Scientia Iranica B (2015) 22(4), 1586–1593

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions B: Mechanical Engineering
www.scientiairanica.com

An approach for secure data exchange: Experiments on android-based mobile device

C.-K. Ke* and Z.-H. Lin

Department of Information Management, National Taichung University of Science and Technology, Taichung City, 40401, Taiwan ROC.

Received 22 May 2014; accepted 9 March 2015

KEYWORDS
Secure data exchange;
Android-based mobile device;
Advanced encryption standard;
Triple data encryption standard;
Data encryption standard.

Abstract. Advances in information and communication technology have made mobile devices very easy for people to share and exchange information. However, such exchanges of data may require privacy, or users may not want that data to be publically accessible. Data exchange security has therefore also become very important in the development of such devices, and, in fact, in mobile wireless network environments, in which the secure exchange of data between mobile devices has become an important issue. This study proposes a novel approach for the secure exchange of data between discrete mobile devices. Cryptographic techniques, including advanced encryption standard, triple data encryption standard, and data encryption standard, are enforced in a data exchange process, and a popular android-based mobile device is used to test the designed mechanism for secure data exchange. The experiment results show the encryption/decryption time of each cryptographic technique working on android-based mobile devices. A comparison of the experiment results also shows which cryptographic technique is appropriate for data exchange between discrete android-based mobile devices. The contribution of this work is to explore the security issue of the above-mentioned data, and to recommend an appropriate cryptographic technique for data exchange between discrete mobile devices.

1. Introduction

Recent advances in information and communication technology have improved the quality of life, and ease of access to information for many people around the world. With these advancements in computing power and high speed network, vast amounts of information are readily and rapidly available to many users. As a result, users are able to more easily perform work or educational tasks. The emergence of mobile devices, such as smart phones and tablet computers has allowed people to more ubiquitously create life data, such as travel notes, photographs, melodies, etc. In addition, the traditional limited resources such as computing power, memory, storage and battery life have been improved by cloud computing, remote storage or application solutions. Mobile devices have thus had a revolutionary influence on daily life and our modern lifestyle.

There is a significant user demand to be able to exchange or share life data with friends in a social community, e.g. Facebook, twitter or Google+, etc. This social behavior of data exchange maintains friendship connections. Due to their mobility and popularity, mobile devices have made it extremely easy for people to share and exchange such information. Some data transmission mechanisms, e.g. Bluetooth [1], Session Initiation Protocol (SIP) [2], Wifi-Direct [3], mobile peer-to-peer (P2P) protocol [4], and cloud applications,

*. *Corresponding author. Tel.: 886-4-2296623; Fax: 886-4-22196311*
E-mail addresses: ckk@nutc.edu.tw (C.-K. Ke); s1800b102@nutc.edu.tw (Z.-H. Lin)

allow users to choose the mobile devices they wish to communicate with when engaging in such social behavior. These data transmission mechanisms allow mobile devices to more closely exchange data with each other over a mobile wireless communication network [5]. With the growing popularity and importance of this kind of data exchange, it is important to note that users may require that their exchanged data remain private, or at least not be publically accessible. Data protection has thus become a critical factor in data exchange processes [6]. Some cryptographic techniques, e.g. Data Encryption Standards (DES) [7], Triple Data Encryption Standards (Triple-DES) [8], Advanced Encryption Standard (AES) [9], tiny encryption algorithm and RC5 [10], are available for the protection of data in wireless communication networks. However, in mobile wireless communication networks, the secure exchange of data between mobile devices remains problematic.

