The Structure of a Signature via the Net

Chuen Chuan Huang, Chuen-Der Huang, and Shin Ya Huang
(Corresponding author: Chuen Chuan Huang)

The Department of Computer and Communication Engineering, Asia University,
500 Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
(Email: sirren@asia.edu.tw)
(Received Jan. 16, 2008; revised and accepted March 4, 2008)

Abstract

The signature can be copied or falsified, especially being sent via the Net. The method being proposed is an algorithm or cryptosystem to guarantee the transmitted information is completed and correct as well as clearly sure of who was the signer. The philosophy is if a sender has a unique codebook and the cipher text can never be written by another person and once the correct receiver uses his unique codebook to solve the cipher text for the plaintext, it is sure that the sender is the correct sender and the receiver is the correct receiver.

Keywords: Cryptosystem, floating codebook, non-repudiation, signature

1 Introduction

Wang [26]surveyed the electronic signature. Among those about electronic signature, an electronic signature is an *object* [25]. And many issues are still ongoing. While, this article proposes a *method* that viewing the signature being happened from the uniqueness of the sender's identity with the cryptosystem acting itself as the unique carrier just for that time mail between solely the sender and the receiver. In other words, in our proposal, the signature is the infrastructure of the system. We think no matter what an electronic signature is, the electronic signature must not be forged and must define the range of its coverage, namely, the range of the mail that the electronic signature covers. We begin from basic math only and a simple analysis on the nature of signature. We realize the essences of a signature are:

1) A signature is so unique to represent the person.

2) A forged signature is possible, and with identification and by law with penalty, the falsification is suppressed to minimum, not zero.

3) It is a habit or custom and also by law, a signature is a declaration of responsibility over an area, normally a sheet of paper.

4) The said area is extended from the ink and traces of the signature with the tangled fibers or molecules to the edge of the sheet of paper.

5) With the continuity of the said tangled fibers or molecules of the sheet of paper, the uniqueness of the identification and the responsibility continue and are extended along the traces and ink of the words and symbols on the sheet of paper. So, the words and symbols are covered by the signature. Therefore, the responsibility of the information carried by the words and the symbols on the sheet of paper with the signature is extended from the signer to those words and symbols.

6) The deficiencies of this system (i.e. the information is covered by an object, the signature) are: if the signer is the correct signer, if the correct signer signed the incorrect signature, if the correct signature signed by the correct signer being planted on the incorrect sheet of paper with incorrect words and symbols, and if the correct signature covers a correct words and symbols without anything about information being added, deleted or moved.

In simple words, when we see a signature we always worry about if the signature is true and if the words or symbols on the sheet of paper with the signature are all true without falsification. And we also wonder if the words or symbols are with a copied and glued correct signature. Think about a person received a letter with the signature of Bill Getz saying he shall give you some money. We doubt if the signature is true signature and if the words on the signed sheet of paper is true. Indeed, we do not recognize the signature, and even we are not so talent to make sure if there is not falsification. We even examine every spot of the sheet of paper to look for any clue of falsification with a microscope. Finally, suppose everything is fine, we are still not so sure the signature is true.

The above explains we do not recognize a signature absolutely. And we recognize there are too many ways to falsify.

This article shall propose a method without worry about the signature or any possibility of falsification on the information. And, our method can warrantee the signature and its coverage both are true, yet not on sheets of paper in handwritings or printings.

The Internet has been worldwide used, and the environment is very complicated. It is not so sure if any attacker is there. We challenge very high security and efficiency in handling our security affair. Normally high security is high cost and inefficient. Therefore we propose a tool can generate all the so many codebooks, each for a person, and with the tool, a cryptosystem is proposed for both security and efficiency.

For the above purpose, in this article we are going to introduce the tool, the so-called onion. Secondly, we shall introduce the cryptosystem based on the unique codebooks. Thirdly, we shall explain why the cryptosystem is so secure. Fourthly, we shall explain the aim, signature, is achieved. And, finally, we shall explain why the tool is so important and efficient. Before explaining what are listed in the above paragraph, we shall introduce some view point about encryption and decryption because it is helpful for the reader to understand the tool, including why the tool is required though it is not a must. We begin just from basic mathematics.

2 The Fact of Cryptosystems

Human language based information is transformed into codes by some coding protocol, such as ASCII, because the codes are numbers, the good operands for computation. Suppose we have a set of the (plain) codes in hand, we use transformations to map the elements of the set into cipher text, a set of encrypted codes not readable. No matter what, the encrypted codes must be reversed to the plain codes at the side of the receiver. We suggest the plain codes from ASCII to be transferred or directly from human language (word by word, not letter by letter) with some transformation (the codebook) to be transferred to the (plain) codes with the form $(n_1, n_2, n_3, 1)$, where "1" is just a scale factor. So, the information is transferred to be a set of (plain) codes. Call this set P, and we have $x_i \in P$, where x_i is a plain codes. We may take a transformation to make $y_i = A(x_i)$, and if possible in matrix form $y_i = A(x_i)$. Now, we can settle a three-dimensional frame in the space to open a three dimensional space [19, 23]. And we may view x_i and y_i as the points in this space, without concerning if the space is metric or not. With the codebook, x_i is a determined matter, but y_i can be arbitrary. Now if we view a pair of x_i and y_i as two ends, we can make infinite numbers of tracks (lines). Many ways of assigning these lines were proposed already with some certain properties [1, 2, 3, 4, 5, 7, 14, 15, 20, 22, 27]. No matter what y_i is, there must be at least one line leading y_i back to x_i. Any such a line or track between (x_i, y_i) or (y_i, x_i) can be composed of plural lines or segments, and some lines or segments probably have some special property.

One of our suggestions is with some special segments, please refer to Figure 1.

