Piracy Detection and Prevention using SIFT based on Earth Mover's Distance (EMD)

B. Srinivas
CSE Department
MVGR College of Engineering,
Vizianagaram, AP, India.

Dr. Koduganti Venkata Rao
Professor and HoD CSE,
Vignan IIT,
Vishakhapatnam, AP, India.

Dr. P. Suresh Varma
Principal and Professor of CS,
Adikavi nannaya university,
Rajahmundry, A.P, India,

ABSTRACT
These days software is not just few lines of code and few number of files, it constitute major part of business logic, and most valuable information. Software is required by all kind of people from individuals to large organizations to carry out important tasks. But it is being pirated on large scale, violating software license and leading to copyright infringement. Almost 50% software licenses are pirated accounting over 51.4 billion dollars loss globally. Piracy is killing many software businesses leading to drastic loss for software developers. Under these circumstances there is a need for anti-piracy methods. This paper discuss about a robust yet efficient method for avoiding software piracy. After introducing software piracy methods and general piracy activities carried out by pirates, a mechanism to validate authorized user using face identity is described. A vector based algorithm is explained which detects Facial features of authorized user and generates a user authentication key, which is used for validation during product activation.

Keywords
Piracy, facial authentication, face recognition, anti-piracy, key generation.

1. INTRODUCTION
Whenever someone buys software, the software publisher delivers software and its associated license to the customer. A complete usage policy and license agreement is clearly mentioned in software license documents. In other words, the customer is actually purchasing license to use that software. Breaking those terms and policies specified in license, some people redistribute this software by creating copies of it and remove any licenses associated with that respective software, resulting in software piracy [2, 3]. This pirated software is similar to the original software except for its license. Using pirated software without a license is against law. Despite piracy being considered a 'felony', people still lean on using pirated software. This is an analogy to theft in real life, just as theft cannot be entirely prevented, we cannot stop people from using pirated software. With increasing usage of broadband internet, peer-to-peer and torrent sharing techniques, piracy of software is becoming easier just as the saying goes 'offered or free food is tastier to mind than food that comes at a cost'. This mass attitude developed by pirates among people leads to drastic increase in piracy due to which is almost one third of software content available today is pirated.

Software Piracy is an unauthorized copy of protected information from one media to another media or conversion from one form to another without appropriate permissions [2]. Software purchasing not only involves owning a software entity but also includes owning a license for handling and using the respective software. This license information includes rules and regulations that should be followed by the end-user or owner of the software. These rules and regulations are laid in order to protect the intellectual properties of the respective software companies involved in developing the software product.

A good quality software product developed by software companies is the fruit of huge effort, stain and tedious hours or months of many software programmers time. Software Company relies on sales of software and thereby programmers. Making illegal copies of software or illegal sharing of the software effects sales causing the respective software company to run into losses. This eventually affects software quality. It is similar to stealing goods and using them without paying. Such losses approximately sum up to 13 Billion dollars every year [8].

One of the primary reason for most of the software being pirated is the fact that software sales can be monitored and recorded while software usage cannot be monitored, this enables unethical software owners or pirates to distribute these programs or copy these copyrighted content using many tools, media and sharing directories [4]. Many underground Forums and discussion groups are filled with copies of copyrighted software; even a novice software user can login to these systems and can download software, free of cost.

Now a day's web is becoming increasingly dynamic after the evolution of web 2.0. A normal internet user with some limited web browsing knowledge can get the premium content for free using Newsgroups, FTP sites, WWW pages and particularly warez sites and also peer to peer, torrent sharing tools makes software piracy and illegal software distribution easier. Software can be pirated at any part of the software distribution process. Software buyer can also be involved in promoting piracy by compromising on his own copy. Software distributions sometimes involve demo versions and trail versions which are controlled based on logics like 30 days expiry or disabling some features, hackers use some reverse engineering techniques to overcome these logics and develop cracks for software. These cracks are also distributed to people using peer to peer networks and warez websites, which may contain malicious programs that can easily damage software at the user end.

