A Model for Embedding and Authorizing Digital Signatures in Printed Documents

Jae-il Lee[1], Takyoung Kwon[2], Sanghoom Song[2], and Jooseok Song[3]
[1]Korea Informations Security Agency, Seoul 138-803, Korea
[2]Sejong University, Seoul 143-747, Korea
[3] Yonsei University, Seoul 120-749, Korea

Abstract. It is a desirable feature in a public key infrastructure (PKI) to include the signature information in a printed document for anthenticity and integrity checks, in a way to bind an electronic document to the printed document. However, it is not easy to preserve the digital signature in the printed document because the digital signature is for the text code (or the whole document file), not for the text image (which can be scanned optically) in printed form. So, we propose a practical and secure method for preserving the authorized digital signatures for printed documents. We will derive a printable digital signature scheme from the Korean Certificate-based Digital Signature Algorithm (KCDSA) for secure transaction and utilize the dense two-dimensional barcode, QRcode, for printing out the signature and data in a small area within a printed document.
Keywords: Computer Security, Distribute Systems Security, Practical Aspects.

1. Introduction

A digital signature is a bit·string which associates a message in electronic form with an original signer. It has various kinds of applications in information security, including authentication, data integrity, and non-repudiation. In order to be accepted widely, the digital signature must be verifiable in an authentic manner. A public key infrastructure (PKI) plays an important role in that sense[2]. However, once a digitally signed electronic document is printed out in tangible form, it is not easy to assert the validity of signatures in the printed document. This is because the digital signature is for the text code (or the whole document file), not for the text image (which can be scanned optically) in printed documents. When we consider the wide acceptance of digital signatures in real world application, it seems a desirable feature in the PKI to print out a digitally signed document and verify the authorized signature in the from of either an electronic file or a tangible paper. For this purpose we may consider a simple method of embedding the digital signature into the printed document by using a ubiquitous bar code scheme[8].

However, this could not easily work because the electronic document may involve a specific a format in a computer system that could never be expressed in printed form, and optical scanning is not a panacea.

In this paper, we propose a practical and secure model for preserving digital signatures of electronic and printed documents. Firstly we will design a security model and build a digital signature protocol according to this model. We will derive the printable digital signature scheme from the Korean certificate-based Digital Signature Algorithm (KCDSA) for secure transaction and utilize the dense two-dimensional barcode, QRcode, for printing out the signature and data in a small area within a printed document[3, 12]. This paper is organized as follows: In Section 2, we will describe a basic model for preserving the authorized signatures of electronic and printed documents. According to this model, a modified version of the KC-DSA will be provided in Section 3, in order to assert the authenticity and integrity of documents in printed form. Subsequently a practical matrix code will be scrutinized in Section 4, so as to embed the printable digital signature in the printed document. Finally we will conclude this paper in Section 5.

2. Basic Model

2.1. Overview

The goal of our study is to build a fundamental model for preserving the authorized signatures of electronic documents even in printed document form. In other words, the authenticity imposed on an electronic document should be validated in a similar manner even if the document is transformed into printed form, i.e., a tangible and readable paper. A digital signature can simply be embedded into a printed document in that sense. However, care must be taken when we devise such a scheme because an adversarial attempt can easily be made to transform the electronic document into the printed document.

We will define the basic components of our model for embedding and authorizing digital signatures even in printed documents. Our strong assumption is that the PKI is provided, so that every participant is able to acquire an authentic public key of another participant in our model.

Definition 1. *Our model is a 5-tuple of* $<\varepsilon, p, S_1, S_2 M)$

- ε: Entities
- P: protocol
- S_1 : Digital signature scheme
- S_2 : Digital signature scheme for printed documents
- M : Matric code scheme

The principal entities run a well-defined protocol for generation and verification of digital signatures. As we see digital signature schemes must be provided for the printed documents as well as the electronic documents, and there must be a distinct relationship between them The same kind of digital signature schemes may be used for S_1 and S_2 with only sight modification. We will discuss further details in Section 3. Finally a matrix code, a kind of bar code, is necessary for embedding the printable digital signature scheme for printed document. Further details will be provided in code Section 4.