In Figure 1, small circles are denoted for points in the space, and the points are codes. After some words being transferred to be ASCII codes a_i, there is a translator translates the ASCII codes a_i into the three-dimension codes x_i, perhaps on word basis. x_i is transformed into e_i (transformed codes), and the ensemble of the transformation is the original codebook. Suppose we have an onion (we shall explain later), and we put x_i at some point e_i on the shell of the onion. The point e_i is a position with a 3-D address mapping to x_i or vice versa. The original codebook is then transformed to be many different business codebooks, each for one business. A business has many users. The business codebook is transformed again into many different fixed codebooks, each for a user. When the user wants to mail a letter, the fixed codebook is transformed with some random variables to be a floating codebook [8, 9, 10, 11, 12, 16]. Only the operands involved in the mail are required to be actually computed, not a whole list or by checking a mapping table or sub-table. The ensemble of Track 1 is the business codebook, and it transforms the codes e_i of the original codebook into the codes of the business codebook b_i. The other option is taking the business codebook as the ensemble of the pairs between x_i and b_i. This means we may view the codes to codes mappings as a codebook or the ensemble of tracks (transformations) as a codebook. In other words, the two kinds are: $\{(x_i, y_i) | 1 \le i \le \overline{m}\}$ or $\{A_{ij} | y_i = A_{ij} x_i, 1 \le i \le \overline{m}, \exists j \in N\}$, where $\overline{m}$ is how many points are concerned, and N is nature number. A track is a transformation for mapping a three-component quantity from one position to the other position, or a point is switched to be the other point. A transformation is independent of x_i and y_i, it may map many operands. However a mapping pair $x_i \to y_i$ carries a part of the information about the transformation until more pairs added without modifying the transformation. Besides, the pairs also reveal what are the operands. For an example, there are pairs [1 0 0] $\to$ [0 1 0] and [0 1 0] $\to$ [−1 0 0], for the first pair, we may write

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = y_1 = A_1 x_1,$$

and for the second pair we may write

$$\begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = y_2 = A_2 x_2.$$

And it is obvious that $A_1 \neq A_2$. But, if we consider these two pairs simultaneously, then there is a (homogeneous)

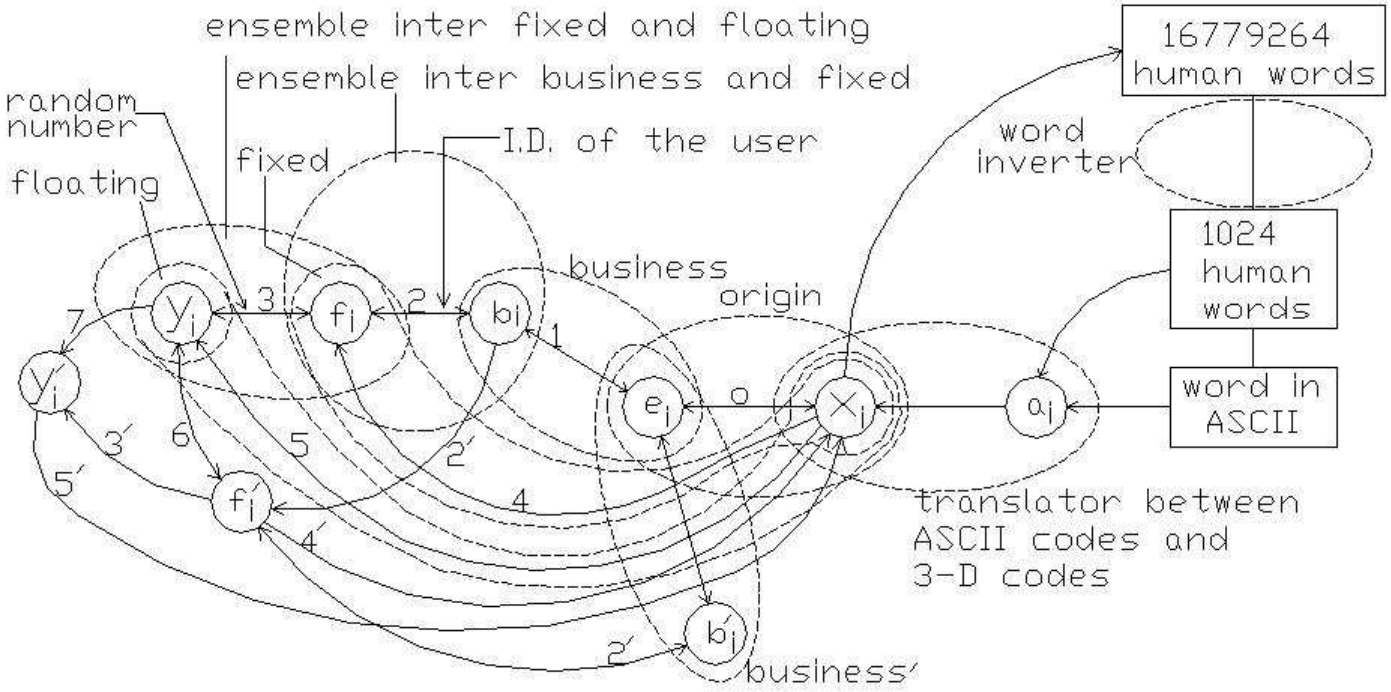

Figure 1: The system diagram

transformation:

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

i.e. $y_1 = Ax_1$ and $y_2 = Ax_2$. A is a rotation matrix, and it is independent of the operands, i.e. for more operands, A is the same. Ensemble of transformation and ensemble of pairs are different conceptually. In the former case, the protocol is Euler's Transformation, the parameter is $(\psi, \varphi, \theta) = (0°, 0°, 90°)$. And the transformation is independent of the operands and possibly expressed as: 0/0/90.

Track 2 transforms codes b_i into the codes of the fixed codebook f_i. From x_i to f_i, there is a direct Track 4, and the ensemble of Track 4 (pair list or operators) is the fixed codebook. Track 4 is the union of the track of the original codebook (Track o), Track 1 and Track 2. Code f_i is transformed to code y_i, the element of the cipher text. The cipher text is the ordered multiplicity set of some codes as y_i. (Analogously, a composition is nothing but the ordered multiplicity subset of a dictionary, an ensemble of pair list.) Track 5 is the union of Track 4 and Track 3. The ensemble of Track 5 is the floating codebook. We suggest Track 5 composes of Tracks o, 1, 2 and 3. We may write $y_i = 5\,(x_i)$. If the space is metric, there is $y_i = 5\,(x_i)$, where 5 is a 4×4 matrix transforming x_i to y_i. We may use f_i as an element of the cipher text, but to do so remaining some weakness for the attacker. The mappings of Track 4 or the fixed (from x_i to f_i is fixed) is given to the user or the sender. And, with some random variables or parameters, fixed codebook is transformed to be the floating codebook. The sender wrote some words, and these words are translated to be a set of $p = \{x_i | 1 \le i \le m, m, i \in N\}$, where m is the number of words or the entities wanted to be mailed, and this work is done by the translator, which is software for public use. We would not talk too much about matters to the right of x_i on Figure 1 here due to the focus of this article.

$q = \{y_i | 1 \le i \le m, m, i \in N\}$ is the cipher text, and it is mailed directly to the receiver (or the receivers) via the Net. When the receiver received the cipher text, depending on the request of the sender, the business shall mail the key to the receiver. The transformation Track 6 is the key mailed. But, in this way there is some weakness and good for the attacker (Track $4'$ is exposed, and so as Track 5; fixed codebooks should be protected). We suggest the key should be Track 7.