Our previous study [5] focused on the influences of data transmission protocol, including Bluetooth, Session Initiation Protocol (SIP), and mobile P2P protocol: Juxtapose (JXTA), on a data exchange process between mobile devices over a mobile wireless communication network. The experiment showed that JXTA was a reasonable solution for data exchange between mobile devices. Another study [11] was then conducted to discuss the optimal selection of candidate mobile devices over a JXTA communication network. It was found that data protection was a critical problem in a data exchange process. This study therefore explores the data exchange status among discrete mobile devices based on the security consideration. Cryptographic techniques including AES, Triple-DES and DES are used to encrypt/decrypt data in a data exchange process over a mobile wireless communication network. A popular android-based mobile device is used as a test device, and a mechanism is designed to carry out the secure data exchange examination. The experiment shows the encryption/decryption time of each cryptographic technique working on the android-based mobile device. A comparison of the experiment results also shows which cryptographic technique is most appropriate for data exchange among discrete android-based mobile devices. The contribution of this work is to explore the security issue of the above-mentioned data, and to recommend an appropriate cryptographic technique for the exchange of data among discrete mobile devices.

The remainder of this paper is organized as follows. Section 2 introduces related literature. Section 3 presents an approach for secure data exchange over a mobile wireless communication network. The experiments and relevant discussions are shown in Section 4, and, finally, Section 5 presents conclusions.

2. Related works

There are many symmetric data encryption algorithms, including data encryption standard, triple data encryption standard, advanced encryption standard, tiny encryption algorithm, RC5, etc. In this section, studies on symmetric data encryption are reviewed, including data encryption standard, triple data encryption standard and advanced encryption standard.

2.1. Data Encryption Standard (DES)

Data encryption prevents non-privileged or illegal users accessing data. Its procedure uses special algorithms or mathematical calculations to convert plain text into an unreadable form in which the important data cannot be directly recognized, called the cipher text. The cipher text must be converted again by the same algorithm for it to be read. Only authorized users now are able to convert the data, and in this way access by unauthorized users is prevented, providing data security. Symmetric data encryption uses the same key for encryption and decryption. In addition, because the key is symmetric, the encryption process is faster. Symmetric data encryption is often used to handle large amounts of data. However, the key must be replaced regularly in order to maintain security and prevent access by illegal users.

Data Encryption Standard (DES) is an encryption algorithm developed by American International Business Machine (IBM) in 1970. The United States Federal Government accepted it as the Federal Information Processing Standard (FIPS) in 1977 [7]. Data encryption standard is a block encryption algorithm, accomplished by several encryption techniques. The purpose is to use diffusion and confusion mode to transform original plain text into a scattered cipher text. This prevents crackers from using statistics or other similar methods to obtain the plain text. The encryption procedure is to cut the plain text into 64 bits as a unit. A 56-bit key is used to conduct the shift computation 16 times. Finally, a 64-bit output cipher text is produced [10,12].

2.2. Triple Data Encryption Standard (Triple-DES)

Triple Data Encryption Standard (Triple-DES) was proposed to solve the key length problem of DES. The key length of DES is only 56 bits. It is possible to crack the DES key by the brute force method of guessing 2^{56} kinds of data, which has become far easier to achieve with modern computer hardware technology. To enhance the security of DES, in 1999, the National Institute of Standards and Technology (NIST) extended and modified the encryption base of DES to form Triple-DES, which then replaced DES [8]. Some procedures of Triple-DES differ from those of DES, but the procedure for plain text processing in Triple-DES

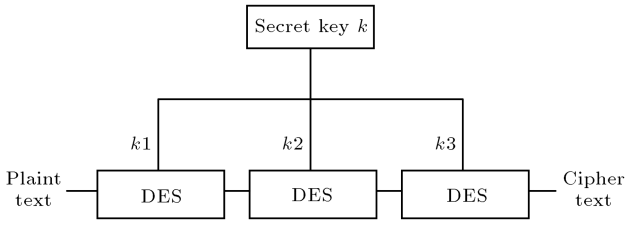

Figure 1. The encryption schema of Triple-DES.

and DES is the same. The procedure divides the plain text into 64 bits as a unit, and then uses three 56-bit keys to enforce three DES encryption procedures to produce the 64-bit cypher text [8]. Figure 1 illustrates the encryption schema of Triple-DES, and its secret key includes three keys. The length of a key is 168-bits. $k1$, $k2$, $k3$ represent three keys, and the length of each key is 56-bits. In an encryption procedure, there are two kinds of key length to choose from, either 112 bits or 168 bits. In addition, the DES of $k2$ in the Triple-DES schema can be used as an encryption or a decryption process [10].

