16.73.22	49	0	Normal IP	0	0	0
45	172.16.73.20	168	0	Normal IP	0	0	0
50	172.16.73.20	54	0	Normal IP	0	0	0
55	172.16.73.22	42	0	Normal IP	0	0	0
60	172.16.73.19	186	1	Attack Carrying IP	1	0	0
65	172.16.73.23	38	0	Normal IP	0	0	0
70	172.16.73.19	168	1	Attack Carrying IP	1	0	0
75	172.16.73.23	41	0	Normal IP	0	0	0

False Positive not detected

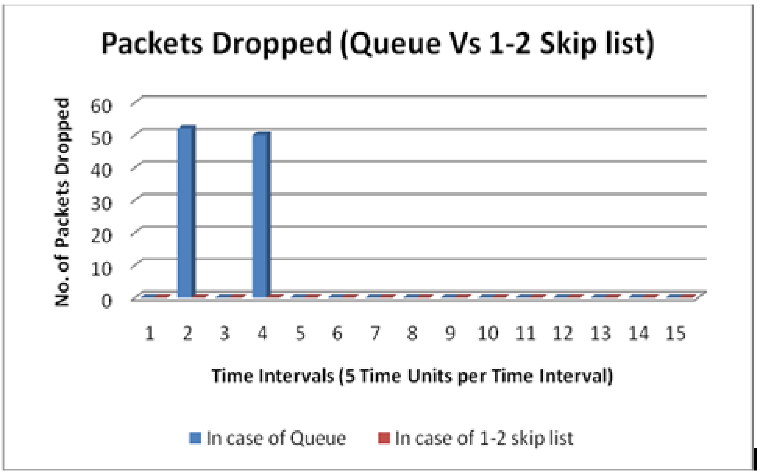


Figure 7: No of Packets dropped in Queue Vs 1-2 skip list implementation

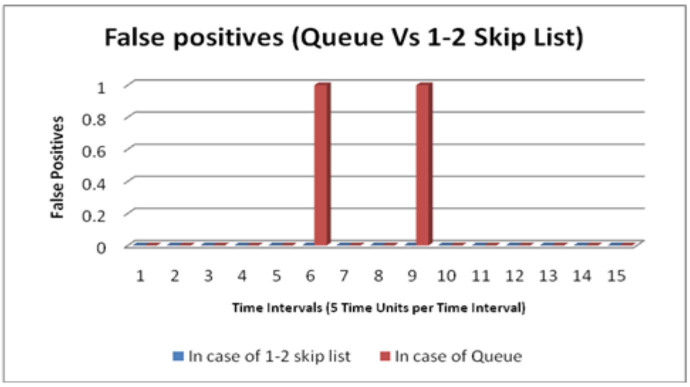


Figure 8: False Positives in Queue Vs 1-2 Skip List Implementation

6 CONCLUSION

In the current era, Ad Hoc computer Networks are having much importance in our day today life and Wireless Mesh Network is one of its types. The paper deals to explain WMN and the usage of 1-2 skip list to handle incoming traffic at Wireless Mesh Gateway node. It also explains that 1-2 skip list in place of linear queue is better to detect flooding or DoS type of attacks with an efficient manner. An appropriate and sufficient analysis with results is also provided in its support.

References

[1] Saini H., Sharma L. K., Panda T. C., and Pratihari H. N., "Extended Cell Planning for Capacity Expansion and Power Optimization by Using MEMETIC Algorithm," *International Journal of Wireless Networks and Broadband Technology (IJWNBT)*, vol-2, Issue-2, pp. 36-46, 2012.

[2] Sharma L. K., Saini H., Panda T.C., and Pratihari H. N., "Taxonomy of Cell Planning," *International Journal on reviews on Computing*, vol. 3, Issue-3, pp. 66-74, 2010.

[3] Cheng K. and Dasgupta P., "Weighted Voting Game Based Multi-Robot Team Formation for Distributed Area Coverage," *in Proceedings of the 3rd International Symposium on Practical Cognitive Agents and Robots (PCAR '10)*. ACM, New York, NY, USA, pp. 9-15, 2010.

[4] Saini H., Sharma K. D., Dadheech P., and Panda T. C., "Enhanced 4-way Handshake Process in IEEE802.11i with Cookies," *International Journal of Information & Network Security (IJINS)*, vol.2, No.3, pp. 229-238, 2013.

[5] Zhou Y., Wang Y., Ma J., Jia J., and Wang F., "A Low-latency GTS Strategy in IEEE802.15.4 for Industrial Applications," *IEEE 2nd International Conference on Future Generation Communication and Networking*, pp. 411-414, 2008.

[6] IEEE 802.16 Working Group on Broadband Wireless Access. *http://wirelessman.org*

[7] Singh V. and Sharma V., "Efficient and fair scheduling of uplink and downlink in IEEE 802.16 OFDMA networks." *Wireless Communications and Networking Conference, IEEE WCNC*, 2006.

[8] Cicconetti C., Lenzini L., Mingozzi E., and Eklund C., "Quality of service support in IEEE 802.16 networks. *IEEE Network*", vol. 20, pp. 50-55, 2006.

[9] Cordero J. A., Yi J., and Clausen T., "Optimization of jitter configuration for reactive route discovery in wireless mesh networks," *International Symposium on Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt)*, 2013 11th, pp. 452- 459, 2013.

[10] Ahmed I, Mohammed A., and Alnuweiri H., "On the fairness of resource allocation in wireless mesh networks: a survey," *Wirel. Netw.* 19, 6, pp. 1451-1468, 2013.

[11] Akyildiz I. F., Wang X. D., and Wang W. L., "Wireless Mesh Networks: A Survey," *Computer Networks*, vol. 47, No. 4, pp. 445-487, 2005

[12] Nandiraju N.S., Nandiraju D. S., Cavalcanti D., Agrawal D.P., "A Novel Queue Management Mechanism for Improving Performance of Multihop Flows in IEEE 802.11s based Mesh Networks Performance," *25th IEEE International on Computing, and Communications (IPCCC 2006)*, pp.161-168, 2006.

[13] Garcia-Luna-Aceves J.J., Menchaca-Mendez, and R. STORM, “A Framework for Integrated Routing, Scheduling, and Traffic Management in Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, vol. 11, No. 8, AUGUST 2012.

[14] Nandiraju N.S., Nandiraju D.S., Santhanam L., and Agrawal D.P., “A Cache Based Traffic Regulator for Improving Performance in WEEE 802.11s based Mesh Networks,” *IEEE Conference on Radio and Wireless Symposium*, pp. 293-296, 2007.

[15] Wikipedia, Skip-List, Retrieved on: Jan, 2014), Available at: *http://en.wikipedia.org/wiki/Skip_list*.

[16] William Pugh, “Skip lists: a probabilistic alternative to balanced trees,” *Communications of the ACM*, 33(6), pp. 668–676, 1990.

[17] Kim, Y. H., Lee, S.K., Koh, J. G., “Enhanced Synchronizing Packet Coalescing Mechanism for Improving Energy Efficiency in Ethernet Switch,” *International Journal of Smart Home*, vol. 7, No. 3, May, 2013.

Hemraj Saini received his PhD in Computer Science from Utkal University, Vani Vihar, Bhubaneswar (ODISHA), M.Tech. degree in Information Technology from the Punjabi University, Patiala, Punjab and B.Tech. in Computer Science & Engineering from National Institute of Technology, Hamirpur (H.P.). Since 1999, he has been actively engaged in Research, Teaching and academic Development activities. Currently he is attached with the Department of Computer Science & Engineering / ICT of Jaypee University of Information Technology, Wakanaghat INDIA. His main professional interests are in Cyber Defense, Software Testing, Enterprise Application Integration, Image processing and Intelligent Techniques. He has played an important role for organizing various National and International Conferences successfully funded by DST, Govt. of India, New Delhi, CSIR, Govt. of India, New Delhi and AICTE, Govt. of India, New Delhi. In addition to it he has published more than 45 research articles in various National/International Journal/Conferences of repute.

Memory-only Selection of Dictionary PINs

Martin Stanek

Department of Computer Science, Comenius University
Mlynská dolina, 842 48 Bratislava, Slovakia
(Email: stanek@dcs.fmph.uniba.sk)

(Received Mar. 7, 2014; revised and accepted Apr. 16, 2014)

Abstract

We estimate the security of dictionary-based PINs (Personal Identification Numbers) that a user selects from his/her memory without any additional aids. The estimates take into account the distribution of words in source language. We use established security metrics, such as entropy, guesswork, marginal guesswork and marginal success rate. The metrics are evaluated for various scenarios – aimed at improving the security of the produced PINs. In general, plain and straightforward construction of memory-only dictionary PINs yields unsatisfactory results and more involved methods must be used to produce secure PINs.

Keywords: authentication, dictionary PIN, metrics, security

1 Introduction

A PIN is frequently used form of user authentication. The PIN is a fixed-length string of digits, usually of length 4, 5 or 6. There are recommendations on how to choose and work with the PINs in secure manner, e.g. [7, 15, 9]. Other proposals try to devise methods for producing sufficiently secure PIN [10, 11]. Even though the users are often informed and aware of PIN security, several studies showed that the significant portion of the users choose weak, easily guessable PINs [2, 4]. Weaknesses can also lie in other aspects of using authentication secrets, e.g. partial password/PIN verification [1].

One possibility of choosing and memorizing the PIN is to use so-called dictionary PIN. Dictionary PIN is derived from a word with the mapping offered by numpads of ATMs, mobile phones or Point-of-Sale terminals. The most commonly used letter to digit mapping is the standard mapping shown in Figure 1.

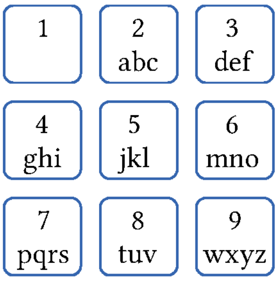


Figure 1: The standard mapping

Certainly, other mappings are possible, covering also digits 0 and 1. Recent study of dictionary PINs [12] analyzed the security of dictionary PINs with respect to various languages and dictionaries. It also described and assessed several methods of improving dictionary PINs selection. The assessment assumed uniform distribution of dictionary words, see [12]:

> “Let us stress that the experiments treat the words in a dictionary as equally probable. This is certainly not true if a user chooses the word from his/her memory. The uniform distribution can be easily achieved with the aid of an application that offers random sets of words for the user to choose from.”

Nevertheless, sometimes it can be impractical to use an external application and some users can hesitate to use or they would not even trust such application for PIN selection. Therefore we focus on dictionary PINs that user selects from his/her memory, without any external aid. We address the question of the security of such user-generated dictionary PINs in this paper. The main findings of our experiments are the following:

– Considering uniform frequencies of dictionary words is inadequate for estimating the security of memory-only selection of dictionary PINs.

– The straightforward word to PIN translation yields unacceptable marginal success rates when frequencies are taken into account.

– Simple blacklisting, prefix, and two-dictionary methods offer only a moderate improvement in security metrics.

– A more demanding methods, such as morphing or combination of multiple methods, are needed to obtain significantly better results.

1.1 Quantifying the Predictability of PIN

Let $N = 10^n$ be the size of an PIN space, for PIN length n. Let X be a random variable over the set $\{0, 1, \ldots, 9\}^n$. Let p_i denotes the probability of choosing a particular PIN x_i. Without loss of generality we assume that PINs are sorted in the descending order of their probabilities, i.e. $p_1 \geq p_2 \geq \ldots \geq p_N$.

Various measures for the PIN choices were proposed and studied, for details, discussions and relations among these metrics see [3, 4, 16]. The most important ones are defined in the following list.

– Shannon entropy, expressed in bits, measures the uncertainty in a random variable:

$$H_1(X) = -\sum_{i=1}^{N} p_i \log_2 p_i.$$

– The guesswork measures the expected number of guesses needed to find a PIN, trying in descending order according their probability:

$$G(X) = \sum_{i=1}^{N} i \cdot p_i.$$

– The marginal guesswork measures the expected number of guesses needed to increase the success probability of finding the PIN to at least α (usually $\alpha = 0.5$):

$$\mu_\alpha(X) = \min\{1 \leq k \leq N \mid \textstyle\sum_{i=1}^{k} p_i \geq \alpha\}$$

– The marginal success rate measures the probability of guessing the PIN given β attempts (β is usually 3 or 6):

$$\lambda_\beta = \sum_{i=1}^{\beta} p_i.$$

Since the guesswork and the marginal guesswork are not directly comparable to Shannon entropy, the values of $G(X)$ and $\mu_\alpha(X)$ are converted into bits using the following formulas [3]: $\tilde{G}(X) = \log_2(2G(X) - 1)$, $\tilde{\mu}_\alpha(X) = \log_2(\mu_\alpha(X)/\lambda_{\mu_\alpha})$.

Since the standard mapping does not cover digits 0 and 1, the ideal security metrics have the following values for dictionary PINs (assuming uniform distribution of PINs):

– PIN length 4: $H_1(X) = \tilde{G}(X) = \tilde{\mu}_{0.5} = 12.00$ bits, $\lambda_6(X) = 0.15\%$,

– PIN length 5: $H_1(X) = \tilde{G}(X) = \tilde{\mu}_{0.5} = 15.00$ bits, $\lambda_6(X) = 0.02\%$.

2 Estimating Metrics for Dictionary PIN Selection

In order to model frequency distribution of words in a language we use frequency lists based on subtitles. This is a respected method for analyzing contemporary languages [5]. We use two frequency lists for English – a carefully prepared SUBTLEXus [14] containing 60,384 words with a frequency higher than 1, and the list compiled from subtitles available from opensubtitles.org [8], containing more than 450,000 words (even words with frequency 1). We will denote the results obtained using the first/second list with label “SUBTLEXus”/“opensub”, respectively.

2.1 Basic Statistics

We compare the metrics for straightforward translation of words to PINs using the standard mapping, see Figure 1, with results obtained in [12]. We consider only the words with the length equal to the PIN length n. The translation starts with stripping the diacritical marks, if they are present. Then, the word is mapped to PIN using the standard mapping, e.g. "love" and "hate" yield 5683 and 4283, respectively. The frequency of particular word contributes to the probability of resulting PIN.

The results for the PIN lengths 4 and 5 are shown in Table 1. The columns labeled "uniform" contain results for PINs derived from a large (spell-checker) English dictionary assuming uniform frequencies of words [12]. The columns labeled "RockYou" contains results for PINs derived from RockYou password database where frequencies of words (passwords) were taken into account. It is easy to notice a striking difference between scenarios that consider the frequencies of words and those that do not.

Table 1: Comparison of metrics for straightforward construction of dictionary PINs

	SUBTLEXus		opensub		uniform [12]		RockYou [12]	
	$n = 4$	$n = 5$	$n = 4$	$n = 5$	$n = 4$	$n = 5$	$n = 4$	$n = 5$
H_1 (bits)	7.23	8.42	7.42	8.88	11.28	13.37	–	–
$\tilde{G}$ (bits)	7.18	8.63	7.49	9.45	10.94	13.08	–	–
$\tilde{\mu}_{0.5}$ (bits)	5.52	6.58	5.64	6.92	10.61	12.68	9.20	10.76
λ_6 (%)	23.93	19.24	22.80	17.34	0.85	0.33	10.81	8.46

A closer look at the most frequent dictionary PINs from SUBTLEXus and opensub reveals the following observations:

- PIN length 4: Top 7 PINs share the same spots in SUBTLEXus and opensub scenarios (the last three spots in top 10 are just permuted, i.e. the top 10 contains exactly the same set of PINs in both scenarios). The high marginal success rates are caused by the frequencies of the following PINs: 8428 (probability 6.65% based on SUBTLEXus, e.g. produced from the word "that"), 9428 (4.64%, "what"), 8447 (3.76%, "this"), 4283 (3.14%, "have"), 9687 (3.04%, "your"), and 5669 (2.70%, "know").
- PIN length 5: The sets of top 10 PINs differ in just two PINs, while the first 5 spots are exactly the same in SUBTLEXus and opensub scenarios. For SUBTLEXus scenario the most frequent PINs are: 84373 (5.12%, e.g. produced from the word "there"), 74448 (3.95%, "right"), 22688 (3.54%, "about"), 84465 (2.62%, "think"), 46464 (2.07%, "going"), and 46662 (1.94%, "gonna").