Track 7 is the union of the tracks, or track $y_i \to f_i \to b_i \to e_i \to b_i' \to f_i' \to y_i'$. Or, we have $y_i' = 7y_i$. Tracks 3, 4 and 5 are in the hand of the sender (the sender can be an attacker), and Track 2, 1 and o are in the hand of the business. (Track o might be in the hand of the general cryptosystem.) After having y_i' in the hand of the receiver, the receiver traces back through Track $3'$ and Track $4'$ to have x_i. Or it is equivalently to trace back through Track $5'$. Tracks $3'$, $4'$ and $5'$ are in the hand of the receiver. In Figure 1, Track $2'$ is redundant between

b_i and f_i'. It is of the case that the businesses are the same one.

Due to Track 7 containing (for neat typing, superscript of 3^{-1} or so being omitted) Tracks 3, 2, 1, $1'$ (between e_i and b_i'), $2'$ and $3'$, Track 7 is secure (useless for the persons other than the correct receiver because at least Track $3'$ in the hand of the receiver). Tracks 2, 1, $1'$ and $2'$ are in the hands of the businesses. Without both Tracks $3'$ and $4'$, y_i' is nothing but still a piece of cipher text. If the sender is the attacker, he can solve Track $5'$. (To do so is the sender attempts to steal some other information mailed to the receiver from the second sender. Track $4'$ should be protected.) However it is meaningless, the reason is Track $3'$ is different from time to time with a parameter. And, the parameter shall be assigned by and mailed from the business of the receiver. After receiving the parameter, Track $3'$ can be obtained by computation at the side of the receiver.

The floating codebooks (5 and $5'$) can be exposed, but that's fine because the attackers other than the sender and the receiver can not reach the floating codebooks. In other word, the two floating codebooks being used in one time communication are not secret to the sender or the receiver. Track 3 is not reachable to the receiver and Track $3'$ is not reachable to the sender. Therefore the fixed codebooks are safe. To keep the business from reaching the operand y_i or y_i' is important. We can not assume the people of the business are honest absolutely. Much work in this article is for preventing information intercepting, attacking, falsifying with or without intention.

The quick reader may question about if the tracks are two ways (bidirectional). They are. The reason is they are all nonsingular because we have the tool to prevent non-one-to-one situation. The tool shall turn out to be the onion.

If we have a multi-shells ball (the onion) with a certain number positions on the shells and each position can only have one point and all positions are occupied, no matter how points are put on the available positions, or points are switched, the transformations are all one-to-one. Therefore, all the transformations of point switching are reversible, i.e. if matrix form is available, the transform matrix is nonsingular.

For advanced consideration, it is adequate to think if y_i is possibly made from f_i' with f_i' being able to be reversed to x_i? The question is equivalent to ask if Track 6 is possibly guessed. The chance is 16779264^{-1}, and if we adopt the principle of avalanche with totally m words, the chance is 16779264^{-m}.To guess an f_i' is meaningless, and no better than guessing from y_i directly to have x_i.Track 6 must be decided with Tracks 3, 2, 1, $1'$ and $2'$. Among them, Tracks $1'$ and $2'$ are already known matters in the hand of the business (primed). How about the primed business is an attacker? No way, it is because Track o is in the hand of the general cryptosystem. (That's why Track o is considered to be in the hand of the general cryptosystem. Besides, this arrangement can stop a business be a faker.) And how about the general cryptosystem is the attacker? Again, no way, it is because the cipher text q is sent directly to the receiver without passing through the business or the general cryptosystem. They (the business and the general cryptosystem) do not have q. We attempt to make the cipher text and the key are separated from each other. Only the government can play a role as the general cryptosystem, the attacker and the hacker. In this case, there is still a weapon, i.e. "dialect", which is a private protocol between the sender and the receiver. We encourage each pair of sender and receiver making their dialect to stop in case the business or the general cryptosystem being a faker. To build a dialect is more secure [8, 9, 16].

With the above explanation, if the business and the general cryptosystem are honest, the problem of information security is solved.

A piece of information via the Internet is particularly difficult with the problem of signature (needing uniqueness and/or correctness). Therefore we shall explore in this environment.

3 The Uniqueness, Signature and Security

Imagine Figure 1 is installed in R^3 space, all entities or the small circles are the points in the space, and the coordinates are the positions of the points in the space. So we realize a transformation (either two points being exchanged or ring rotation, never non-one-to-one) is actually a point to move to a new position. Suppose y_i is first selected, different Track 6 (union of Track 7 and $3'$) shall map y_i finally to different x_i. The correct Track 6 is composed of Track 3, 2, 1, $1'$ and $2'$. These tracks are belonged to the correct sender and the correct receiver (and the correct businesses). We can conclude, if a piece of cipher text is written by the correct sender and is received by the correct receiver, x_i is surely correct. If the sender is correct but the receiver is incorrect (the receiver is the attacker), is it possible, by chance, a different Track 6 is chosen to have another f_i' and is there another Track $4'$ to reverse y_i to x_i? It is possible, but the chance is as low as $(16779264)^{-1}$, practically zero. It is because y_i and x_i are two points (occasionally, coincide), from a given point to seek the other having that chance. On the other hand, there is a counter part question, i.e. can y_i be generated by wrong sender (the falsification maker) but with this y_i to have the correct meaning x_i if with respect to the correct sender's? The chance is still $(16779264)^{-1}$.

If there are m words of a plaintext or the cipher text, the chance shall be $(16779264)^{-m}$, and if m goes very large, finally, the chance is actually approaching to $(16779264!)^{-1}$. And if avalanche effect is designed to be embedded inside, the chance goes to $(16779264!)^{-m}$. Even for just one word in a plaintext or the cipher text, i.e. with chance $(16779264)^{-1}$, falsification for some receivers is just impractical.

With the above explanation, basically, for the cor-

rect receiver to have the correct x_i can only be transformed from the unique y_i (the reason behind is our transformations are all one-to-one correspondence), and the chance for a wrong sender (the faker) to be successful is just $(16779264)^{-m}$. This means basically (due to $(16779264)^{-m}$ being not actually zero) a cipher text can stand for the sender, i.e. a sender can not deny that the cipher text is not sent by him, non-repudiation. Hence, if and only if there is a cipher text can be reversed by the correct receiver to have the correct or meaningful plaintext, the sender must be the correct sender. This is equivalent to put a signature with verification (having a signature with truth verified by the cryptosystem). For simplicity, if a receiver can successfully solve a cipher text for a meaningful plaintext, the plaintext must be correct and the sender must be the correct sender.

As for security, the chance for a wrong receiver (the attacker) to have the correct plaintext is the same as $(16779264)^{-m}$. It is quite good. If higher security is required, the receiver can write a cipher text to be a sender to ask the original sender to answer something to have absolute confirmation. The other way is to make 16779264 and m larger as being satisfied.