Therefore, Triple-DES has four kinds of key usages (E is encryption, and D is decryption):

(1) *EEE* (three keys). Use three different keys (key length is 168 bits) to enforce three encryption actions;

(2) *EDE* (three keys). Use three different keys to first enforce an encryption action. Then use three different keys to enforce an encryption action. Finally, use three different keys to enforce an encryption action;

(3) *EEE* (two keys). Use two different keys (key length is 112 bits); select one of two keys to clone and produce the third key. Select any two keys from the three keys to set as the same key and perform three encryptions;

(4) *EDE* (two keys). Use two different keys (length of key is 112 bits) for encryption/decryption. Select one of the two keys to clone and produce the third key. Select any two keys from the three keys to set as the same key. The first step carries out the encryption; second step carries out the decryption, and third step carries out the encryption.

2.3. Advanced Encryption Standard (AES)

With the rapid development of computer technology, Triple-DES was also inevitably replaced. In order to meet future demand, the National Institute of Standards and Technology (NIST) began to openly solicit the advanced encryption standard in 1997. NIST announced the functionality requirements for the advanced encryption standard specification, and hoped that cryptography experts would conduct the relevant research. After three years, NIST chose Rijndael, designed by Joan Daemen and Vincent Rijmen, as the AES algorithm in 2000 [13,14].

The advanced encryption standard algorithm is an iteration computation encryption algorithm. Each encryption enforces four procedures, including *AddRoundKey*, *SubByte*, *ShiftRow* and *MixColumn* [9,10].

(1) *AddRoundKey*: Use the state and round key to enforce the XOR computation; each round will change the state;

(2) *SubByte*: After the XOR computation, put the byte into a 16×16 two dimension matrix to enforce the nonlinear byte exchange computation;

(3) *ShiftRow*: Circle shift the row of state;

(4) *MixColumn*: Use linear transformation mix 4 bytes of each column.

The number of rounds is determined by the length of the encryption block and the length of the secret key. The round key is produced from all round keys before encryption or decryption, then selected based on the number of rounds. The state is a $4 \times Nb$ matrix to store the bit value, and Nb is obtained from the results of the encryption block length, which is divided by 32 bits. In other words, the data is divided into Nb blocks, and each block is 32 bits in length. For example, a 256-bit data block can be divided into 8 blocks. The state can then be presented as a 4×8 matrix. The secret key is the key used for encryption, which is a $4 \times Nk$ matrix, and the length of Nk is obtained from the key length, which is divided by 32 bits. For example, the length of a key is 128-bits, and it can be divided into 4 blocks. The secret key can be presented as a 4×4 matrix. Figure 2 shows the encryption procedure of AES.

Review of the previous literature shows that many researches use symmetric encryption to encrypt data and enhance data security. Jiujun Cheng et al. (2009) [15] combined HMAC One Time Password algorithm (HOTP) and mobile system time to produce an authentication image for user identification authentica-

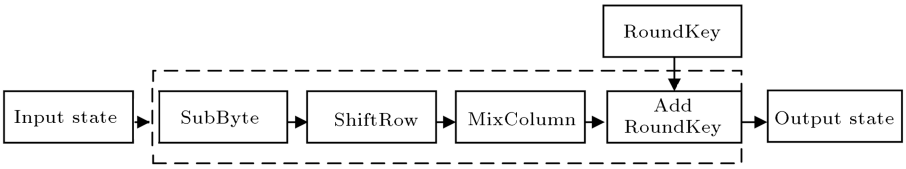

Figure 2. The encryption procedure of AES [10,13].