We can conclude that considering uniform frequencies of dictionary words is inadequate for estimating the security of memory-only selection of dictionary PIN. The results show the deficiency of such PINs clearly – the marginal success rates are unacceptable. Moreover, it seems that this strategy is worse (in average) than strategies currently employed by users. In order to compare memory-only dictionary PINs with "common" PIN selection strategies, we present estimates of PIN metrics based on RockYou password database and iPhone unlock codes [4] in Table 2. On the other hand, the lack of digits 0 and 1 in the standard mapping ensures that these digits do not appear in the resulting PIN. Therefore the PINs that are often blacklisted, e.g. 0000, 1111 or 1234, or those the users are warned not to use, e.g. birthday or anniversary years, cannot be selected this way.

Table 2: Estimation of PIN metrics (PIN length 4) [4]

	RockYou	iPhone
H_1 (bits)	10.74	11.42
$\tilde{G}$ (bits)	11.50	11.83
$\tilde{\mu}_{0.5}$ (bits)	9.11	10.37
λ_6 (%)	12.29	12.39

We explore few possibilities for improving memory-only dictionary PINs in the following sections.

2.2 Blacklisting

Blacklisting is a common method for improving the security of user-selected PINs. Even if not strictly enforced (by forbidding the selection of some PINs), at least there are recommendations what PINs a user should not choose, e.g. see [15]:

> "Select a PIN that cannot be easily guessed (i.e., do not use birth date, partial account numbers, sequential numbers like 1234, or repeated values such as 1111)."

There are two possibilities for blacklisting in dictionary PIN scenario: first, blacklisting the most frequent words; and second, blacklisting the most frequent PINs. The PIN blacklisting is easier to enforce in practice, and the values of security metrics are comparable for both approaches. Figure 2 shows the entropy and the marginal success rate (λ_6) for PIN blacklisting (based on SUBTLELXus) ranging from 0 to 100 blacklisted PINs. The results show only a moderate improvement in security metrics, therefore blacklisting alone is not a satisfactory method for improving memory-only dictionary PINs.

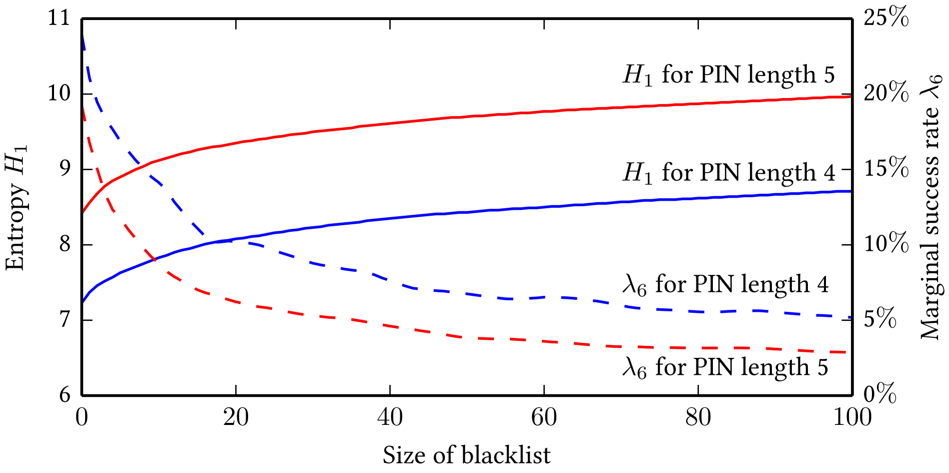


Figure 2: The effect of PIN blacklisting on entropy and marginal success rate

2.3 Modifications of PIN Construction

In order to improve the security metrics of resulting PIN, some natural modifications to basic translation of dictionary word to PIN were proposed in [12]. We evaluate these modifications when applied to our "frequency-aware" experiments:

- Stretched mapping – in order to cover digits 0 and 1, we can stretch the standard mapping. Our estimates for this modification use the following mapping: a, b ↦ 1; c, d ↦ 2; e, f ↦ 3; g, h, i ↦ 4; j, k, l ↦ 5; m, n ↦ 6; o, p, q ↦ 7; r, s, t ↦ 8; u, v, w ↦ 9; x, y, z ↦ 0.

- Prefix – instead of taking just words with the desired PIN length, i.e. n, we use any word with the length greater or equal to n and we use its prefix for translation to PIN.

- Morphing – the standard word to PIN translation is enriched by random change of one character. Assuming that user can choose a random position in a word/PIN and a random digit, the resulting PIN is formed by replacing this position by the chosen digit. For example "this" can be translated to "t1is"/8147, "0his"/0447, "thi9"/8449, etc. Certainly, this method is more demanding than the straightforward use of dictionary words. Our estimates assume uniform distribution of positions and digits for this method.

The results for all above methods are presented in Table 3. The stretched mapping yields no improvement at all – the most frequent PINs changed their values, but their frequencies remained almost unchanged. The prefix method

Table 3: Modifications of PIN constructions – results based on SUBTLEXus

	Stretched map.		Prefix		Morphing	
	$n = 4$	$n = 5$	$n = 4$	$n = 5$	$n = 4$	$n = 5$
H_1 (bits)	7.28	8.43	9.03	10.36	11.08	12.96
$\tilde{G}$ (bits)	7.28	8.67	8.89	10.39	10.73	12.84
$\tilde{\mu}_{0.5}$ (bits)	5.52	6.58	7.56	8.77	9.88	11.53
λ_6 (%)	24.04	19.31	11.30	8.01	2.77	2.17

offers a moderate improvement in security metrics. Obviously, the morphing is the most successful approach by a wide margin.

Comparing these results with the estimates from Table 2, we can notice that the prefix method for dictionary PINs offers slightly better marginal success rate but worse entropy, guesswork and marginal guesswork. An interesting observation is that the morphing offers much better marginal success rate while keeping other security metrics comparable to real-word estimates from Table 2.

Interestingly, the prefix method and the morphing yield better results than the estimates of PIN entropy by NIST [6]: 9 and 10 bits for PIN lengths 4 and 5, respectively. On the other hand, plain memory-only dictionary PINs offer less entropy than these estimates.

2.4 Blacklisting the Prefix and the Morphing Methods

We expect that blacklisting of the most frequent PINs can further improve the security metrics of promising methods from the previous section (i.e. prefix and morphing methods). Indeed, our experiments confirm this expectation. Table 4 shows the values of the entropy and the marginal success rate for various sizes of the blacklist (0, 10, and 20).

Table 4: Combination of PIN blacklist and the prefix/morphing method

		Prefix		Morphing	
	blacklist	$n = 4$	$n = 5$	$n = 4$	$n = 5$
H_1 (bits)	0	9.03	10.36	11.08	12.96
	10	9.37	10.68	11.15	13.06
	20	9.53	10.82	11.19	13.09
λ_6 (%)	0	11.30	8.01	2.77	2.17
	10	6.62	4.28	1.74	0.97
	20	4.69	2.95	1.56	0.85

The blacklisting substantially improves the marginal success rate, but offers only a moderate improvement of the entropy. A disadvantage of the blacklisting is that it complicates the implementation of authentication.

2.5 Two-dictionary PINs

Many people know more than one language. In such case, it is easy to adopt a strategy where a user randomly choses a language and then (s)he selects a word for PIN construction. We expect obviously an improvement in security metrics values. In order to assess the improvement we use English and Dutch frequency dictionaries SUBTLEXus and SUBTLEXnl [13]. We use words with frequency above 1 in both dictionaries, and we assume that the user selects the dictionary with probability 1/2. The results for this two-dictionary scenario are shown in Table 5, where "basic" denotes the construction using words with the length n, "prefix" denotes the prefix method, and "prefix (BL)" denotes the combination of the prefix method with the blacklist of the length 10.

Table 5: Security metrics for two-dictionary scenario

	Basic		Prefix		Prefix (BL 10)	
	$n = 4$	$n = 5$	$n = 4$	$n = 5$	$n = 4$	$n = 5$
H_1 (bits)	7.84	9.35	9.62	11.21	9.84	11.37
$\tilde{G}$ (bits)	7.72	9.41	9.37	11.09	9.50	11.17
$\tilde{\mu}_{0.5}$ (bits)	6.24	7.63	8.27	9.68	8.55	9.85
λ_6 (%)	18.00	11.07	7.78	4.37	4.09	2.32

As expected, the results are better than results for corresponding single-dictionary scenario methods. However, even with the blacklisting the results cannot match the morphing method results for single dictionary.

3 Conclusion

We analyzed the security of memory-only selection of dictionary PINs. The results show that plain construction of dictionary PINs is unsatisfactory and more involved methods should be used for improved security metrics.

Acknowledgment

The author acknowledges support by VEGA 1/0259/13.

References

[1] David Aspinall and Mike Just. "Give me letters 2, 3 and 6!": Partial password implementations and attacks. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 126–143. Springer Berlin Heidelberg, 2013.

[2] Nick Berry. PIN analysis. DataGenetics, www.datagenetics.com/blog/september32012/index.html, 2012. Accessed: 2014-03-31.

[3] Joseph Bonneau, Mike Just, and Greg Matthews. Whats in a name? In Radu Sion, editor, *Financial Cryptography and Data Security*, volume 6052 of *Lecture Notes in Computer Science*, pages 98–113. Springer Berlin Heidelberg, 2010.

[4] Joseph Bonneau, Sren Preibusch, and Ross Anderson. A birthday present every eleven wallets? The security of customer-chosen banking PINs. In Angelos D. Keromytis, editor, *Financial Cryptography and Data Security*, volume 7397 of *Lecture Notes in Computer Science*, pages 25–40. Springer Berlin Heidelberg, 2012.

[5] Marc Brysbaert and Boris New. Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, 41(4):977–990, 2009.

[6] William E. Burr, Donna F. Dodson, Elaine M. Newton, Ray A. Perlner, W. Timothy Polk, Sarbari Gupta, and Emad A. Nabbus. NIST SP 800-63-1. Electronic Authentication Guideline. Technical report, National Institute of Standards & Technology, Gaithersburg, MD, United States, 2011.

[7] PCI Security Standards Council. Payment Card Industry PIN Security Requirements, Version 1.0. www.pcisecuritystandards.org/security_standards, 2011. Accessed: 2014-03-31.

[8] Hermit Dave. Frequency word lists. http://invokeit.wordpress.com/frequency-word-lists/, 2012. Accessed: 2014-03-31.

[9] ISO. Financial services – Personal Identification Number (PIN) management and security, Part 1: Basic principles and requirements for PINs in card-based systems. ISO 9564-1, 2011.

[10] Markus Jakobsson and Debin Liu. Bootstrapping mobile PINs using passwords. http://w2spconf.com/2011/papers/mobilePIN.pdf, 2011. Accessed: 2014-03-31.

[11] Markus Jakobsson and Debin Liu. Your password is your new PIN. In *Mobile Authentication*, SpringerBriefs in Computer Science, pages 25–36. Springer New York, 2013.

[12] Lubica Staneková and Martin Stanek. Analysis of dictionary methods for PIN selection. *Computers & Security*, 39, Part B:289 – 298, 2013.

[13] SUBTLEXnl, Database of Dutch Word Frequencies. Ghent University, http://crr.ugent.be/programs-data/subtitle-frequencies/subtlex-nl, 2010. Accessed: 2014-03-31.

[14] SUBTLEXus, Word Frequency American English. Ghent University, http://expsy.ugent.be/subtlexus/, 2009. Accessed: 2014-03-31.

[15] VISA. Issuer PIN Security Guidelines. https://usa.visa.com/download/merchants/visa-issuer-pin-security-guideline.pdf, 2010. Accessed: 2014-03-31.

[16] Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 162–175, New York, NY, USA, 2010. ACM.

Martin Stanek is an Associate Professor in the Department of Computer Science, Comenius University. He received his PhD. in computer science from Comenius University. His research interests include cryptography and information security.

Shielding the Grid World: An Overview

Christos Chrysoulas
(Corresponding author: Christos Chrysoulas)

Electrical & Computer Engineering Deapartment, University of Patras
(Email: cchrys@ece.upatras.gr)

(Received Mar. 20, 2014; revised and accepted Apri. 30, 2014)

Abstract

Continues research and development efforts within the Grid community have produced protocols, services, and tools that address the challenges arising when we seek to build scalable virtual organizations (VOs). The technologies that have evolved from the Grid community include security solutions that support management of credentials and policies when computations span multiple institutions; resource management protocols and services that support secure remote access to computing and data resources and the co-allocation of multiple resources; information query protocols and services that provide configuration and status information about resources, organizations, and services; and data management services that locate and transport datasets between storage systems and applications.

Keywords: Grid computing, Security, Web services

1 Introduction

Grid computing [1-2] is the aggregation of networked connected computers to form a large-scale distributed system used to tackle complex problems. By spreading the workload across a large number of computers, Grid computing offers enormous computational, storage, and bandwidth resources that would otherwise be far too expensive to attain within traditional supercomputers. High-performance computational Grids involve heterogeneous collections of computers that may reside in different administrative domains, run different software, be subject to different access control policies, and be connected by networks with widely varying performance characteristics. The security of these environments requires specialized Grid-enabled tools that hide the mundane aspects of the heterogeneous Grid environment without compromising performance.

These tools are possible to make use of existing solutions or can implement completely new models. In either case, research is required to understand the utility of different approaches and the techniques that may be used to implement these approaches in different environments. Grid computing is distinguished from conventional distributed computing by its focus on large-scale pervasive resource sharing, virtual and pluggable high-performance orientation. The electrical power grid’s pervasiveness and reliability inspired computer scientists in the mid-1990s to explore the design and development of a new infrastructure, computational power grids for network computing. The real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. The sharing is not just file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of resource brokering strategies emerging in industry, science, and engineering.

The heterogeneous nature of resources and their differing security policies are complicated and complex in the security schemes of a Grid computing environment. These computing resources are hosted in different security domains and heterogeneous platforms. The major security requirement for the Grid is centered on the dynamic configuration of its security services [3], such as data integrity, confidentiality, and information privacy in potentially volatile environments.

The rest of the document is structured as follows. Section 2 presents the Grid security model. Section 3 and Section 4 discusses security binding and security associations respectively. Section 5 presents authentication in Grid systems. Section 6 gives an insight on available security standards. Security in web service is provided in Section 7 while Section 8 analyzes the grid security infrastructure. Finally Section 9 concludes the document.

2 Grid Security Model

Web services (WS) [4] is an emerging architecture that has the ability to deliver integrated, interoperable solutions. Ensuring a) integrity, b) confidentiality, and c) security of Web services through the use of a comprehensive security model is critical, both for organizations and their customers. The secure “transactions” between virtual organizations demands interoperable solutions using heterogeneous systems. For instance, the secure messaging model proposed by the

Web Services Security roadmap [5] document supports both public key infrastructure (PKI) and Kerberos mechanisms as particular embodiments of a more general facility that can be extended to support additional security mechanisms. The security of a Grid environment must take into account the security of various aspects involved in a Grid service invocation. This is depicted in Figure 1.