In the following, we would like to introduce our tool. The aim of our tool is to realize what we report above with ease and the efficiency.

4 The Problem of Realization and Efficiency on Transformation

Please refer to Figure 1 again. We said all the small circles are the points of the 3-D codes space. Now we introduce a ball has totally 1024 shells (onion), and on each shell there are 16386 points equally spaced on the surface of each shell. Please refer to Figure 2. We may just imagine all the small circles now must be at the said points of the 1024 shells, i.e. totally 16779264 points. Suppose we have $16779264x_i$, each for a point of the ball, and the possibility (how many kinds of permutation) is 16779264!. A transformation of a point is to make the point be moved to some other position (a point site), and the original point at the position is moved to the position of the point (two points exchange their positions). In Figure 1, any track is a point, for instance, x_i, to be moved to position y_i, etc. Such a ball point structure (points with operations or transformations) can provide as large as space of what just said 16779264!. Generally, we say the sample space is this figure. Namely, we have totally 16779264! permutation tables for mapping x_i to y_i. However, if a plaintext is short, the true possibility (the size of event space) is just $16779264!/(16779264-m)!$, not 16779264! or $(16779264!)^m$ (depending on some details of how to make a cipher text). This discussion explains the basic number (such as 16779264) should not be too small. And this is not science, and it should be judged by art. This phenomenon (not so large neither so little as the case of having "!") is contraction. The reason is though there are 16779264! permutation tables for 16779264 different elements (words) to be mapped, and only the subsets of some tables are used. A full table contains 16779264 mapping pairs, and a subset just contains no more than m mapping pairs. And, therefore, there is easier chance, some sub-tables (sub-permutations) or the subsets are hence equal.

Though the true possibility or chance is not so large or so low, the number 16779264! provides a chance to make the faker and the attacker busy because this system can actually provides as many as 16779264! different kinds of floating codebooks and/or fixed codebooks. For the above reason, we encourage the way to enlarge m. Therefore among those 16779264 words, there are many words are virtue words or null words. In this policy, m is enlarged quite.

As for efficiency, the ball has a quite good efficiency in scrambling the points (new permutation), it is not anything of theory but device therefore [6, 13, 15, 17, 18, 21].

Suppose we have a mechanical ball partially being mounted in a socket and can be freely rotated around the socket internally. We can rotate some selected shells or sets of points (select and highlight them) to transform some or all the points into new permutation [10, 11, 12]. And this mechanical device may be electronically simulated in a computer or so. The efficiency is high, and distribution is even, without concentration (inhomogeneous) too much. Besides, the points always keep all on the shells and each point to the address is always one-to-one before and after permutation, which guarantees the tracks are all bidirectional.

For any rotation or exchange of points, the results are just characterized by some angles and the point sets being involved. It is not necessary to combine many matrices into one matrix right the way, but it just makes a record of these quantities until the user to decide which transformations need fusion. These quantities may even be transmitted via the Net securely because the attacker doesn't know how to use them. For more details, please refer to the section of "The structure of a signature", where matrix A having infinite possibility, and the attacker doesn't have the codebook. Such transformations start from some status of the ball. It is our wish that the device at the user's side is portable, small and light, and we wish our work is a part for personal mobile communication. And, hence we concern if the computation can be performed in a small device, even being integrated into a PDA. Before adopting parameters for computing the fixed codebook and the floating codebook, let see how about just using pairs.

After reorder the pairs, there are totally 16779264 pairs, and each pair needs 16779264 in number, so, $16779264^2 \approx 2.9 \times 10^{14} = 35193G - bytes$. For hand held device, we may just take one layer. Each layer contains 16386 words. We still have 16386!, which still is a very large number. And the memory required for a reduced fixed codebook is 0.034G-bytes. We may easily take more than one shell (about 5 shells, 0.84 G-bytes; for computa-

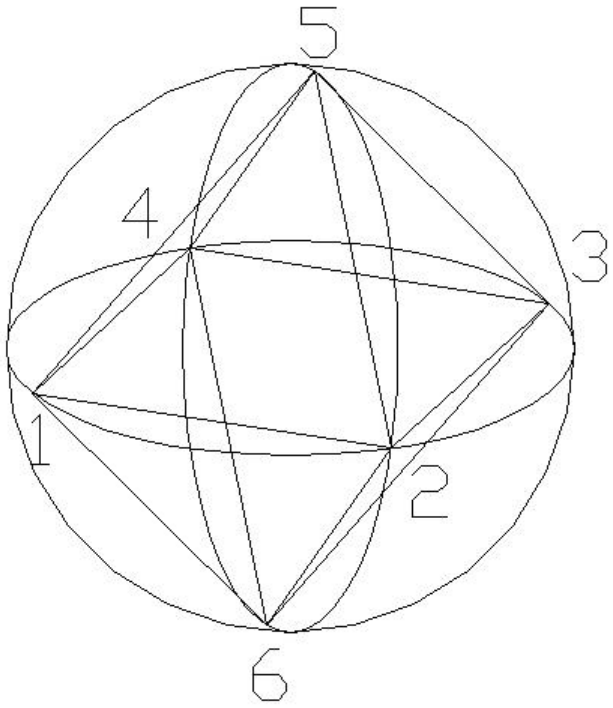

Figure 2: The simplest equally spaced grids on a sphere

**Two equal pyramids are put together and internally connect with the corners to the ball. The arcs between each two points can be halved again and again. And, equally spaced points are obtained.*

tion, some times of this amount is required to return the cipher text to the human words or voice) if currently popular 4G-bytes SD card is used. Further, a RS $\mu - Card$ has smaller size, up to 2TB, with bandwidth 120MB/s is available, which may bring a full size fixed codebook realized if the codebook is not computed with the parameters at the side of the user. Our wish is even the fixed codebook is not all computed, and only the part being needed in a mail is computed. Currently, a user's device needs a fixed codebook ready, and some parameters are transmitted via the Net.

The ball can quicken the receiver's hand held device to transform to be the part of the floating codebook (sub-permutation) of his own and the sender's as well as receiving the parameters via the Net [10, 11, 12]. Due to the range of this article being limited, how to treat our algorithm feasible in real time on a very small device still needs more effort (if we still appreciate 1024 shells). The proposed system probably is good for PC or notebook level device if the technique of sub-permutation is not used. Please refer to the section of "Discussion" to check more about the sub-permutation.