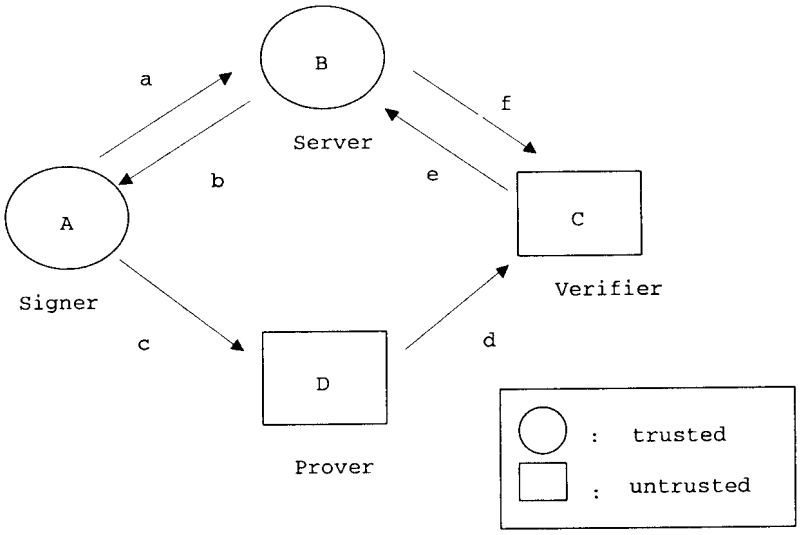

Fig. 1. Relation among Principals

2.2 Principal and Adversarial Entities

First we define the following entities who principally participate and run a protocol, called a Digital but Printable Signature Protocol (DPSP), in our model. In addition, two kinds of adversarial entities must be defined in our model.

Definition 2. *ε includes a 4-tuple of <A, B, C, D> as principal entities and a 2-tuple of <E, E'> as adversarial entities.*

- *A*: Signer who may generate digital signatures
- *B*: Sever who may control the verification process.
- *C*: Verifier who may verify digital signatures.
- *D*: Prover who may attempt to prove digital signatures.
- *E*: Passive adversary who may eavesdrop the network.
- *E'*: Active adversary if E defines a principal entity.

Figure 1 shows the DPSP relation among the principals. As depicted in Figure 1, A first agrees on signature generation with *B* (see flows *a* and *b*) and then sends *D* a digitally signed document(see flow *c*). Then, *D* may print out the document and try to prove its validity to *C* (see flow *d*). Finally *C* verifies the printed signature with *B* (See flows *c* and *f*). Here are the clear requirements for the relationship of the

principal entities.
- A trusts B in that B may control the verification process.
- A does not need to reveal the primary data[1)] to B for the trust.
- A lets D print out the signed document.
- A does not trust C and D in terms of integrity (A may suspect that C and D could compromise the signed document.)
- B trusts A in that A may generate a valid signature.
- B does not trust C and D in terms of integrity. (B may suspect that C and D could compromise the signed document.)
- C trusts B in that B may control the verification process.
- C does not need to reveal the primary data to B for the trust.
- C does not trust D. (C may suspect that D could modify or reuse the A-signed document.)
- D trusts A in that A may generate a valid signature.
- D lets C verify the A-signed document.

As we can see, we need to define two kinds of clear paths in our mode1. They are path A-B-C for controlling a printable signature, and path A-D-C for performing signature transaction. In a sense of concatenating the principal entities in our model, the path A-B-C must be provided with secrecy on a document and the path A-D-C with integrity of the document. In other words, A and C should not reveal the document to B on path A-B-C, while A and C should not allow D to modify the document on path A-D-C.

All entities, A, B, C, and D, are considered to be separated in the above requirements. In order to assert flexibility, however, we can consider the following cases in our model.
- Each entity is separated. In this case, the above requirements are completely applied.
- A and B are the same entity. In this case, C must trust A as it did B in the above.
- A and D are the same entity.
- A, B, and C are the same entity.

2.3 Basic Protocol

A protocol P is called the DPSP, and can be defined in general as shown in Figure 1 above.

Definition 3. *P is composed of signature generation, signature transaction, and signature verification.*

1. Signature generation

---

1)In this paper, the primary data means the data that must be approved in terms ofauthenticity and integrity. Also the primary field means a field in which the primary data is located. Such a field could be the printable text or binary image.

(a) Make an electronic document.
(b) Select primary fields.
(c) Generate a data matrix code for the primary fields.
(d) Generate a signature matrix code for the data matrix code.
(e) Insert the matrix codes into the electronic document.
(f) Generate a digital signature for the complete document.
(g) Attach the digital signature to the document.

