On the Validity of Digital Signatures

Jianying Zhou and Robert Deng
Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore 119613
{jyzhou,deng}@krdl.org.sg

ABSTRACT

An important feature of digital signatures is to serve as non-repudiation evidence. To be eligible as non-repudiation evidence, a digital signature on an electronic document should remain valid until its expiry date which is specified by some non-repudiation policy. As signature keys may be compromised and the validity of signatures may become questionable, additional security mechanisms need to be imposed on digital signatures. This paper examines the mechanisms for maintaining the validity of digital signatures, and provides a guideline on the use of these mechanisms in various context of applications.

Keywords

digital signature, non-repudiation, secure electronic commerce

1. INTRODUCTION

The fast development of computer networks has facilitated electronic commerce. Many business transactions are shifting to the Internet. On the other hand, dispute of transactions is a common problem that could jeopardise business. For instance, the following disputes may arise when transferring an electronic message M from Alice to Bob.

- Alice claims that she has sent M to Bob while Bob denies having received it;
- Bob claims that he received M from Alice while Alice denies sending it.

In order to settle these disputes by a third party arbitrator, Alice and Bob need to present evidence to prove their own claims.

Digital signature is an important security mechanism for generating non-repudiation evidence [8], and is receiving legal recognition [7]. To be eligible as non-repudiation evidence, a digital signature on an electronic document should remain valid until its expiry date which is specified by some non-repudiation policy. As signature keys may be compromised and the validity of signatures may become questionable, additional security mechanisms need to be imposed on digital signatures. In this paper, we examine the mechanisms for maintaining the validity of digital signatures, and provide a guideline on the use of these mechanisms in various context of applications.

The rest of the paper is organised as follows. In the next section, we compare different properties of digital signatures and hand-written signatures, and analyse security requirements on digital signatures. After that, we investigate four approaches for maintaining the validity of digital signatures, i.e. time-stamping approach in Section 3, auditing approach in Section 4, one-way sequential link approach in Section 5, and temporary certificate approach in Section 6. We compare these approaches in Section 7, and conclude the paper in Section 8.

The following basic notation is used throughout the paper.

- X, Y: concatenation of two messages X and Y.
- $H(X)$: a one-way hash function applied to message X.
- V_A and S_A: principal A's public verification key and private signature key.
- $sS_A(X)$: principal A's digital signature on message X. The algorithm is assumed to be a 'signature with appendix', and the message is not recoverable from the signature.
- $A \rightarrow B : X$: principal A dispatches message X addressed to principal B.

2. DIGITAL VS HAND-WRITTEN SIGNATURE

The concept of digital signature, which was invented by Diffie and Hellman [4], forms an important basis for non-repudiation services. To understand how digital signatures may achieve the effect of hand-written signatures, it is worthwhile first observing the properties of hand-written signatures.

Hand-written signatures on paper documents have long been used as proof of authorship of, or in agreement with, the contents of a document since the signature holds the following properties[14].

- The signature is *hard to forge.* The process of generating a signature is a 'trained reflex', which is not subject to conscious muscular control. Thus signature imitation is difficult, especially at normal writing speed; this explains why bank clerks often ask for documents to be signed while they are watching.
- The signature is *easily verifiable.* The traditional verification technique is based on the visual inspection of a written signature. However, verification may become difficult for persons who are very inconsistent with their signatures.
- The signature is *not reusable.* The signature is part of the document. Other persons cannot cut and paste the signature to other documents.
- The signed document is *unalterable.* However, there is a limitation. The signature on a document can only guarantee the origin and integrity of the single sheet of the document bearing the signature. Thus, for a multi-sheet legal document, it has to be signed on each sheet to guarantee the origin and integrity of the whole document.
- The signature is a piece of *non-repudiation evidence.* The signature on a document is a physical object which can be presented for dispute resolution. For a newly generated signature to be acceptable, it should be compared with a notarized sample like the signature on a credit card. To enforce non-repudiation and simplify dispute resolution, the signature on a document can be witnessed by a (trusted) third party.

