Limited Verifier Signature from Bilinear Pairings

Xiaofeng Chen[1], Fangguo Zhang[2], and Kwangjo Kim[1]

[1] International Research center for Information Security (IRIS)
Information and Communications University(ICU),
103-6 Munji-dong, Yusong-ku, Taejon, 305-714 KOREA
{crazymount,kkj}@icu.ac.kr

[2] Department of Electronics and Communication Engineering,
Institute of Information Security Technology,
Sun Yat-Sen University,
Guangzhou 510275, P.R.China
isdzhfg@zsu.edu.cn

Abstract. Motivated by the conflict between authenticity and privacy in the digital signature, the notion of limited verifier signature was introduced [1]. The signature can be verified by a limited verifier, who will try to preserve the privacy of the signer if the signer follows some specified rules. Also, the limited verifier can provide a proof to convince a judge that the signer has indeed generated the signature if he violated the predetermined rule. However, the judge cannot transfer this proof to convince any other party. Also, the limited verifier signature should be converted into an ordinary one for public verification if required.
In this paper, we first present the precise definition and clear security notions for (convertible) limited verifier signature, and then propose two efficient (convertible) limited verifier signature schemes from bilinear pairings. Our schemes were proved to achieve the desired security notions under the random oracle model.

Keywords: Undeniable signature, Designated verifier signature, Limited verifier signature, Bilinear pairings.

1 Introduction

Undeniable signature, introduced by Chaum and van Antwerpen [10], is a kind of digital signature which cannot be verified without interacting with the signer. It is useful in a case where the validity of a signature must not be verified universally. For example, a software vendor might embed his signature into his products and only allow the paying customers to verify the authentication of the products. If the vendor signed a message (product), he must provide some proofs to convince the customer of the fact. Also, these proofs must be non-transferable, *i.e.*, once a verifier (customer) is convinced that the vendor signed (or did not sign) the message, he cannot transfer these proofs to convince any third party. After the initial work of Chaum and van Antwerpen, several undeniable signature schemes were proposed [9,17,15,22]. Also, Boyar *et al.* [5] introduced the notion of convertible undeniable signature.

In some cases, it will be a disadvantage that the signature can be verified only with the cooperation of the signer. If the signer should be unavailable, or should refuse to cooperate, then the recipient cannot make use of the signature. This facilitates the concept of "designated confirmer signature" [8]. The designated confirmer can confirm the signature even without the cooperation of the signer when a dispute occurs.

In some applications, it is important for the signer to decide not only *when* but also *by whom* his signatures can be verified due to the blackmailing [13, 20] and mafia [12] attacks. For example, the voting center presents a proof to convince a certain voter that his vote was counted while without letting him to convince others (*e.g.*, a coercer) of his vote, which is important to design a receipt-free electronic voting scheme preventing vote buying and coercion. This is the motivation of the concept of "designated verifier signature" [21]. The designated verifier will trust the signer indeed signed a message with a proof of the signer. However, he cannot present the proof to convince any third party because he is fully capable of generating the same proof by himself.

Recently, motivated by privacy issues associated with dissemination of signed digital certificate, Steinfeld *et al.* [26] introduced the conception of "universal designated verifier signature", which can be viewed as an extended notion of designated verifier signature. Universal designated verifier signature allows any holder of the signature (not necessarily the signer) to designate the signature to any desired designated verifier. The verifier can be convinced that the signer indeed generated the signature, but cannot transfer the proof to convince any third party. For example, a user Alice is issued a signed certificate by the CA. When Alice wishes to send her certificate to a verifier Bob, she uses Bob's public key to transfer the CA's signature into a universal designated verifier signature to Bob. Bob can verifier the signature with CA's public key but is unable to use this designated signature to convince any third party that the certificate is issued by the CA, even if Bob is willing to reveal his secret key to the third party.

In some applications, it is also important for the recipient to decide *when* and *whom* the signer's signature should be verified. For example, a credit company will try his best to preserve the client's privacy in order to get his trust, provided that the client obeys the rules of the company. So, it is sufficient for the company only to be convinced the validity of the client's signature for his dishonorable message such as a bill. Furthermore, the company will preserve the client's privacy if he pays the bill in a certain time. However, if the client violated the rules, the company can provide a proof to convince a Judge of the client's treachery while the Judge cannot transfer the proof to convince any other third party.

