Online Identity Theft and Its Prevention Using Threshold Cryptography

Syedur Rahman[1] **, A. M. Ahsan Feroz**[2]**, Md. Kamruzzaman**[3] **and Md. Azimuddin Khan**[4]
[1]*Jaxara IT Limited, Dhaka, Bangladesh*
[2]*M&H Informatics, Dhaka, Bangladesh*
[3]*Jaxara IT Limited, Dhaka, Bangladesh*
[4] *Business Intelligence in Axiata, Dhaka, Bangladesh*

Summary
The increase in online activities and e-commerce requires user's identification information which leads to certain identity theft have becomes a widespread computer security issue. Identity theft is a term used to refer to a fraudulent activity that involves stealing money or getting benefits by pretending to be someone else. It is the misuse of personal information and identity. This paper incorporated current research on identity theft attacks and prevention techniques to find a solution to ensure higher security in encrypting finger prints in the identification card. This paper discusses about identity thefts and methods of identity theft prevention in computing and networked environments. It focuses on use of biometrics in identity authentication mainly fingerprints. Using Threshold Cryptography, the efficiency of encryption of fingerprints was researched here.

Key words:
Cryptography, Threshold Cryptography, Identity Theft

1. Introduction

The Internet and World Wide Web have become important parts of most people's lives. Users regularly conduct various transactions over the Web that involves personal information. These transactions include, among others, on-line banking, use of E-health services, and engaging in E-commerce. Personal identity is mostly referred to as the essence of a human being which makes a person a unique individual despite superficial modification which further persists at different points in time. Identity is the fundamental concept of uniquely identifying an object (person, computer, etc.) within a context. Digital identity can be defined as the digital representation of the information known about a specific individual or organization. As such, it encompasses not only login names, but much additional information, referred to as identity attributes or identifiers. Managing identity attributes raises a number of challenges, due to conflicting requirements. On the other hand, identity attributes need to be shared to speed up and facilitate authentication of users and access control. On the other hand, they need to be protected as they may convey sensitive information about an individual and can be a target of attacks like identity theft. Most common identity theft attacks are perpetrated through password cracking, pharming, phishing, and database attacks where the attackers captures the personally identifying information of individuals and user them to commit fraud. Our paper incorporates current research on identity theft attacks and prevention techniques using cryptographic algorithm and suggests an algorithm that can securely encrypt the information stored in the identification card.

2. Identity Theft

Identity Theft – the term is relatively new and is actually a misnomer, since it is not inherently possible to steal an identity, only to use it. Identity theft is used to describe a particularly harmful form of the familiar criminal activity of passing oneself off as somebody else. In this paper, by identity theft we mean the act of impersonating others' identity by presenting stolen identifiers or proofs of identities. It occurs when one person obtains data or documents belonging to another – the victim – and then passes himself off as the victim. The misappropriated data could include some or all of: full name and date of birth, credit card number and expiry date, passport number, computer password, etc while the documents could include identifying documents such as a passport or a driver's license, but can also include such items as electricity bills or even delivered envelopes – essentially anything which suffices to convince some other authority or company that the thief is indeed the victim. According to the Identity Theft Resource Center, identity theft is sub-divided into four categories:

- Financial identity theft (using another's identity to obtain goods and services)
- Criminal identity theft (posing as another when apprehended for a crime)
- Identity cloning (using another's information to assume his or her identity in daily life)
- Business/commercial identity theft (using another's business name to obtain credit)

Manuscript received September 5, 2010
Manuscript revised September 20, 2010

3. Techniques to Obtain Identity

In most cases, a criminal needs to obtain personally identifiable information or documents about an individual in order to impersonate them. They may do this by:

- Stealing mail or rummaging through rubbish.
- Retrieving information from redundant equipment which has been disposed of carelessly, e.g. at dump sites, given away without proper sanitizing etc.
- Researching about the victim in government registers, internet search engines, or public record search services.
- Stealing payment or identification cards, either by pick pocketing or by skimming through a compromised card reader.
- Remotely reading information from an RFID chip on a smart card, RFID-enabled credit card, or passport.
- Eavesdropping on public transaction to obtain personal data (shoulder surfing).
- Stealing personal information in computer databases (Trojan horses, hacking).
- Advertising bogus job offers (either full time or work from home based) to which the victims will reply with their full name, address, curriculum vitae, telephone numbers and banking details.
- Infiltration of organization that store large amounts of personal information.
- Impersonating a trusted organization in an electronic communication (phishing).
- Obtaining castings of fingers for falsifying fingerprints identification.
- Browsing social network (MySpace, Facebook, Bebo etc.) sites, online for personal details that have been posted by users.

4. Biometrics

Despite long-lasting efforts in the security community, authentication of people remains a weak link in the security chain. Passports with low-quality photographs, credit cards whose handwritten signatures are usually not checked, easily guessed passwords to access computer systems, etc. Biometrics is under study to see whether they will allow this weak link to be significantly strengthened. Biometric methods automatically confirm the identity of a person using either a distinctive aspect of their appearance (e.g. a fingerprint or an iris pattern) or a unique action by that individual (such as an electronically captured written signature or a spoken phrase). Among the many applications that hope to use these technologies are more secure passports, replacement of passwords for computers, and removal of the need for keys in homes and cars. The biometric approach is based on the fact that many characteristics of an individual are unique and hardly change over a lifetime. Various governments are now independently or jointly tabling plans for the introduction of biometric identifiers (e.g. a fingerprint, or an iris scan) in a move to reinforce border controls. This could for instance involve the storage of biometric information by means of a code or a chip card on a passport or other document. It is important to stress that biometrics can be applied in at least two different ways:

- When used for authentication purposes, a 1-to-1 verification of an identity of an individual is performed, by verifying the presented biometrics data of the individual with a pre-defined reference value.
- When used for identification purpose instead, the actual presented biometrics data of an individual is compared with all the stored reference data of a set of individuals (1-to-n verification) and the system returns a best match or a selection of best matches.

5. Cryptographic Algorithms

A basic definition of cryptography would be – a process or skill of communicating in or deciphering secret writings or ciphers. In our paper we will be using such an algorithm to encrypt biometric data such that it can only read after deciphering it with keys. Perhaps the most important areas that cryptography encompasses are:

- Encryption/Decryption
- Message Authentication Codes (MACs)
- Digital certificates and signatures

5.1. Encryption and Decryption

There are many encryption and decryption algorithms and they have different strengths and weaknesses. Cryptographic algorithms are further divided into two. They are Symmetric and Asymmetric Algorithms.

5.1.1. Symmetric versus Asymmetric algorithms

The process of encrypting and decrypting data almost always uses a single symmetric key or a pair of asymmetric keys. A symmetric key can be used to both encrypt and decrypt information. Examples of symmetric key algorithms include DES, Rijndael, AES, triple-DES and Blowfish.

Unlike symmetric keys, asymmetric keys come in pairs. Probably the best known asymmetric key algorithm is RSA. Its two keys are called the 'public key' and the 'private key'. Either key in the pair can be used either to encrypt or decrypt a given 'cleartext'. If information is encrypted with one of the keys, the other key is required to decrypt it. An advantage of RSA is that one key can be kept private and the other made available to the general public. This eliminates a common problem with symmetric keys: if two people want to encrypt and decrypt information they send to each other, first they encrypt it using the recipient's public key. The recipient will then decipher it using his private key and can also authenticate it using the sender's public key.

In addition, symmetric keys suffer from the problem that anyone who has the key can encrypt and decrypt information. This makes it hard to know who has done the encryption or decryption when more than one person has the key.

One interesting characteristics of RSA is that it is limited in how much data it can encrypt or decrypt. Essentially, it cannot encrypt or decrypt more information than the size of its keys. Because of this, RSA is usually combined with a symmetric algorithm to support the encryption and decryption of larger amount of data (since RSA key sizes are relatively small compared to documents or messages exchanged). By having the public/private keys encrypt/decrypt the relatively small secret key and by using the secret key to encrypt and decrypt the actual cleartext, RSA becomes practical for use with larger cleartexts. Digital certificates, which build upon public/private keys, were created to facilitate the reliable exchange of public keys between correspondents.