tion. DES encrypts the authentication image to protect the data from counterfeiting. Gongbin Qian et al. (2009) [16] proposed a new image encryption schema based on DES and chaotic Encryption to protect image security and reliability. Joan Arnedo-Moreno and Herrera Joancomartí (2009) [6] investigated the security of applications developed by Juxtapose (JXTA) peer-to-peer protocol and found that a message transmitted in the protocol should be encrypted by DES in advance in order to preserve the security of the message transmission process. In addition, DES is often used in E-mail, online transactions and ATM network transfers [12], while eXtensible Markup Language (XML) uses symmetric data encryption to encrypt the tag, including DES, Triple-DES and AES [4,17]. In a mobile wireless network environment, users require a security mechanism to protect their exchanged data. Symmetric encryption is thus a viable method of maintaining data security when users transmit or share data on mobile devices.

2.4. Social data exchange over a mobile wireless network

Social Networking (SN) [18,19] is the concept of a real human community based in a virtual environment on the Internet. According to their interests or preferences, people construct various social networks to join, connect, and share information together. Recently, service provider have begun to offer various social network platforms for people to store files and share information with each other, e.g. Facebook, Twitter, Google+, etc. A social network is a peer group. Peer can be a user, mobile devices, computers or servers. Generally, a social network is a peer-to-peer (P2P) mechanism. Peers thus own files or information, and can share these files or information with each other in a group based on a specific data transmission technique, e.g. P2P transmission [4,20,21], cloud transmission, Bluetooth [1], Wifi-Direct [3], Session Initiation Protocol (SIP) [2], etc. Various mobile social network applications have also emerged, e.g. remote healthcare and e-Learning. In America, social networks are combined with Personal Health Record Systems (PHRS) for people to share self-healing experiences of the same disease [22]. German Fraunhofer FOKUS developed an Electronic Case Record (ECR) sharing system mechanism. The ECR mechanism is a P2P sharing platform. Each hospital in the treatment social network plays a role, sharing and using the ECR, and exchanging ECR information on the social network [23]. If a mobile device is used for data exchange, securely exchanging data between mobile devices is clearly an important issue. Reasonable solutions are therefore required for the enforcement of secure data exchange over mobile wireless communication networks.

3. Approach for secure data exchange over a mobile wireless communication network

The recent rapid development of mobile device technology has facilitated large scale and ubiquitous creation and sharing of personal user data and information. The mobility and popularity of such devices means that people are more easily able to share and access such information. With the growing importance of the data exchange process, the security of data has also become a vital aspect of data exchange. However, while there are many methods for protecting data in wireless networks, the secure exchange of data between mobile devices in a mobile wireless communication network remains an issue to be resolved. Our previous study [5] discussed the influences of data transmission protocol, including Bluetooth, Session Initiation Protocol (SIP), and mobile peer-to-peer protocol: Juxtapose (JXTA), on a data exchange process between mobile devices. Another study [11] was then conducted to explore the optimal selection of candidate mobile devices over a JXTA communication network. It was found that data security was a critical problem in the data exchange process. This work proposes an approach using symmetric data encryption AES to encrypt plain text into a cypher text. The process makes the data more secure for transmission or exchange over a mobile wireless network. The symmetric encryption key is used to decrypt the cypher text and obtain the plain text. Then the proposed approach is then illustrated for experimental implementation, including a key generator expansion, data encryption and data decryption algorithms.

3.1. The key generator expansion algorithm

Before applying a symmetric encryption/decryption process, a symmetric key must be generated. The key generation requires inputting a key seed and configuring the key length. The generated key is used to generate a round key, including key expansion and round key selection periods. In this work, the permission and user profile are used as the key seed to generate key. The key is used for secure mobile data exchange over a mobile wireless network. The relevant parameter definition of key generator expansion algorithm [10] is shown in Table 1.