If somebody insists always 16779264!, it is reachable. It is to map each point to a state of full permutation. So each point may have a possibility of one of 16779264! choices. Therefore the declaration of 16779264! is true. It is because when a point maps to the other point, the information contained at each point is changed to be mapped to the other one of the 16779264!. The structure of the signature is the structure of the ball. This is why the subject of the article is named. And it is seen that the signature is so solid from any falsification or attack. The following is the structure of the ball.

5 The Introduction and the Structure of the Ball

It is proposed, not necessary, on the surface of a ball, there are many points equally spaced with the other points. A way for doing so is making two pyramids with the square bottoms coincided with the other and all the edge points internally connected to the surface of a ball, please refer to Figure 2. Hence the ball has six (1+4+1=6) points on the surface and each point is equally spaced with each other surround it, i.e. for any point, there are four points around it and all the distances between each close pair of points are equal. Now, we may see there are totally twelve arcs between each two close pair of points. Each of the arcs is now halved. Hence, there are 18 (1+4+8+4+1=1+4(1+2+1)+1=18) points equally spaced. The same process is repeated again and again, there will be:

- $n = 0 : 1 + 4 + 1 = 1 + 4(1) + 1 = 6$ (points);

- $n = 1$ (one halving from the two-pyramid status): $1 + 4 + 8 + 4 + 1 = 1 + 4(1 + 2 + 1) + 1 = 18$;

- $n = 2$:

$$\begin{aligned} & 1 + 4 + 8 + 12 + 16 + 12 + 8 + 4 + 1 \\ = \; & 1 + 4(1 + 2 + 3 + 4 + 3 + 2 + 1) \\ = \; & 66; \end{aligned}$$

- $n = 3$:

$$\begin{aligned} & 1 + 4 + 8 + 12 + 16 + 20 + 24 + 28 + 32 \\ & \quad +28 + 24 + 20 + 16 + 12 + 8 + 4 + 1 \\ = \; & 1 + 4(1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 7 \\ & \quad +6 + 5 + 4 + 3 + 2 + 1) + 1 \\ = \; & 258; \end{aligned}$$

- $n = 4$:

$$\begin{aligned} & 1 + 4 + 8 + 12 + 16 + 20 + 24 + 28 + 32 \\ & \quad +36 + 40 + 44 + 48 + 52 + 56 + 60 + 64 \\ & \quad +60 + 56 + 52 + 48 + 44 + 40 + 36 + 32 \\ & \quad +28 + 24 + 20 + 16 + 12 + 8 + 4 + 1 \\ = \; & 1 + 4(1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 \\ & \quad +11 + 12 + 13 + 14 + 15 + 16 + 15 + 14 \\ & \quad +13 + 12 + 11 + 10 + 9 + 8 + 7 + 6 + 5 \\ & \quad +4 + 3 + 2 + 1) + 1 \\ = \; & 2 + 2^{2(4+1)} = 2 + 2^{10} = 2 + 1024 = 1026; \end{aligned}$$

- $n = 5$:

$$\begin{aligned} & 1 + 4(1 + ... + 4 + ... + 8 + ... + 16 + ... + 32 \\ & \quad +... + 16 + ... + 8 + ... + 4 + ... + 1) + 1 \\ = \; & 2 + 2^{2(5+1)} = 2 + 2^{12} = 4098; \end{aligned}$$

- $n = 6$:

$$\begin{aligned} & 1 + 4(1 + ... + 4 + ... + 8 + ... + 16 + ... \\ & \quad +32 + ... + 64 + ... + 32 + ... + 16 + ... \\ & \quad +8 + ... + 4 + ... + 1) + 1 \\ = \; & 2 + 2^{14} = 16386. \end{aligned}$$

There are rings $0, 1, 2, ..., 63, 64, ..., 128$. totally 129 rings, always $2^{n+1} + 1$ rings. For n, the number of the equally spaced points is $2 + 2^{2(n+1)}$.

For n=6, the points around the equator of the ball are $4 \times 64 = 256 = 2^8 = 4 \times 2^6$, the n=6 means based on four points originally on the equator, we halve the equator six times from the status of $n = 0$, the status of the two pyramids. A dice with triangular facets can have 6, 18, 66, 258, 1026, 4098, 16386,..., $2 + 2^{2(n+1)}$ points corresponding to 8, 32, 125, 512, 2096, 8384, 33536,..., 2^{2n+3} facets, respectively. As n going larger, the ratio of the number of the facets to the number of the points approaches 2.

It is not necessary, but we would like to adopt n=6. So, on the surface of the ball, there are 16386 points equally spaced. We also suggest that the ball has 1024 shells, and on each shell there are 16386 grids [15]. 1024 is not necessary, and it is just a suggestion.

Consider all the corresponding points on each shell lined up on a radius always. And, any shell rotates about the center of the ball with points always on radii. For a shell, after rotation, any point shall just be moved to be right at some point which was a point before. In other words, a point can just be moved to some discrete position in the space. It is obviously, a rotation in this way is position exchange. If each point of this ball is used as an address, each point can stand for a set of three integers as its address or position. The rotation of any shell of this ball keeps all the addresses always be occupied. It is clear that the sample space $S_j \subset S \subset R^3$, where S_j is a set of all the addresses or points of the j^{th} shell. For any shell, any two points can be exchanged because it is possible to move one point to any position of a third point without affecting each other. The details of how rotation rules or operations shall not be discussed in this article due to being not required for our purpose in aiming at structuring a set of data on the points or addresses. The required knowledge is just figuring out that points can always be at new positions and right at the points. So, it is clear the addresses can be changed after some rotations on a single shell. We may move a point or some points between two different shells. Therefore, any point can be moved to a new position. In other words, all the points or the addresses of the ball can be reassigned to new positions. And it is a permutation. That is all the old positions always being occupied by only one point each, and any point always occupy its own unique address. Anyway, each point can be reassigned by some ways to have a new address, and no address is vacant.

If we think transformation or encryption is limited to reassignment of the points and addresses, no matter what kind transformation being adopted, all the computed values are just limited to three integers. This explains the computation has two features: the range of computation within some numbers, say 256, 256 and 1024, and all computations are integer computations without error accumulation. A small device with a simple microchip and limited memory can handle the computations and limited algorithm or a generating equation.