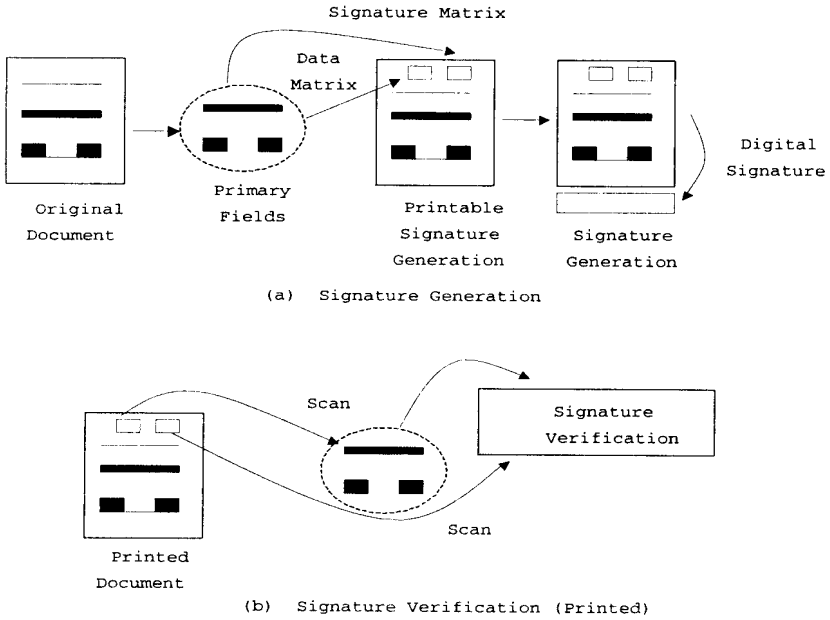

Fig. 2. Signature Generation and Verification

2. Signature transaction
(a) Verify the digital signature of the electronic document.
(b) Print out the digitally signed document.
3. Signature verification
(a) Scan the matrix codes of the printed document.
(b) Verify the primary fields of the data matrix code.
(c) Verify the digital signature of the primary fields.

The signature generation corresponds to *a* and *b* in Figure 1 and the signature verification to *e* and *f*. Similarly the signature transaction corresponds to *c* and *d* in Figure 1. The primary fields mean chosen data that must be signed in an authentic manner. The whole data may not need to be signed in general, so that we propose to generate a printable signature on the chosen data only, as depicted in Figure 2. The primary fields must be clearly marked in the original document, so that a verifier can easily detect the primary fields in the printed document on verifying them after scanning the data matrix. Note that the data matrix should be scanned for showing the signed primary field data on verifier's display. The Data could be

displayed, so long as the signature is correctly verified. In that sense, the data verification might be performed manually while the signature verification should be performed automatically. The automatic comparison of the printed and displayed primary fields is not considered in this paper.

As for choosing primary fields, it would be advantageous to a verifier to make location information involved in a data matrix. That means one encodes the primary fields with the location information of each, for example, a 3-tuple of {*data, x-location, y-location*}, We omit the details in this paper. Note that it is optional to choose the primary fields. One can generate a signature on the whole data of the document. The data matrix code must encode the primary fields along with the document identity information. Finally, the signature matrix code must encode the printable signature on the data matrix code. Both matrix codes should be made printable on the original document. Figure 2 summarizes these concepts.

2.4 Signature Schemes

Any kinds of digital signature schemes can be used for S_1 and S_2 However, the chosen scheme must carefully be considered and modified so as to satisfy the following requirements on S_2. Note that this is the reason why we clearly separated S_1 and S_2 in our model definition.

1. A should not reveal the primary data to B on signature generation.
2. D should not be able to modify the data matrix on signature transaction.
3. D should not be able to modify the signature matrix on signature transaction.
4. D should not be able to reuse the signed document unless it is permitted.
5. C should not reveal the primary data to B on signature verification.
6. C should not be able to modify the data matrix after signature verification.
7. C should not be able to modify the signature matrix after signature verification.
8. C should not be able to reuse the signed document unless it is permitted. We have given the critical requirements for the printable signature scheme, S_2, in our mode1. We will introduce a carefully derived scheme, P-KCDSA (Printable KCDSA), in Section 3.2.