The key generator expansion algorithm [10] is shown in Figure 3. The initial step of the algorithm is to input permission, user profile (*PUInfo*) and the expected length of a generated key (*len*). The algorithm then determines whether the input values of permission and user profile are null. If the input value (*PUInfo*) is null, then it returns a null value message. If the input value (*PUInfo*) is not null, then it configures the key generation seed (*seed*) of the permission and user profile

The key generator expansion algorithm
Input: Permission and user profile ($PUInfo$)
Output: Expansion key matrix (W) or message

```
00   KeyGeneratorExpansion (PUInfo)
01   {
02     if (PUInfo is not empty)
03     {
04       Set up seed value that is PUInfo;
05       Set up key length that is len;
06       Generate the role's key K;
07       Nk = len/32;
08       Nb = State/32;
09       for (i = 0; i < Nk, i++)
10       {
11         W[i] = (Key[4 × i], Key[4 × i + 1], Key[4 × i + 2], Key[4 × i] + 3)
12         for (i = Nk, i < Nb × (Nr + 1); i++)
13         {
14           temp=W[i − 1]
15           if (i%Nk = 0)
16             temp=ByteSub (RotByte(temp))^Rcon[i/Nk] ;
17             W[i] = W[i − Nk]^temp;
18         }
19       }
20       return W;
21     }
22     Else
23     {
24       return the null value;
25     }
26   }
```

Figure 3. The key generator expansion algorithm (Pseudo Code).

Table 1. The relevant parameter definition of key generator expansion algorithm.

Parameter definition	
$PUInfo$	Permission and user profile;
Len	The length of key shall be 192 bits;
$Seed$	The seed to generate a key;
K	The generated symmetric key;
$State$	The encrypted data block shall be 128 bits;
Nb	The bock length of each encryption;
Nk	The key block;
$ByteSub()$	S-Box exchange function;
$RotByte$	The function for the left cycle of a word-set;
W	The array of a key expansion;
$Rcon$	The array of a round.

Table 2. The relevant parameter definition of data encryption algorithm.

Parameter definition	
K	The generated symmetric key;
Nr	The number of round;
$State$	The calculating matrix which stores the encrypted information and encrypted results generated from final round;
$ExpKey$	The array of key expansion;
$AddRoundKey()$	Round key encryption execution;
$ByteSub()$	Byte exchange;
$ShiftRow()$	Shift row;
$MixColumn()$	Mix column.

($PUInfo$), and sets the expected key length (len). The permission and user profile ($PUInfo$) and key length (len) are used to generate a symmetric key (K). The generated key is used to carry out an expansion process. The expansion process calculates Nk and Nb from a symmetric key (K) and encryption block ($State$), and then the process enters a for-loop. If the key expansion value is larger than the number of key blocks (Nk), the process stops the for-loop and inputs the symmetric key (K) to the expansion key array (W). The process then uses a for-loop to calculate the sub key block in an expansion key array (W) until $Nb \times (Nr + 1)$ number of times to stop. The expansion key array is then returned.

3.2. Data encryption algorithm

The relevant parameter definition of data encryption algorithm [10] is shown in Table 2.

The data encryption algorithm [10] is shown in Figure 4.

When the approach transfers the information or data, it inputs the plain text of the data to the matrix ($State$) and the symmetric key (K) of the data to apply the encryption process. The approach determines whether the input matrix ($State$) and symmetric key (K) is null. If the input value is null, then it returns a null message. If the input value is not null, it uses the matrix ($State$) and initial round key ($ExpKey[0]$) to apply round key encryption ($AddRoundKey$). The

```
Data encryption algorithm
Input: Value (State), symmetric key (K)
Output: encrypted value (State) or message

00   DataEncrypt (State, K)
01   {
02     if (State and K are both empty)
03       {
04         Add RoundKey (State, ExpKey [0]);
05         for (i = 1; i < Nr; i++)
06         {
07           ByteSub (State);
08           ShiftRow (State);
09           MixColumn (State);
10           AddRoundKey (State, ExpKey[i])
11         }
12           ByteSub (State)
13           ShiftRow (State)
14           AddRoundKey (State, ExpKey[Nr]);
15                   return (State);
16         }
17         else
18         {
19           return the null value;
20         }
21    }
```