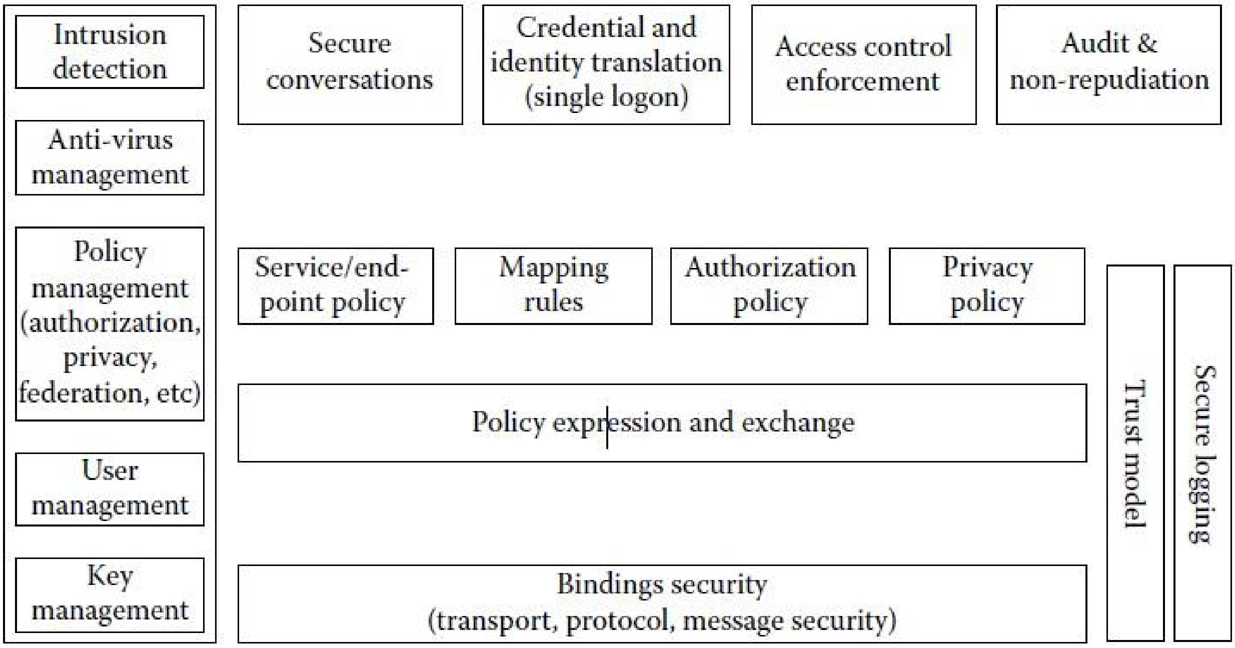

Intrusion detection
Anti-virus management
Policy management (authorization, privacy, federation, etc)
User management
Key management
Secure conversations
Credential and identity translation (single logon)
Access control enforcement
Audit & non-repudiation
Service/end-point policy
Mapping rules
Authorization policy
Privacy policy
Policy expression and exchange
Bindings security (transport, protocol, message security)
Trust model
Secure logging

Figure 1: Grid security model components

3 Binding Security

The set of bindings to be considered includes SOAP (SOAP/HTTP, SOAP over a message queue or SOAP over any other protocol) and IIOP bindings. The security of a binding is based on the security characteristics of the associated protocol and message format. If new protocols or message formats are introduced, care should be taken to address security requirements in those bindings so that, at a minimum, suitable authentication, integrity, and confidentiality can be achieved.

HTTP is an important protocol to consider because of its transparency to firewalls and wide adoption. In the case of bindings over HTTP, requests can be sent over SSL (i.e., https) and thus SSL can provide authentication, integrity, and confidentiality. However, SSL ensures these qualities of service only among participating SSL connection end points. If a request needs to traverse multiple intermediaries (firewalls, proxies, etc.), then end-to-end security needs to be enforced at a layer above the SSL protocol.

In the case of SOAP messages, security information can be carried in the SOAP message itself in the form of security tokens defined in the WS-Security specification [5]. SOAP messages can also be integrity and confidentiality protected using XML Digital Signature and XML Encryption support, respectively. Signature and encryption bindings defined in WS-Security can be used for this purpose.

Web services can be accessed over IIOP when the service implementation is based on CORBA. In the case of IIOP, the security of the message exchange can be achieved by using the Common Secure Interoperability specification, version 2 (CSIv2). This specification is also adopted in J2EE.

In addition to, binding-level security requirements, network security solutions (e.g., firewalls, IPSec, VPN, DNSSEC, etc.) remain useful components for securing a Grid environment. Firewalls can continue to enforce boundary access rules between domains and other network-level security solutions can continue to be deployed in intradomain environments. Grid services deployment can take the topology into consideration when defining security policies. At the same time, deployment assumptions may be surfaced as policies attached to firewalls and network architecture.

The Grid security model must be able to leverage security capabilities of any of these underlying protocols or message formats. For example, in the case of SOAP over HTTP requests, one can use WS-Security for end-to-end security functionality, HTTPs for point-to-point security, and SSL, TLS, or IPSec for other purposes. Security requirements for a given Web service access will be specified and honored based on the set of policies associated with the participating end points. For example, a policy associated with a Web service can specify that it expects SOAP messages to be signed and encrypted.

Thus, service requestors accessing that service would be required to use WS-Security to secure their SOAP requests. Addressing the security of the service bindings will address the requirements related to integrity and confidentiality of messages, achieving delegation facilities, and facilitating firewall traversal.

4 Secure Associations

A service requester and a service provider are likely to exchange more messages and submit requests subsequent to an initial request. In order for messages to be securely exchanged, policy may require the service requester and service provider to authenticate each other. In that case, a mechanism is required so that they can perform authentication and establish a security context.

This security context can be used to protect exchange of subsequent messages. As an added benefit, using the established security context will improve the performance of secure message exchanges. The period of time over which a context is reused is considered a session or association between the interacting end points. Security context establishment and maintenance should be based on a Web service context (to be) defined within Web or Grid service specifications.

The notion of a context is tightly coupled with the bindings. Many existing protocols (e.g., IPSEC, SSL, IIOP) and mechanisms (e.g., Kerberos) already support secure association contexts. For example, in the case of IIOP, context establishment is based on the CSIv2 specification. In the case of SOAP, the context can be carried and secured as part of the SOAP messages.

WS-Secure Conversation will describe how a Web service can authenticate service requestor messages, how service requestors can authenticate service providers, and how to establish mutually authenticated security contexts. WS-Secure Conversation will be designed to operate at the SOAP message layer so that the messages may traverse a variety of transports and intermediaries. Therefore, in the case of SOAP bindings, the Grid security model should adopt WS-Secure Conversation to establish security contexts and exchange messages securely. Alternatively, depending on the constraints of a VO's other technologies (e.g., SASL, BEEP, etc.) may be used. Therefore, the mechanism used to establish security contexts between end points will be based on the bindings used as well as the policy associated with the end points.

Facilitating secure association is required to establish the identity of a requestor to the service provider (and vice versa) so that the service provider (and service requestor) can satisfy the requirements to authenticate the identity on the other end and then enforce authorization and privacy policies based on the established identity. The identities of the requestor and service provider are required for auditing purposes, so that audit logs will contain information about accessing identity.

5 Authentication in Grid Systems

A computational Grid has been defined as “a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.” Typically, Grid resources are provided by various organizations and are used by people from diverse sets of organizations. A Grid may support (or define) a single virtual organization or it may be used by more than one virtual organization. Individual pieces of hardware may be used in more than one Grid, and people may be members of more than one virtual organization. The different resources in a Grid may have different access policies, including how they authenticate and authorize users. If no common or overlapping authorizations exist among the resources, however, they do not form a usable Grid.

Users, hosts, and services need to be able to authenticate themselves in the Grid environment. Experience in using Grids for remote computations has demonstrated the need for unattended user authentication in addition to interactive authentication. Unattended authentication of users is needed when a user is making frequent requests to remote servers and does not want to repeatedly type in a pass phrase and when a long-running job may need to authenticate itself after the user has left. Servers specific to a single host mays need to be started at system boot time and run with their own or the host's identity. Some services may need to be started periodically on many different hosts and be able to authenticate themselves with a known identity.

Basically, authentication between two entities on remote Grid nodes means that each party establishes a level of trust in the identity of the other party. In practical use an authentication protocol sets up a secure communication channel between the authenticated parties, so that subsequent messages can be sent without repeated authentication steps, although it is possible to authenticate every message. The identity of an entity is typically some token or name that uniquely identifies the entity.

6 Relationship to Security Standards

The Grid environment and technologies address seamless integration of services with existing resources and core application assets. As discussed in the Grid Security Model section, the Grid security model is a framework that is extensible, flexible, and maximizes existing investments in security infrastructure. It allows the use of existing technologies such as X.509 public key certificates, Kerberos shared-secret tickets, and even password digests.

Therefore, it is important for the security architecture to adopt, embrace, and support existing standards where relevant. Given Grid services are based on Web services, Grid security model will embrace and extend the Web services security standards proposed under the WS Security roadmap [5].

Specifically, given that OGSA is a service-oriented architecture based on Web services (i.e., WSDL-based service definitions), the OGSA security model needs to be consistent with Web services security model. The Web services security roadmap provides a layered approach to address Web services, and also defines SOAP security bindings. Figure 2 illustrates the layering of security technology and standards that exist today and how they fit into the Grid security model.

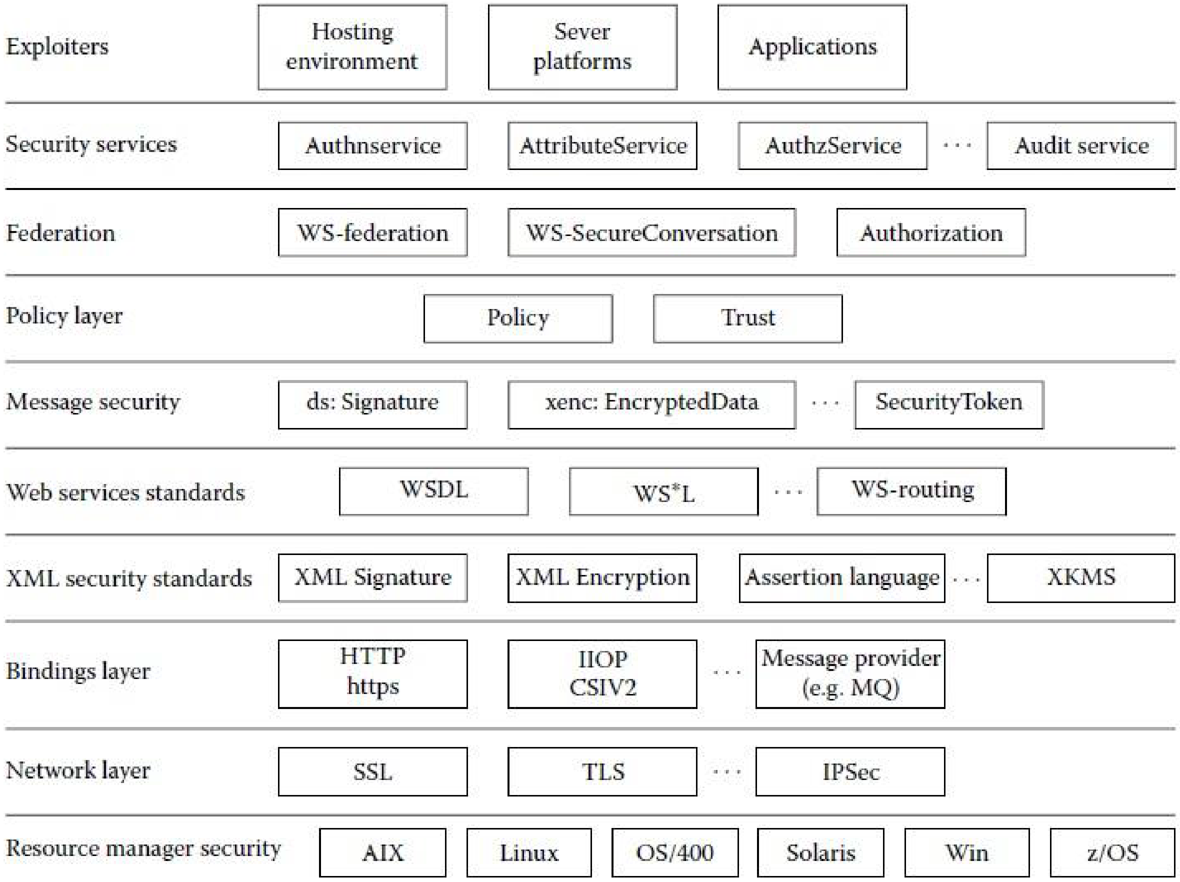


Figure 2: Building blocks for grid security architecture

7 What about Security in Web Services

Web services offer an interoperable framework for stateless, message-based, and loosely coupled interaction between software entities. These entities can be spread across different companies and organizations, can be implemented on different platforms, and can reside in different computing infrastructures. Web services expose functionality via XML messages, which are exchanged through the SOAP protocol. The interface of a Web service is described in detail in an XML document using the “Web Service Description Language” (WSDL).

In order to provide security, reliability, transaction abilities, and other features, additional specifications exist on top of the XML/SOAP stack. The creation of the specifications is a cross-industry effort, with the participation of standardization bodies such as W3C and OASIS. A key element in the Web services specifications is the so-called combinability. Web services specifications are being created in such a way that they are mostly independent of each other; however, they can be combined to achieve more powerful and complex solutions. In this section we describe some individual specifications, specifically focusing on those dealing with secure and reliable transactions.

8 Grid Security Infrastructure

In grid computing environments, the mutual authentication and information service are serious issues. Before their applications are running, the users need to choose hosts based on security, availability, and many other aspects. The GSI (Grid Security Infrastructure) is designed as one very important part of the Globus grid toolkit. The GSI uses public key cryptography (also known as asymmetric cryptography) as the basis for its functionality.

The primary motivations behind the GSI are the need for secure communication (authenticated and perhaps confidential) between elements of a computational Grid and the need to support security across organizational boundaries, thus prohibiting a centrally managed security system. Finally, a fundamental issue is to support “single sign-on” for users of the Grid, including delegation of credentials for computations that involve multiple resources and/or sites.

GSI, which is designed to solve the security in the Globus system, is based on RSA encystations algorithm and employs a standard (X.509v3) for encoding credentials for security principals, and thus enables secure authentication and communication over open network. At the same time GSI enables Interoperable with local security solutions without changing anything. The Grid Security Infrastructure (GSI) is a specific implementation of an OGSA-based Grid security architecture that include as part of the Globus Toolkit Version 5 (GT5).

9 Conclusions

Grid computing has really distinct security themes in comparison to other traditional computing systems. The most important security problems [7] include, but not limited to the following ones: a) Impact on Local Host: Grid computing involves running an alien code in the host system. This external code can hamper jobs running locally, and compromise local data security, b) Vulnerable Hosts: Clients using the grid remain in danger from the local hosts. The major vulnerabilities include the local hosts shutting down resulting in denial of service, viruses, or other malware in the local host affecting the entire process, and local hosts compromising client data integrity and confidentiality, c) Interception: One major security risk with grid computing is an attacker intercepting the resources and data in the grid. The attack can take various forms such as a distributed denial-of-service (DDOS) attack, and d) Packet Losses: Interruption of nodes during the routing process to send packets from source to destination decreases total packet delivery and loss or corruption of data.

The extent of security risks when using smart grid depends on the intellectual property put in the hosted environment

Most of the security issues regarding the a grid system can be solved with the use of a monitoring agency that deals with: a) the monitoring of the resource, b) the creation, management and negotiation of trust among the different actors in the grid and c) the establishment of an authorization system in order to authorize user access to specific resources.

The user running an application on a remote machine in the grid-computing network requires assurance of the machine retaining its integrity, to ensure that proprietary application remains safe. The local host requires a similar assurance regarding the client data and processes that run on the host. While the safeguards of a traditional system aim at protecting the system and data from its users, the security orientation of grid systems need to go a step ahead and also protect applications and data from the system where the computation takes place.

Grid computing security requires strong authentication and restrictions on local execution from remote systems. Some of the solutions can be: a) Secure grid communication using public key cryptography, b) Authentication or verifying identity of the participant, c) Single sign on in order to reduce the number of times a user needs to enter password, d) Filtering and auditing of data, and e) Erasing of data after use.

The gains resulting from grid computing already surpass the security risks, and since security problems find nowadays easy and realistic solutions, grid computing is becoming more and more a commonplace.

References

[1] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the Grid enabling scalable virtual organizations". International Journal of Supercomputer Applications, vol. 15, pp.200-222, 2001.

[2] I. Foster, C. Kesselman, J. M. Nick and S. Tuecke, "The physiology of the grid, an open grid services architecture for distributed systems integration", Open Grid Service Infrastructure WG, Global Grid Forum, 2002. www.globus.org/alliance/.

[3] Y. Xiao, "State-of-the-art security in grid computing". in Security in distributed, grid, mobile and pervasive computing. Auerbach, New York, pp.207-236, 2007.

[4] N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, F. Siebenlist,V.Welch, S. Tuecke, and I. Foster. "Security architecture for open grid Services", GWD-I (draft-ggf-ogsa-sec-arch-01), July, 2002.

