Mobile Diary & Data Management

Manjunath M[1] & Vandana B S[2]

ABSTRACT

Smartphone application, which supports to record daily activities locally are important in this busy world. People would prefer to record other than textual data. While developing this kind of application memory matters a lot. So compressing recorded files helps save memory. An interactive user interface based on hierarchical media segmentation incorporates user memory and intelligence data. This kind of system for retrieval and summarization of multimedia data for home-like environment continuously captures audio and video. With advances in technology methods for recording human experience have become available and amount of multimedia data that individual's record has greatly increased. Automated capture and retrieval of experiences taking place at home is interesting for several reasons. Captured media can serve as a "memory assistant" to help recall things that were forgotten. The interaction and experiences recorded can provide valuable information for studies.

Keywords

Blowfish, MD5, H.264, AAC, Mobile Diary, Data Management, Automated capture

[1] Dept. of CS&E, KVGCE, SULLIA, DK – 574327
VTU, Belgaum. Karnataka, India, yakruth@yahoo.co.in
[2] Asst. Prof., Dept. of CS&E
KVGCE, SULLIA, DK – 574327, VTU, Belgaum, Karnataka, India, vandanamata@yahoo.com

INTRODUCTION

Technological advances in recent years have meant that digital display media are becoming more “paper-like”. Wireless, lightweight digital displays are now more mobile and portable than ever. The quality and readability of the display screens are, approaching the brightness, resolution and contrast of plain paper. New techniques for stylus-based entry are making possible more paper-like interactions, such as allowing richer and more flexible mark-up and manipulation of digital documents.

Such advances have sparked a flurry of interest in developing digital devices for reading, as evidenced by a growing number of new projects using terms such as “portable document reader”, “electronic book”, “electronic encyclopaedias’’, “ virtual paper”, and “electronic paper”. However, developers of these devices are facing some difficulties in decision-making. If they are to design devices for reading, what are the kinds of reading they should aim to support, which are the people that they should support, and how should they best support these reading activities? The fundamental problem is that the task of “reading” is far too general and ubiquitous: reading takes on a range of forms, is done for a variety of purposes, and is embedded within and related to many other document-based activities. These activities, of course, will also vary depending on whether one is considering reading at work or at home, and will depend on the kind of work environment one considers. The purpose of the study reported in this paper is to begin to answer these questions in a systematic, empirical way, within real work settings. While not discounting for leisure, this study focuses on work-related activities, whether they happen “at work” or in the home. Our approach was to ask subjects to keep daily logs of their document activities, and then to unpack and expand these descriptions through structured interviews. From this diary study comes a rich database of both descriptive materials as well as quantitative measures. In this paper, we present some selected findings addressing the following issues, which we feel have important implications for the design of digital devices for reading:

- A description and taxonomy of the range of reading activities and writing activities that occurred in the working lives of our sample.
- A discussion of common findings across our sample, most notably the predominance of reading in conjunction with writing, the predominance of paper-based document activity, and the degree to which reading activities occurred across multiple, independent reading surfaces.
- A discussion of the ways in which these people in different professions varied with respect to their reading activities.

Throughout these discussions we draw attention to what these findings mean in terms of design requirements for devices, which aim to support online writing and reading.

RELATED WORK

Mobile Diary is a way of keeping/managing journals, photos and voice memos securely with encryption in User’s Mac OS X device. This allows the user to browse their appointments, and mark appointments on Mobile Diary Application pages. Since the device holds the details of each appointment users can be reminded in advance of meetings and appointments. The use of Mac OS X device also introduces flexibility into the format of diary, presenting different views, such as by year, month or week. Unlike a paper diary, the Mobile Diary Application automatically adds new pages when needed, extra room for each day, and retains copies of diaries for years gone past [1].

While diary is useful, the most important advantage of a mobile diary over a paper one is in its ability to be shared. Unlike a paper diary, which is bound to one place, staff can access a mobile diary application over a Mac OS X device network, checking to see when other staffs are free [2]. Moreover, staff can share a diary, allowing a secretary to run a manager's diary, while still allowing the manager access over the computer network. Mobile Diary Application held on desktop computers, then, make booking meetings much easier. A mobile diary allows meetings times to be found, arranged, and confirmed, all from the computer, saving time and bother [3].

There are four key uses of Mobile Diary Application.