Digital signatures have advantages over hand-written signatures, and are being accepted as legal evidence within the same general guidelines as hand-written signatures. Several well-known digital signature schemes (e.g. RSA [11] and ElGamal algorithm [5]) exist. A digital signature on an electronic document is generated by using a public-key cryptography algorithm with the private signature key held by the signer. As the signature key is kept secret, others cannot forge the signature. But the signature can be verified by others with the corresponding public verification key, and the verification is more accurate than hand-written signatures. Since the digital signature is applied to the whole document, any change to the signed document will be detected, which is more convenient than hand-written signatures. Because a valid digital signature can only be generated by the signer holding the signature key, it can also serve as non-repudiation evidence.

In practice, however, a signature key may be compromised and a digital signature could be forged with a compromised key. Therefore, the compromised key needs to be revoked so that all signatures generated after revocation of the compromised key will be deemed invalid. On the other hand, digital signatures generated before revocation of the compromised key should remain valid. Otherwise, the signer who wants to repudiate signatures that he has generated may deliberately compromise his signature key and falsely claim those signatures as forged by somebody else.

There are several approaches to maintain the validity of digital signatures generated before revocation of the signature key. Each of them has its own feature and is applicable in a dedicated environment.

3. APPROACH A: TIME-STAMPING

A typical approach to maintain the validity of digital signatures as non-repudiation evidence relies on the existence of an *on-line* trusted time-stamping authority [1; 2; 3; 9; 13]. Each newly generated digital signature will be time-stamped by a time-stamping authority so that the trusted time of signature generation can be identified.

For instance, a user A's signature on a message M could be sent to a trusted time-stamping authority TS to certify that the signature was generated at the time of T_g [1].

1. $A \rightarrow TS: \quad sS_A(M)$
2. $TS \rightarrow A: \quad T_g, sS_{TS}(sS_A(M), T_g)$

TS simply adds a time stamp to A's signature without any verification of A's request. A may check TS's signature to see whether A's signature has been time-stamped correctly.

To check the validity of A's signature $sS_A(M)$, the verifier needs to go through the following steps.

1. The verifier should check TS's signature on $(sS_A(M), T_g)$.
2. The verifier should check the expiry date T_e of A's public key certificate C_A. If $T_e < T_g$, A's signature is invalid.
3. The verifier should check the *Certificate Revocation List* (*CRL*). If there exists a record showing that C_A was revoked at the time of T_r and $T_r < T_g$, A's signature is invalid.
4. The verifier should use A's public key certified in C_A to check A's signature $sS_A(M)$.

Only if all of the above checks are successful, will A's signature be regarded as valid non-repudiation evidence.

This approach is secure against disputes over the validity of digital signatures caused by (accidental or deliberate) compromise of signature keys, and can be employed in high value business transactions where security is an important requirement. However, it is likely to be much too expensive to support non-repudiation for large volume of low risk business transactions in electronic commerce.

4. APPROACH B: AUDITING

A less secure approach to maintain the validity of digital signatures as non-repudiation evidence is to combine the

[1] Here we use a simplified time-stamping process for a concise description. A more robust time-stamping service [6] could be used in commercial applications.

use of digital signatures with auditing processes, whereby the audit trails of transaction networks and of parties are used to establish whether a signature was generated at a specific time [10].

For example, a bank could establish audit logs to record the history of transactions and the related signatures supplied by its customers. Later on, if disputes arise regarding the validity of a digital signature, the bank may claim that, despite the subsequent revocation of a signature key by an individual, the audit logs show that the signature was generated at a time prior to the revocation, and hence the signature should be accepted as non-repudiation evidence.

In this approach, physical security of the audit trail is replacing the time-stamping service. Ultimately the success of this approach depends on how well adjudicators trust the integrity of the bank's auditing systems.

As an auditing system usually demands considerable storage resources, this approach is not applicable for ordinary users with limited storage capacity.

5. APPROACH C: ONE-WAY SEQUENTIAL LINK

The idea behind this approach is to link all digital signatures generated by a user in a way that any change to the order of the linked signatures or insertion of a new signature to the link will be detected.