[5] "Security in a Web Services World: A Proposed Architecture and Roadmap", http://www-106.ibm.com/developerworks/library/ws-secmap/.

[6] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, and C. Kesselman. "Grid Service Specification". Draft 2, 6/13/2002, http://www.globus.org.

[7] "A Look at Security Problems with Grid Computing", Accessed 22/4/2014 http://www.brighthub.com/environment/green-computing/articles/94587.aspx

Christos Chrysoulas received his PhD in Electrical & Computer Engineering from the Electrical & Computer Engineering Dept., University of Patras, Greece in 2009. He received his Diploma in Electrical & Computer Engineering from the Electrical & Computer Engineering Dept., University of Patras, Greece in 2003. Since 2009, he is working as an Adjunct Professor, in the Technological Educational Institute of Patras, HELLAS. He is teaching in the Informatics & MM Department and in Museology and Museography Department. From July 2013 he is with the CISTER Research Center as a Research Associate. His research interests include Computer Networks, High Performance Communication Subsystems Architecture and Implementation, Wireless Networks, New Generation Networks Architectures, Resource Management and Dynamic Service Deployment in New Generation Networks and Communication Networks, Grid Architecture, Semantics. During the last year he is intensively working in the Smart Grid area, as a system architect expert. He is also interested in Cyber physical systems. Christos Chrysoulas has published more than 10 technical papers in these areas. He has also participated as Senior Engineer in European Research Projects.

Efficient X-box Mapping in Stego-image Using Four-bit Concatenation

Manoharan Shobana

Department of Electronics and Communication Engineering
Karpagam College of Engineering,Coimbatore
(Email: divyashobana.m@gmail.com)

(Received Mar. 20, 2014; revised and accepted Apr. 29, 2014)

Abstract

The approach of hiding the secret information in the covert image is called image steganography. Its goods like strength and defense is broadly used for covert communication, where the secret message is used to hide in the covert image. The least-significant-bit (LSB)-based method is a popular type of steganographic algorithms in the spatial domain. In image steganography the LSB based method is used extensively for hiding the secret data due to its simplicity and high embedding capability. In an existing method for hiding the secret data in the LSB substitution method four X-boxes are used to hide two secret bits in each pixel of the cover image. In this method using two X-box, 4 bits of secret message is hidden. This enhances the security of the secret message, with no movement of the PSNR value of the cover image.

Keywords: Decoding, encoding, image steganography, network security, X-boxes

1 Introduction

Due to copyright violation, counterfeiting, forgery, and fraud, transmitting the digital data in open networks such as the Internet is not consistently safe. Thus, for protecting the secret data many approaches are forward for protecting essential digital data [3]. Cryptographic methods are used for transmitting the secret data encrypted by cryptosystems and used for secret communication. The meaningless form of the encrypted data may draw the thought of hackers. This confidential data can be protected by using information hiding techniques such as watermarking and steganography, which hides the secret information into a cover object and create an embedded object. Figure 1 shows a basic steganography model.

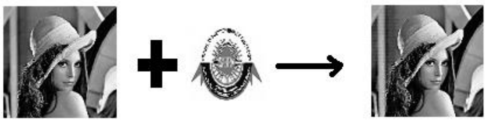


Figure 1: Basic steganography model

Watermarking is used for screen monitoring, copyright defense, tracking transaction and similar activities. In contrast, steganography is used primarily for secret communications. This method invisibly alters a cover object to mask a covert message. Thus, it can hide the very existence of concealed communications. For further protection, a cryptographic technique is used before embedding process [2].

Image steganography techniques are further classified into Image Domain and Transform Domain [6]. In image domain embedding process is done by using its pixel intensity, this manipulates to hide the secret data in more significant areas. Image domain is also known as spatial domain is more robust. Alternately transform domain, uses transformed embedding data to hide in the secret communication and independent the image format and the embedded message. Transform domain is otherwise known as frequency domain is the most secure method [2].

In image steganography LSB substitution method that is the least significant bit (LSB) placing is a common, simple approach for hiding covert information in a cover image. In this method, some or all of the bits in the covert image are changed to a bit of the secret message [4, 5].

2 Existing Method

In previous method mapping based image-image steganography is developed. Only grayscale image is used here for both secret message and covert image. The gray scale secret image is converted to binary where each pixel has 8-bit value.Mapping method is nothing but assigning encoded value for the pixels in the secret image using four kinds of X-boxes such as b1, b2, b3, b4.The 8-bit value is further divided into four equal parts of two bits. Each two bits will get equivalent value from X – box in the sequence of the first part from b1, second part from b2 and so on.Then the new values get embedded in the pixels of cover image.As a result only two bits of message gets embed in each pixel of cover image [1].

3 Proposed Method

Step 1 (procedure for filling values in X-box):
This mapping method used two X-boxes which are capable of holding values from 0 to 15 which is explained in Figure 2.

00	01	10	11	
0	1	3	2	0
7	6	4	5	1

B1

00	01	10	11	
15	14	12	13	0
8	9	11	10	1

B2

Figure 2: Mapping using X-box

The steps for inserting values in the X-box are described below using an example:

7 => 0111 = 100
0 XOR 1 = 1
1 XOR 1 = 0
1 XOR 1 = 0.

Now take a look at B1 where 7 are placed in the 2nd row and 1st column. This procedure is similar for the X-box B2. Main purpose of using X-box is decoding the pixels of secret image.

Step 2 (Message encoding):
Convert the Secret image into its binary values each of 8-bit length. Split each pixel into two equal parts convert that into 3-bit using XOR operation which is mentioned above. Let us see with an example:

$(123)_{10} = (01111011)_2$

Divide the above values into two equal parts:

$$\underbrace{0111}_{\mathbf{b1}}\underbrace{1011}_{\mathbf{b2}}$$

Using XOR operation b1 and b2 is converted to 100, 110 respectively.

Step 3 (Mapping):
Now we just map the values of b1, b2 from the X-box. First, we take b1 = 100. Then we search the value of 1st row and 00th column of the X-I box; After mapping we get the value (7)10 = (0111)2; Similarly, we get mapping values for b2 = 11.

Step 4 (Embedding):
After getting the new mapping values we insert these values into the cover image. We placed these values into the 4 bit LSB of cover image sequentially (See Figure 3). First, we take the pixels one by one from the cover image. The 4 LSB bits are replaced by 7, 11 sequentially Here the message bits get embedded in the sequential manner:

$(200)_{10}$ => $(11001000)_2$
$(215)_{10}$ => $(11010111)_2$

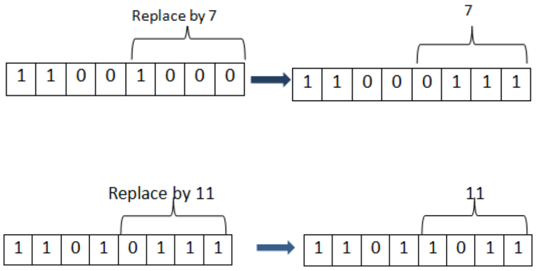


Figure 3: Bits embedding in the cover image

As a result, we get two new stego pixels which are given below:

$(200)_{10}$ => $(196)_{10}$
$(215)_{10}$ => $(219)_{10}$

This process will continue until all secret bits get embedded in the given cover image. The following are encoding procedures:

Input: A grayscale image of 128x128 sizes
Output: A stego-image with secret message of 64x64 sizes
1. Convert the image into binary format.
2. Divide each pixel of cover image into 2 parts each of length 4-bit.
3. Reduce the each resultant 4-bit into 3-bit using XOR operation.
For example: 1001 => (1 XOR 1 = 0, 0 XOR 0 = 0, 0 XOR 1 = 1) = 001.
4. Retrieve the corresponding value for these two 3-bit of the X-box.
5. Insert each new value into its corresponding pixel's LSB position.
6. Thus the Stego image has been obtained.

Step 5 (Extraction method):
After getting stego-image its value is converted to a binary value. In that extract all four LSB and perform the consecutive XOR operation to convert that all 4-bit to 3-bit.Then get equivalent values for all 3-bit from both B1 and B2 box alternatively. Convert all the decimal values to binary in the size of 4-bit.Concatenate the successive two 4-bit to form 8-bit. Thus the Stego pixel has been arrived to get the secret image. The concatenation of four bits is given below:

$$\overbrace{\underbrace{0111}_{7}\,\underbrace{1011}_{11}}^{123}$$

The following are decoding procedures:

Input: A grayscale Stego image of 128x128 sizes
Output: Secret image (64x64)
1. Convert the Stego image into binary format.
2. Extract LSB of 4-bit from the pixels.
3. Convert all the 4-bit into 3-bit using XOR operation.
4. Get all the equivalent 4 –bit values from two X-boxes.
5. Concatenate 4-bit values to 8 (Stego pixel).
6. Arrange all stego pixels to get the full secret image.

4 Experimental Results

To evaluate the efficiency of the X-boxes encoding, two parameters are used, that is PSNR and MSE. Here we took 128 x 128 images as cover object (See Figure 4) and 64x64 images as secret data (See Figure 5). After hiding the secret image in the cover image the PSNR and MSE were calculated using the following equations:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

$$PSNR = 10\log_{10}\left(\frac{R^2}{MSE}\right),$$

where R is a maximum disturbance in the resultant image. Table 1 shows performance measures of new stego image.

Table 1: Performance measures of new stego image

Image	PSNR	MSE	Bits per pixel	Maximum embedding capacity
Lena	36.6268	14.1383	4	50%
Gandhi	36.4388	14.7640	4	50%
Temple	36.5729	14.3148	4	50%
Baboon	36.4001	14.3148	4	50%

Figure 4: Original images

Figure 5: Stego images

5 Conclusions

In this approach the number of secret bits in each pixel is raised up to 4 bits, where in previous work only 2 bits are used. Thus the embedding capacity is increased and high value of PSNR value is achieved. This design uses only two Xbox instead of using four Xbox when compared to existing methods. Another advantage is it uses mapping based steganography for security and high image quality, thus without knowledge of the Xbox values the secret image cannot be retrieved correctly. This approach can be further enhanced by using color images instead of using grayscale image.

References

[1] Amitava Nag, Saswati Ghosh, Sushanta Biswas, Debasree Sarkar, Partha Pratim Sarkar,”An Image Steganography Technique using X-Box Mapping,” *in the proceedings of the International Conference On Advances In Engineering, Science And Management (ICAESM 2012),*pp.709-713,Nagapattinam,India,March 2012.

[2] C. Chang, T. D. Kieu,”A reversible data hiding scheme using complementary embedding strategy,” *Information Sciences*, vol. 180, no. 16, pp.3045-3058, 2010.

[3] A. D. Ker, “Steganalysis of LSB matching in grayscale images,” *IEEE Signal Processing Letters,* vol. 12. no. 6, pp.441-444, 2005.

[4] W. Luo, F. Huang, J. Huang, “Edge adaptive image steganography based on LSB matching revisited,” *IEEE Transaction on Information Forensics Security*, vol. 5, no. 2, pp.201-214, 2010.

[5] J. Mielikainen, “LSB matching revisited,” *IEEE Signal Processing Letters*, vol. 13, no. 5, pp.285-287, 2006.

[6] M. Shobana, R. Manikandan, “Efficient method for hiding data by pixel intensity,” *International Journal of Engineering and Technology,* vol. 5, no. 1, pp.75-81, 2013.

M.Shobana is currently working as a Assistant Professor, Karpagam College of Engineering, Coimbatore, India. She obtained B. Tech (Computer and science Engineering) in SASTRA University and then finished M. Tech (VLSI design) in SASTRA university Thanjavur, Tamil Nadu, India. She is interested in the area of Steganography and Network security.

Increase the Performance of Mobile Smartphones using Partition and Migration of Mobile Applications to Cloud Computing

Diaa Salama AbdElminaam[1], Hatem M. Abdul Kader[2], Mohie M. Hadhoud[3], Salah M El-Sayed[4]
(Corresponding author: Hatem M. Abdul Kader)

Information Systems Department, Faculty of Computers and Informatics, Banha University[1]
Diaa.salama@fci.bu.edu.eg
Information Systems Department, Faculty of Computers and Informatics, Menofyia University[2]
hatem6803@yahoo.com
Information Technology Department, Faculty of Computers and Informatics, Menofyia University[3]
mmhadhoud@yahoo.com
Scientific Computing Department, Faculty of Computers and Informatics, Banha University[4]
ms4elsayed@fci.bu.edu.eg

(Received Dec. 20, 2013; revised and accepted Feb. 17, 2014)

Abstract

With the increasing use of smartphones devices, mobile applications with richer functionalities are becoming ubiquitous but mobile devices are limited by their resources for computing and power consumption. Cloud the place for abundant resources. Clouds provide opportunity to do huge computations quickly and accurately so we can use cloud for mobile computations. Mobile Cloud Computing (MCC) which combines mobile computing and cloud computing, has become one of a major discussion thread in the IT world in the recent few years. We developed an architecture that uses cloud to do computations that consume resources badly on mobiles. It aims at finding the right spots in an application automatically where the execution can be partitioned and migrated to the cloud. Thus, an elastic application can augment the capabilities of a mobile device including computation power, storage, and network bandwidth, with the light of dynamic execution configuration according to device's status including CPU load, memory, and battery level. We demonstrate results of the proposed application model using data collected from one of our elastic application.

Keywords: Cloud computing, GPS, Mobile cloud computing (MCC), Offloading, Partitioning and migration

1 Introduction

Together with an explosive growth of the mobile applications and emerging of cloud computing concept, mobile cloud computing (MCC) has been introduced to be a potential technology for mobile services. MCC integrates the cloud computing into the mobile environment and overcomes obstacles related to the performance (e.g., battery life, storage, and bandwidth), environment, and security (e.g. Confidentiality, reliability and privacy) discussed in mobile computing. Mobile devices (e.g., smartphone, tablet pcs, etc) are increasingly have become the primary computing platform for many users [1, 2, 3]. Mobile users accumulate rich experience of various services from mobile applications (e.g., iPhone apps, Google apps, etc), which run on the devices and/or on remote servers via wireless networks. The rapid progress of mobile computing (MC) becomes a powerful trend in the development of IT technology as well as commerce and industry fields. However, the mobile devices are facing many challenges in their resources (e.g., battery life, storage, and bandwidth) and communications (e.g., mobility and security). The limited resources significantly impede the improvement of service qualities. Various studies have identified longer battery lifetime as the most desired feature of such systems. Cloud computing (CC) has been widely recognized as the next generation's computing infrastructure. CC offers some advantages by allowing users to use infrastructure (e.g., servers, networks, and storages), platforms, and software (e.g., application programs) provided by cloud providers (e.g., Google, Amazon, and Salesforce) at low cost. In addition, CC enables users to elastically utilize resources in an on-demand fashion [4, 5]. As a result, mobile applications can be rapidly provisioned and released with the minimal management efforts or service provider's interactions. With the explosion of mobile applications and the support of CC for a variety of services for mobile users, mobile cloud computing (MCC) is introduced as an integration of cloud computing into the mobile environment.

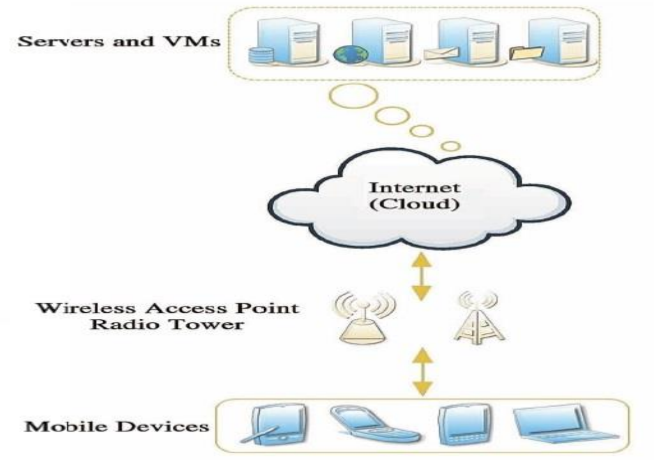


Figure 1: Mobile Cloud Computing (MCC)

Mobile cloud computing (Figure 1) brings new types of services and facilities for mobile users to take full advantages of cloud computing. This paper introduces the basic terminology of cloud computing and mobile cloud computing, its background, key technology, current research status, and its further research perspectives as well.