It is obvious that undeniable signature and designated verifier signature are unsuitable for these situations. In the undeniable signatures, the signature can be verified only the cooperation of the signer. In the designated verifier signature, the designated verifier can never transfer the signature or the proof to convince any third party even he would like to reveal his secret key. This is because the designated verifier is fully capable to generate a “signature” himself which is indistinguishable from the real signature of the signer.

Araki *et al.* [1] introduced the concept of “limited verifier signature” to solve these problems. The limited verifier signature can only be verified by a limited verifier, who will try to preserve the signer’s privacy (especially some dishonorable message) unless the signer violated some rules. When a later dispute occurs, the limited verifier can convince a third party, usually a Judge, that the signer indeed generated a signature. We argue that the goal of the limited verifier is not to make the signature to be verified publicly, but force the signer to obey the rules. In some cases, the signer may not intentionally violate the rules and the limited verifier should give the signer some chances to correct his fault. Therefore, the Judge should not transfer this proof to convince any other party.

In some situations, the signer’s privacy is closely related to the recipient’s privacy. For example, a spy, Carol, has a certificate with a signature of the President, which can be verified by Carol herself. Also, Carol can provide a proof to prove her real identity to a third party in case of an emergency. However, the signature and the proof cannot be transferred by the third party to convince any other party in order to ensure Carol’s safety. Therefore, limited verifier signature can be used in any cases that the signer’s signature *should* be protected by the recipient.

Some official documents, which is treated as limited verifier signature, should be verified by everyone after a period of time if necessary. This is the motivation of “convertible limited verifier signatures”, also introduced by Araki *et al.* [1]. Convertible limited verifier signatures enable the limited verifier to convert the signature into an ordinary one for public verification.[1]

In the convertible limited verifier signature [1], the conversion of the signature requires the cooperation of the original signer, who must release some information. This might not be workable if the original signer is unwilling or inconvenient to cooperate. Furthermore, Zhang and Kim [28] proposed a universal forgery attack on this scheme. Wu *et al.* [24] proposed a convertible authenticated encryption scheme, which overcomes some disadvantages of Araki *et al.*’s scheme. However, if the recipient publishes the message and signature together, anyone can be convinced that the signer generated the signature. It does not satisfy the non-transferability. There seems no secure convertible limited verifier signature scheme to the best of our knowledge.

In this paper, we first present the precise definition and clear security notions for (convertible) limited verifier signature. Based on the power of different adversaries, we then propose two efficient (convertible) limited verifier signature schemes from bilinear pairings. Moreover, the conversion of the proposed limited verifier signature schemes does not need the cooperation of the original signer.

The rest of the paper is organized as follows: Some preliminary works are given in Section 2. In Section 3, the precise definition and notions of security for

[1] Convertible limited verifier signature is different from the notion of converted undeniable signature, where only the signer can release some information to convert his originally undeniable signature into an ordinary one.

limited verifier signature are presented. Our efficient limited verifier signature schemes from bilinear pairings are given in Section 4. In Section 5, the security and efficiency analysis of our schemes are given. Finally, conclusions will be made in Section 6.

2 Preliminary Works

In this section, we will briefly describe the basic definition and properties of bilinear pairings and gap Diffie-Hellman group.

2.1 Bilinear Pairings

Let G_1 be a cyclic additive group generated by P, whose order is a prime q, and G_2 be a cyclic multiplicative group of the same order q. Let a and b be elements of Z_q^*. We assume that the discrete logarithm problems (DLP) in both G_1 and G_2 are hard. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

1. Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$.
2. Non-degenerate: There exists P and $Q \in G_1$ such that $e(P, Q) \neq 1$.
3. Computable: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

2.2 Gap Diffie-Hellman Group

Let G_1 be a cyclic additive group generated by P, whose order is a prime q. Assume that the inversion and multiplication in G_1 can be computed efficiently. We introduce the following problems in G_1.

1. Discrete Logarithm Problem (DLP): Given two elements P and Q, to find an integer $n \in Z_q^*$, such that $Q = nP$ whenever such an integer exists.
2. Computation Diffie-Hellman Problem (CDHP): Given P, aP, bP for $a, b \in Z_q^*$, to compute abP.
3. Decision Diffie-Hellman Problem (DDHP): Given P, aP, bP, cP for $a, b, c \in Z_q^*$, to decide whether $c \equiv ab \bmod q$.
4. Bilinear Diffie-Hellman Problem (BDHP): Given P, aP, bP, cP for $a, b, c \in Z_q^*$, to compute $W = e(P, P)^{abc} \in G_2$.