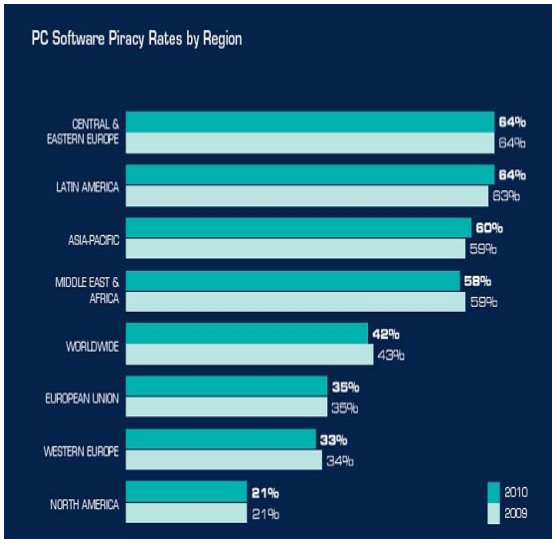

Fig. 1: Region wise piracy in % during 2009 and 2010 by BSA

Recent survey released by the Business software alliance (BSA) [8], the Fig.1. shows year and region wise piracy statistics. If this situation continues for a few more years, software would no longer be commercial thus crashing the existence of 'quality of software'. To maintain software development, growth of software business and their respective intellectual properties safe, Anti-piracy methods must be employed with greater strength.

2. RELATED WORKS

This section provides an overview of the related work that has been done to stop software privacy and identifies their fundamental weaknesses.

Software companies are facing huge losses due to software piracy, which instigates them to develop technical measures to prevent piracy [10]. The main idea behind the efforts to cease piracy is to implement a identifier or token which can be either a software based license key, serial number, license file or hardware based components like Dongle, smartcard, CD etc… There are situations where even the link between software and identifier can be weak or strong. Companies mainly concentrate on activating and authenticating user, using these identifiers and giving full access to software. Although many measures are taken to prevent illegitimate user to access software, pirates use extra ordinary technical skills to skip these activation techniques.

Although software identifiers like license key, file, activation code etc., are used to protect software assert, techniques like watermarking and fingerprinting provide better security and privacy thereby increasing the chances of pirate being caught even though both techniques carry their own advantages and disadvantages [13]. Another way is to provide updates to legitimate users, so that product verification identifies illegitimate users and restrict product access or usage. As the software ageing, with time increases its fragileness towards upcoming technologies, products and times, updates are necessary for any business software which in turn provides software protection.

Now a day's, many software companies including Microsoft are implementing software and hardware based approach that could make piracy a bit difficult and far- fetched for experienced pirates.

Piracy techniques are very powerful. Pirates are experienced coders and researchers who can easily break software protection using reverse engineering techniques and machine level programming [4]. According to a recent survey, newbie pirates are also using artificial intelligence concepts to automatically crack any software. A simple technique used by pirates to by-pass activation by editing machine level code using decompiles and changing jumps to desired memory locations.

2.1 Pitfalls in the existing software protection

Most of the software activations are performed at user end, such as key validation and hardware integration, so anyone can easily get a hold on executable software and perform some tapping activity that can easily lead to overcome this activation or validation process. This problem can be related to finite state machine problem where every state of object can be examined and these states can be varied, ultimately to achieve a success state. Time and efforts after some attacks result in cracking.

The disadvantage of static protection mechanism is that, once a copy is available that undoes the static copy protection or no longer carries the identification of the perpetrator, it can be distributed virtually to an unlimited extent due to which the software provider can no longer enforce its copyright. In short it is the static nature of existing defense mechanisms that are reason for them to fail.

Another reason why static protection techniques are so susceptible to attacks is that, while the first copy is very expensive to produce, subsequent copies are inexpensive to reproduce and distribute. This is an important facilitating condition for software privacy and hence its elimination will make software privacy less attractive. As in the world of physical objects where each objects is unique and cost to reproduce it is nonzero, we believe that the only way to achieve useful reproduction at nonzero cost is to make each legitimate copy unique. This is most obvious for the hardware based mechanism as they combine the software with a unique hard method to duplicate, physical object. The software approaches also use of the part that is unique for each installation copy such as license number, license file, activation code, decryption key or fingerprint. Software aging uses a key to identify legal owners of a copy and TCPA identifies the host computer and operating system. A fundamental drawback of these schemes however is that these unique parts are not part of original program instead they are added for the purpose of copyright protection. We believe that this is one of the reasons why they have been proven to be relatively easy to be removed or circumvented.