The rest of the paper is organized as follows. Section 2 present cloud computing definitions and basic terminology of mobile cloud computing and its architectures Following that, respectively in the next section the discussion of related work of mobile cloud computing. Following that, Section 4 presents problem definition and system model, and the description of partition cost module and the evaluation. Finally, the conclusion lies in the last section.

2 Overview of Mobile Cloud Computing

In order to help us better understanding of Mobile Cloud Computing, let's start from the two previous techniques: Mobile Computing and Cloud Computing followed by mobile cloud computing.

A. *Mobile Computing*
B. *Cloud Computing*
C. *Mobile Cloud computing*

A. *Mobile Computing*

Mobile computing exactly is described as a form of human-computer interaction by which a computer is expected to be transported during normal usage [6, 7]. Mobile computing is based on a collection of three major concepts: hardware, software and communication. The concepts of hardware can be considered as mobile devices, such as smartphone and laptop, or their mobile components. Software of mobile computing is the numerous mobile applications in the devices, such as the mobile browser, anti-virus software and games. The communication issue includes the infrastructure of mobile networks, protocols and data delivery in their use.it should have the following feature (mobility, Diversity of network conditions, frequent disconnection and consistency, Dis-symmetrical network communication, and Low reliability)

- ***Current status of mobile applications***

Several researchers, [8, 9], have identified the fundamental challenges in mobile computing. Mobile computing environments are characterized by severe resources constraints and frequent changes in operating conditions.

(1) *Offline Applications*: Fat Clients with presentation and business logic processed locally. In offline applications data downloaded from backend. Its advantages(Well Integrated, Optimized Performance, Availability :even without network connectivity).its disadvantages (No Portability, Complex

(2) *Online Applications:* Online Applications: Only presentation layer at the client. All processing done online. Assume constant connectivity with backend. Its advantages: Multiplatform, Direct and Instantaneous Accessibility to better services and its disadvantages: Excessive latency for real time responsiveness, no access to device features, sometimes difficult to maintain sessions for a long time.

B. *Cloud Computing*

[10, 11, 12] Cloud computing refers to the hardware, systems software, and applications delivered as services over the Internet (Figure 2). When a cloud is made available in a payas-you-go manner to the general public, we call it a

Public Cloud. The term Private Cloud is used when the cloud infrastructure is operated solely for a business or an organization. A composition of the two types (private and public) is called a Hybrid Cloud, where a private cloud is able to maintain high service availability by scaling up their system with externally provisioned resources from a public cloud when there are rapid workload fluctuations or hardware failures.

In general, cloud providers fall into three categories:

(1) Infrastructure as a Service (IaaS): offering web-based access to storage and computing power. The consumer does not need to manage or control the underlying cloud infrastructure but has control over the operating systems, storage, and deployed applications.

(2) Platform as a Service (PaaS): giving developers the tools to build and host web applications (e.g., APPRIO [13], software as a service provider, is built using the Force.com [14] platform while the infrastructure is provided by the Amazon Web Service [15]). The users host an environment for their applications. The users control the applications, but do not control the operating system, hardware or network infrastructure, which they are using.

(3) Software as a Service (SaaS): where the consumer uses an application, but does not control the operating system, hardware or network infrastructure. In this situation, the user steers applications over the network. Applications that are accessible from various client devices through a thin client interface such as a web browser.

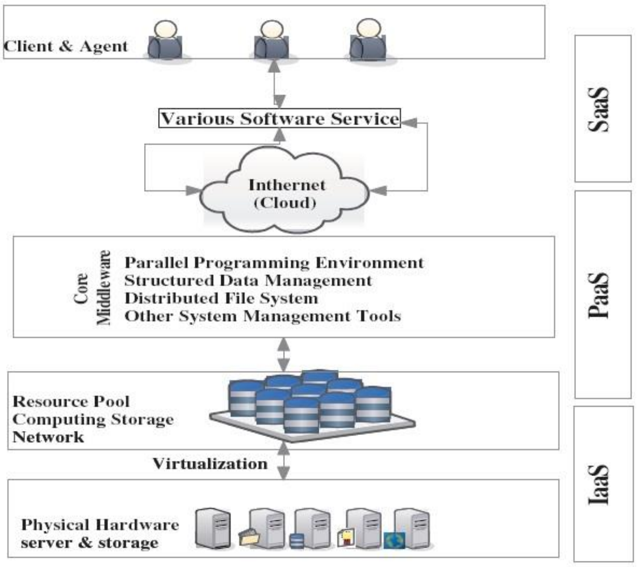


Figure 2: The framework of cloud computing

C. *Mobile Cloud computing*

There are several definitions of mobile cloud computing [16, 17], and different research refers to different concepts of the ‘mobile cloud:

The term mobile cloud computing means to Mobile Cloud computing a combination of cloud computing, wireless infrastructure, portable devices, and location based services have given rise to it. Mobile cloud computing is a model for transparent elastic augmentation of mobile device’s capability.The main objective of mobile cloud computing is to provide a convenient and rapid method for users to access and receive data from the cloud, such convenient and rapid method means accessing cloud computing resources effectively by using mobile devices.

3 Related Works

To give more prospective about the Mobile Cloud Computing, this section discusses the results obtained from other resource.

It was shown in [18] executes video games in the cloud and delivers video stream to resource-poor clients without interrupting the game experience. Many other examples where the cloud can augment mobile devices can be envisioned, e.g. virus scan, mobile file system indexing, augmented reality applications.

In [19] uses VM migration to offload part of their application workload to a resourceful server through either 3G or WiFi. CloneCloud (Figure 3) was tested using Android phones with the clones executing on a Dell desktop running Ubuntu. The system is a flexible application partitioned and execution runtime. It enables unmodified mobile applications to offload part of their execution from mobile devices onto device clones operating in a computational cloud.

It was presented in [20] 'Hyrax' for Android smartphone applications which are distributed both in terms of data and computation based on Hadoop ported to the Android platform. Hyrax (Figure 4) explores the possibility of using a cluster of mobile phones as resource providers and shows the feasibility of such a mobile cloud. As a sample application, they present 'HyraxTube'; which is a simple distributed mobile multimedia search and sharing program. The objective of HyraxTube is to allow users to search through multimedia files in terms of time, quality, and location. There are several of researches about Mobile Cloud Computing can be found in [21, 22, 23, 24]

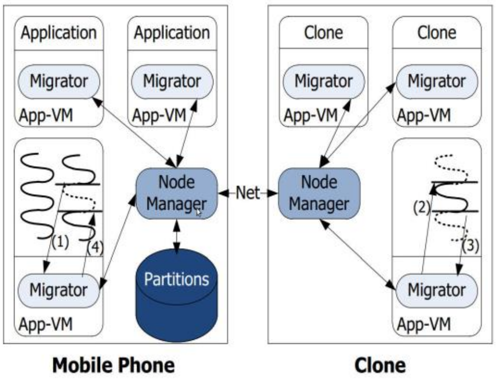


Figure 3: CloneCloud migration overview

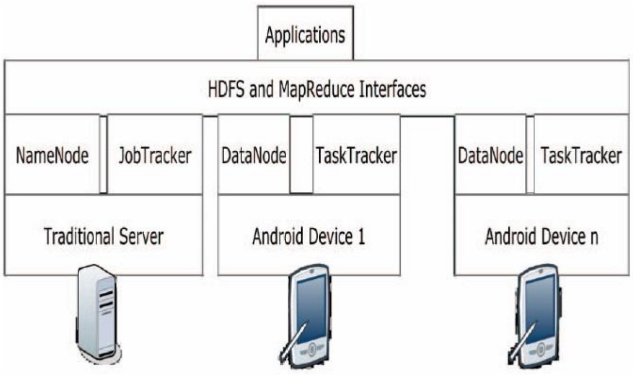


Figure 4: Hyrax infrastructure

4 Problem Definition and System Model

In this section, we present a model for application execution on the cloud-assisted mobile application platform. Application system architecture as shown in Figure 5, first, we define a mobile application profile. Then, we calculate consuming resources for application execution, including resources consumed for computation on mobile execution and a transmission computation to cloud execution. The following Sequence steps for our framework application as shown in Figure 6. We use a mobile smartphone SAMSUNG GALAXY GRAND 1.2 GHz Dual Core CPU, and Android 4 Operating Systems in which performance data is collected and tested. In the experiments, the GPS Test Performance application smartphone calculate some of GPS calculations such as distance between two points or more till 100 points using different algorithms. For our experiment, we calculate the effects of Sending computation to cloud web service and back with results and studying the Offloading computation to save energy on power consumption for smartphone mobile in case of running all processes of application on mobile or by partition and offloading processes to cloud

In the first step: Comparison is conducted using two different types of GPS mode (using mobile GPS), and using mobile network.for each type of GPS, we can get latitude and longitude for each point (it can be calculated by mobile GPS satellite or by mobile network)

In the second step: After selecting GPS mode of operations, we have to choose between manual or automatic calculation to get latitude or longitude for each point

- If automatic calculation is selected, we have to enter number of points and system get points every thirty second;
- If manual calculation is selected, we have to click to get points.

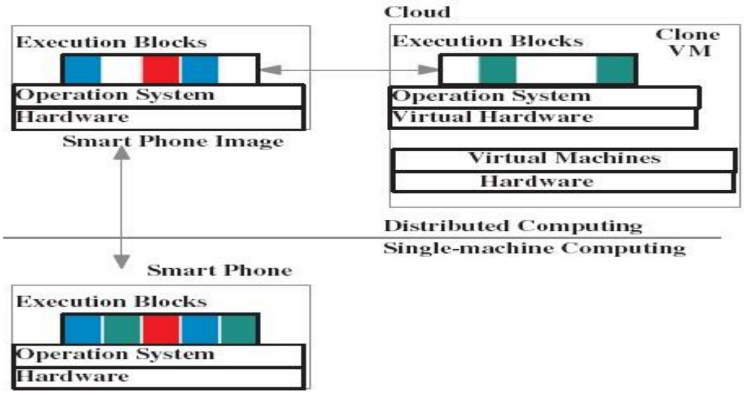


Figure 5: The application system architecture

In the third step: After selecting method to get points either manually or automatic, we have to choose between calculation way on mobile or by partition and offloading to perform part of calculation on mobile and part on cloud server.

A. **In case calculation on mobile**, calculation is conducted in case of getting points manually or automatic Mobile Application will take GPS reading and perform calculations over certain period of time.
   (1) GPS reading to determine latitude and longitude for each point either by GPS for mobile (smart phone, satellite) or from mobile network.
   (2) Then calculate the distance between two point or more using different algorithms.
   (3) The Application will perform all calculations on smart phone device and calculate the results and the consuming resources such as Memory consumed, CPU usage, Time consumed for calculation, battery consumed to perform the processes, time consumed for calculations and for getting points.

B. **In case of partition and offloading calculation on cloud and mobile**, we implement cloud clone application that enables the mobile applications developers to take decision of performing all application processes on an android mobile device or to divide the application processes to execute on mobile & cloud.
   (1) GPS reading to determine latitude and longitude for each point either by GPS for mobile (smart phone, satellite) or from mobile network (this step execute on mobile device).
   (2) Then data (longitude and latitude for each point) is transmitted to cloud server to perform calculation on cloud.
   (3) The distance between two points or more using different algorithms calculations performed on cloud the distance between two points or more using different algorithms.
   (4) The Application will perform distance calculations on cloud server and calculate the results and the consuming resources such as Memory consumed for sending and receiving results, Memory consumed for distance calculations only, Memory consumed for all process from getting points till receive results, CPU usage, Time consumed for calculation, battery consumed to perform the transmitting data, time consumed for calculations and for getting points.

4.1 Mathematical Calculation

Distance using Haversine formula: For our experiment, distance calculations between two point using the 'haversine' formula to calculate the great-circle distance between two points – that is, the shortest distance over the earth's surface. The formula assumes that the earth is a sphere, (we know that it is "ellipse" shaped) – giving an 'as-the-crow-flies' distance between the points (ignoring any hills, of course!).

Haversine Formula:

$$\begin{aligned} a &= \sin^2(\Delta\phi/2) + \cos(\phi_1).\cos(\phi_2).\sin^2(\Delta\lambda/2) \\ c &= 2.\mathrm{atan2}(\sqrt{\boxed{a}}, \sqrt{\boxed{(1-a)}}) \\ d &= R.c \end{aligned}$$

$\Delta\phi$ is latitude difference (lat2– lat1), $\Delta\lambda$ is longitude difference (long2– long1), R is earth's radius(mean radius = 6,371km).

Distance using Spherical low of Cosines: When Sinnott published the haversine formula, computational precision was limited. Nowadays, most modern computers & languages use IEEE 754 64-bit floating-point numbers, which provide 15 significant figures of precision. With this precision, the simple spherical law of cosines formula gives well-conditioned results down to distances as small as around 1 metre.

Spherical law of cosines formula:

$$d = \mathrm{acos}(\ \sin(\varphi1).\sin(\varphi2) + \cos(\varphi1).\cos(\varphi2).\cos(\Delta\lambda)\ ).R$$

Distance using Equirectangular approximation: If performance is an issue and accuracy less important, for small distances Pythagoras' theorem can be used on anequirectangular projection.

Formula:

$$x = \Delta\lambda.\cos(\varphi)$$

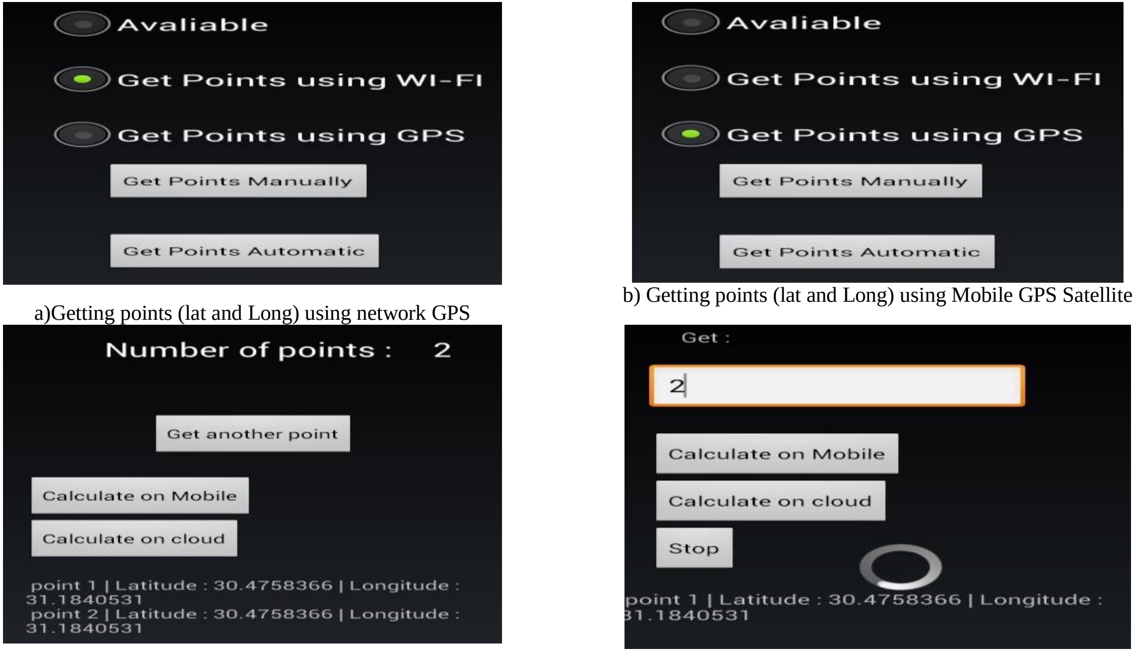
$$y = \Delta\varphi$$

$$d = R.\sqrt{(x^2 + y^2)}$$

4.2 Experiment (Manual Calculations)

4.2.1 Getting Points using GPS Satellite

Table 1 shows memory consumed in case of manual calculation in case of getting longitude and latitude for each point manually using GPS satellite for execution application on cloud web services and the for different number of points range from two points till ten points [GPS calculation on mobile smartphone and calculation moved to cloud and return results to mobile).



point 1 | Latitude : 30.4758366 | Longitude : 31.1840531
point 2 | Latitude : 30.4758366 | Longitude : 31.1840531

point 1 | Latitude : 30.4758366 | Longitude : 31.1840531

a)Getting points (lat and Long) using network GPS

b) Getting points (lat and Long) using Mobile GPS Satellite

c) Getting points manually

d)Getting Points Automatic(every 30 Sec)

Figure 6: Snapshot of elastic GPS application on Samsung Galaxy Grand

Table 1: Resources consumed for execution application on cloud web services (getting points using gps satellite)

	Calculations on Cloud			
# of points	GPS memory consumed	connection to cloud memory	Memory consumed for calculation only	Total Memory on Mobile
2	5.050582	1.343775	85.58896	6.394357
3	7.625728	0.7984205	90.784704	8.4241485
4	10.564673	1.27652	102.56821	11.841193
5	13.60835	0.045329	103.8482823	13.653679
6	12.671564	0.431785	106.742794	13.103349
7	20.9655	0.57854	106.8743	21.54404
8	25.56983	0.2678985	101.769765	25.8377285
9	20.87329	0.45698	97.894871	21.33027
10	22.5389	0.53489	108.87314	23.07379

Table 2: Resources consumed in case of partition and offloading to cloud web services (GPS Satellite)

Calculations on mobile Smartphone			
# of points	GPS Memory consumed	Calculation Memory consumed	Total Memory
2	15.027344	31.971785	46.999129
3	14.988281	12.453125	18.10156225
4	12.3945313	12.3125	15.4726563
5	16.328125	8.484375	18.44921875
6	20.3945313	16.3125	24.4726563
7	22.378906	23.34375	28.2148435
8	29.296875	24.3125	35.375
9	26.800781	28.3125	33.878906
10	23.34375	32.484375	31.46484375

Table 2 shows memory consumed for execution all application on mobile smartphones. The memory unit is bytes.