5.1.1.1. The Complexities of encryption and decryption

In addition to the difference between symmetric and asymmetric algorithms, many algorithms can be configured in different ways (for example, AES and RSA support key of varying lengths). There are also various 'modes' in which algorithm can operate and various means of padding clear texts which are not a multiple of the bit size required by the algorithm. Finally, because some algorithms need to get 'jumpstarted' they may require some initialization. This initialization involves parameters specifying exactly how the encryption/decryption is to be done and/or may require initialization information.

5.2. Message Authentication Codes (MACs)

A Message Authentication Code is an algorithm applied to some cleartext which produces a large number. This large number is of a fixed size and is usually represented as an array of bytes. The number is referred to as the 'hash result' or the 'message digests'. In some ways a MAC is similar to a checksum, except it doesn't simple result in the same number for the same document input. Because they are based upon one-way hashing, MACs have some useful properties beyond checksums:

- Given a text, we always generate the same result (just like a checksum)
- Given the digest, we can't recreates the original text
- Different texts produce quite different results, even if the texts differ only a little (very different from a simple checksum)
- Given the digest, you can't determine anything about the text that was used to produce the digest

A true MAC is cryptographically secure, meaning that you can know the MAC used in every detail and still find it very hard or impossible to break one of the 'rules' listed above. Sometimes a MAC is called a 'digital fingerprint' because it produces a small, essentially unique number representing the original clear text.

5.3. Digital Certificates and Signature

A digital certificate is essentially the public key of an asymmetric algorithm (like RSA) combined with some identifying information to specify the owner of the private key. As long as the owner of the certificate keeps the private key to themselves, they can 'sign' data with the private key and anyone who possesses the digital certificate can verify that the data was signed by that person/program.

The area of digital certificates is vast, including many issues relating to the administration of certificates such as issuing certificates, validating certificates signed by a series of trusted parties and determining the certificate is valid (and to what degree to trust it), revoking certificates, the format for certificates etc.

6. Distributed Cryptography

Distributed cryptography spreads the operation of a cryptosystem among a group of servers (or parties) in a fault-tolerant way. I will consider only the threshold failure model with n servers, of which up to t are faulty;

such distributed cryptosystems are called threshold cryptosystems.

There are two hard problems, factoring big number and discrete logarithm, which I will describe in this section. Everything in this protocol is based on the assumption that there two problems are computationally infeasible to be solved in polynomial time.

6.1. Factoring Problem

The problem states that given a big number, about 1000 bits, it is computationally infeasible to factorize the number into prime factors. The best known algorithm has exponential complexity in term number of bits of the number.

For example: If N = pq where p and q are big prime numbers (500 bits) then it is hard to find p and q given the value of N. This problem will be the basis of security of RSA encryption and decryption scheme.

6.2. Discrete Logarithm Problem

Given a big number N, of size 1000 bits, and y, g in the interval [1,…,(N-1)] where gcd (g,N) = 1. It is hard to find x that satisfies the equation: y = g^x (mod N)

This problem will be the basis of threshold decryption and the signing scheme implemented in this project.

Now, multiples parties, say n, will come together to generate a module N and make N and the encryption exponent, e public. Nobody knows the prime factors of N but everyone is convinced that N is a product of two large prime numbers. The scheme is n-out-of-n threshold scheme, and that means decryption requires the presence of all parties because each party keeps an additive share, d_i of the decryption exponent, d. As a result, this scheme allows a new Threshold Singing Scheme, (k,n), to be added later on in this project. Also note that the value of d is unknown to all parties and after any number of decryptions. Throughout the protocol, a trusted third party is not required and all stages in the protocol need the contribution of all individual parties.

6.3. Secret Sharing

Secret sharing forms the basis of threshold cryptography. A secret is shared among n parties such that the cooperation of at least t+1 is needed to recover s.