2.5 Matrix Codes

A well-chosen encoding and code representation method must be used in our model. As we mentioned already, the matrix code, M, is defined as follows.

Definition 4. *M is a 2-tuple of $<M_D, M_S>$.*

- M_D : Data matrix that encodes the primary data fields and the printed document specific information, for example, a document serial number and an expiration date.
- M_S : Signature matrix that encodes the printable digital signature of the data matrix.

We will introduce the chosen schemes in Section 4.

3. Digital Signatures

In this paper, we consider the KCDSA as a possible instance of the DPSP for practical use. The other established signature schemes could be considered for the same purposes.

3.1 KCDSA

The KCDSA (Korean Certificate-based Digital Signature Algorithm) is one of the ElGamal-type signature schemes in which security is based on the hard problem of finding discrete logarithms over finite fields[3]. Two famous variants of the ElGamal signature scheme include the DSS (Digital Signature Standard)and GOST 34.10[6]. Readers are referred to [3] for the details of the KCDSA.

In this paper we utilize the KCDSA as a base signature scheme for S_1 and For S_2 S_1 we can utilize the KCDSA without any modification. However, we have to reconsider and modify it carefully in order to run S_2. Remember that the S_2 specific requirements were necessary in our model (See Section 2.4).

3.2 Printable KCDSA

We call the modified version of the KCDSA the Printable KCDSA, and abbreviate it to P-KCDSA. That is, the P-KCDSA is a slightly modified version of KCDSA satisfying the S_2 specific requirements described in Section 2.4. Figure 3 depicts the P-KCDSA message nows. In the figure, a message parenthesized by {and} is assumed to be a message encrypted under the recipient's public key for confidentiality, while a message parenthesized by [and] is assumed to be a message with a MAC (message authentication code) or a digital signature for integrity. In that sense, the primary feature of our protocol in P-KCDSA must be guaranteeing the integrity of r and s' in printed form because they are not encrypted nor digitally signed as shown in Figure 3. The P-KCDSA is as follows.

Parameter Setup. Client entity A should do the following things for choosing user parameters:

1. Select a large prime p such that $|p| = 512 + 256i$ where I = 0,…,6. We denote the bit-wise length by $|\ |$.
2. Select a prime q such that $|q| = 128 + 32j$ where j = 0,…,4 with the property that $q \mid (p-1)$.
3. Select a generator g of the unique cyclic subgroup of order q in Z_p^+ such that $g = \alpha^{(p-1)q} \bmod p$ for an element $\alpha \in Z_p^+$

4. Select an integer x at random from Z_p^+, Note that x is a private key in electronic KCDSA.

5. Compute a public key y such that $y = g^{x-1}$ mode p.

6. Acquire a certificate from a CA (Certificate Authority).

7. Compute a hash value of the certificate such that $z = h(CertData)$. Here we denote by *CertData* the signer's certificate data.

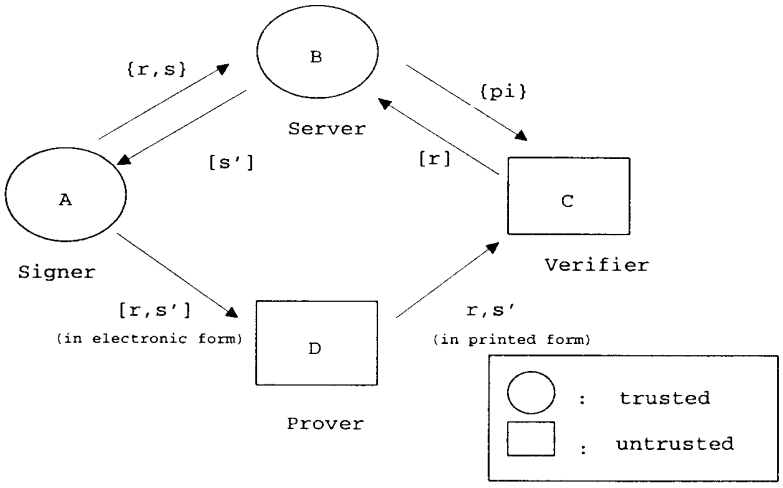

Fig. 3. Printable KCDSA Message Flows

Signature Generation. We assume A signs a binary message m of arbitrary length. Here m means the information encoded in the data matrix, i.e., the primary fields and the document specific data. Entity A should do the following:

1. Select random secret integer k such that $0 < k < q$.
2. Compute $r = h(g^k \bmod p)$.
3. Compute $s = x(k - r \oplus h(z, m)) \bmod q$.
4. Submit (r, s) and the certificate to a trusted server B in a confidential manner.