Experimental results are shown in Figures 7, 8, and 9 for different data calculations using manual method for getting longitude and latitude for each point rang from calculating distance between two points till ten points in case of distance range from approximately 100 meter till 16 kilo meter in case of all calculation done on mobile device or application is partitioned and offloading on cloud to perform distance calculation on cloud.

(Figure 7) shows the results of portioned and offloading application and getting latitude and longitude for each point on mobile smart phone and executes distance calculations on cloud. The results include the following matrices:

- Memory consumed on mobile (bytes) for GPS calculation only (getting longitude and latitude for each point).
- Memory consumed on cloud (bytes) for calculating distances between points on cloud.
- Memory consumed for send points longitudes and latitudes to web services or receive results from web services to mobile smart phone.
- Total memory consumed on mobile smartphone for calculations.

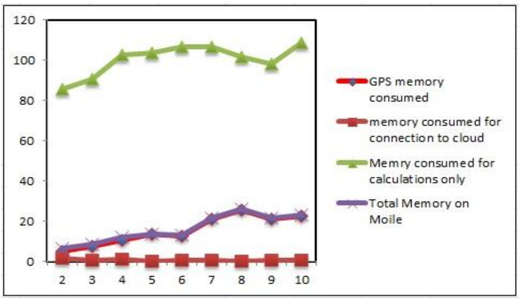


Figure 7: Memory consumed for execution application on cloud (Bytes)

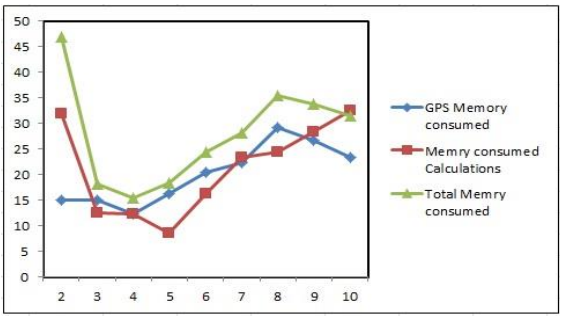


Figure 8: Memory consumed for execution application on mobile (Bytes)

Figure 8 shows the results of execute all application processes on mobile smartphone only. The results include the following matrices:

- Memory consumed in bytes for GPS calculation only (getting longitude and latitude for each point).
- Memory consumed in bytes for calculating distances between points.
- Total Memory consumed to execute application in Mega Hertz.

From the previous two figures: Figures 7 and 8. The performance of execute application on mobile or cloud in terms of memory consumed using different distance and number of points are shown in Figure 9. The total memory consumed on mobile in the case of cloud or in the case of executed all the application on mobile only.

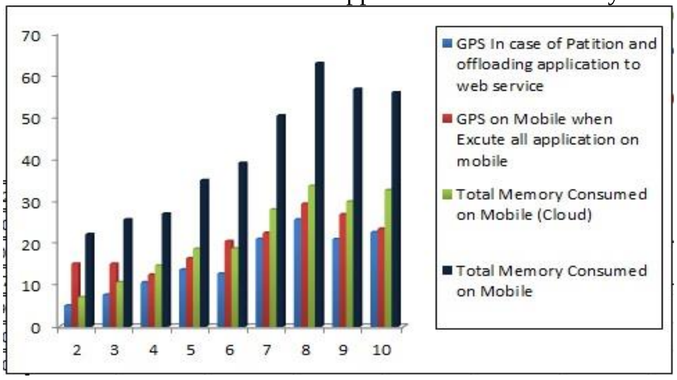


Figure 9: Total memory consumed on mobile for execution application on cloud and on mobile

Results analysis:

According to partition algorithm, most of resources consumed on mobile smartphone will decrease to approximately to half as shown in Figure 9. In case of partition and offloading application most of resources consumed on cloud and minimize the resources consumed in mobile smartphone as shown in Figures 7 and 9.

4.2.2 Getting Points using Network GPS

Table 1 shows memory consumed in case of manual calculation in case of getting longitude and latitude for each point manually using Network GPS and Table 2 shows memory consumed for execution all application on mobile smartphones. The memory unit is bytes.

Experimental results are shown in Figures 10, 11, and 12 for different data calculations using manual method for getting longitude and latitude for each point rang from calculating distance between two points till ten points in case of distance range from approximately 100 meter till 16 kilo meter in case of all calculation done on mobile device or application is partitioned and offloading on cloud to perform distance calculation on cloud.

Table 3: Resources consumed for execution application on cloud web services (Getting points using Network GPS)

Calculations on Cloud				
# of points	GPS memory consumed	connection to cloud memory	Memory consumed for calculation only	Total Memory on Mobile
2	**14.050781**	**1.109375**	**89.058594**	**15.160156**
3	**10.60156**	**0.765625**	**93.7709375**	**11.367185**
4	**1.0546875**	**1.3242188**	**103.6210937**	**2.3789063**
5	**3.609375**	**0.02734375**	**103.0632813**	**3.63671875**
6	**2.8515625**	**0.546875**	**103.4296875**	**3.3984375**
7	**25.8125**	**0.515625**	**80.5**	**26.328125**
8	**15.6645**	**0.23828125**	**91.7345**	**15.90278125**
9	**20.324219**	**0.484375**	**86.8710935**	**20.808594**
10	**9.949219**	**0.51953125**	**98.64975**	**10.46875025**

Table 4: Resources consumed in case of partition and offloading to cloud web services (Network GPS)

Mobile Calculations			
# of points	GPS Memory consumed	Calculation Memory consumed	Total Memory
2	**15.027344**	**10.84375**	**25.871094**
3	**14.988281**	**13.3127304**	**28.3010114**
4	**2.3945313**	**15.56688**	**17.9614113**
5	**6.328125**	**16.9151816**	**23.2433066**
6	**2.3945313**	**21.501**	**23.8955313**
7	**37.378906**	**17.818392**	**55.197298**
8	**9.296875**	**25.515936**	**34.812811**
9	**26.800781**	**21.91938**	**48.720161**
10	**19.34375**	**29.1054**	**48.44915**

Figure 10 shows the results of portioned and offloading application and getting latitude and longitude for each point on mobile smart phone and executes distance calculations on cloud. The results include the following matrices:

- Memory consumed on mobile (bytes) for GPS calculation only (getting longitude and latitude for each point).
- Memory consumed on cloud (bytes) for calculating distances between points on cloud.
- Memory consumed for sending points longitudes and latitudes to web services or receive results from web services to mobile smart phone.
- Total memory consumed on mobile smartphone for calculations.

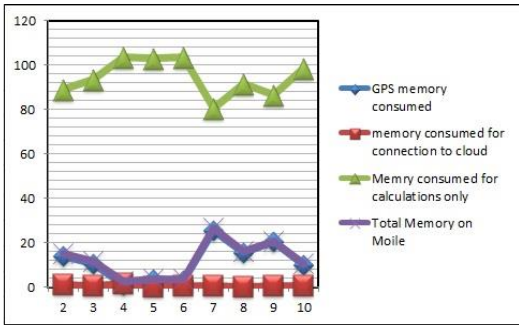


Figure 10: Memory consumed for execution application on cloud (Bytes)

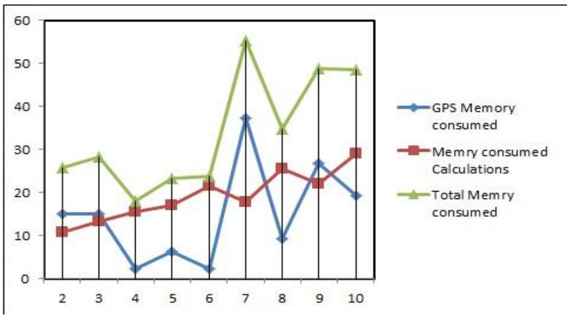


Figure 11: Memory consumed for execution application on Mobile (Bytes)

Figure 11 shows the results of execute all application processes on mobile smartphone only. The results include the following matrices:

- Memory consumed in bytes for GPS calculation only (getting longitude and latitude for each point).
- Memory consumed in bytes for calculating distances between points.
- Total Memory consumed to execute application in Mega Hertz.

From the previous two figures: Figures 10 and 11. The performance of execute application on mobile or cloud in terms of memory consumed using different distance and number of points are shown in Figure 12. The total memory consumed on mobile in the case of cloud or in the case of executed all the application on mobile only.

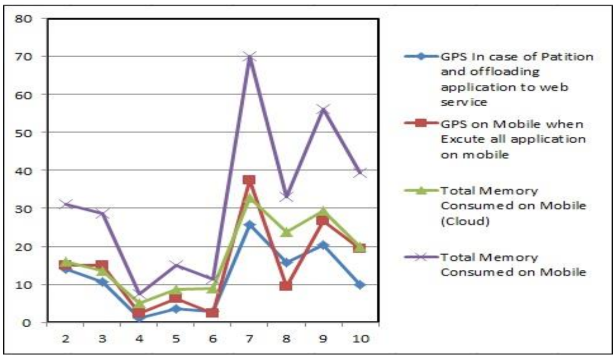


Figure 12: Total Memory consumed on mobile for execution application on cloud and on mobile

Results analysis:
According to partition algorithm, most of resources consumed on mobile smartphone will decrease to approximately half as shown in Figure 11. In case of partition and offloading application, most of resources consumed on cloud and minimized the resources consumed in mobile smartphone as shown in Figures 10 and 12.

5 Conclusion and Future Work

In this paper, we proposed the elastic partition algorithm and partition cost module. Sending computation to another machine is a good idea. Thus, cloud computing can save energy for mobile users through computation offloading. Virtualization, a fundamental feature in cloud computing, lets applications from different customers run on different virtual machines, thereby providing separation and protection. The advantages of Mobile Cloud Computing: Cloud computing can be a promising solution for mobile computing due to many reasons (e.g., mobility, communication, and portability so that cloud can be used to overcome obstacles in mobile computing, thereby pointing out advantages of MCC. Mobile cloud computing can extending battery lifetime (Battery is one of the main concerns for mobile devices). And can reduce power consumption for mobile devices. these solutions require changes in the structure of mobile devices, and mobile application .Computation offloading technique is proposed with the objective to migrate the large computations and complex processing from resource-limited devices (i.e., mobile devices) to resourceful machines (i.e., servers in clouds). This avoids taking a long application execution time on mobile devices which results in large amount of power consumption. The results demonstrate that the remote application execution can save energy significantly
In mobile cloud computing, application offloading is implemented as a software level solution for augmenting computing capabilities of smart mobile devices. In this paper, we present a survey of the energy-efficient technologies in mobile cloud computing, provide the definitions and architectural designs of MCC. We summarize related works in energy-efficient wireless transmission.

We believe that there are still great opportunities for researchers to make ground-breaking contributions in this field, thus bringing significant impacts to the development in the industry. We hope our work will provide a better understanding of design challenges surrounding energy-efficient MCC.

References

[1] X. Zhang, S. Jeong, A. Kunjithapatham, S. Gibbs, “Towards an elastic application model for augmenting computing capabilities of mobile platforms”, The 3rd International ICST Conference on Mobile Wireless Middleware, Operating Systems, and Applications (MobileWare), vol. 48(4), pp.161–174, 2010.

[2] X. Fan, J. Cao, “A survey of mobile cloud computing”, ZTE Communications, vol. 9, no. 1, pp.4–8, 2011.

[3] Le Guan, K. Xu, S. Meina, and S. Junde, “A survey of research on mobile cloud computing”, IEEE/ACIS 10th International Conference on Computer and Information Science (ICIS), pp. 387-392, 2010.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, M. Zaharia, “Above the clouds: A berkeley view of cloud computing”, Technical Report UCB/EECS-2009-28, University of California, Berkeley, Feb. 2009.

[5] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, G. Alonso, “Calling the cloud: Enabling mobile phones as interfaces to cloud applications”, Bacon, J.M., Cooper, B.F. (eds.), Springer, Heidelberg, LNCS, vol. 5896, pp. 83–102, 2009.

[6] F. Niroshinie, W. L. Seng, R. Wenny, “Mobile cloud computing: A survey”, Future Generation Computer Systems, vol. 29, no. 1, pp. 84-106, Jan. 2013.

[7] B. G. Chun, P. Maniatis, “Augmented smartphone applications through clone cloud execution”, USENIX HotOS XII, 2009.

[8] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, S. Jeong, “Securing elastic applications on mobile devices for cloud computing”, Proc. of ACM Cloud Computing Security Workshop, 2009.

[9] M. H. Tang, et al., “A dynamic mechanism for handling mobile computing environmental changes,” InfoScale, no. 7, pp. 1-9, May 2006.

[10] N. R. Vallina, E. J. Crowcroft, “Achieving energy savings in mobile OS”, Proceedings of the Sixth International Workshop on MobiArch, MobiArch’11, ACM, New York, NY, USA, pp. 37–42, 2011.

[11] L. Xinhui, L. Ying, L. Tiancheng, “The method and tool of cost analysis for cloud computing”, Proceedings of IEEE International Conference on Cloud Computing, CLOUD’09, pp. 93–100, 2009.

[12] K. Kumar, Y. Lu, “Cloud computing for mobile users: can offloading computation save energy?”, Computer, vol. 43, pp. 51–56, 2010.

[13] APPRIO Homepage : last accessed 13, October, 2013 http://www.appirio.com/

[14] Force.com Homepage : last accessed 13, October, 2013 http://www.salesforce.com/platform/

[15] Amazon Web Services : last accessed 13, October, 2013 http://aws.amazon.com/

[16] E. P. Daniela, M. L. Alina, “Mobile cloud computing”, Book Chapter in “New Trends in Mobile and Web Development 2012”, Publication series of Lahti University of Applied Sciences, ISBN 978-951-827-141-6, Chapter 10, pp. 287-336, 2012

[17] F. Xiaopeng, C. Jiannong, and M. Haixia, "A survey of mobile cloud computing", ZTE Communications, Special Issue on Mobile Cloud Computing and Applications, vol. 9, no. 1, pp.4-8, 2011.

[18] OnLive Inc., “OnLIve.” [Online]. Available: http://www.onlive.com

[19] G. C. Byung, I. Sunghwan, M. Petros, “Clonecloud: elastic execution between mobile device and cloud”, Proceedings of the Sixth Conference on Computer Systems, EuroSys’11, ACM, New York, NY, USA, pp. 301–314, 2011.