Suppose we assign one selected word or one selected character on one unique shell, there are totally 1024 matters (1024 is already enough for installing a key board) can be mapped to 1024 shells with one to one correspondence. Now a shell still has 16385 vacancies since only one point is occupied. The remained 16385 addresses can be assigned to some datum each. We are not going to suggest how to fill data to the addresses, but it is clear so many addresses can fill data with many addresses vacant. Please be noted, the addresses are expressed by three (positive) integers less or equal to 256, 256 and 1024. For individual user, his device does or does not contain all the data of

as many as 16386 times 1024. For a sender, he can write a plaintext, and the plaintext shall be translated as codes with respect to the ball arrangement of his device. And the codes can be sent to the receiver whose device having different arranged ball. The sender reported to the upper level, the business. The business shall mail a set of key to the receiver for the receiver to compute the (true) key or the cipher text and reversely translated with the receiver's ball arrangement to have the plaintext back. Please referred to Figure 1, if a piece of the plaintext in codes form of some original ball is x_i, we see we can take any one point in the space as the codes of the plaintext of the sender and the other point in the space as the codes of the receiver with respect to the users' ball arrangements, respectively. The tracks from the codes of the sender's and the codes of the receiver's to x_i are the transformation between the balls of the business and the sender and the receiver. After all, the codes shall be translated in to human language readable to the sender and the receiver. We may view the codes x_i is the codes on ASCII. The secrecy is the track which links the codes of the sender's and the receiver's. This track can be arbitrary or being formed with some announced blocks (ensembles or a subset of an ensemble). In the last case, the secrecy is some parameters which are found with respect to the blocks. After all, some secret data are what needed without being in the hands of the attacker. Our aim is not to discuss what's easy or known as proposing a cryptosystem, but pointing out the possibility for build up many cryptosystems. However, for introduction, we can not but introduce a cryptosystem later.

If the shells are exchanged and rotated, the ball provides an image for knowing some points are moved to some new positions within S (all the points of the union of all shells).

6 A Proposed Cryptosystem

Please refer to Figure 1. Suppose a piece of information written in human language is denoted as $w = \{w_i | 1 \leq i \leq m, m \in N\}$, and though w could be an ordered multiplicity set, we just think it is a set. And, suppose w is translated to be a set $x = \{x_i | 1 \leq i \leq m, m \in N\}$. x is transformed to be a set f, where $f = 2(1(o(x)))$, 2, 1 and o are the transformations or the tracks. If 2, 1 and o are matrices of transformation, precisely there are $f_i = 2_i 1_i o_i x_i$ (or denoted as $f_i = 21o_i x_i$) and $f = \{f_i | 1 \leq i \leq m, i, m \in N\}$, the cipher text encrypted with the fixed codebook. Let $y_i = 3_i 21o_i x_i$, and $y = \{y_i | 1 \leq i \leq m, i, m \in N\}$. y is the cipher text based on the floating codebook. For an x_i, f_i always is the same because Track 2 is a certain transformation of the invariant identification of the sender. And Track 3 is a transformation (function) of time random variable. When an f_i is going to be encrypted again, the software shall select a time value to decide a transformation (Track 3). We suggest a three-minute cyclic interval is divided into 10^8 sub-intervals. When a specified button is pressed, the moment shall decide the time is in what time sub-interval. Therefore a value is decided. A simple example is to make 16386 points be in a certain order, and the number of the sub-interval is to make the corresponding point be the rotation center to make the part of the shell or shells to rotate 90 degrees (two number sets are required) and/or something like this. We may make an x_i be rotated individually, or a subset of x be rotated by a transformation.

The key Track 7 and the parameter for Track 3' are mailed via the Net from the business (primed) to the receiver. Hence the receiver can have f_i' or f' directly, without computing for y_i'. Even the receiver may compute directly for x_i because Track 4′ is in the hand of the receiver. If the receiver can successfully solve for the plaintext (x or the related human language), as being checked before, the plaintext is true and the sender is the correct sender himself. With a legal agreement, a sender is aware of his responsibility to put a certain signal (codes; shall be encrypted with a floating codebook) together with the cipher text or just to send his agreement to the notarization authority so as his name in a list saying that he is one of the persons declared to take the responsibility.

Such a ball has some nature good for carrying a whole batch transformation. Think about, for instance, the ball is rotated as a whole, and some portion of the ball is exchanged with the other counter portion of the ball. There are many points being moved (that's transformations concerning many points at one time). Among the points, the sender selects some. If a transformation must be specified and then can be computed, which one or ones are selected is no other than telling the business what the plaintext is. And if the cipher text should be processed by the business, no matter how hard being tried, the security design is never secure-no secrecy is preserved from being transferred via the Net. In the proposed cryptosystem, the secrecy is the fixed codebooks, which are never transmitted via the Net. We may even conclude, without the tool (the ball) with the good nature, we can not easily separate transformations (the operators) from the codes (the operands) in engineering, i.e. critical computation of operators and the limited operands being performed at the user's end only.

Without the separation, the security is not good enough to reach uniqueness about the cipher text. Without the uniqueness, signature can be mixed up, though perhaps not deadly harmful to the information security. Since signature assurance is not possible, without non-repudiation, no responsibility can be confirmed. Therefore, it is simply that signature is failed. Separation of the transformations (operators) and the codes (operands) is essential for security and signature. The success of signature is the foundation of information assurance, including authentication.