Then entity B should do the following:

1. Select random secret integer t such that $0 < t < q$.
2. Compute $s' = s + t \bmod q$.
3. Send s' to A in an authentic manner.
4. Compute and store a verification permit π such that $\pi = y^{-t} \bmod p$ along with r.

A's printable signature for m is (id, r, s'). We can say s' is an encrypted form of s while the transient key t may be shared by A and B. Note that t must be a private key while the corresponding public key g^{-t} (= π) will be provided to C in encrypted form in the next transaction. Also note that A and B could exchange a count C for restricting the number of verifiers (see below). Finally A should put it into the document by encoding it in a signature matrix code.

Signature Verification. To verify A's signature (id, r, s') on m, entity C should do the following:

1. Obtain the user's certificate.
2. Verify its authenticity; if not, then reject the signature.
3. Verify that $0 < r < q$ and $0 < s' < q$; if not, then reject the signature.
4. Request a verification permit to B by sending r in an authentic manner.

Then entity B should send the permit π to C in a confidential manner if it is allowed. Note that if B exchanged C with A above, B should decrease it when (s)he gives the permit to a verifier. If C is equal to zero, B should deny sending the permit. Though we utilized r as an index for maintaining π, one can devise more concrete scheme for the purpose. Finally C should do the following:

1. Compute $z = h(CertData)$
2. Compute $u = r \oplus h(z, m) \bmod q$.
3. Compute $w = y^{s'} \bmod p$.
4. Compute $v = h(w\pi g^u \bmod p)$.
5. Accept the signature if and only if $v = r$.

3.3 Analysis

We assume the KCDSA is secure in the random oracle model[3]. In that sense, a passive adversary, E, is not given any verifiable information on signature transaction. On the basis of the security of KCDSA, we will examine how the P-KCDSA satisfies the S_2 specific requirements described in Section 2.4, as a way of removing the threats of an active adversary, E'. Note that an active adversary who impersonates C is denoted by C' Assuming the respective private keys of A and B are safe, we consider C' and D' only in the following examination.

1. A should not reveal the primary data to B on signature generation.

- A sends B the initial signature r and s only in the P-KCDSA.
- B is not able to acquire any information on m from $r(=h(g^k \bmod p))$ and $s(=x(k-r \oplus h(z, m)) \bmod q)$ only.

2. D should not be able to modify the data matrix on signature transaction.

- The information in the data matrix is all signed by A and encoded in the signature matrix.
- D and D' are not able to generate a new signature of A on signature transaction without having x.

3. D should not be able to modify the signature matrix on signature transaction.

- The signature matrix contains r and s' rather than s, so that D cannot even verify the signature on signature transaction.
- D and D' cannot remove t from $s'(=s+t \bmod q)$ without having previous knowledge of s or t. In fact. D' cannot acquire t from s' because D' cannot decrypt out s from $\{r, s\}$.

4. D should not be able to reuse the signed document unless it is permitted.

- In order to use the printable signature, an intervention of B is always required in the protocol. That means a verifier C should ask B for a permit on signature verification.

5. C should not reveal the primary data to B on signature verification.

- C gives B the information r only on signature verification.
- B is not able to derive m from $r(= h(g^k \bmod p))$.

6. C should not be able to modify the data matrix after signature verification.

- For the same reasons as in 2 above.

7. C should not be able to modify the signature matrix after signature verification

- D cannot derive t from $\pi(= g^{-t} \bmod p)$ because of the hardness of solving the discrete logarithm problem.
- D' cannot decrypt out t from s' because of encryption on s.
- D and D' cannot remove t from s'.

8. C should not be able to reuse the signed document unless it is permitted.

- In order to use the printable signature, an intervention of B is always required. That means another verifier C^* should ask B for a permit on signature verification.

4. Embedding Digital Signatures with 2D bar code

4.1. 2D Bar Code

Bar code and human-readable text are often printed together, so little additional cost is associated with the inclusion of a bar code symbol. A bar code scanner can extract all of the information by scanning through a conventional bar code symbol. Because of the simplicity in data entry, bar code has become the dominant automatic identification technology [8].