[20] E. E. Marinelli, “Hyrax: cloud computing on mobile devices using MapReduce”, Masters Thesis, Carnegie Mellon University, 2009.

[21] G. Huerta-Canepa, D. Lee, “A virtual cloud computing provider for mobile devices”, Proc. of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, 2010.

[22] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, “The case for VM-based cloudlets in mobile computing”, Proc. IEEE Pervasive Computing, vol. 8, no. 4, pp. 14–23, 2009.

[23] D. Kovachev, Y. Tian, R. Klamma, "Adaptive computation offloading from mobile devices into the cloud," 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications (ISPA), pp.784-791, July 2012.

[24] X. Gu, A. Messer, I. Greenberg, D. Milojicic, K. Nahrstedt, “Adaptive offloading for pervasive computing”, IEEE Pervasive Computing, vol. 3, no. 3, pp. 66-73, 2004.

Diaa Salama Abdul-Minaam was born on November 23, 1982 in KafrSakr, Sharkia, Egypt. He received the B.S from Faculty of Computers &Informatics, Zagazig University, Egypt in 2004 with grade very good with honor, and obtains master degree in information system from faculty of computers and information, menufia university, Egypt in 2009 and submitted for PhD from October 2009. He is working in Benha University, Egypt as teaching assistance at Faculty of Computer and informatics. Diaa has contributed more than 18+ technical papers in the areas of wireless networks, wireless network security, Information security and Internet applications in international journals, international conferences, local journals and local conferences. He majors in Cryptography and Network Security. (Mobile: +20166104747 E-mail: ds_desert@yahoo.com).

Hatem. M. Abdul-kader vice Dean of Faculty of Computers and Information, Menoufia university, Shebin Elkom, Egypt. Prof Hatem obtained his BSC. And M.SC. (by research) both in Electrical Engineering from the Alexandria University, Faculty of Engineering, Egypt in 1990 and 1995 respectively. He obtained his Ph.D. degree in Electrical Engineering also from Alexandria University, Faculty of Engineering, in 2001 specializing in neural networks and applications. Since 2009 he is the Head of the department of Information Systems (IS). Prof. Hatem has published more than 100 papers in international journals, international conferences, local journals and local conferences. He is currently a Professor in Information systems department, Faculty of Computers and Information, Menoufya University, Egypt since 2004.

Mohiy Mohamed Hadhoud, Former vice president of Menoufia university for education and student affairs and former dean of Faculty of Computers and Information, University, Shebin Elkom, Egypt. Currently, he is the dean of Canadian International College (CIC) in New Cairo. He is a member of National Promotion committee for professors, he is a member of National Computers and Informatics Sector Planning committee, and is the University training supervisor. Prof Hadhoud graduated from the department of Electronics and Computer Science, Southampton University, UK, 1987. Since 2001 he worked as a Professor of Multimedia, Signals and image processing and Head of the department of Information Technology (IT), He was a member of the university council. He is the recipient of the university supremacy award for the year 2007. He, among others are the recipient of the Most cited paper award from the Digital signal processing journal, Vol.18, No. 4, July 2008, pp 677-678, ELSEVIER Publisher. Prof. Hadhoud has published more than 160 papers in international journals, international conferences, local journals and local conferences. His fields of Interest: Digital Signal Processing, 2-D Adaptive filtering, Digital Image Processing, Digital communications, Multimedia applications, Information security and data hiding.

Salah M. Elsayed, Dean, Faculty of Computers and Information, head of Scientific Computing Department, Benha University, Benha, Egypt. His PhD degree, in Numerical Analysis from the department of Numerical, Theory and Algorithms of Numerical Linear Algebra, and numerical methods of ordinary and partial differential equations (multi-integral and finite difference methods. A domain decomposition method and chebychev pseudo spec trail methods. Prof Salah obtain Egyptian incentive prize of science in mathematics 2002,and Scopus prize of Best Author have higher citation and H-Index in Scopus 2008 in the last ten years. Prof. Salah has published more than 150 papers in international journals, international conferences, local journals and local conferences. His fields of Interest: Numerical Analysis, numerical methods of ordinary and partial differential equations, and Information security and data hiding.

A Comparative Study of Lossless Compression Algorithm on Text Data

Amit Jain[1], Kamaljit I. Lakhtaria[2]
(Corresponding author: Amit Jain)

Department of Computer Science and Engineering[1]
Sir Padampat Singhania University, Udaipur (Raj.) India

Department of Computer Science and Engineering[2]
Sir Padampat Singhania University, Udaipur (Raj.) India
(Email: amitscjain@gmail.com)

(Received Nov. 20, 2013; revised and accepted Jan. 30, 2014)

Abstract

With increasing amount of text data being stored rapidly, efficient information retrieval and Storage in the compressed domain has become a major concern. Compression is the process of coding that will effectively reduce the total number of bits needed to represent certain information. Data compression has been one of the critical enabling technologies for the ongoing digital multimedia revolution. There are lots of data compression algorithms which are available to compress files of different formats. This paper provides a survey of different basic lossless data compression algorithms on English text files: LZW, Huffman, Fixed-length code (FLC), and Huffman after using Fixed-length code (HFLC). All the above algorithms are evaluated and tested on different text files of different sizes. To find the best algorithm among above, comparison is made in terms of compression: Size, Ratio, Time (Speed), and Entropy. The paper is concluded by the decision showing which algorithm performs best over text data.

Keywords: Data Compression, Huffman Coding, LZW, RLE.

1 Introduction

Data compression is a technique that transforms the data from one representation to another new compressed (in bits) representation, which contains the same information but with smallest possible size [1]. The size of data is reduced by removing the excessive information. The data to be stored or transmitted at reduces storage and/or communication costs. When the amount of data to be transmitted is reduced, the effect is that of increasing the capacity of the communication channel for more data transmission. Similarly, compressing a file to half of its original size is equivalent to doubling the capacity of the storage medium. It may then become feasible to store the data at a higher, thus faster, level of the storage hierarchy and reduce the load on the input/output channels of the computer system.

Benefits of compression

It provides a potential cost saving associated with sending less data over switched telephone network where cost of call is usually based upon its duration.

It not only reduces storage requirements but also overall execution time.

It also reduces the probability of transmission errors since fewer bits are transferred.

It also provides a level of security against illicit monitoring [2].

Data compression can be lossless, Lossless data compression makes use of data compression algorithms that allows the exact original data to be reconstructed from the compressed data. Lossless data compression is used in many applications. For example, it is used in the popular ZIP file format and in the Unix tool gzip. Lossless compression is used when it is important that the original and the decompressed data be identical, or when no assumption can be made on whether certain deviation is uncritical. Typical examples are executable programs and source code. Some image file formats, notably PNG, use only lossless compression [3].

Another family of compression is lossy compression. A lossy data compression method is one where compressing data and then decompressing it retrieves data that may well be different from the original, but is "close enough" to be useful in some way. Lossy data compression is used frequently on the Internet and especially in streaming media and telephonic applications. These methods are typically referred to as codec in this context. Most lossy data compression formats suffer from generation loss: repeatedly compressing and decompressing the file will cause it to progressively lose quality. This is in contrast with lossless data compression [4].

Following are some definitions that are used in this research:

Compression size

Is the size of the new file in bits after compression is complete?

Compression ratio

Is a percentage that results from dividing the compression size in bits by the original file size in bits and then multiplying the result by 100%.

$$\text{Compression Ratio} = 100 * \frac{\text{Size after Compression}}{\text{Size before Compression}}$$

Compression time

Time taken for the compression and the time taken for decompression is considered separately. Compression time is the time in millisecond that we need for each symbol or character in the original file for compression, it results from dividing the time in millisecond that is needed for compressing the whole file by the number of symbols in the original file and scales as millisecond / symbol. If the compression and decompression times of an algorithm are less or up to an acceptable level then it implies that the algorithm can be accepted with respective to the given time factor.

The paper is organized as follows: Section 1 contains a brief Introduction about Compression and its types, Section 2 presents a brief explanation about different compression techniques, Section 3 has its focus on comparing the performance of compression techniques and the final section contains the Conclusion.

2 Data Compression Techniques

Various kind of text data compression algorithms have been proposed till date, mainly those algorithms is lossless algorithm. This paper examines the performance of the Run Length Encoding Algorithm (RLE), Arithmetic Encoding Algorithm, Huffman Encoding Algorithm, Adaptive Huffman Encoding Algorithm and Shannon Fano Algorithm [5]. Performance of above listed algorithms for compressing text data is evaluated and compared.

2.1 Run Length Encoding Technique (RLE)

One of the simplest compression techniques known as the Run-Length Encoding (RLE) is created especially for data with strings of repeated symbols (the length of the string is called a run). The main idea behind this is to encode repeated symbols as a pair: the length of the string and the symbol [6]. For example, the string ‘abbaaaaabaabbbaa’ of length 16 bytes (characters) is represented as 7 integers plus 7 characters, which can be easily encoded on 14 bytes (as for example ‘1a2b5a1b2a3b2a’). The biggest problem with RLE is that in the worst case the size of output data can be two times more than the size of input data. To eliminate this problem, each pair (the lengths and the strings separately) can be later encoded with an algorithm like Huffman coding.

2.2 Huffman Coding

The Huffman coding algorithm [7] is named after its inventor, David Huffman, who developed this algorithms a student in a class on information theory at MIT in1950. It is a more successful method used for text compression. Huffman’s idea is to replace fixed-length codes (such as ASCII) by variable-length codes, assigning shorter codewords to the more frequently occurring symbols and thus decreasing the overall length of the data. When using variable-length codewords, it is desirable to create a (uniquely decipherable) prefix-code, avoiding the need for a separator to determine codeword boundaries. Huffman coding creates such a code. Huffman algorithm is almost same as Shannon - Fano algorithm. Both the algorithms employ a variable bit probabilistic coding method. Both the algorithms differ slightly in the manner in which the binary tree is built. Huffman uses bottom-up approach and Shannon Fano uses Top-down approach. The Huffman algorithm is simple and can be described in terms of creating a Huffman code tree.

The procedure for building this tree is:

1). Start with a list of free nodes, where each node corresponds to a symbol in the alphabet.
2). Select two free nodes with the lowest weight from the list.
3). Create a parent node for these two nodes selected and the weight is equal to the weight of the sum of two child nodes.
4). Remove the two child nodes from the list and the parent node is added to the list of free nodes.
5). Repeat the process starting from step-2 until only a single tree remains.

After building the Huffman tree, the algorithm creates a prefix code for each symbol from the alphabet simply by traversing the binary tree from the root to the node, which corresponds to the symbol. It assigns 0 for a left branch and 1 for a right branch. The algorithm presented above is called as a semiadaptive or semi-static Huffman coding as it requires knowledge of frequencies for each symbol from alphabet. Along with the compressed output, the Huffman tree with the Huffman codes for symbols or just the frequencies of symbols which are used to create the Huffman tree must be stored. This information is needed during the decoding process and it is placed in the header of the compressed file.

2.3 Shannon Fano Coding

Shannon – Fano algorithm is developed by Claude Shannon and R. M. Fano [14, 15]. It is used to encode messages depending upon their probabilities. It allots less number of bits for highly probable messages and more number of bits for rarely occurring messages. The algorithm is as follows:

1). From the given list of symbol, develop either frequency or probability table.
2). Sort the table according to the frequency, with the most frequently occurring symbol at the top.
3). Divide the table into two halves with the total frequency count of the upper half being as close to the total frequency count of the bottom half as possible.
4). Assign the upper half of the list a binary digit '0' and the lower half a '1'.
5). Recursively apply the steps 3 and 4 to each of the two halves, subdividing groups and adding bits to the codes until each symbol has become a corresponding leaf on the tree.

Generally, Shannon-Fano coding does not guarantee that an optimal code is generated. Shannon – Fano algorithm is more efficient when the probabilities are closer to inverses of powers of 2.

2.4 Arithmetic Encoding

This encoding technique developed by Jorma Rissane. It provides extremely high coding efficiency and superior Compression to the better-known Huffman algorithm. Arithmetic coding is a method to ensure lossless data compression. It is indeed a form of variable length entropy encoding. In the case of other entropy encoding techniques, the input message is separated into its component symbols and each symbol is replaced by a code word. But arithmetic coding encodes the entire message into a single number, a fraction n where (0.0_n< 1.0) [8].

The coding algorithm is symbol wise recursive; i.e., it operates upon and encodes (decodes) one data symbol per iteration or recursion. On each recursion, the algorithm successively partitions an interval of the number line between 0 and 1, and retains one of the partitions as the new interval. Thus, the algorithm successively deals with smaller intervals, and the code string, viewed as a magnitude, lies in each of the nested intervals. The data string is recovered by using magnitude comparisons on the code string to recreate how the encoder must have successively partitioned and retained each nested subinterval.

2.5 Adaptive Huffman Coding

The basic Huffman algorithm suffers from the drawback that to generate Huffman codes it requires the probability distribution of the input set which is often not available. Moreover it is not suitable to cases when probabilities of the input symbols are changing. The Adaptive Huffman coding technique was developed based on Huffman coding first by Newton Faller [9] and by Robert G. Gallager [10] and then improved by Donald Knuth [11] and Jefferey S. Vitter [12, 13].

In this method, a different approach known as sibling property is followed to build a Huffman tree. Here, both sender and receiver maintain dynamically changing Huffman code trees whose leaves represent characters seen so far. Initially the tree contains only the 0-node, a special node representing messages that have yet to be seen. Here, the Huffman tree includes a counter for each symbol and the counter is updated every time when a corresponding input symbol is coded. Huffman tree under construction is still a Huffman tree if it is ensured by checking whether the sibling property is retained. If the sibling property is violated, the tree has to be restructured to ensure this property. Usually this algorithm generates codes that are more effective than static Huffman coding.

Storing Huffman tree along with the Huffman codes for symbols with the Huffman tree is not needed here. It is superior to Static Huffman coding in two aspects: It requires only one pass through the input and it adds little or no overhead to the output. But this algorithm has to rebuild the entire Huffman tree after encoding each symbol which becomes slower than the static Huffman coding.

3 Methodologies

In order to test the performance of above mentioned compression algorithms e.g. the Run Length Encoding Algorithm, Shannon Fano Algorithm, Adaptive Huffman Encoding Algorithm, Huffman Encoding Algorithm and Arithmetic Encoding, the algorithm were implemented and tested with a various set of text files. Performances of the algorithm were evaluated by computing the compression ratio, compression time.

The performances of the algorithms depend on the size of the source file and the organization of different symbols and text patterns in the source file. Therefore, research work done to include text files of different types such as notepad files, source codes, e-books in pdf files, etc, and of different file sizes are used as source files. A chart is drawn in order to verify the relationship between the file sizes after compression, the compression and decompression time.

An algorithm which gives an acceptable saving percentage with minimum time period for compression and decompression is considered as the best algorithm.

4 Results/Comparison

Five lossless compression algorithms are tested on ten different types, size and contents of text files. All the text files were of different size. The first 3 text files were in normal English language. The next 2 files are computer programs, having more repeating set of words. The last 5 file are the pdf files written in normal English language.

Followings are the results for 10 different text files.

4.1 Results

Arithmetic coding algorithm result has not been considered as results were not accurate due to overflow problem. Results of all other 4 algorithms and their comparisons are given below.

According to result of Table 1, the compression ratio of RLE algorithm is very low. For the file number 1, 3 and 7, we see that the size of compressed file is larger than original file size. Among the given 4 algorithm, we can see that the size of compressed file created by Adaptive Huffman algorithm is very less in compare to other algorithm.

Table 1 – Comparison based on compressed file size

Original File			**Compressed File Size**			
S. No.	***File Name***	***File Size***	***RLE***	***Adaptive Huffman***	***Huffman Encoding***	***Shannon Fano***
1	Paper1	22,094	22,251	13,432	13,826	14,127
2	Paper2	44,355	43,800	26,913	27,357	27,585
3	Paper3	11,252	11,267	7,215	7,584	7,652
4	Prog1	15,370	13,620	8,584	8,961	9,082
5	Prog2	78,144	68,931	44,908	45,367	46,242
6	Book1	39,494	37,951	22,863	23,275	23,412
7	Book2	118,223	118,692	73,512	74,027	75,380
8	Book3	180,395	179,415	103,716	104,193	107,324
9	Book4	242,679	242,422	147,114	147,659	150,826
10	Book5	71,575	71,194	44,104	44,586	44,806

From the Table 2, its shows that compression ratio achieved by RLE algorithm is not more than 2% of original file, that is not a reasonable compression. In the Adaptive Huffman algorithm, the compression ratio of selected files is within the range of 55% to 65%. The compression ratio does not depend on file size but it depends on structure and contents of file. In the Huffman Encoding algorithm, the compression ratio range within 58% to 67%. The compression ratios for Shannon Fano approach are in the range of 59% to 64% which is slightly equivalent to the Huffman Encoding algorithm.