3. PROPOSED SYSTEM

A robust system with a centralized activation server is designed which is capable enough to make secured validations using image matching and features extraction based on Earth movers distance algorithm and SIFT algorithms [13,14,18.]. Software producer integrates an activation tool along with software distribution which will be used to perform validations and activation at user end. Whenever a buyer purchases software, buyer will submit his image, this image is stored in database and used for the purpose of matching, features are extracted from this image using SIFT and stored in database[19]. During software activation using integrated tool, user submits the same image which he had submitted during software purchase, or takes a photo using system webcam. This image is sent to activation server along with

user system properties. The resulted image is then compared with image in database using earth mover distance algorithm after verification and validation, a unique key of 2MB size is generated based on features and system properties along with a tracking key to track user software activation process and usage. This key is shared with user and key file is stored in user system, and software is now activated successfully.

3.1 Face Detection and Feature Extraction

Face recognition can be applied for a wide variety of problems like image and film processing, human-computer interaction, criminal identification etc. This has motivated the researchers to develop computational models to identify the faces, which are relatively simple and easy to implement.

3.2 SIFT Approach

Scale Invariant Feature Transform (SIFT) features are features extracted from images to help in reliable matching between different views of the same object [18]. The extracted features are invariant to scale and orientation and are highly distinctive of the image. They are extracted in four steps. The first step computes the locations of potential interest points in the image by detecting the maxima and minima of a set of Difference of Gaussian (DoG) filters, applied at different scales all over the image. Where DoG image $D(x, y, \boldsymbol{\sigma})$ is given by $D(x, y, \sigma) = L(x, y, ki\sigma) - L(x, y, ki\sigma)$ where $L(x, y, K\sigma)$is convolution of the original image $I(x, y)$ with the Gaussian blur $G(x, y, k\sigma)$ at scale k$\boldsymbol{\sigma}$ that is $L(x, y, k\sigma) = G(x, y, ki\sigma) * I(x, y)$ Then, these locations are refined by discarding points of low contrast. An orientation is then assigned to each key point based on local image features. Finally, a local feature descriptor is computed at each key point. This descriptor is based on the local image gradient that is transformed according to the orientation of the key point to provide orientation in variance. Every feature is a vector of dimension 128, distinctively identifying the neighborhood around the key point.

SIFT approach is followed and features are extracted for user submitted image, because using SIFT approach we can easily compare images despite their orientation in different directions. These generated features are used for image comparison and the generation of unique key for validation.

This system is built in two phases. In the first part, image features are generated from user submitted image during purchase phase. Generated image features are added to database along with software ID.

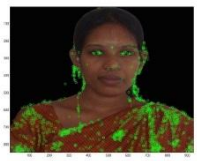

Fig. A. **Fig. B.**
Features Extracted Using SIFT.

Second part is building a software installer that is used for activation of software this is build and distributed along with software, this installer connects to activation server and performs activation process.

3.3 EMD based Feature Comparison

Computing the EMD is based on a solution to the well-known transportation problem (Hitchcock, 1941) a.k.a. the Monge-Kantorovich problem which goes back to 1781 when it was first introduced by Monge (Rachev, 1984) where a set of several suppliers are considered, each with a given amount of goods, who are required to supply several consumers, each with a given limited capacity [11, 12]. For each supplier-consumer pair, the cost of transporting a single unit of goods is given. Here, the transportation problem is to find a least-expensive flow of goods from the suppliers to the consumers that satisfy the consumers' demand.
Signature matching can be naturally cast as a transportation problem by defining one signature as the supplier and the other as the consumer, and by setting the cost for a supplier-consumer pair to equal the ground distance between an element in the first signature and an element in the second. Intuitively, the solution is then the minimum amount of "work" required transforming one signature into the other [19].