We call G_1 a gap Diffie-Hellman group if DDHP can be solved in polynomial time but there is no polynomial time algorithm to solve CDHP with non-negligible probability. Such group can be found in supersingular elliptic curve or hyperelliptic curve over finite field, and the bilinear pairings can be derived from the Weil or Tate pairings. For more details, see [3,7,14,19].

3 Limited Verifier Signature (LVS) Scheme

3.1 Precise Definition

The limited verifier signature scheme involves a signer, a limited verifier (the designated recipient of the signature) and a certain third party (the Judge). It consists of six algorithms and a specific protocol.

- **System Parameters Generation:** on input a security parameter k, outputs the common system parameters SP.
- **Key Generation:** on input the common system parameters SP, outputs a secret/public key pair (sk, pk) for each user.
- **Limited Verifier Signing:** on input the key pair (sk_s, pk_s) of the signer, message m and the public key pk_v of the limited verifier, outputs a limited verifier signature σ.
- **Limited Verifier Verification:** on input the key pair (sk_v, pk_v) of the limited verifier, the public key pk_s of the signer, and a limited verifier signature σ, outputs a verification decision $b \in \{0, 1\}$. If $b = 1$, the verifier accepts the signature.
- **Confirmation Protocol:** a protocol between the limited verifier and a third party such as a Judge. The limited verifier provides a proof to convince the third party that a signature is indeed generated by a certain signer while the third party cannot transfer this proof to convince any other party even he can always eavesdrop the information between the signer and the limited verifier.
- **Convertible Limited Verifier Signing:** on input the secret key sk_v of the limited verifier, the public key pk_s of the signer, the message m and a limited verifier signature σ, outputs a convertible limited verifier signature σ'.
- **Public Verification:** on input the public key pk_v of the limited verifier, the public key pk_s of the signer, the message m and a convertible limited verifier signature σ', outputs a verification decision $b \in \{0, 1\}$. If $b = 1$, anyone can be convinced that the signer indeed generated the signature σ' for the message m.

3.2 Adversarial Model

The only assumption in the LVS scheme is that the limited verifier will try his best to preserve the signer's privacy unless the signer violates some rules or an emergency occurs. But the limited verifier should never be able to forge a signature of the signer to frame him. Therefore, "unforgeability" is the basic cryptographic requirement of LVS scheme. There are three kind of forgers in LVS scheme: "limited verifier", "outsiders" and "colluders". In the proposed schemes, we only consider the strongest adversarial model for unforgeability: an adversary can collude with the limited verifier.

On the other hand, an adversary should not be able to forge a proof to convince any other party that the signer indeed generated a signature. "Non-transferability" is another basic cryptographic requirement in LVS scheme. Similarly, we think the adversary can collude with the Judge. Also, we suppose the adversary can also eavesdrop all the information between the limited verifier and signer. This is the strongest adversarial model for non-transferability. In this case, the adversary should not collude with the limited verifier anymore because the limited verifier wants to convince only the Judge of the fact.

3.3 Security Requirements

3.3.1 Unforgeability

Similar to universal designated verifier signature scheme, there are two type of unforgeability in LVS scheme. The first is identical to the usual existential unforgeability notion under the chosen message attack. This prevents an adversary to frame the signer by "generating" a signature of the signer. The second requires that it is difficult for an adversary (usually the limited verifier) to forge a proof, which can be used to convince a third party (usually a Judge) that the signer generated a signature for a message. Because LVS scheme should be converted into an ordinary one for public verification when necessary, the limited verifier only forges a proof to frame a signer is meaningless even he can.[2] In this sense, we only consider the first unforgeability in LVS scheme.