Suppose C is a regular customer of a service provider S. If C is going to send signatures on messages $M_1, M_2, \cdots, M_i$ to S in a series of transactions, C can establish a one-way sequential link of his digital signatures $\sigma_1, \sigma_2, \cdots, \sigma_i$ as follows.

$$\begin{aligned} \sigma_1 &= sS_C(M_1) \\ \sigma_2 &= sS_C(M_2, H(\sigma_1)) \\ &\cdots \\ \sigma_i &= sS_C(M_i, H(\sigma_{i-1})) \end{aligned}$$

For $1 < j \leq i$, S will check whether σ_j is linked properly to σ_{j-1} before accepting C's jth signature. Such a link has the following properties [15].

- $\sigma_1, \sigma_2, \cdots, \sigma_i$ are *sequential.* That means, for $1 < j \leq i$, σ_j is generated later than σ_{j-1}.
- $\sigma_1, \sigma_2, \cdots, \sigma_i$ are *one-way linked.* That means, for $1 < j \leq i$, it is computationally infeasible to generate a valid signature σ' which is linked between σ_j and σ_{j-1}.

If C wants to revoke his signature key, C only needs to ask S to countersign C's latest digital signature.

$$\begin{aligned} &1.\ C \rightarrow S: && \sigma_i \\ &2.\ S \rightarrow C: && sS_S(\sigma_i) \end{aligned}$$

After receiving C's revocation request, S checks whether σ_i is C's latest digital signature. If so, S will confirm C's revocation request.

With S's countersignature, C can deny other signed messages which are generated with his revoked key but not in the countersigned link. Hence, S should not accept C's digital signatures generated with his revoked key once S has confirmed C's revocation request.

Like Approach B, there is no trusted third party involved in maintaining the validity of digital signatures. This approach is applicable in an environment satisfying the requirements below.

- Two parties have a regular business transaction relationship, thus a one-way sequential link of signatures can be established.
- The party providing the countersignature will not deny the validity of its countersignature. Otherwise, this approach has to be used in combination with Approach A, i.e. time-stamping the countersignature.

6. APPROACH D: TEMPORARY CERTIFICATE

In a low risk business transaction where digital signatures are used as non-repudiation evidence to settle disputes only within a certain period, the efficiency of the system can be significantly improved if digital signatures will remain valid within that period without being time-stamped.

The temporary certificate approach [16] can reach the goal. It defines two different types of signature keys.

- *Revocable signature keys* – the corresponding verification key certificates are issued by a certification authority (CA), and can be revoked as usual.
- *Irrevocable signature keys* – the corresponding verification key certificates are issued by users themselves and time-stamped by a time-stamping authority (TS). Such certificates cannot be revoked before their expiry.

The revocable signature key is used as a long-term master key to issue irrevocable verification key certificates while the irrevocable signature key is used as a temporary key to sign electronic documents. The digital signatures generated in such a way will remain valid until the corresponding irrevocable verification key certificates expire, thus can be exempted from being time-stamped by a time-stamping authority during on-line transactions.

6.1 Certificate Generation

Suppose S_A and V_A are user A's *revocable* signature and verification key pair, and C_A is A's *revocable* verification key certificate with expiry date T_e which is issued by a certification authority CA in the form of [2]

$$C_A = A, V_A, T_e, sS_{CA}(A, V_A, T_e)$$

Suppose S'_A and V'_A are user A's *irrevocable* signature and verification key pair. TS is an *off-line* time-stamping authority. User A can generate its *irrevocable* verification key certificate as below.

$$\begin{aligned} &1.\ A \rightarrow TS: && sS_A(V'_A, T'_e) \\ &2.\ TS \rightarrow A: && T'_g, sS_{TS}(sS_A(V'_A, T'_e), T'_g) \end{aligned}$$

[2] Other information that is required in the practical implementation is omitted here.