Figure 4. The data encryption algorithm (Pseudo Code).

```
Data decryption algorithm
Input: Value (State), symmetric key (K)
Output: Decrypted value (State) or message

00   DataDecrypt (State, K)
01   {
02     if (State is not empty and K is not empty)
03     {
04       AddRoundKey (State, ExpKey[Nr]);
05       for (i = Nr − 1; i > 0; i − −)
06       {
07         Inv Shift Row (State);
08         Inv ByteSub (State);
09         Add RoundKey (State, ExpKey[i]);
10         Inv MixColumn (State)
11       }
13         Inv ShiftRow (State)
12         Inv ByteSub (State)
14         Add RoundKey (State, ExpKey[0]);
15         return (State);
16       }
17       else
18       {
19         return the null value;
20       }
21   }
```

Figure 5. The data encryption algorithm (Pseudo Code).

Table 3. The relevant parameter definition of data decryption algorithm.

Parameter definition	
K	The generated symmetric key;
Nr	The number of round;
$State$	The calculating matrix which stores the decrypted information and decrypted results generated from final round;
$ExpKey$	The array of key expansion;
$AddRoundKey()$	Round key decryption execution;
$InvByteSub()$	Invert byte exchange;
$InvShiftRow()$	Invert shift row;
$InvMixColumn()$	Invert mix column.

for-loop executes until Nr round to stop, or the matrix carries out byte exchange, shift row, mix column and round key encryption (*AddRoundKey*), until the value of the matrix (*State*) is the final encrypted results.

3.3. Data decryption algorithm

The relevant parameter definition of data decryption algorithm [10] is shown in Table 3.

The data decryption algorithm [10] is shown in Figure 5. When the approach receives the encrypted data, it copies the cypher text of the data to the matrix (*State*), and inputs the symmetric key (K) of the data to apply the decryption process. The approach determines whether the input matrix (*State*) and symmetric key (K) are null. If the input value is null, then it returns a null message. If the input value is not null, it uses the matrix (*State*) and final round key (*ExpKey[Nr]*) to apply round key encryption (*AddRoundKey*). The for-loop executes Nr -1 rounds and each round -1 until it reaches 0 round and stops, or the matrix carries out invert shift row (*InvShiftRow*), invert byte exchange (*InvByteSub*), round key encryption (*AddRoundKey*) and invert mix column (*InvMixColumn*). When it reaches the 0 round, it carries out the round key encryption (*AddRoundKey*), invert shift row (*InvShiftRow*) and invert byte exchange (*InvByteSub*), until the value of the matrix (*State*) is the final decrypted result.

4. Experiments and result discussions

Our previous study [5] discussed the influences of data transmission protocol on a data exchange process between mobile devices. Another study [11] was then conducted to explore the optimal selection of candidate mobile devices over a JXTA communication network. Based on the researches [5,11], we have constructed an experimental platform for relevant services or approaches implementation. The development environment used is Google Android SDK 4.0.3 [24], Google Android Development Tools (ADT) version 20.0.3 (Google, 2012), Java Development Kit (JDK) version 1.7 and Eclipse Classic version 4.2.1. The peer to peer protocol library is JXTA version 2.6. The encryption schema used in this experiment includes Advanced Encryption Standard (AES), triple data encryption standards (Triple-DES) and Data Encryption Standards (DES). The sizes of test data are 1 KB, 10 KB, 100 KB, 1 MB, 10 MB, 20 MB, 50 MB, 100 MB and 200 MB. The encryption and decryption times of the encryption schema are compared. The experiment over a testing platform is Asus Eee Pad Transformer TF201, and the library is security.jar and crypto.jar provided by Java. Table 4 is the comparison of encryption times. Table 5 is the comparison of decryption times. The units used are milliseconds.

Table 4. Comparison of encryption times (milliseconds).