Definition 1. *A LVS scheme is said to secure against an existential forgery for adaptive chosen message attack if no polynomial bounded adversary $\mathcal{A}$ win the following game with a non-negligible advantage.*

1. The challenger $\mathcal{C}$ runs the System Parameter Generation algorithm with a security parameter k and sends the system parameters SP to the adversary $\mathcal{A}$.
2. The limited verifier $\mathcal{V}$ runs the Key Generation algorithm to generate his key pair $(pk_{\mathcal{V}}, sk_{\mathcal{V}})$ and publishes $pk_{\mathcal{V}}$. Also, the adversary $\mathcal{A}$ is allowed to access the secret key $sk_{\mathcal{V}}$.
3. The adversary $\mathcal{A}$ performs a polynomial bounded number of queries to challenger $\mathcal{C}$.
4. Finally, the adversary $\mathcal{A}$ outputs a valid message-signature pair (m, s). We said that $\mathcal{A}$ wins the game if m is never queried by $\mathcal{A}$ in step 3.

3.3.2 Non-transferability

The property of non-transferability in LVS scheme can be automatically reduced from universal designated verifier signature scheme.

[2] This is different from universal designated verifier signature scheme, where it is enough for the third party to be convinced by such a proof.

Definition 2. *Let $P(\mathcal{V}, \mathcal{J})$ be a protocol between the limited verifier $\mathcal{V}$ and a Judge $\mathcal{J}$. The outputs of $P(\mathcal{V}, \mathcal{J})$ is a proof $\mathcal{P}$ presented by $\mathcal{V}$ which can convince $\mathcal{J}$ the truth of a statement Θ. We said the proof is non-transferable if $\mathcal{J}$ is fully able to generate an indistinguishable proof $\mathcal{P}'$. In this case, no one can be convinced of the truth of a statement Θ even if $\mathcal{J}$ would like to reveal his secret key $sk_{\mathcal{J}}$.*

4 Our Proposed LVS Schemes from Bilinear Pairings

In this section, we propose two efficient LVS schemes from bilinear pairings based on the power of different adversaries. Furthermore, we present a general construction of LVS scheme.

4.1 Our Scheme (I)

- **System Parameters Generation:** Let G_1 be a gap Diffie-Hellman group generated by P, whose order is a prime q, and G_2 be a cyclic multiplicative group of the same order q. A bilinear pairing is a map $e : G_1 \times G_1 \to G_2$. Define two cryptographic hash functions $H_1 : \{0,1\}^l \to G_1$, $H_2 : G_2 \to Z_q$ and $h : \{0,1\}^l \times G_2 \to Z_q$, where l denotes a bound on the message bit-length. The system parameters are $SP = \{G_1, G_2, e, q, P, H_1, H_2, h, l\}$.
- **Key Generation:** The user U randomly chooses $r_U \in_R Z_q^*$ as the secret key and computes the public key $r_U P$.
- **Limited Verifier Signing:** Suppose Alice wants to sign the message m for Bob. She does as follows:
  - Randomly choose a point $Q \in_R G_1$ and compute $c = e(Q, r_A P)$.
  - Compute $s = Q - r_A k H_1(m)$, where $k = h(m, e(Q, P))$.
  - Compute $t = H_2(e(r_A Q, r_B P))^{-1} s$.

  The signature for message m is the pair $S = (c, k, t)$.
- **Limited Verifier Verification:** On receiving the limited verifier signature S, Bob computes:
  - $s = H_2(c^{r_B})t$.
  - $d = e(s, P)e(H_1(m), r_A P)^k$.
  - Output "accept" if and only if $k = h(m, d)$.
- **Confirmation Protocol:** When Alice does not obey some rules, **only** Bob can provide a proof to convince a Judge that Alice indeed signed a message with a confirmation protocol.[3] However, the Judge cannot transfer this proof to convince any other party.
  - Bob computes $a = e(s, r_J P)$.
  - Bob sends (a, d) and the message m to Judge.
  - Let $k = h(m, d)$. Judge computes $l = (d^{r_J}/a)^{k^{-1}}$ and accepts the proof if and only if $l = e(H_1(m), r_A P)^{r_J}$.

[3] Note that any adversary cannot compute s without the information of r_B even he can eavesdrop all the information between Bob and Alice and Judge unless he can solve CDHP in G_2.