Thus, the irrevocable verification key certificate C'_A can be defined as

$$C'_A = V'_A, T'_e, T'_g,\ sS_A(V'_A, T'_e),\ sS_{TS}(sS_A(V'_A, T'_e), T'_g)$$

where T'_e is the expiry date of C'_A, and T'_g is the time that C'_A was generated. C'_A is valid only if S_A is valid at T'_g, and will remain valid until T'_e even if S_A becomes invalid after T'_g. It is important to note that the validity of S_A can only be checked *before* the revocable certificate C_A's expiry date T_e because the revocation information of expired certificates is not maintained in the *Certificate Revocation List* (*CRL*). Hence C'_A's expiry date T'_e should not be later than T_e [3], i.e. $T'_e \leq T_e$.

The aforementioned process of certificate generation by the use of a time-stamping authority is different from the one using a certification authority. *TS* need not check who is sending what to be time-stamped while *CA* has to authenticate who is sending the request for a public key certificate. Obviously, it is inefficient and undesirable to change a user's signature key frequently. In this approach, S_A will be used as user A's long-term master key for generating C'_A, and only needs to be changed after its expiry or revocation. S'_A will be used as user A's temporary key to sign messages, and can be changed periodically.

As C'_A does not contain any explicit reference to the name of A, another party B may ask *CA* to issue a revocable verification key certificate C_B in which $V_B = V_A$. Then the origin of a message signed with S'_A could be vague since a signature verifier may regard B as the originator of C'_A if C_B is used in the verification of C'_A. This concern will be much reduced if *CA* further verifies that B knows the private key S_B corresponding to V_B before *CA* signs C_B [9; 10; 12].

6.2 Signature Generation and Verification

With an irrevocable verification key certificate C'_A, user A can generate its signature on a message M using the corresponding irrevocable signature key S'_A. To form complete non-repudiation evidence, the certificate C'_A should be appended to the signature. Thus, A's signature on message M can be represented as

$$C'_A, sS'_A(M)$$

The expiry date of such a signature is defined the same as C'_A's expiry date T'_e.

To verify the above signature, the verifier should first check the validity of C'_A. The verification steps are the same as those outlined in Section 3. In addition, the verifier needs to check whether C'_A's expiry date T'_e meets the non-repudiation policy since T'_e will also decide the expiry date of signatures generated with S'_A. Once C'_A is checked to be valid, the verifier can use V'_A to check $sS'_A(M)$. If the result is positive, A's signature can be regarded as valid non-repudiation evidence for the settlement of possible disputes. Since the valid C'_A will remain valid until T'_e, the verifier may store it and directly use it later to check A's signatures appended with the same certificate C'_A.

6.3 Protection against Key Compromise

Although user A is not allowed to revoke its irrevocable signature keys, it becomes much easier to protect against the compromise of such keys than that of its revocable signature key. A need not keep its irrevocable signature keys until their expiry. Instead, A can destroy its used irrevocable signature keys and generate new irrevocable signature keys at any time it wishes, thus reducing the requirement of key management and the risk of key compromise.

Furthermore, by imposing additional restrictions on the irrevocable verification key certificate C'_A, user A can limit the loss even if its irrevocable signature key S'_A is compromised. The restrictions may include

- the types of transactions that S'_A can be applied to,
- the set of legitimate recipients of the signatures generated with S'_A,
- the maximum amount of a transaction which can be authorised by S'_A.

Thus, user A can generate an irrevocable verification key certificate C'_A as follows, which is limited to verify A's signatures on payment orders for recipients R_1, R_2 or R_3 with the maximum amount of \$1000.

$$\begin{aligned} limit &= (payment\ order, R_1, R_2, R_3, \$1000) \\ C'_A &= V'_A, T'_e, limit, T'_g,\ sS_A(V'_A, T'_e, limit), \\ &\quad sS_{TS}(sS_A(V'_A, T'_e, limit), T'_g) \end{aligned}$$

Suppose R_1 receives the following signed message from A in a transaction where C'_A is defined with the above limit.

$$C'_A, sS'_A(pay\ \$500\ to\ R_1 \cdots)$$

In addition to the process of signature verification described in Section 6.2, R_1 should also check whether C'_A is authorised for the verification of A's signatures on payment orders, whether R_1 is specified as a legitimate recipient in C'_A, whether the amount of this payment is within the limit of C'_A. Only if the signed message meets these restrictions, can R_1 accept it safely as non-repudiation evidence.