Keeping a datebook

As mentioned above, a mobile diary lets the user keep track of their appointments in an electronic form. Advantage here is that the datebook can be viewed in many different formats – by day or by year, for example. Although a user may put information in to the datebook on a day-by-day basis, they can then view their commitments "by year" when wanting to book holidays, or plan overtime. There is no need to have a separate year planner. Users can even keep multiple entries in a single day.

Sharing a datebook

The second usage that mobile diary offer is the ability to share diaries between other datebook users. Post journals as Twitter or Facebook status. User can share photos and videos from mobile diary application.

Viewing free times

While these features help in the management of time, mobile diary also help in booking meetings. Since datebook are on-line, it is possible to view the details of other staff member's free time. This way, it is possible to quickly suggest suitable meeting times. The usual way this is done is by picking a list of meeting participants, and the computer displaying the time where all the participants are free.

Booking meetings electronically

Bringing these features together, most systems allow user to book meetings completely automatically. That is, using the datebook to find a free time, and using the system to send invitations to the meeting's attendees. Most systems send these invitations through electronic mail.

Security and authenticity has been given to mobile device from accessing of information from third person or from private place. To overcome this digital signature has been implemented in mobile terminals. Symmetric cryptography and Asymmetric cryptography are used between key pair of communication for encrypting and decrypting. Use of digital certificates on mobile devices provides high security for data/information on mobile devices [8].

PROPOSED METHOD

The steps followed in developing this application, the market dynamics, selection of app category, defining the feel and nature of the content and its monetization makes it a complicated task. However, despite the challenge, if developers could see through every aspect of the application in a unified manner then a roadmap towards building a great application is secured. The data is stored securely within the application.

Blowfish Algorithm

Blowfish is a keyed, symmetric block cipher, designed in 1993 by Bruce Schneier and included in a large number of cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. However, the Advanced Encryption Standard now receives more attention. Schneier designed Blowfish as a general-purpose algorithm, intended as an alternative to the aging DES and free of the problems and constraints associated with other algorithms. At the time Blowfish was released, many other designs were proprietary, encumbered by patents or were commercial or government secrets. Schneier has stated that, "Blowfish is unpatented, and will remain so in all countries

[9]. The algorithm is hereby placed in the public domain, and can be freely used by anyone."

1. Generate_Subkeys.
2. Encryption:
   Blowfish is a Feistel network (Figure 1) consisting of 16 rounds. The input is a 64-bit data element X.
3. Divide X into two 32-bit halves: X_L, X_R
4. For i = 1 to 16:
5. $X_L = X_L$ XOR P_i
6. $X_R = F(X_L)$ XOR X_R
7. Swap X_L and X_R
8. Next I
9. Swap X_L and X_R (Undo the last swap.)
10. $X_R = X_R$ XOR P_{17}
11. $X_L = X_L$ XOR P_{18}
12. Recombine X_L and X_R (to get the cipher text)
13. Function F:
14. Divide X_L into four 8-bit quarters: a, b, c and d
15. $F(X_R)$ = ((S_1,a + S_2,b mod 232) XOR S_3,c) + S_4,d mod 232
16. Decryption is exactly the same as encryption, except that P_1, P_2, P_{18} are used in the reverse order.

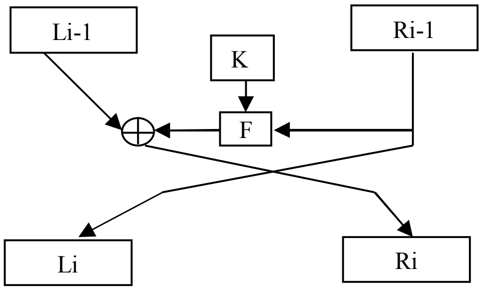


Figure 1: Feistel network

Generate_Subkeys.
The sub keys are calculated using the Blowfish algorithm.
The exact method is as follows:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of P_i (less the initial 3).
2. XOR P_1 with the first 32 bits of the key, XOR P_2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P_{14}). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits.
3. Encrypt the all-zero string with the Blowfish algorithm, using the sub keys described in steps (1) and (2).
4. Replace P_1 and P_2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified sub keys. Replace P_3 and P_4 with the output of step (5).
6. Continue the process, replacing all entries of the P-array, and then all four S-boxes, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required sub keys.