Actually, $l = e(r_A H_1(m), r_J P)$, which is a universal designated verifier signature for the message m [26]. Therefore, the Judge will be convinced that Alice signed the message while he cannot transfer this proof to convince any other party.
We explain this in more details. The Judge can simulate Bob to generate an indistinguishable pair (a, d) for any message m as follows:
  - He randomly chooses an element $d \in G_2$, and computes $k = h(m, d)$.
  - He computes $l = (e(H_1(m), r_A P)^{r_J}$.
  - He computes $a = d^{r_J}/l^k$, and outputs (a, d).
- **Convertible Limited Verifier Signing:** In some situations, the limited verifier signature should be converted into an ordinary signature for public verification. In Araiki *et al.*'s scheme, the conversion of the signature requires the cooperation of the original signer. However, it might be unworkable if the signer is unwilling or inconvenient to cooperate. In our scheme, both the signer and limited verifier can convert a limited verifier signature into an ordinary one:
  - Alice (or Bob) publishes the message m and the pair (k, s).
- **Public Verification:** Anyone can be convinced that the signer indeed generated the the signature for the message m:
  - The verifier computes $d = e(s, P)e(H_1(m), r_A P)^k$.
  - Output "accept" if and only if $k = h(m, d)$.

4.2 Our Scheme (II)

In some situations, the message m, *e.g.*, an official document, also should be confidential. Signcryption, firstly introduced by Zheng [29], provides simultaneously both message confidentiality and unforgeablity at a lower computational and communication overhead compare to *Encrypt-and-Sign* method. Signcryption protocol usually should satisfy the property of public verifiability, *i.e.*, if a recipient Bob can recover the signer Alice's signature, anyone can verify the signature based on a given signature scheme.[4] However, in the limited verifier signcryption algorithm, the signature can only be verified by himself even after the recipient recovered the message-signature pair. Also, it should satisfy the property of non-transferability, *i.e.*, the recipient can provide a proof to convince a third party that the signer generated a signature while the third party cannot transfer the proof to convince any other party. Therefore, the signature on the message must be invisible in the ciphertext because the adversary can eavesdrop all the information between the recipient and others. If the adversary knows the signature, the message and the proof, he can convince any party that the signer indeed generated the signature. We will explain this later in more details.

We construct limited verifier signature protocol based on "*Sign-then-Encrypt*" methodology [6]. Without loss of generality, let Alice is the signer and Bob is the recipient (limited verifier).

[4] Shin *et al.* [25] defined this "**SIG-verifiability**".

- **System Parameters Generation:** Let G_1 be a gap Diffie-Hellman group generated by P, whose order is a prime q, and G_2 be a cyclic multiplicative group of the same order q. A bilinear pairing is a map $e : G_1 \times G_1 \to G_2$. Define five cryptographic hash functions $H_1 : \{0,1\}^l \to G_1$, $H_2 : G_2 \to Z_q$, $H_3 : Z_q \to G_1$, $H_4 : G_1 \to \{0,1\}^l$, and $h : \{0,1\}^l \times Z_q \to Z_q$, where l denotes a bound on the message bit-length. The system parameters are $SP = \{G_1, G_2, e, q, P, H_1, H_2, H_3, H_4, h, l\}$.
- **Key Generation:** The user U randomly chooses $r_U \in_R Z_q^*$ as his secret key and computes the public key $r_U P$.
- **Limited Verifier Signing (Signcryption):** Suppose Alice wants to sign the message m for Bob. She does as follows:
  - Randomly choose an integer $c \in_R Z_q$ and compute $S = cr_A H_1(m)$.
  - Compute $k = h(m, c)$.
  - Compute $U = H_2(e(r_A P, r_B P)^k) \oplus c$.
  - Compute $V = H_3(c) \oplus S$.
  - Compute $W = H_4(S) \oplus m$.

  The signature for message m is the ciphertext $C = (kP, U, V, W)$.
- **Limited Verifier Verification (Unsigncryption):** On receiving the limited verifier signature C, Bob computes:
  - $c = U \oplus H_2(e(r_A P, kP)^{r_B})$.
  - $S = V \oplus H_3(c)$.
  - $m = W \oplus H_4(S)$.
  - Verify that $kP = h(m, c)P$. If not, output "reject".
  - Output "accept" if and only if $e(S, P)^{c^{-1}} = e(H_1(m), r_A P)$.
- **Confirmation Protocol:** Bob can convince a Judge that Alice indeed signed a message with the following confirmation protocol.[5] From the property of universal designated verifier signature, the Judge cannot transfer this proof to convince any other party.
  - Bob computes $a = e(S, r_J P)^{c^{-1}}$.
  - Bob sends a and the message m to Judge.
  - Judge outputs "accept" if and only if $a = e(H_1(m), r_A P)^{r_J}$.