6.3.1. Algorithm

To share s ε 2 F_q, a dealer P_d not ε {P_1,…, P_n} chooses uniformly at random a polynomial f (X) ε F_q[X] of degree t subject to f(0) = s, generates shares s_i = f(i), and sends s_i to P_i. To recover s among a group of t+1 server with indices S, every server reveals its share and they publicly recover the secret

$$S = f(0) = \Sigma_{i \varepsilon S} (\lambda^{s}_{0,i} S_{i)}$$

where $$\lambda^{S}_{0,i} = \Pi_{j \varepsilon S, j \neq i} j/(j-1)$$

are they (easy-to-compute) Lagrange coefficients. The scheme has perfect security, i.e., the shares held by every group of t or fewer servers are statistically independent of s (as in a one-time pad).

6.4. Verifiable Secret Sharing

If the dealer P_d is also faulty (malicious), we need a verifiable secret sharing (VSS), a fault-tolerant protocol to ensure that P_d distributes “consistent” shares which define a unique secret. VSS is an important building block for secure multi-party computation.

6.5. Distributed Key Generation

There are also distributed key generation protocols (DKG) for generating a public key and a sharing of the corresponding secret key. They must ensure that the corrupted parties learn no information about the secret key. Such protocols exist and have been implemented for the common public key types, discrete logarithm and RSA. Usually these algorithms assure synchronous networks and passive adversaries. With weaker assumptions (active adversary), they are less practical, however.

7. Threshold Cryptography

Threshold decryption has been studied a lot for the last two decades. It is a branch of public key cryptography in general and multi party computation in particular. Essentially, in a k-out-of-n threshold crypto system, denoted (k,n) where 1<k<=n, for the RSA function, my aim is to generate and then split the secret decryption/signing exponent d into n different pieces, which are then distributed privately to n parties. This enables:

- Any k or more out of n total parties, when they come together, they can “reconstruct” the secret d in a way which enable them to decrypt or sign a message. This should be done in a way that does not reveal the value of d and its shares to anyone in the scheme.
- Secondly, signing or decryption will be totally impossible in the circumstance where less than k parties are present.

The area of threshold cryptography has been pioneered by Adi Shamir in his 1978 paper; however the idea only took off when the problem was formally stated by Desmedt.

Since then there has been much work devoted to the topic such as Desmedt and Frankel, Pedersen, Gennaro et. al. and many more. However, the majority of these solutions are only for discrete logarithm based system that has a direct application to the Elgamal encryption and decryption algorithm. The reason why discrete logarithm based threshold systems are easier to design is because the group in which one works has a publicly known order.

7.1. Discrete Logarithms

Let G = <g> be a group of prime order q, such that g is a generator of G. The discrete logarithm problem (DLP) means, for a random $y \varepsilon G$, to compute $x \varepsilon Z_q$ such that $y = g^x$. The Diffie-Hellman problem (DHP) is to compute g^{x1x2} from random $y_1 = g^{x1}$ and $y_2 = g^{x2}$.

It is conjectured that there exist groups in which solving the DLP and DHP is hard, for example, the multiplicative subgroup $G \zeta Z_q^*$ of order q, for some prime p = mq+1, where |p|=1024 and |q|=160 (recall the q is prime).

The language of complexity theory says that, a problem is hard means that any efficient algorithm solves it only with negligible probability. (Formally, this is defined using complexity-theoretic notions: there is a security parameter k, an efficient algorithm is a probabilistic that runs in time bounded by a fixed polynomial in k, and a negligible function is smaller than any polynomial fraction).

7.2. ElGamal Encryption

The ElGamal cryptosystem is based on the Diffie-Hellman problem. Key generation chooses a random secret key $x \varepsilon Z_q$ and computes the public key as $y = g^x$. The encryption of $m \varepsilon \{0,1\}^k$ under public key y is the tuple $(c_1,c_2) = (g^r, m \Theta H(y^r))$, computed using a randomly chosen $r \varepsilon Z_q$ and a hash function $H : G \rightarrow \{0,1\}^k$. The decryption of a cipher text (c_1,c_2) is **m** $= H(c_1^x) \Theta c_2$. One can easily verify that **m** = m because $c_1^x = g^{rx} = g^{xr} = y^r$, and therefore, the argument to H is the same in encryption and decryption. The scheme is widely considered to be secure against passive adversaries.