MD5 ALGORITHM

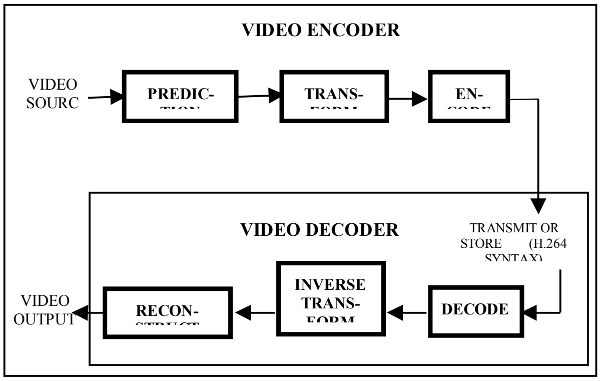
A MD5 hash is created by taking a string of an any length and encoding it into a 128-bit fingerprint. Encoding the same string using the MD5 algorithm will always result in the same 128-bit hash output. MD5 hashes are commonly used with smaller strings when storing passwords, credit card numbers or other sensitive data in databases such as the popular Sqlite3. This tool provides a quick and easy way to encode a MD5 hash from a simple string of up to 256 characters in length [10].

MD5 hashes are also used to ensure the data integrity of files. Because the MD5 hash algorithm always produces the same output for the same given input, users can compare a hash of the source file with a newly created hash of the destination file to check that it is intact and unmodified. A MD5 hash is NOT encryption. It is simply a fingerprint of the given input. However, it is a one-way transaction and as such it is almost impossible to reverse engineer a MD5 hash to retrieve the original string.

```
//M = ($Y_0$, $Y_1$, $Y_2$ ...$Y_{n-1}$), message to hash, after padding
//Each $Y_i$ is a 32-bit word and N is a multiple of 16
MD5 (M)
//Initialize (A, B, C, D) = IV
(A, B, C, D) = (0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476)
For I – 0 to N/16-1
//Copy block I into X
$X_j = Y_{16i+j}$, for j = 0 to 15
//Copy X to W
$W_j = X_{\delta(j)}$, for j = 0 to 63
// initialize Q
($Q_{-4}$, $Q_{-3}$, $Q_{-2}$, $Q_{-1}$) = (A, D, C, B)
//Rounds 0, 1, 2, and 3
Round0(Q, W)
Round1(Q, W)
Round2(Q, W)
Round3(Q, W)
//Each addition is module 232
(A, B, C, D) = (($Q_{60} + Q_{-4}$), ($Q_{63} + Q_{-1}$), ($Q_{62} + Q_{-2}$), ($Q_{61} + Q_{-3}$)) next i
return A, B, C, D
end MD5

Round0 (Q, W)
//Steps 0 through 15
for i = 0 through 15
$Q_i = Q_{i-1} + ((Q_{i-4} + F(Q_{i-1}, Q_{i-2}, Q_{i-3}) + W_i + K_i) <<< S_i)$ Next i
End Round0
```

H.264



H.264 is an industry standard for video compression, the process of converting digital video into a format that takes up less capacity when it is stored or transmitted. Video compression (or video coding) is an essential technology for applications such as digital television, DVD-Video, mobile TV, videoconferencing and Internet video streaming. Standardizing video compression makes it possible for products from different manufacturers (e.g. encoders, decoders and storage media) to interoperate. An encoder converts video into a compressed format and a decoder converts compressed video back into an uncompressed format.

Recommendation H.264: Advanced Video Coding is a document published by the international standards bodies ITU-T (International Telecommunication Union) and ISO/IEC (International Organisation for Standardisation / International Electrotechnical Commission). It defines a format (syntax) for compressed video and a method for decoding this syntax to produce a displayable video sequence. The standard document does not actually specify how to encode (compress) digital video this is left to the manufacturer of a video encoder but in practice the encoder is likely to mirror the steps of the decoding process. Figure 2 shows the encoding and decoding processes and highlights the parts that are covered by the H.264 standard.

Figure 2: H.264 Video encoding and decoding process.

Advanced Audio Coding (AAC)

Advanced Audio Coding (AAC) is a standardized, loss compression and encoding scheme for digital audio. Designed to be the successor of the MP3 format, AAC generally achieves better sound quality than MP3 at similar bit rates [12]. AAC has been standardized by ISO and IEC, as part of the MPEG-2 and MPEG-4 specifications [13] [14]. Part of the AAC known as High Efficiency Advanced Audio Coding (HE-AAC) which is part of MPEG-4 Audio is also adopted into digital radio standards like DAB+ and Digital Radio Mondiale, as well as mobile television standards DVB-H and ATSC-M/H.