  Note that the Judge is fully able to generate the indistinguishable proof $e(H_1(m), r_A P)^{r_J}$. Therefore, he cannot use this proof to convince any other party.
- **Convertible Limited Verifier Signing:** Both the signer and limited verifier can convert a limited verifier signature into an ordinary one:
  - Alice (or Bob) publishes the message m and the signature $T = c^{-1}S$.
- **Public Verification:** Anyone can be convinced that the signer indeed generated the the signature for the message m:
  - Outputs "accept" if and only if $e(T, P) = e(H_1(m), r_A P)$.

[5] Any adversary cannot compute c to recover $r_A H_1(m)$ without the information of r_B even he can eavesdrop all the information between Bob and Alice and Judge unless he can solve BDHP in G_1.

4.3 Generalization

Our Scheme (II) can be extended to design a general construction of (convertible) limited verifier signature.[6] The signer generates a universal designated verifier signature s on the message m and then encrypts the concatenation of m and s with the limited verifier's public key PK_v by using a semantically secure probabilistic encryption algorithm ENC. The ciphertext $C = \text{ENC}_{PK_v}(m||s)$ is the limited verifier signature for the message m.

The limited verifier decrypts the ciphertext with his secret key and can then designate it to any Judge as in the universal designated verifier signature scheme. For public verification, the limited verifier (or the signer) publishes m and s, and anyone can be convinced that the signer generated the the signature s for the message m.

Recently, Steinfeld *et al.* [27] extended standard Schnorr/RSA signatures into universal designated verifier signatures. Therefore, we can use the general construction to design (convertible) limited verifier signature scheme without pairings.

5 Analysis of the Proposed Schemes

5.1 Security

Lemma 1. *Under the strongest adversarial model, if an adversary $\mathcal{A}$ in scheme (I) can forge a valid signature (m, c, k, t) with the advantage ε within time T, then he can forge the valid signature (m, c, k, s) with the same advantage ε within time T, and vice versa.*

Proof. Suppose the adversary $\mathcal{A}$ can forge a valid signature (m, c, k, t) with the advantage ε within time T, then he can compute $s = H_2(c^{r_B})t$ since he can access the secret key r_B of the limited verifier Bob, *i.e.*, he can forge the valid signature (m, c, k, s) with the same advantage ε within time T, and vice versa. □

Theorem 1. *In the random oracle, if there exists an adversary $\mathcal{A}$ that can succeed in an existential forgery against the proposed LVS scheme (I) with an advantage ϵ within a time T and when performing n queries on signature oracle and hash oracles h and H_1, then there exists an algorithm $\mathcal{C}$ can solve the CDHP in G_1 with an advantage $\epsilon' \geq \epsilon/n$ within a time $T' \leq 84480nT/\varepsilon$.*

Proof. Let P is a generator of G_1, the following algorithm $\mathcal{C}$ can be used to compute abP for a randomly given triple (P, aP, bP). Define the public key of the signer is aP.

Randomly choose $x_i \in Z_q, y_i \in Z_q$ and $k_i \in Z_q$ for $i = 1, 2, \cdots, n$. Denote by m_i the (partial) input of the i-th query to h and H_1. We show how the queries of $\mathcal{A}$ can be simulated.

[6] An anonymous reviewer suggested the general approach.

Choose an index $r \in \{1, 2, \cdots, n\}$ randomly. Define

$$c_i = e(x_i P, aP)$$

$$h(m_i, e(x_i P, P)) = k_i$$

$$H_1(m_i) = \begin{cases} bP, & \text{if } i = r \\ y_i P, & \text{if } i \neq r \end{cases}$$

$$s_i = \begin{cases} \text{“Fail”}, & \text{if } i = r \\ x_i P - y_i k_i (aP), & \text{if } i \neq r \end{cases}$$

Suppose the output of $\mathcal{A}$ be (m, c, k, s). If $m = m_r$ and (m, c, k, s) is valid, output (m, c, k, s). Otherwise, output “Fail” and halt.