6.4 Dispute Resolution

As digital signatures generated in this approach have a limited valid period, all disputes related to a specific transaction that require user A's signature $sS'_A(M)$ as non-repudiation evidence should be brought to an arbitrator before this signature expires at time T'_e (referring to the arbitrator's clock). Users should be aware of the non-repudiation policy of business transactions that they will be involved in, and generate/accept digital signatures with appropriate expiry dates.

Suppose there is a dispute which requires $sS'_A(M)$ as a piece of non-repudiation evidence whose expiry date is T'_e. To prove the validity of this signature, the disputing party is required to submit the following messages to the arbitrator:

$$C_A, C'_A, M, sS'_A(M)$$

Suppose the arbitrator received the submission at time T_s. The arbitrator will make the following checks.

[3] A tighter restriction on C'_A's expiry date T'_e may be imposed because of administrative reasons.

1. The arbitrator checks that C_A was issued by CA with the expiry date of T_e.

2. The arbitrator checks that C'_A was signed by A and certified by TS, and C'_A satisfies $T'_g < T'_e \leq T_e$ [4].

3. The arbitrator checks that $sS'_A(M)$ was signed by A, and the signature does not breach the restrictions defined in C'_A.

4. The arbitrator checks that $T_s \leq T'_e$.

Only if all of the above checks are successful, will A's signature on message M be accepted as valid non-repudiation evidence.

This approach can significantly improve the efficiency of mass on-line transactions. However, as the generation of irrevocable signature/verification key pairs will incur extra computation overheads, it may not be feasible for applications to be implemented in computing resource limited environments.

7. COMPARISON

In the earlier sections, we have investigated four approaches to maintain the validity of digital signatures. Here we compare the performance of these approaches from several aspects. The comparison result is summarised in Table 1.

7.1 Security

In Approach A, each newly generated digital signature is time-stamped by a trusted time-stamping authority, thus the time of signature generation is guaranteed. If disputes arise, the time-stamp is sufficient to prove whether the signature was generated before revocation of the signature key. Hence, this approach has the highest security.

The security of Approach B relies on the integrity of the auditing system. As the auditing system is under the control of the party running the system, it is very hard to prove the integrity of the auditing system to a third party. Therefore, the security of this approach is low.

In Approach C, all digital signatures generated by a party are chained with a one-way sequential link. Once such a link is countersigned by another party, the validity of signatures in the link is non-repudiable. This is based on the assumption that the party providing the countersignature will not deny the validity of its countersignature.

Digital signatures generated in Approach D have a limited valid period. The party accepting such a signature as non-repudiation evidence should be aware of its valid period, and settle possible disputes before its expiry date. As a temporary certificate is irrevocable, the issuer of the certificate will bear the risk of compromise of the temporary signature key, though mechanisms are available to minimize the risk.

7.2 TTP's Involvement

In Approach A, each newly generated digital signature has to be time-stamped by a trusted time-stamping authority. Hence, an on-line trusted third party is needed. On-line time-stamping service may cause a substantial delay in a transaction since every time-stamping request incurs two rounds of communication with the TTP.

In Approach B, apart from the certification authority providing public key certificate service, no trusted third party is involved to maintain the validity of digital signatures. This is true in Approach C as well, provided that the party countersigning the one-way sequential link will not deny the validity of its countersignature.

In Approach D, time-stamping service will be invoked only when a temporary certificate is generated. Therefore, the trusted third party could be off-line. No extra communication with the TTP is needed during an on-line transaction.

7.3 Computation

In Approach A, the time-stamping authority needs to countersign each newly generated digital signature with a time stamp. The verifier needs to check signatures of both the original signer and the time-stamping authority. Hence, the computation overheads are high in terms of signature generation and verification.

Approach B has the lowest computation overheads since there are no extra computation costs in the generation and verification of a digital signature.

Approach C slightly increases the computation overheads for signature generation with an extra hash operation on the last signature to be linked. To check the validity of a signature, two adjacent signatures in the link will be used in verification as well.