Two-dimensional codes provide much higher information density than conventional bar codes. Due to the low information density, conventional bar codes usually function as keys to databases. However, the increased information density of 2D bar codes enables the applications that require encoding of explicit information rather than a database key. A 2D bar code symbol can hold up to about 4,300 alphanumeric characters or 3,000 bytes of binary data in a small area [12]. With the immense storage capacity, the development of 2D bar codes enables the data exchange under off-line condition[8]. The 2D bar code may work as a portable data file because the information can be received without access to a database. The 2D bar codes also have an excellent data restoration capability for a damaged symbol.

EDI(Electronic Data Interchange) has been proposed as a solution for quick exchange of large amounts of data in business. However, EDI faces serious practical limitations in an area with unreliable communication network or without

network connection. A potable data file containing detailed information can bean alternative to EDI. Many business models have been developed using 2D bar codes in the fields of logistics, construction, automobile, semiconductors and chemicals.
There are four widely used 2D bar codes that are ISO standard: PDF417, DataMatrix, QRcode and Maxicode. QRcode(Quick Response code) is particularly developed for high data capacity, reduced printing space, and high speed reading[12].

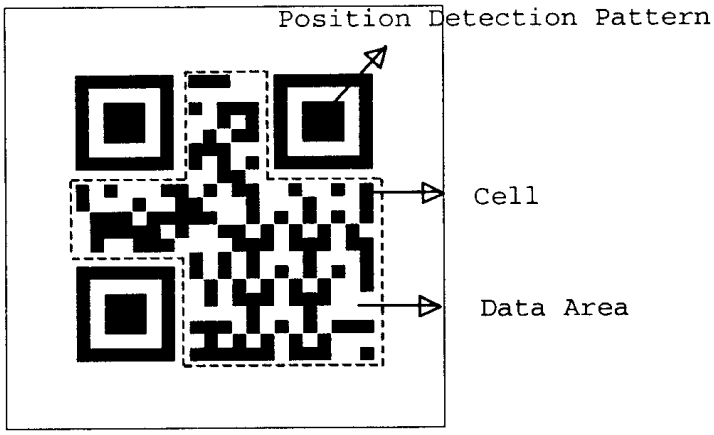

Fig. 4. QRcode Structure

4·2 QRcode

QRcode is a 2D matrix symbol which consists of square cells arranged in a square pattern. It allows three models - Model 1, Model 2, and MicroQR. Model 1 and Model 2 each have a position detection pattern in three corners while the MicroQR has it in only one corner. The position detection pattern allows code readers to quickly obtain the symbol size, position and tilt. Mode1 2 is developed for enhanced specification with improved position correction and large volume of data capacity. MicroQR model is suitable for small amounts of data. A QRcode symbol can encode up to 7,089 characters (numeric data), 4,296 alphanumeric characters, and 2,953 8-bit bytes[12].
The symbol size is determined by the number of characters to encode. It can grow by 4 cells/side from 21X21 cells to 177X177 cells. The physical size of a symbol is determined by the cell pitch. The minimum cell pitch is the width of the smallest printed element that can also be resolved by the reader. With current printing and reader technology, the minimum cell pitch can be as low as 0.1 mm and the signature data of 1024-bit(128 bytes) can be printed in an area less than 10 mm sq[14]. The Qrcode employs a Reed-Solomon algorithm to detect and correct data errors due to a dirtied or damaged area. There are four levels of error-correction capability that users can select. The error-correction level determines the maximum recoverable rate that is from 7

- Embedding the digital signature in printed documents will simplify the work-flows that require verifying the authenticity and integrity of documents. One example of a promising recent application is an electronic notice system using a 2D bar code in mobile phones. When we buy a ticket for a movie in advance, we can get the receipt in 2D bar code with the mobile phone.
The primary data of the 2D bar code in the receipt contains information about the viewer's name, the name of the movie, the data and time, seat number, etc. The theater can verify the 2D bar code receipt in a mobile phone using a system with 2D bar code scanner or may be equipped with a kiosk that issues a paper ticket for the 2D bar code receipt.