7.3. Threshold ElGamal Encryption

The following threshold ElGamal cryptosystem tolerates the passive corruption of t < n/2 parties.

Let the secret key x is shared among $P_1,\ldots,P_n$ using a polynomial f of degree t over Z_q such that P_i holds a share $x_i = f(i)$. The public key $y = g^x$ is global and known to all parties (and clients), and encryption is as in ElGamal above. For decryption, a client sends a decryption request containing c_1, c_2 to all servers. Upon receiving a decryption request, server P_i computes a decryption share $d_i = c_{1i}^x$ and sends it to the client. Upon receiving decryption shares from a set of t+1 servers with indices S, the client computes the message as

$$m = H (\Pi_{j \varepsilon S}\, d_i \wedge \lambda^S_{0,i}) \Theta c_2$$

This works because

$$\Pi_{j \varepsilon S}\, d_i \wedge \lambda^S_{0,i} = \Pi_{j \varepsilon S}\, c_i \wedge x_i\lambda^S_{0,i} = c_i \wedge \Pi_{j \varepsilon S}\, x_i\lambda^S_{0,i} = C_i^x$$

from the properties of Algorithm 6.3.1. Note that the decryption operation only requires the cooperation of n-t servers. This is an example of a non-interactive threshold cryptosystem, as no interaction among the parties is needed. It can also be made robust, i.e., secure against an active adversary. Such threshold cryptosystem can easily be integrated in asynchronous distributed systems; but many threshold cryptosystem are only known under the stronger assumption of synchronous networks with broadcast.

8. Implementation

8.1. Threshold Algorithm

Assuming that a set of n users wishes to generate a number of threshold signatures, the n users can generate a shared module N, a public exponent e and n shares d_i of the secret exponent d, such that

$$d = d_1 + d_2 + \ldots + d_n.$$

This shared key generation protocol can be executed without the need for a trusted dealer. The parties now wish to use these shares so as to generate a threshold signature scheme, with threshold value k. This means that we want any k parties to come together so as to be able to sign a document. We let

$$I = \{t_1, \ldots, t_k\} \zeta \{1, \ldots, n\}$$

denote the set of parties who wish to come together to sign the document and

$$I' = \{1, \ldots, n\} \setminus I = \{t_{k+1},\ldots, t_n\}$$

denote the other parties. There are essentially two existing ways of doing this, both with disadvantages.

This means that after signing one message, if a different subset is going to sign for the second message, one need to re-key in some way. The signing stage requires the interaction of all k signing parties, so as to reconstruct the n-k missing secrets, and the share refreshing protocol requires the interaction of all n parties. Now the following algorithm will be defined, with the following properties:

8.1.1. Dealing Algorithm

This is an interactive protocol amongst the n users. Each user has an input d_i, which is their share of the unknown private key d. At the end of this protocol the users agree on a threshold value k and some global public information S.

In addition each user also obtains a public/private share (P_i, S_i) of the data needed to implement the threshold signature scheme.

8.1.2. Subset Presigning Algorithm
This is an interactive protocol amongst k member I = {t1, …, tk} of the n parties. The protocol results in public data DI which is used by the share combining algorithm to generate a valid full signature from the signature shares. The protocol results in each of the k parties holding some secret information $S_{I,ti}$ which depends on the sunset I. This protocol is interactive, but only needs to be run once for each sunset I.

8.1.3. Signature Share Generation Algorithm
This algorithm takes as input a subset I as above, the secret information $S_{I,ti}$ and a message m. The result is a partial signature $\sigma_{I,ti}$ on the message m.