File size	AES	Triple-DES	DES
1 KB	3	14	8
10 KB	18	74	28
100 KB	178	717	247
1 MB	1735	4237	2449
10 MB	6270	22746	9551
20 MB	10375	50212	17641
50 MB	22049	120195	42772
100 MB	41707	231628	81583
200 MB	82191	503941	163045

Table 5. Comparison of decryption times (milliseconds).

File size	AES	Triple-DES	DES
1 KB	3	6	5
10 KB	19	27	33
100 KB	161	289	365
1 MB	1658	2806	3075
10 MB	5677	27186	14991
20 MB	10000	54318	26413
50 MB	25588	135572	60024
100 MB	52829	271217	132126
200 MB	107390	542292	321813

Some scholars have illustrated the differences and comparisons of various symmetric encryption algorithms. Alanazi et al. (2010) [25] compared some factors of the AES, Triple-DES and DES, including key length, encryption schema, the block size for encryption, encryption algorithm proposed year, feature, security, etc. The security degree of AES is the highest. Elminaam et al. (2009) [26] measured the encryption/decryption times of AES, Triple-DES and DES. AES uses the shortest time to apply the encryption, and DES uses the shortest time to apply the decryption. Our previous study [5] conducted some experiments to show the influences of data transmission protocol on a data exchange process between mobile devices over a mobile wireless communication network. An optimal selection of candidate mobile devices was then proposed for application over a mobile P2P communication network [11]. Data protection is a critical factor in data exchange processes. The experiments in this study show that the encryption/decryption speed of AES is higher than that of DES, and DES is faster than Triple-DES in a mobile device over a mobile wireless communication network.

5. Conclusion

Our previous study focused on the influences of data transmission protocol on a data exchange process between mobile devices over a mobile wireless communication network. Another study was then conducted to discuss the optimal selection of candidate mobile devices. It was found that data protection is a critical factor in a data exchange process. This work, therefore, proposes an approach for the secure exchange of data between discrete mobile devices. Cryptographic techniques including Advanced Encryption Standard (AES), triple data encryption standard (Triple-DES) and Data Encryption Standard (DES) were applied in a data exchange process. A popular android-based mobile device was used as a test device, and a mechanism was designed to carry out the secure data exchange examinations. The experiment results show the encryption/decryption time of each cryptographic technique working on an android-based mobile device. The comparison of the experiment results also shows which cryptographic technique is appropriate for data exchange among discrete android-based mobile devices. The contribution of this work is to explore the mobile data exchange security issue, and to recommend a suitable cryptographic technique for exchanging data among discrete mobile devices. Future work should focus on advanced cryptographic techniques to make the proposed approach more secure.

Acknowledgments

This research was supported in part by the National Science Council of Taiwan (Republic of China) with a NSC grant 102-2410-H-025-017.

References

1. Bluetooth Special Interest Group "Bluetooth specification version 4.0" (2010).
2. Handley, M., Schulzrinne, H., Schooler, E. and Rosenberg, J. "SIP: Session Initiation Protocol", RFC 2543 (1999).
3. Wi-Fi Alliance "Wi-Fi peer-to-peer (P2P) technical specification" (2010).
4. Li, G. "Project JXTA: A technology overview" (2001). http://www.jxta.org/project/www/docs/TechOverview.pdf.
5. Ke, C.K. and Lin, Z.H. "Secure resource synchronization of mobile peer-to-peer techniques: experiments on the android platform", *Lecture Notes in Electrical Engineering*, **308**, pp. 289-294 (2014).
6. Arnedo-Moreno, J. and Herrera-Joancomartí, J. "A survey on security in JXTA applications", *Journal of Systems and Software*, **82**(9), pp. 1513-1525 (2009).
7. National Institute of Standards and Technology, *Data Encryption Standard (DES)*, Federal Information Processing Standard Publication 46-2 (1976).
8. National Institute of Standards and Technology, *Data Encryption Standard (DES)*, Federal Information Processing Standard Publication 46-3 (1999).