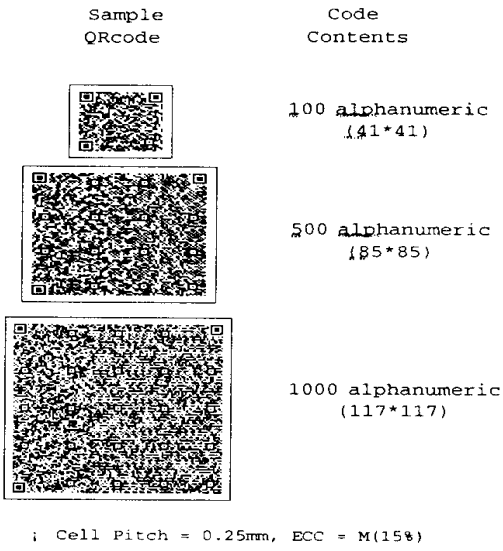

Fig. 5. QRcode Samples

If the mobile phone is not available, we can print the 2D bar code receipt on plain paper and bring it with a personal ID card to the theater. As another example, we can apply to a transcript service in a university. When a university graduate needs his official transcript for job applications, he may request the university to send his official transcript to him on line. He then prints the received electronic transcript on plain paper instead of the paper with the university official seal. The university embeds the primary data and signature data in printable 2D bar codes so that any third party can verify the authenticity and integrity of the printed transcript. The primary data has all the grade information including his personal information, such as name, student id, and birthday. If all the primary data cannot fit into one small 2D bar code, we can either make several additional 2D bar codes on one page or increase the size of the 2D bar code to hold more data.

5. Conclusion

In this paper, we proposed a practical and secure model for embedding and authorizing digital signatures in printed documents.
For this purpose, we carefully derived a printable signature scheme from the KCDSA, and selected an appropriate matrix code scheme. In a future study, we will implement the proposed model and analyze its applicability in more detail.
When a digitally signed document is printed out in a human-readable text image, it is useful to include the signature information in the text image for authenticity and integrity checks. With the development of dense 2D bar codes, we can put the digital signature in 2D bar code form into a small area of the printed document. Also, we can include several hundreds or thousands of alphanumeric characters in a small 2D bar code. We have proposed a practical and secure method to preserve authorized digital signatures in printed documents. The proposed model utilized KCDSA for secure transaction and the dense QRcode for printing out the signature and primary data in a small area.

Acknowledgement

We would like to thank anonymous referees for their invaluable comments on this work. Also we would like to express our deep appreciation to the committee members of ICISC 2002.

References

1. H. E, Burke, "Handbook of bar Coding Systems," *Van Nostrand Reinhold*, New York. N Y 1984
2. T. Kwon, "Digital signature algorithm for securing digital identities," *Information Processing Letters*, Vol. 82, Iss.5, pp 247-252, May 2002.
3. C Lim, "A study on the proposed Korean digital signature algorithm," *Advances in Cryptology-ASIACRYPT'98*, LNCS 1514, Spinger-Verlag, pp.175-186, 1998.
4. S, Lin and D. J. Costello Jr., "Error Control Coding, Fundamentals and Applications,'' *Prentice Hall*, Englewoo Cliffs, N.J ,1983.
5. A. Longacre, Jr., "Stacked Bar Code Symbologies,'' *Identification J.*, Vol. 11, No. 1, Jan /Feb., pp. 12-14, 1989.
6. M. Michels, D. Naccache and H. Pertersen, "GOST 34.10 - A. brief overview of Russia's DSA" *Computer Security* Vol. 15, No,8, pp.725-732, 1996.
7. NIST, "Digital signature standard,'' *Federal Information Processing Standards*

Pubication 186, 1994.
8. Roger C. Palmer, "The Bar Code Book," *Helmers Publishing*, Peterborough, N.H., 3rd Ed., 1995.
9 Theo Pavlidis, Jerome Swartz, and Ynjiun P. Wang, "Fundamentals of Bar Code Information Theory," *IEEE Computer*, Vol. 23, No. 4, pp 74-86, April 1990.
10. Y.P. Wang and T. Pavlidis, Optinlal Correspondence of String SubSequences," *IEEE Trans Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, No. 11, pp.1080-1087, Nov. 1990.
11. Y. P. Wang, "PDF417 Specification, "*Symbol Technologies*, Boemia, N. Y.,1991,
12, "QRmaker: User's Manual," *Denso Corporation*, Aichi, Japan, 1998.
13. "A Business Case Study QRcode, *Denso Wave Inc.*, Kariya, Japan, 2001.
14. "2D Code Solution," *Sunwoo Information Inc.*, Seoul, Korea, 2002.