EMD is calculated using these equations

Producer:

P= {(P1,Wp1), (P2,Wp2),….(Pm ,Wpm)} -(1)

Consumer:

C = {(C1,Wc1), (C2,Wc2), ….(Cn,Wpn)} - (2)

Ground distance matrix (distance b/w p,q)

D = [dij] -(3)

Flow matrix(that minimizes overall cost)

F = [fij] - (4)

we find F using following equations

$$WORK(P, Q, \boldsymbol{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} dij \,.\, fij \qquad -(5)$$

$$fij \;\geq\; 0 \; where \; 1 \leq i \;\leq m \; and \; 1 \leq j \leq n \qquad -(6)$$

$$\sum_{j=1}^{n} fij \;\leq wpi \; where \;\; 1 \;\leq\; \mathrm{i} \;\leq\; \mathrm{m} \qquad -(7)$$

$$\sum_{j=1}^{m} fij \;\leq wqj \; where \;\; 1 \;\leq\; \mathrm{j} \;\leq\; \mathrm{n} \qquad -(8)$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} fij = Min\left(\sum_{i=1}^{m} wpi \sum_{j=1}^{n} wqj\right) \qquad -(9)$$

System Setup:

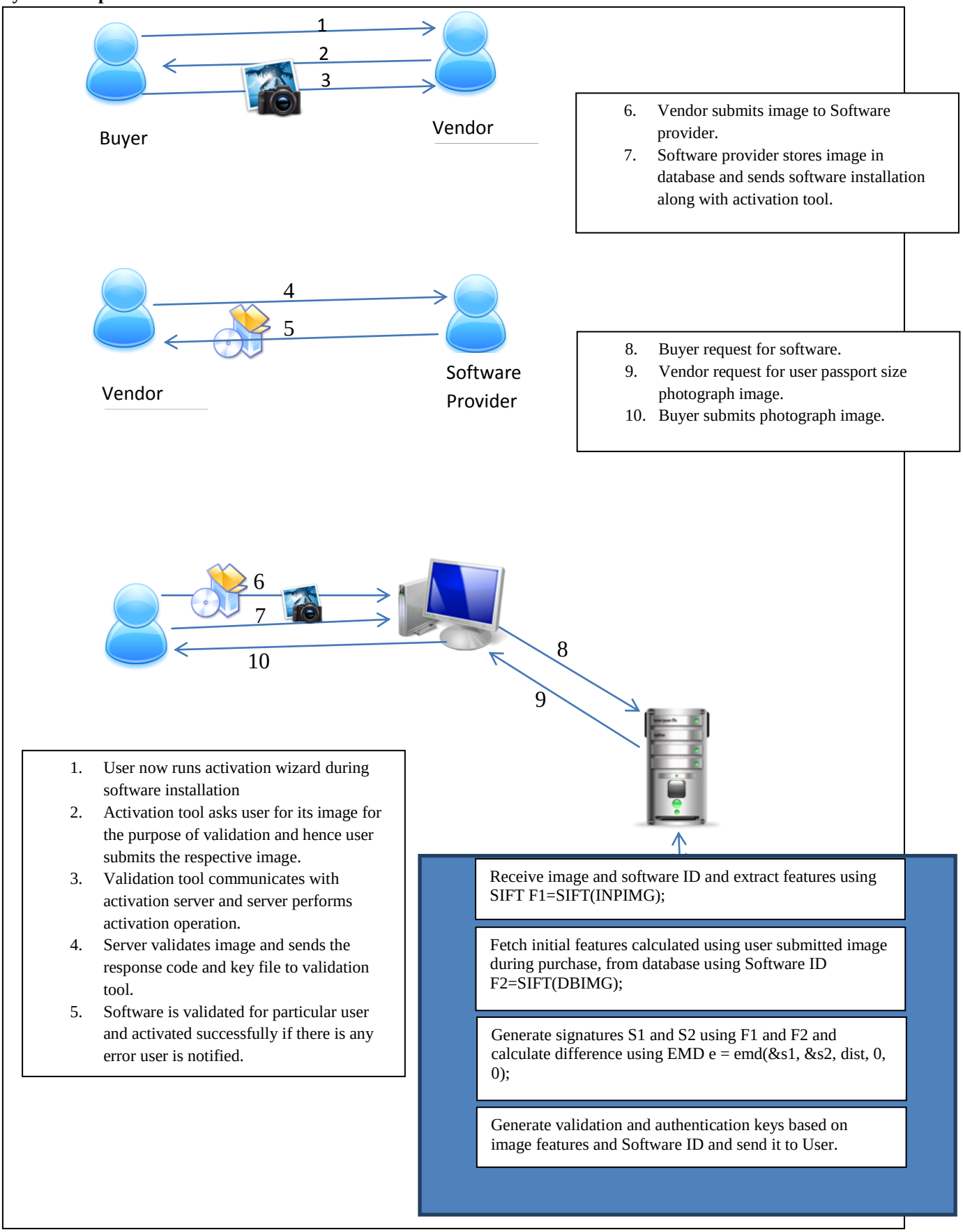

Fig. 2: Architecture of Anti-Piracy System.

Now EMD is calculated:

$$EMD(P,Q) = \sum_{i=1}^{m} \sum_{j=1}^{n} dij \cdot fij \Big/ \sum_{i=1}^{m} \sum_{j=1}^{n} fij \qquad -(10)$$

We generated a signature s1 (f1, w1) with f1, features and w1, weights and save that in our activation server and while authentication we capture shots of face and generate feature signature, s2 (f2, w2) where f1, f2 features of face (values or positions corresponding to iris, eyebrow positions etc..) And w2, w1 are weights associated.

Now we calculate EMD like this

```
float dist(feature_t *F1, feature_t *F2)
{
 intdX = F1->X - F2->X, dY = F1->Y - F2->Y, dZ = F1->Z - F2->Z;
 return sqrt(dX*dX + dY*dY + dZ*dZ);
}

feature_t f1[4] = { {100,40,22}, {211,20,2}, {32,190,150}, {2,100,100} },
 f2[3] = { {0,0,0}, {50,100,80}, {255,255,255} };
 float      w1[5] = { 0.4, 0.3, 0.2, 0.1 }, w2[3] = { 0.5, 0.3, 0.2 };
 signature_t s1 = { 4, f1, w1},     s2 = { 3, f2, w2};