8.1.4. Signature Share Verification Algorithm
This takes as input a signature share $\sigma_{I,ti}$ on the message m and verifies that it is validly formed using the public information D_I and P_{ti}.

8.1.5. Share Combining Algorithm
This takes as input the public information S, P_i and D_I, plus a message m and the partial signature shares $\sigma_{I,ti}$ for all t_i ε I, and then produces a valid full RSA signature σ. Or returns fail if one of the signature shares is invalid.
Hence the main advantages are that it does not require a trusted dealer and it does not require re-keying or interacting once the Subset Presigning algorithm has been implemented for a given subset I. The main disadvantage is that the signature share generation algorithm needs to know which subsets of shares are going to be combined later on.

8.2. Benefits of threshold algorithm

8.2.1. Inter-activeness
If the k parties I want to sign a different message, they need to come back to the Signature Share Generation Algorithm. This is because the values of x_I, S_{ti} etc can be reused. If there is change in the threshold set parties. From I to I', then all the shares $S_{t'I}$, for t'_I ε I', will have to be calculated again as their values depend on the set I' so that the parties need to come back to Subset Presigning Algorithm stage. A new value of $x_{I'}$ needs to be determined. Note that the threshold value k, still remains the same in this case. If all parties agree to decrease the threshold value k, then all of them need to run the protocol from the beginning, i.e. the Dealing phase. Also note that the threshold value can only be decreased, but not increased.

8.2.2. Share Refreshing
Since the value of d still remains the same, the individual shares of each party need to be refreshed. If they do not renew their shares, then a certain single party might end up knowing the shares of all other parties, for example, if we have three parties P_1, P_2 and P_3 and a (2,3) threshold scheme:

- In the first signature, party P_2 is away and its share is reconstructed by party P_1. So P_1 now knows shares d_1 and d_2.
- In the second signature, party P_3 is away and its share is reconstructed by party P_1 again. So P_1 now knows share d_1, d_2 and d_3. That means the first party knows the shares of all parties.

To remove such errors, each individual share, d_i, is reconstructed in a way that does not further reveal its value to any party after any number of changes in the set of threshold parties. As a result, this not only does not leak any information about the share to anyone else but also avoids the refreshing procedure that requires the interaction of all parties in the scheme.

8.2.3. Robustness
Even though the protocol is always (k-1) private, i.e. no information is leaked when up to (k-1) parties corrupt, it does not mean the combining signature process will be always successful. If the number of dishonest parties is l, and $(n-k) < l < k$, then the number of honest parties is (n-l) and $n-l < n-(n-k) = k$. That means a valid signature on a message cannot be obtained. As a result, to make sure that this never happens we require that $(n-k) > k$, i.e. $k < [n/2]$.

9. Analysis of the Problem: Robustness of Threshold Cryptography

The reason why threshold decryption/signing is very useful in practice is because not only does it provide secrecy and reliability but also flexibility. In addition, the property of sharing the secret is ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate, for example, in an electronic voting system, identification cards or any gambling games. On the other hand, sharing the secret key by multiple parties, each holds a share of the secret, can guarantee that decryption is done if and only if all parties agree to do it and therefore the scheme can give us a much higher level of security.

10. Conclusion

Identity theft exploits a systemic weakness (of society's mechanisms for identifying people), and as such a systemic analysis can be of value. In principle, there would seem to be two approaches possible for reducing identity theft: either making it more difficult, or making it less profitable, However, when the systemic response is taken into account, there may be conflicts between these two approaches: for example, actions which make identity theft more difficult may also have the effect of making it more profitable.

Thus, for example, a single barrier approach would involve developing a proof of identity which was extremely hard to forge. However, a single barrier is also a single point of vulnerability, and a very good proof of identity would induce so much confidence that if the rouge could forge it, he could get away with anything – thus making such a forgery, although difficult, very profitable.