The major computation overheads in Approach D is the generation of temporary certificates. The cost is high if a temporary certificate is used for only a few digital signatures. However, for mass on-line transactions, once a temporary certificate is generated and verified, there are no extra costs in the generation and verification of digital signatures with the temporary certificate.

7.4 Storage

Approach B needs to maintain an audit log which usually demands considerable storage resources. By contrast, the storage requirements in other approaches are low. They only need to store digital signatures plus some auxiliary data, e.g. the time-stamping authority's signatures in Approach A, and the temporary certificates in Approach D.

From the above analysis, we can derive a guideline on the use of these approaches in a given application.

- Approach A is recommended for high value business transactions.
- Approach B is applicable when one of the transacting parties is able to run secure and reliable auditing systems.

[4] If user A wants to deny the validity of C'_A, A is required to present evidence of revocation, e.g. the CRL, to prove that S_A was revoked before T'_g.

Table 1: A Comparison of Performance

Approach	A: Time-stamping	B: Auditing	C: One-way Sequential Link	D: Temporary Certificate
Security	high	low	medium	medium
TTP's Involvement	on-line	no	no	off-line
Computation	high	low	medium	high (single transaction) medium (mass transactions)
Storage	low	high	low	low

- Approach C is preferred when transaction parties have a regular business relationship so that a one-way sequential link can be established.
- Approach D is especially efficient for mass on-line transactions.

8. CONCLUSION

Like the role that hand-written signatures have been playing in physical business transactions, digital signatures serve as non-repudiation evidence with legal effect in electronic commerce. The security of digital signatures relies not only on the cryptographic strength of signature algorithms, but also on the management of signature keys. It is vital to maintain the validity of digital signatures in case signature keys are compromised.

We analysed four approaches for maintaining the validity of digital signatures. Depending on the requirements of a given application, an appropriate approach could be selected.

9. ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for helpful comments.

10. REFERENCES

[1] S. G. Akl. *Digital signatures: a tutorial survey.* Computer, 16(2):15–24, February 1983.

[2] K. S. Booth. *Authentication of signatures using public key encryption.* Communications of the ACM, 24(11):772–774, November 1981.

[3] R. DeMillo and M. Merritt. *Protocols for data security.* Computer, 16(2):39–50, February 1983.

[4] W. Diffie and M. E. Hellman. *New directions in cryptography.* IEEE Transactions on Information Theory, IT-22(6):644-654, November 1976.

[5] T. ElGamal. *A public-key cryptosystem and a signature scheme based on discrete logarithms.* IEEE Transactions on Information Theory, IT-31(4):469–472, July 1985.

[6] S. Haber and W. S. Stornetta. *How to time-stamp a digital document.* Journal of Cryptology, 3(2):99–111, 1991.

[7] L. Hollaar and A. Asay. *Legal recognition of digital signatures.* IEEE Micro, 16(3):44–45, June 1996.

[8] ISO/IEC 13888-3. *Information technology - Security techniques - Non-repudiation - Part 3: Mechanisms using asymmetric techniques.* ISO/IEC, 1997.

[9] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone. *Handbook of applied cryptography.* CRC Press, 1996.

[10] C. J. Mitchell. *Private communication.* May 1998.

[11] R. Rivest, A. Shamir and L. Adelman. *A method for obtaining digital signatures and public-key cryptosystems.* Communications of the ACM, 21(2):120–126, February 1978.

[12] M. Roe. *Cryptography and evidence.* PhD Thesis, University of Cambridge, 1997.

[13] B. Schneier. *Applied cryptography – Protocols, algorithms, and source code in C.* New York: John Wiley & Sons, 1996 (second edition).

[14] J. Zhou. *Non-repudiation.* PhD Thesis, University of London, December 1996.

[15] J. Zhou and K. Y. Lam. *A secure pay-per-view scheme for web-based video service.* Lecture Notes in Computer Science 1560, Proceedings of 1999 International Workshop on Practice and Theory in Public Key Cryptography, pages 315–326, Kamakura, Japan, March 1999.

[16] J. Zhou and K. Y. Lam. *Securing digital signatures for non-repudiation.* Computer Communications, 22(8):710–716, Elsevier, May 1999.