 float     e;
 e = emd(&s1, &s2, dist, 0, 0);
```

3.4 Key generation and distribution

A large key with varying size 2MB, 4MB, 6MB key file is generated using image features and software ID, some system properties are also added to the key at the client side to make it unique and this key contains or represents software validation and authentication. Even though a hacker or cracker tries to bypass this activation process, this key file should be present to run software, this key file adds more advantage so that software usage can also be tracked and if any misuse of software or piracy attempts can be recorded and reported.

This key file is distributed from activation server to activation setup system at client end. A three step process is undergone by client and server to exchange this key file. After authentication at server, it sends a key ready signal to client and client respond with a key request by encapsulating system properties. Now server generates a unique number using system properties and encrypts key file contents with that unique number and transfers it to client. Upon receiving key file client decrypts it and loads key into secure place and adds a reference to software executable and generates a loader executable. This executable should be launched to execute the original software.

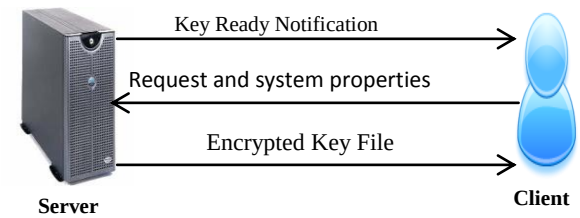

Fig. 3. Key Sharing Mechanism

4. RESULTS AND ANALYSIS

This system is tested for 15 software distributions and used 25 faces as user submitted images and performed activation on 6 individual systems, during activation process 25 correct and 25 wrong images are given as input to system and got 98% successful activation. Following are the results obtained for 7 input images.

Table 1: EMD and feature comparison between user input images.