9. National Institute of Standards and Technology, *Specification for the Advanced Encryption Standard (AES)*, Federal Information Processing Standard Publication, 197 (2001).
10. William, S, *Cryptography and Network Security: Principles and Practice*, Fifth Edn., Prentice Hall Press Upper Saddle River, NJ, USA (2010).
11. Ke, C.K. and Lin, Z.H. "An optimal mobile service for telecare data collection", *Int. Conf. on Business, Information, and Cultural Creative Industry*, Taipei, Taiwan (2014).
12. Raphael, C.W.P. "Reducing the exhaustive key search of the Data Encryption Standard (DES)", *Computer Standards & Interfaces*, **29**, pp. 528-530 (2007).
13. Jason, V.D. and José, G.D.F. "FPGA schemes for minimizing the power-throughput trade-off in executing the Advanced Encryption Standard algorithm", *Journal of Systems Architecture*, **56**(2-3), pp. 116-123 (2010).
14. Daemen, J. and Rijmen, V., *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer-Verlag, pp. 1-238 (2002).
15. Cheng, J.J., Zhang, F., Yu, K.F. and Ma, J. "The dynamic and double encryption system based on two-dimensional image", *Int. Conf. on Computational Intelligence and Security*, Beijing, pp. 458-462 (2009).
16. Qian, G.B., Jiang, Q.F. and Qiu, S.S. "A new image encryption scheme based on DES algorithm and Chua[s circuit", *Int. Workshop on Imaging Systems and Techniques*, pp. 168-172 (2009).
17. Sea, C.S. and Ng, K.S. "A file-based implementation of XML encryption", *5th Malaysian Conference in Software Engineering*, Johor Bahru, pp. 418-422 (2011).
18. Ariyarathna, H.M.T.A. and Ranasinghe, D.N. "JXTA based parallel service invocation model for peer to peer web service composition", *Industrial and Information Systems*, pp. 1-6 (2012).
19. Cachia, R., Compañó, R. and Da Costa, O. "Grasping the potential of online social networks for foresight", *Technological Forecasting and Social Change*, **74**(8), pp. 1179-1203 (2007).
20. Barolli, L. and Xhafa, F. "JXTA-overlay: A P2P platform for distributed, collaborative, and ubiquitous computing", *IEEE Transactions on Industrial Electronics*, **58**(6), pp. 2163-2172 (2011).
21. Tsai, F.S., Han, W.C., Xu, J.W. and Chua, H.C. "Design and development of a mobile peer-to-peer social networking application", *Expert Systems with Applications*, **36**(8), pp. 11077-11087 (2009).
22. Carrión, I., Alemán, J.L.F. and Toval, A. "Personal health records: New means to safely handle health data?", *IEEE Computer Magazine*, pp. 27-33 (2012).
23. Kuhlisch, R., Kraufmann, B. and Restel, H. "Electronic case records in a box: Integrating patient data in healthcare networks", *IEEE Computer Magazine*, pp. 34-40 (2012).
24. Google Android SDK, http://developer.android.com/sdk/index.html
25. Alanazi, H.O., Zaidan, B.B., Zaidan, A.A., Jalab, H.A., Shabbir, M. and Al-Nabhani, Y. "New comparative study between DES, 3DES and AES within nine factors", *Journal of Computing*, **2**(3), pp. 152-157 (2010).
26. Elminaam, D.S.A., Kader, H.M.A. and Hadhoud, M.M. "Performance evaluation of symmetric encryption algorithms", *Communications of the IBIMA*, **8**, pp. 58-64 (2009).

Biographies

Chih-Kun Ke received the MS degree in Computer Science and Information Engineering from National Taiwan University of Science Technology, Taiwan, in 1999, and the PhD degree from the Institute of Information Management, National Chiao Tung University, Taiwan, in 2006. He is currently an Associate Professor of the Department of Information Management, National Taichung University of Science and Technology. His research interests include intelligent information systems, knowledge support systems, mobile and web services.

Zheng-Hua Lin received the MS degree in Information Management of National Taichung University of Science Technology, Taiwan. His research interests include information systems, role-based access control, information security, and mobile app development.