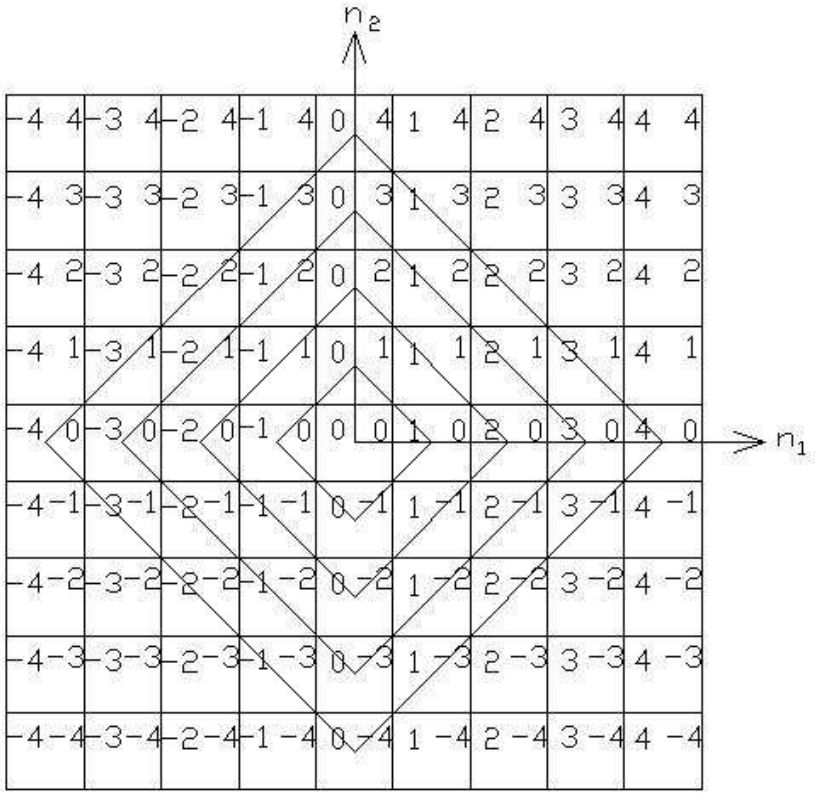

Figure 3: A view of the spatial informative lock emphasizing the loops

**At the very center of this figure, there is a small circle #0 with 00 in it which stands for the original point. Around the origin, there are four grids to form the loop #1. Outside the loop #1, there are eight grids to form loop #2, and so forth (until #64 and shrinking to #129, just one point). No matter which grid is selected as the origin, around the origin, there are always four points. The reader may view this figure as a top view. This figure can be rotated by 90-degree. The other way to view this figure is to think what seen being just a part of a big ball. It is seen that the sum of the two absolute values of the two components on the same ring is the number of the ring. Please refer to "Discussion", and this table is good for making subsets for sub-permutation (like playing a Russian Cube). The user just encircles the set or points on the screen to pick up the points of the set (being highlighted), and specifies two (adjacent) points, one linking to a new position as the rotation center and the other for specifying the angle, making clockwise circle for the points of the upper semi-sphere, and counterclockwise circle for the points of the lower semi-sphere. This flattened figure makes all kinds of rotation very easy in specifying and computing. This figure is even good for handwriting in helping to characterize points. With this figure, any transformation of the ball is simplified, including layer exchange. The settings are recorded as the parameters for encrypting, and decrypting is always reversible.*

7 The Structure of a Signature

Please refer to Figure 3. In fact in the way of extending the ball points topologically into a planar grid array, the points on the equator of the ball can also be extended as making the array of Figure 3 larger. We may hand write a signature on a plane and make the two planes coincided. Therefore the traces of the handwriting shall pick up many points of the array with a protocol of the nearest points to the traces being selected.

These points will provide the addresses (positions of the points selected) as a set of codes. We may view the codes as a part of the plaintext, like x, denoted as $\overline{x}$. And we shall have the relating $\overline{y}$. After the receiver solved these codes in $\overline{x}$, the codes shall be returned to the traces with curve fitting of least squared minimization. Therefore the handwritings are back.

his work does not have any extra true meaning, but it can give the receiver better feeling due to having the handwritings. If we would like such a work having better meaning, we may analyze the handwritings ($\overline{x}$) to have the characteristic features of the sender's signature [24]. It is beyond our range in this article.

If avalanche design is included, it is not hard to make the plaintext and the signature (traces of handwritings) be kept from falsification simultaneously, i.e. both the plaintext and the signature are true otherwise both are incorrect. Here we suggest algebraic simultaneous equation set as an example in order to persuade the readers how easy the principle of avalanche can be performed. Suppose there is a set of simultaneous ($m = n$):

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{12} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{12} & y_{22} & y_{23} \\ \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & y_{n3} \end{bmatrix}$$

$$= \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} & b_{m3} \end{bmatrix} = AY = B.$$

For nonsingular A, always A^{-1}, Y is the codes combined matrix of the codes of the cipher text and/or codes from the signature, and B is a matrix of codes under avalanche treatment. B shall be transmitted via the Net in the form: $...b_{ij}...$, only the correct receiver knows how to get correct B (after correct partitioning) and has A to have Y, a portion of the cipher text. If y_{ij} is x_{ij}, without all related codes b_{ij} being solved (decrypted), x_{ij} can not be solved.

Due to a mail with one floating codebook (just being used one time; the indirect source of the signature; the direct source of the signature is the fixed codebook) being just like a sheet of paper with the identity verification and true signature being just like the imprinted background on the sheet of paper already, any sender just writes and mails the mail without toil. The signature or the identity comes out from the floating codebook being used and extends to all the codes up to the moment of a button of the keyboard is pressed. After that, even the sender can not change anything. The software at this moment encrypts the plaintext with the floating codebook and mails it to the receiver.

We encourage that the sender may put a special sentence with common memory between the sender and the receiver ("dialect" is one of them), a social security number or post address, a secret or plain logo known by the receiver, just asking (via the Net or telephone, with or without encrypting) the receiver what the sender should add in the mail, the name of the sender or the receiver (particularly in the case of endorsement (the signature is an object), some critical points of the endorsed document should enter the cryptosystem to provide certain codes and ensure the range by the endorsement), a public information such as public key or so (these are determined information) in the mail to check if they are correctly solved. The function is similar to a stranger sends a letter to you with a signature, it is just meaningless because we don't have some known information to compare with. Besides, doing so is to increase the confidence on the mail subjectively and objectively.

Though we wouldn't say our system is versatile, it might be possible to build a true ball with sensors at the points of some shells of the ball. Making the sender to grasp the ball, the induced stress or strain shall be sensed and recorded as the codes. The status of grasping might be unique because the muscles and the characteristics of the sender's hand should be unique. It is analogous to the handwriting.

8 Discussion

The fact of the ensemble transformation in codebook form is a long list of $u_k \longleftrightarrow v_k$, each u_k or v_k is a three-component code. A code u_p (for instance $u_p \in x$) is then converted to v_p, and the business doesn't know u_p is converted or v_p included in the cipher text. It is good that the business or the third party does not know that, but the pair list is so long as16779264. So the transformations themselves are adopted and even with parameters to be specified, and so as even they may be transmitted via the Net among all parties. Normally, for ensemble transformation, there is $s_j \subseteq S$, and s_j is transformed, where s_j containing some or none $y_i \in y$. There are many ways

$$\prod_{j=1}^{K} [C_{\|s_j\|}^{16779264} P_{\|s_j\|}^{16779264}]$$

$$= \prod_{j=1}^{K} [16779264!/(16779264 - \|s_j\|)!]^2/\|s_j\|!$$

of $s_j \subseteq S$ to transform (with exchange operation), and therefore it not harmful to security if a few exchanges and/or rotations are well designed, where $\|s_j\|$ is the number of the points being selected at j^{th} time, and K is how many times of the selecting a subset and transforming.

This number approaches its upper bound 16779264! efficiently. The sub-permutation is good enough, and the sub-permutation can be represented by parameters to define s_j, $1 \leq j \leq K$ (for instance, a circle just defined by the center and the radius) and exchanges. The communication quantity is short, and separation of the operators and the operands is feasible. Besides, the rotation and exchange are cyclic discrete and forming a system with closeness. Hence the ball is one of the ideal tools for executing the ensemble transformation and/or specified transformations.