Another aspect worthy of systemic analysis is the creation of centralized databases of identifying data. Any such database represents in principle a single point of vulnerability for large-scale identity theft, and it would be reasonable, on these grounds alone, to try and minimize the number of such databases. Given that customer or client databases of some sort are an essential element of much business and other (e.g. government) activity, and that these databases are usually centralized for operational reasons, the best approach may be to encourage such a database to hold only the information strictly necessary for the operations the database supports. This could be done by the database owner, by encouragement - or if necessary legislation – by the public authorities, or, to some extent at least, by the customer or client restricting the information supplied. In this context the development and widespread deployment of identity management systems may help to reduce these vulnerabilities. Overall, any systemic analysis has to take into consideration both technological and socioeconomic aspects, in order to analyze questions addressing public policy. Threshold Cryptography, even though popular and effective, is a complex field of study. The limitation of this study is the insufficiency of time and resources for the required algorithm.

References

[1] Rainbow Technologies, 2003. Password survey results (June 2003). Retrieved November 14, 2005, from http://mktg.rainbow.come/mk/get/pwsurvey03S.

[2] Matt Bishop, Sathyanarayana S.Venkatramanayya, Introduction to Computer Security, O.Goldreich, Foundations of cryptography, vol. I & II, Cambridge University Press, 2001-2004.

[3] Privacy Rights Clearinghouse, Criminal ID Theft, <http://www.privacyrights.org/fs/fs17g-CrimIDTheft.htm> March, 2004.

[4] Identity Theft Resource Center, <http://www.idtheftcenter.org>

[5] Identity Theft Research, <http://www.identitytheftreseach.com/> March, 2008.

[6] European Biometric Forum – Biovision. <http://www.eubiometricforum.com/> The Computer Bulletin, British Computer Society, September 2002.

[7] The Java Cryptography Architecture. <http://java.sun.com/j2se/1.4.2/docs/guide/security/CryptoSpec.html>

[8] Y.Desmedt. Society and group oriented cryptography: a new concept. Advances in Cryptography – CRYPTO '87, Springer-Verlag LNCS 293, 120-127, 1987.

[9] Y.Desmedt and Y.Frankel. Threshold Crypto-System, Advances in Cryptography – CRYPTO '89, Springer-Verlag LNCS 435, 307-315, 1989.

[10] V.Shoup and R.Gennaro, Securing threshold Cryptosystems against chosen ciphertext attach, Journal of Cryptology 15 (2002), no.2, 75-96.
V.Shoup, Practical threshold signatures, Advances in Cryptology: EUROCRYPT 2000 (B.Preneel, ed.), Lecture Notes in Computer Science, vol.1087, Springer, 20000, pp.207-220.

Syedur Rahman received the B.S. degree in Computer Engineering from North South University in 2007. During 2007-2008, he stayed in North South University (NSU) as a Teacher Assistant and as a Lab Instructor. From 2009 he is working as a Software Engineer in Jaxara It Limited (An USA based Software Company), Bangladesh.

A.M. Ahsan Feroz received the B.S. degree in Computer Science and Information Technology from Islamic University of Technology (IUT) in 2007. During 2007-2008, he worked as a research assistant on a research based firm. From 2009 he is working as a Software Engineer in M&H Informatics (An IMS Health Company), Bangladesh.

Md. Kamruzzaman received the B.Sc. degree in Computer Science and Engineering from Khulna University in 2005. After graduation, he joined EVOKNOW Bangladesh Ltd. and worked there as a Software Engineer for one year. Afterwards he joined United IT Global Net as a Software Engineer and stayed for about 8 months. Currently he is working as a Senior Software Engineer in Jaxara IT Limited (An USA based Software Company), Bangladesh.

Md. Azimuddin Khan accomplished the degree of Bachelor of Science in Computer Science & Information Technology in 2006 from Islamic University of Technology, Bangladesh. He is working as Analyst, Business Intelligence in Axiata (Bangladesh) Ltd. from September'2010. Previously he worked as Software Engineer in IMS Health (Dhaka Office) from June'2008 to August'2009. He was also Lecturer of Department of Computer Science & Engineering in Northern University Bangladesh from February'2007 to April'2008. Recently he has completed Masters of Business Administration from Institute of Business Administration, University of Dhaka.