S.No	Attempt (User Image)	Number of SIFT features	EMD Distance	% of validity
1	Image 1	24	6.5186e-015	96.4
2	Image 2	32	4.6476e-015	97.2
3	Image 3	21	1.8214e-015	98.6
4	Image 4	16	10.5186e-015	95.4
5	Image 5	28	24.4686e-015	93.2
6	Image 6	40	8.1184e-015	95.1
7	Image 7	35	17.6186e-015	94.8

SIFT features are extracted from images for individual inputs these SIFT features are used to generate key and these acts as unique identifiers for particular activation. Fig. A and Fig. B shows the features points considered for key gen process.

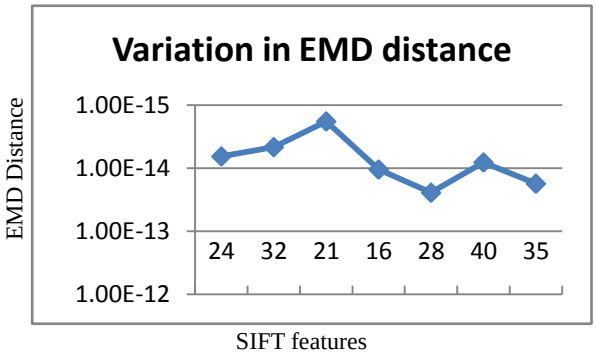

Fig. 4: EMD distance variation.

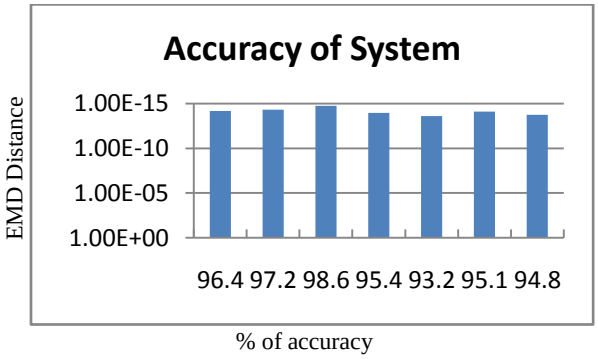

Fig. 5: Accuracy in the system.

We have tested our system where hacking is possible by considering a case where some another image is used for

activation after it is activated by some other person. Following are results obtained.

Table 2: Comparison between impersonated and original image

S.no	Image Input	SIFT features	EMD Distance	% of accuracy	Valid
1	Image1	28	2.56	10%	False
2	Image2	36	1.29	20%	False

Here image1 is another person image, actually did not purchased software and image2 is morphed image, which is modified to get near match. But both attempts failed because EMD distance is larger and thus making these attempts invalid.

We want to consider another case where user submits same image twice by running activation wizard in another computer. Following are results obtained.

Table 3: Tested input based on SIFT and EMD.

	Image 1	Image 2
SIFT features	42	42
EMD distance	2.9426e-015	2.9412e-015
% accuracy	98%	98%
Key File	Generated	Not Generated
Valid	True	False
Remarks	Key is based on system properties	Key is already generated and attempt is invalid

Here image 2 is same image used for activation once again to use same software in another computer. This may be attempted by other user who acquired buyer's image and used it for activation. In this case key generation fails because it is already generated and system properties vary because software activation is initiated form another computer.

However our system fails when we attempt to start activation simultaneously in two computers using same image. This can be solved by binding IP address or using GPS based location detection.

5. CONCLUSION AND FUTURE WORK

This technique can be extended to web services, web applications and can be deployed in a centralized system. Using an API, software companies can easily integrate this service or application. This technique makes hacking or cracking software using native methods very difficult to achieve their respective purpose. Our experiment results prove that it reduces hacking attempts and improves software quality.