So, from the table, we can derive the decision that RLE has lowest compression ratio and Adaptive Huffman has best compression ratio, although the compression ratio achieved by Adaptive Huffman is relatively same as achieved by Huffman Encoding and Shannon Fano, the difference is not more than 2%.

Table 2 – Comparison based on compression ratio

Original File			**Compression Ratio**			
S. No.	***File Name***	***File Size***	***RLE***	***Adaptive Huffman***	***Huffman Encoding***	***Shannon Fano***
1	Paper1	22,094	100.7106	60.7947	62.5780	63.9404
2	Paper2	44,355	98.7487	60.6763	61.6773	62.1914
3	Paper3	11,252	100.1333	64.1219	67.4013	68.0056
4	Prog1	15,370	88.6141	55.8490	58.3018	59.0891
5	Prog2	78,144	88.2102	57.4682	58.0556	59.1753
6	Book1	39,494	96.09307	57.8898	58.9330	59.2798
7	Book2	118,223	100.3967	62.1807	62.6164	63.7608
8	Book3	180,395	99.4567	57.4938	57.7582	59.4938
9	Book4	242,679	99.89409	60.6208	60.8453	62.1504
10	Book5	71,575	99.4676	61.6192	62.2927	62.6000

From the Table 3, it shows that the compression time of RLE algorithm is relatively low but for the Adaptive Huffman algorithm, the compression time is relatively high. The Compression time of Huffman Encoding algorithm and Shannon Fano algorithm is relatively low in compare to Adaptive Huffman algorithm but higher than RLE algorithm.

Table 3 – Comparison based on compression time

Original File			**Compression Time (ms)**			
S. No.	***File Name***	***File Size***	***RLE***	***Adaptive Huffman***	***Huffman Encoding***	***Shannon Fano***
1	Paper1	22,094	359	80141	16141	14219
2	Paper2	44,355	687	223875	54719	55078
3	Paper3	11,252	469	30922	3766	3766
4	Prog1	15,370	94	41141	5906	6078
5	Prog2	78,144	1234	406938	156844	162609
6	Book1	39,494	141	81856	13044	12638
7	Book2	118,223	344	526070	134281	153869
8	Book3	180,395	2766	611908	368720	310686
9	Book4	242,679	2953	1222523	655514	549523
10	Book5	71,575	344	231406	42046	42997

From the Table 4, it shows that the decompression time of RLE algorithm is relatively low but for the Adaptive Huffman algorithm, the decompression time is relatively high. The decompression time of Huffman Encoding algorithm and Shannon Fano algorithm is relatively low in compare to Adaptive Huffman algorithm but higher than RLE algorithm.

Table 4 – Comparison based on decompression ratio

Original File			**Decompression Time (ms)**			
S. No.	***File Name***	***File Size***	***RLE***	***Adaptive Huffman***	***Huffman Encoding***	***Shannon Fano***
1	Paper1	22,094	2672	734469	16574	19623
2	Paper2	44,355	2663	1473297	20606	69016
3	Paper3	11,252	2844	297625	6750	8031
4	Prog1	15,370	2500	406266	9703	9547
5	Prog2	78,144	17359	2611891	224125	229625
6	Book1	39,494	2312	1554182	12638	12022
7	Book2	118,223	1469	1271041	99086	114187
8	Book3	180,395	2250	1554182	288232	255933
9	Book4	242,679	1828	2761631	470521	441153
10	Book5	71,575	1532	633117	34293	32869

4.2 Comparison of Result

In order to compare the performance of selected algorithm, the compressed file size, compression ratio, compression and decompression time are compared. Figure 1 shows the compression file size of selected 10 files for the entire algorithms.

The sizes of compressed files are compared with original file size and result is shown in Figure 1. The figure shows that saving percentage of RLE algorithm is very less. The compressed files of all other 3 algorithms are relatively similar. The *compressed file size increased according to original file size that indicates the saving percentage of algorithm depends on the redundancy of file.*

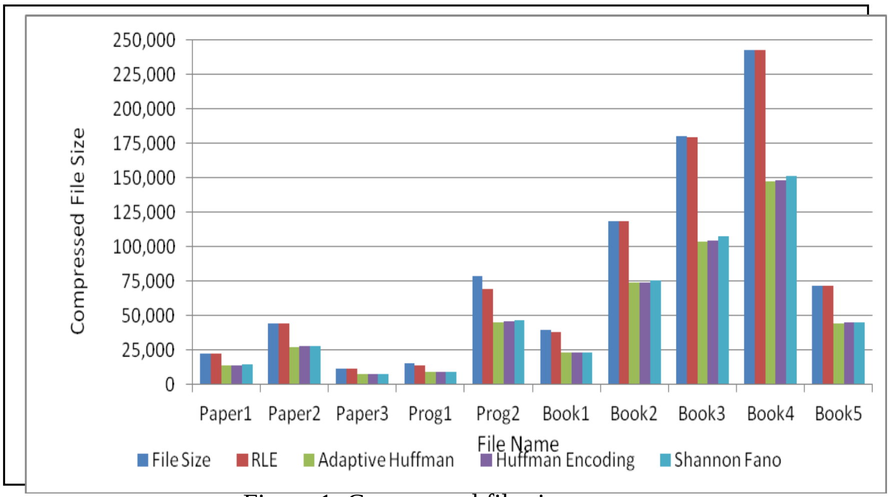


Figure 1: Compressed file size

Figure 2 shows the comparison of compression time of all 4 algorithms. Compression time increase with the increase of file size. For RLE algorithm the compression time does not depends on the size of file, it remain almost constant. Compression time for RLE is very low but for Adaptive Huffman algorithm, the compression time is very high.

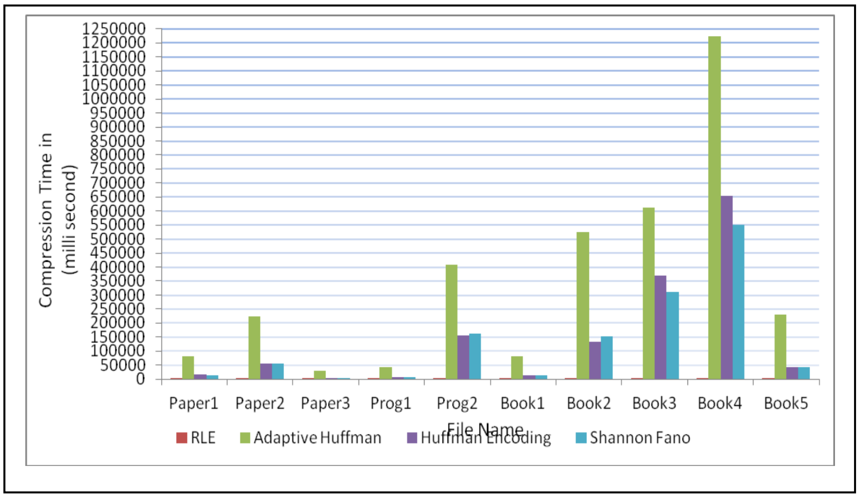


Figure 2: Compression time

Figure 3 shows the decompression time of all the algorithm. The decompression time of RLE algorithm is almost negligible and almost same for all the files of different size. the decompression time of Huffman Encoding and Shannon Fano is relatively same but for Adaptive Huffman algorithm, the decompression time is very high.

5 Conclusions

We have taken statistical compression techniques for our study to examine the performance of compression algorithm over English text data. This text data are available in the form of different kind of text file which contain different text patterns. By considering the compression time, decompression time and compression ratio of all the algorithms we have drawn the graph and table. From the above comparison and graph, it can be derived that the Huffman Encoding can be considered as the most efficient algorithm among selection ones.

We also note that; the contents of file (i.e. the number of different character or symbols and the frequency for each symbol) are effective factor on the performance of the data compression techniques. So, we suggest to make another test for the four techniques that we study but on the other sample tested files that contain different number of symbols.

Data compression stills an important topic for research these days, and has many applications and useful needed. So, we suggest continuing searching in this field and trying to combine two techniques in order to get best one.

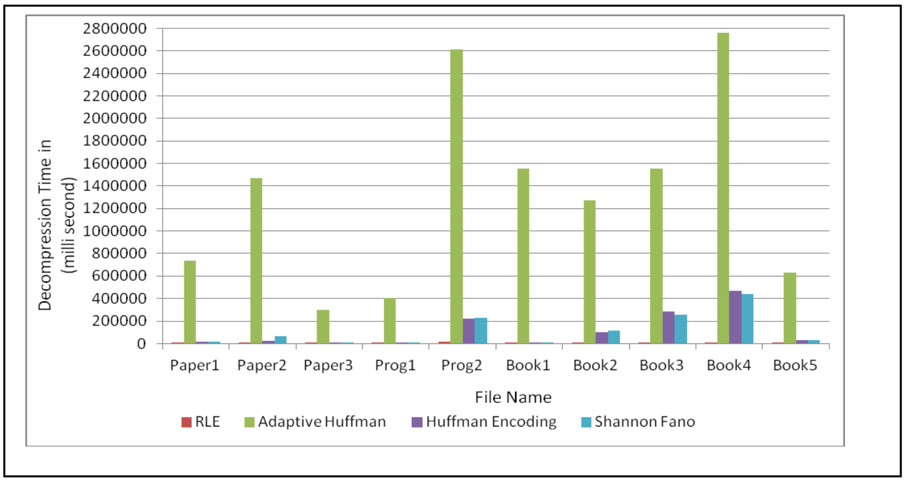


Figure 3: Decompression time

References

[1] I. M. Pu, Fundamental Data Compression, Elsevier, Britain, 2006.

[2] Data Compression: Advantages and Disadvantages: http://www.esrf.eu/computing/Forum/imgCIF/PAPER/advantages_disadvantages.html, last accessed on Feb. 2013.

[3] Lossless Compression: http://en.wikipedia.org/wiki/Lossless_compression, last access on Feb. 2013.

[4] Lossy Compression: http://en.wikipedia.org/wiki/Lossy_compression, last access on Feb. 2013.

[5] W. Kesheng, J. Otoo and S. Arie, “Optimizing bitmap indices with efficient compression”, ACM Trans. Database Systems, vol. 31, pp. 1-38, 2006.

[6] E. Blelloch, Introduction to Data Compression, Computer Science Department, Carnegie Mellon University, 2002.

[7] D. A. Huffman, “A method for the construction of minimumredundancy codes”, Proceedings of the Institute of RadioEngineers, vol. 40, no. 9, pp. 1098–1101, 1952.

[8] A. S. E. Campos, Basic arithmetic coding by Arturo Campos Website, Available from: http://www.arturocampos.com/ac_arithmetic.html. (Accessed 02 February 2009)

[9] N. Faller, “An adaptive system for data compression”, In Record of the 7th Asilornar Conference on Circuits, Systems and Computers, IEEE Press, Piscataway, NJ, pp. 593-597, 1973.

[10] R. G. Gallager, “Variations on a theme by Huffman”, IEEE Transactions on Information Theory, vol. IT-24, no. 6, pp. 668-674, Nov. 1978.

[11] D. E. Knuth, “Dynamic Huffman coding”, Journal of Algorithms, vol. 6, no. 2, pp. 163-180, June 1985.

[12] J. S. Vitter, “Design and analysis of dynamic Huffman codes”, Journal of the ACM, vol. 34, no. 4, pp. 825-845, October 1987.

[13] J. S. Vitter, “Dynamic Huffman coding”, ACM Transactions on Mathematical Software, vol. 15, no. 2, pp. 158-167, June 1989.

[14] R. M. Fano, “The Transmission of Information”, Technical Report No. 65, Research Laboratory of Electronics, M.I.T., Cambridge, Mass.; 1949.

[15] K. Lakhtaria, "Protecting computer network with encryption technique: A Study." Ubiquitous Computing and Multimedia Applications. Springer Berlin Heidelberg, pp. 381-390, 2011.

[16] C. E. Shannon, “A mathematical theory of communication,” Bell Sys. Tech. Jour., vol. 27, pp. 398-403, July 1948.

Amit Jain is working in CSE Department, Sir Padampat Singhania University, Udaipur, India. He is having 17 years of teaching experience. He has taught to post-graduate and graduate students of engineering. He is pursuing Ph.D. in Computer Science, in the area of Information Security. He has presented 3 papers in International Journal, 5 papers in International Conference and 8 papers in National Conference.

Dr. Kamaljit I Lakhtaria is working in CSE Department, Sir Padampat Singhaniya University, India. He obtained Ph. D.

in Computer Science; area of Research is “Next Generation Networking Service Prototyping & Modeling”. He holds an edge in Next Generation Network, Web Services, MANET, Web 2.0, Distributed Computing. His inquisitiveness has made him present 18 Papers in International Conferences, 28 Paper in International Journals. He is author of 8 Reference Books. He is member of Life time member ISTE, IAENG. He holds the post of Editor, Associate Editor in many International Research Journal. He is reviewer in IEEE WSN, Inderscience and Elsevier Journals.

Guide for Authors
International Journal of Electronics and Information Engineering

International Journal of Electronics and Information Engineering (IJEIE) will be committed to the timely publication of very high-quality, peer-reviewed, original papers that advance the state-of-the art and applications of Electronics and Information Engineering. Topics will include, but not be limited to, the following: Algorithms, Bioinformatics, Computation Theory, AI, Fuzzy Systems, Embedded Systems, VLSI/EDA, Computer Networks, Information Security, Database, Data Mining, and Information Retrieval, Digital Content, Digital Life, and Human Computer Interaction, Image Processing, Computer Graphics, and Multimedia Technologies, Information Literacy, e-Learning, and Social Media, Mobile Computing, Wireless Communications, and Vehicular Technologies, Parallel, Peer-to-peer, Distributed, and Cloud Computing, Semiconductor, Software Engineering and Programming Languages, Telecommunication, etc.

1. Submission Procedure

Authors are strongly encouraged to submit their papers electronically by using online manuscript submission at http://ijeie.jalaxy.com.tw/.

2. General

Articles must be written in good English. Submission of an article implies that the work described has not been published previously, that it is not under consideration for publication elsewhere. It will not be published elsewhere in the same form, in English or in any other language, without the written consent of the Publisher.

2.1 Length Limitation:

All papers should be concisely written and be no longer than 30 double-spaced pages (12-point font, approximately 26 lines/page) including figures.

2.2 Title page

Title page should contain the article title, author(s) names and affiliations, address, an abstract not exceeding 100 words, and a list of three to five keywords.

2.3 Corresponding author

Clearly indicate who is willing to handle correspondence at all stages of refereeing and publication. Ensure that telephone and fax numbers (with country and area code) are provided in addition to the e-mail address and the complete postal address.

2.4 References

References should be listed alphabetically, in the same way as follows:

For a paper in a journal: M. S. Hwang, C. C. Chang, and K. F. Hwang, ``An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445--446, 2002.

For a book: Dorothy E. R. Denning, *Cryptography and Data Security*. Massachusetts: Addison-Wesley, 1982.

For a paper in a proceeding: M. S. Hwang, C. C. Lee, and Y. L. Tang, ``Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13--16, Xian, China, 2001.

In text, references should be indicated by [number].

2.5 Author benefits

No page charge is made.

Subscription Information

Individual subscriptions to IJEIE are available at the annual rate of US$ 200.00 or NT 6,000 (Taiwan). The rate is US$1000.00 or NT 30,000 (Taiwan) for institutional subscriptions. Price includes surface postage, packing and handling charges worldwide. Please make your payment payable to "Jalaxy Technique Co., LTD." For detailed information, please refer to http://ijeie.jalaxy.com.tw or Email to ijeieoffice@gmail.com.