Signature is imprinted by the sender at Track 3, and with the sender's identity verified at Track 2 by the business (a kind of authentication and/or notarization; due to reporting to the business who was the receiver and the random parameters were chosen for Track 3). Besides, each codebook is unique, different from another, is the base why it can carry the signature. And, the security is high as all what mentioned in the article. These fulfill our knowledge of the signature in "Introduction".

9 Conclusion

The invisible signature is embedded inside a sturdy cryptosystem with the uniqueness of the fixed codebook, and the signature concerns the correctness of the sender's identity and the correctness of the plaintext which promised by the signature. Falsification or incorrectness is not allowed is essential. A powerful tool is helpful to construct good cryptosystems, and the floating codebooks carry the signature and the associated plaintext, one for each time. Finally the ball is the tool for performing the signature. The structure of signature is the whole system allowing each person's identity being uniquely specified by and with what he sends via the Net.

References

[1] A. Aziz, and W. Diffie, "Privacy and authentication for wireless local area networks," *IEEE Personal Communications*, pp. 25-31, 1994.

[2] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," *42nd FOCS*, pp. 136-145, 2001.

[3] W. Diffie, and M. E. Hellman, "New directions in cryptography," *IEEE Transactions Information Theory*, pp. 444-654, 1976.

[4] W. Diffie, and M. E. Hellman, "Privacy and authentication: An introduction to cryptography," *Proceedings, IEEE*, pp. 397-427, 1979.

[5] C. F. Gerald, and P.O. Wheatly, *Applied Numerical Analysis*, 6th Edition, Addison-Wesley Publishing Company, ISBN 978-0201870725, New York, 1999.

[6] M. E. Hellman, "An extension of the shannon theory approach to cryptography," *IEEE Transactions on Information Theory*, vol. IT-23, pp. 289-294, 1977.

[7] D. Huang, and D. Medhi, "A Key-chain based keying scheme for many-to-many secure group communication," *ACM Transaction on Information and System Security (TISSEC)*, pp. 523-552, 2004.

[8] C. C. Huang, C. Huang, and P.Liu, "An encryption method Based on dynamic codebook," *The 17th International Conference on Information Management (ICIM)*, I-SHOU University, 2006.

[9] C. C. Huang, C. Huang, and P. Liu, "The encryption of floating codebook," *The Second International Conference on Information Management and Project Management Schedule*, Kainan University, 2006.

[10] C. C. Huang, C. Huang, and S. Y. Huang, "Floating password in mutual date link," *Crass-Strait Conference an Next Generation Internet Services and Applications*, pp. 105-110, Asia University, 2007.

[11] C. C. Huang, C. Huang, and S. Y. Huang, "Mutual data link as a floating password," *e-CASE 2007 International Joint Conference on e-Commerce, e-Administration, e-Society, and e-Education*, pp. 50, Hong Kong, 2007.

[12] C. C. Huang, C. Huang, S. Y. Huang, and W. H. Liu, "The mutual link floating Ppassword," *2007 National Computer Symposium, NCS*, pp. 635-644, 2007.

[13] A. Kahate, *Cryptography and Network Security*, International Edition, McGraw-Hill, ISBN 007-123477-2, Singapore, 2003.

[14] E. Kreyszig, *Advanced Engineering Mathematics*, 7nd ed., John Wiley & Sons, ISBN 0-471-59989-1. New York, 1993.

[15] C. L. Liu, *Elements of Discrete Mathematics*, 2nd ed., McGraw-Hill International, ISBN 0-07-100544-7, 1998.

[16] P. Liu, *Randomly Dynamic Cryptosystem*, Department of Information Science and Applications, Master Thesis, Asia University, 2006.

[17] M. Mcloone, and J. V. Mccanny, "Generic architecture and semiconductor intellectual property cores for advanced encryption standard cryptography," *IEE Proceedings-Computers and Digital Techniques*, pp. 239-244, 2003.

[18] K. G. Paterson, and A. K. L. Yau, "Lost in translation: Theory and practice in cryptography," *In IEEE Security & Privacy*, pp. 69-72, 2006.

[19] R. P. Paul, *Robot Manipulators*, ISBN 978-0-262-16082-7, The MIT Press, Cambridge, MA, 1981.

[20] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, pp. 379-423, 1948.

[21] C. E. Shannon, "Communication theory of secret systems," *Bell System Technical Journal*, pp. 665-715, 1949.

[22] G. J. Simmons, "An introduction to the mathematics of trust in security protocols," *Proceedings of Computer Security Foundations Workshop VIs*, pp. 121-127, IEEE Computer Society Press, Los Alamito, California, 1993.

[23] L. W. Tsai, *Robot Analysis*, ISBN 0-471-32593-7, John Wiley & Sons., INC., 1999.

[24] D. Venugopal, "An efficient signature representation and matching method for mobile devices," *Proceedings of the 2nd annual international workshop on Wireless internet WICON' 06*, vol. 120, article no. 16, 2006.

[25] N. J. Victory, *Electronic Signatures: A Review of the Exceptions to the Electronic Signatures in Global and National Commerce Act* U.S. department of commerce National Telecommunications and Information Administration, 2003.

[26] M. Wang, "A review of electronic signatures regulations: Do they facilitate or impede international electronic commerce?," *Proceedings of the 8th International Conference on Electronic commerce: The New E-commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet (ICEC '06)*, pp. 548-552, 2006.

[27] I. Zeid, *Mastering CAD/CAM*, ISBN 0-07-286845-7, 1st ed, McGraw-Hill International, 2005.

Chuen Chuan Huang received a Ph.D. in Engineering Mechanics from the Dept. of Engineering Science and Mechanics, Virginia Tech, Blacksburg, VA, 24060, USA, 1992, a Master of Science in Mechanical Engineering from the Institute of Mechanical Engineering, Tamkang University, Tamsui, 25137, Taiwan, 1983, and a Bachelor of Science in Engineering Science from the Dept. of Engineering Science, National Chen Kung University, Tainan, 70101, Taiwan, 1977. Currently he is an associated professor of the Dept. of Computer and Communication Engineering, Asia University, Wufong, Taichung, 41354, Taiwan. Mostly he worked in CSIST for SAM in guidance and control and civilian manufacturing factories before teaching. Currently he works for cryptosystems, intelligent road system and mono-frequency communication.

Chuen-Der Huang is an associate professor of the Department of Electrical Engineering at Hsiuping Institute of Technology, Taiwan. He received a Bachelor of Science in Electrical Engineering and a Master of Science in Control Engineering from FangChia University, Taiwan, and a Ph.D. in Electrical and Control Engineering from National ChiaoTung University, Taiwan. His research interest includes control, measurement, bioinformatics, information fusion, and intelligent system.

Shin Ya Huang is a graduated student of Dept. of Computer and Communication Engineering, Asia University, Taiwan.