6. REFERENCES

[1] Yawei Zhang, Lei Jin, Xiaojun Ye, "Software Piracy Prevention: Splitting on Client", IEEE International Conference on Security Technology, December 13-15, 2008

[2] Sivasubramanyam Y, Deepak RanjanShenoy, "Computer Hardware and System Software Concepts" Vol -1, Version 1.0, March 2007

[3] Mikko T. Siponen, TeroVartiainen, "Unauthorized Copying of Software - An Empirical Study of Reasons For and Against", SIGCAS Computers and Society, Volume 37, No. 1, June 2007, PP 30 -43

[4] Petar Djekic, Claudia Loebbecke, "Preventing application software piracy: An empirical investigation of technical copy protections", The Journal of Strategic Information System, Volume 16, Issue,June 2007, PP 173-186

[5] B. Fu, G. Richard III and Y. Chen. Some New Approaches for Preventing Software tampering. In ACM SE, 06, Melbourne Florida, USA. 2006

[6] Q. Yang and X.O. Tang, "Recent advances in subspace analysis for face recognition," in Advances in Biometric Person Authentication, pp. 275–287. 2005

[7] David G. Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60, 2004

[8] Business Software Alliance. http://www.bsa.org

[9] Ginger Myles, Christian Collberg. "Detecting Software Theft via Whole Program path Birthmarks."LNCS 3225, pp. 404-415, ISC 2004

[10] Joshphberg, K., Pollack, J., Victoriano, J., &Gitig, O. (2003). Software piracy carries heavy cost for US. Intellectual Property & Technology Law Journal, 15(5), 22-23

[11] E. Levina and P. Bickel, "The Earth Mover's Distance is the Mallows Distance: Some Insights from Statistics," Proc. IEEE Int'l Conf. Computer Vision, vol. 2, 2001

[12] S. Cohen and L. Guibas, "The Earth Mover's Distance under Transformation Sets," Proc. IEEE Int'l Conf. Computer Vision, vol. 2, pp. 1076-1083, 1999

[13] "International software piracy: Analysis of key issues and impacts," Inform. Syst. Res., vol. 9, no. 4, pp. 380–397, Dec. 1998

[14] I. Al-Jabri and A. Abdul-Gader, "Software copyright infringements: An exploratory study of the effects of individual and peer beliefs," Omega, Int. J. Manage. Sci., vol. 25, no. 3, pp. 335–344, June 1997

[15] R. D. Gopal and G. L. Sanders, "Preventive and deterrent controls for software piracy," J. Manage. Inform. Syst., vol. 13, no. 4, pp. 29–47, Spring 1997

[16] Cheng, H. K., Sims, R. R., &Teegen, H. (1997). To purchase or to pirate software: an empirical study. Journal of Management Information Systems, 13(4), 49-60

[17] M. Givon, V. Mahajan, and E. Muller, "Software piracy: Estimation of lost sales and the impact on software diffusion," J. Marketing, vol. 59, no. 1, pp. 29–37, January 1995

[18] M. Fleming and G. Cottrell, "Categorization of faces using unsupervised feature extraction," Proc. IJCNN-90, Vol. 2

[19] Yu Meng and Bernard Tiddeman, "Implementing the Scale Invariant Feature Transform (SIFT) Method", Department of Computer Science University of St. Andrews

7. AUTHORS PROFILE

Srinivas Baggam received M.Tech in (Computer Science & Engineering) from R.V.R & J.C college of Engineering, Guntur, Affiliated to Acharya Nagarjuna University. Currently working as an Assistant Professor in M.V.G.R. College of Engineering. He got two and half years of Industrial and Three and half years in teaching Experience.

Prof. Koduganti Venkata Rao received Ph.D in Computer Science and Engineering from Andhra University, M.Tech in (Computer Science and Technology) from Andhra University and M.Sc (computer science) from Nagarjuna University, 2008, 1999, 1994 respectively. Currently Working as Head of the Department and Professor in Computer Science and Engineering Vignan institute Vishakapatnam.

Prof. P. Suresh Varma received Ph.D. in Computer Science and Engineering with specialization in Communication Networks from Acharya Nagarjuna University. M.Tech in (Computer Science and Technology) from Andhra University in 1998. A.M.I.E (Computer Science and Engineering from Institute of Engineers. M.Sc (Nuclear Physics from Andhra University from 1993.Working as Principal and Professor in Computer Science, Adikavi Nannaya University, Rajahmudry from Oct 2008 to till date.