AAC supports inclusion of 48 full-bandwidth (up to 96 kHz) audio channels in one-stream plus 16 low frequency effects (LFE, limited to 120 Hz) channels, up to 16 "coupling" or dialog channels, and up to 16 data streams. The quality for stereo is satisfactory to modest requirements at 96 kbit/s in joint stereo mode; however, hi-fi transparency demands data rates of at least 128 kbit/s (VBR). The MPEG-2 audio tests showed that AAC meets the requirements referred to as "transparent" for the ITU at 128 kbit/s for stereo, and 320 kbit/s for 5.1 audio.

AAC is a wideband audio coding algorithm that exploits two primary coding strategies to dramatically reduce the amount of data needed to represent high-quality digital audio.

1. Signal components that are perceptually irrelevant are discarded.
2. Redundancies in the coded audio signal are eliminated.

The actual encoding process consists of the following steps:

- The signal is converted from time-domain to frequency-domain using forward modified discrete cosine transform (MDCT). This is done by using filter banks that take an appropriate number of time samples and convert them to frequency samples.
- The frequency domain signal is quantized based on a psychoacoustic model and encoded.
- Internal error correction codes are added.
- The signal is stored or transmitted.
- In order to prevent corrupt samples, a modern implementation of the Luhn mod N algorithm is applied to each frame [15].

PERFORMANCE DISCUSSION

There are plenty of applications available to record daily activities, which store data in unsecured way. Our application uses the available encryption technique to store the data in secure way. As data are stored locally using sqlite database, memory has to be reduced. H.264 compression technique is used to compress the video data and AAC encoding is used to compress the audio data. By using compression technique huge amount of memory can be saved on the device’s disk. The image captured is compressed to required resolution. All pictured and audios files are zipped. Searching saved files is easy. It first locates zipped files and then gets the content of the given file name. This saves the time of searching. Each user’s data are stored separately on disk. No access permission is granted to view another user’s data.

CONCLUSION

Currently this application supports compression of textual data, pictures and audios. Enhancements have to be done to video files. Also cloud feature needs to be implement to store the data. This is a mobile application and desktop application and is a form of personal information management system that allows users to record notes and ideas in a digital format. User records are kept secure. Deaf and dumb people can also use this application. Even they can store and share audio/image files. Long-term motivation could also be promoted with social support features, such as data sharing and encouraging messages. Even very busy people can enter text by speaking i.e., speech recognition. This feature helps to enter textual data very fast. If this feature supports other than English then number of users increases.

REFERENCES

[1] Kim Mooseop, Ryou Jaecheol, and Jun Sungik, “Compact Implementation of SHA-1 Hash Function for Mobile Trusted Module,” Information Security Applications, Volume 5379/2009, and pp: 292-304, 2009. http://www.springerlink.com/content/f2v64u64324w6q47/

[2] SG Punja. "Mobile device analysis". Small Scale Digital Device Forensics Journal. (2008)

[3] M. Becher et al., “Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of

[4] Mobile Devices,” *Proc. 2011 IEEE Symp. Security and Privacy,* IEEE CS, 2011, pp. 96–111.

[5] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4249824&tag=1

[6] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5165553

[7] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5510894&tag=1

[8] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5783997

[9] http://en.wikipedia.org/wiki/Blowfish_%28cipher%29#The_algorithm

[10] http://www.cocoawithlove.com/2009/07/hashvalue-object-for-holding-md5-and.html

[11] Apple Corporation. "iOS Security". 2012. http://images.apple.com/ipad/business/docs/iOS_Security_May12.pdf

[12] JeffreyB.Layton."AnatomyofSSDs"LinuxMagazine.2009. http://www.linux-mag.com/id/7590/2/

[13] T. Yamazaki, “Ubiquitous Home: Real-Life Testbed for Home Context-Aware Service,” *Proc. Tridentcom2005*, IEEE Press, 2005, pp. 54-59.

[14] Sinha, D. and Johnston, J. D., "Audio compression at low bit rates using a signal adaptive switched filter bank", IEEE ASSP, 1996, pp. 1053-1057.

[15] Johnston, J. D., Sinha, D., Dorward, S. and Quackenbush, S., "AT&T perceptual audio coder (PAC)" in Collected Papers on Digital Audio Bit-Rate Reduction, Gilchrist, N. and Grewin, C. (Ed.), Audio Engineering Society, 1996.

[16] Herre, J. and Johnston, J. D., "Enhancing the performance of perceptual audio coders by using temporal noise shaping", AES 101st Convention, no. preprint 4384, 1996

[17] US patent application 20070297624 Digital audio encoding