By replays of with the same random tape but different choices of oracle h, as done in the Forking Lemma [23], we can obtain two valid signatures (m, c, k, s) and (m, c, k', s') with respect to different hash oracles h and h'. Note that $s = Q - akH_1(m)$ and $s' = Q - ak'H_1(m)$, we have $abP = (s - s')/(k' - k)$.

Because h and H_1 are the random oracles, the adversary $\mathcal{A}$ cannot distinguish the simulation of algorithm $\mathcal{C}$ from the real signer. Also, since r is independently and randomly chosen, the success of probability of $\mathcal{C}$ is ε/n. The total running time T' of algorithm $\mathcal{C}$ is equal to the running time of the Forking Lemma [23] which is bound by $84480nT/\varepsilon$. □

In our scheme (II), the proposed signcryption algorithm is based on “Sign-then-Encrypt” methodology, which can be viewed as the standard version of Boyen’s ID-based signcryption algorithm [6]. Therefore, we have

Theorem 2. *In the random oracle, the proposed signcryption algorithm in our scheme (II) is semantically secure against adaptively chosen ciphertext attacks and unforgeable secure against adaptively chosen message attacks based on the assumption BDHP is intractable.*

Theorem 3. *Our proposed LVS schemes are both satisfy the property of non-transferability based on the assumption of BDHP is intractable.*

Proof. Firstly, the third party can be convinced by the proof that the signer indeed generate a signature. From the result of [26], we know that it is impossible for the limited verifier to forge a universal designated verifier signature to cheat the Judge.

Secondly, the Judge cannot transfer the proof to convince any other party. In scheme (I), the proof is the pair (a, d). We have proved that the Judge is fully able to generate an indistinguishable pair. In scheme (II), the proof is just a universal designated verifier signature. Therefore, the non-transferability of both schemes is obvious. □

5.2 Efficiency

We compare the efficiency of our schemes with that of Araki *et al.*'s scheme. In Table 1, we denote $\mathcal{P}$ the pairings operation, $\mathcal{M}$ the point scalar multiplication in G_1, $\mathcal{E}$ exponentiation in G_2 and $\mathcal{R}$ reversion in Z_q. We ignore other operations such as hash in all schemes.

Table 1. Comparison of computation cost

	Araki's scheme	Our scheme (I)	Our scheme (II)
Signing	$1\mathcal{E}+2\mathcal{R}$	$2\mathcal{P}+3\mathcal{M}+1\mathcal{R}$	$1\mathcal{P}+2M$
Verification	$2\mathcal{E}+1\mathcal{R}$	$2\mathcal{P}+1\mathcal{M}+1\mathcal{E}$	$2\mathcal{P}+1\mathcal{E}+1\mathcal{R}$
Confirmation	$11\mathcal{E}+1\mathcal{R}$	$2\mathcal{P}+2\mathcal{E}+2\mathcal{R}$	$2\mathcal{P}+2\mathcal{E}$
Denial	$24\mathcal{E}$	/	/
Convertion	$3\mathcal{E}+1\mathcal{R}$	$2\mathcal{P}+1\mathcal{E}$	$2\mathcal{P}$

In Araki *et al.*'s scheme, both of the confirmation and denial protocol need rounds of interactive communication. However, the confirmation protocol in our schemes is performed in a non-interactive manner. Moreover, our scheme does not require the denial protocol. The Judge can be convinced by a proof that the signer indeed generated a signature. Because the proposed scheme can be converted into an ordinary one for public verification when necessary, the signer cannot repudiate his signature.

Suppose the length of a point in G_1 is $|q|$, and the length of an element of G_2 and the message m is $|p|$. Table 2 presents the comparison of communication cost between Araki *et al.*'s scheme and ours.

Table 2. Comparison of communication cost

	Araki's scheme	Our scheme (I)	Our scheme (II)
Signing	$1\|p\|+1\|q\|$	$2\|p\|+2\|q\|$	$1\|p\|+3\|q\|$
Confirmation	$3\|p\|+3\|q\|$	$3\|p\|$	$2\|p\|$
Denial	$6\|p\|+6\|q\|$	/	/
Convertion	$2\|p\|+1\|q\|$	$1\|p\|+2\|q\|$	$1\|p\|+1\|q\|$

6 Conclusions

The ordinary digital signature provides the functions of integration, authentication, and non-repudiation for the signed message. Anyone can verify the signature with the signer's public key. However, it is unnecessary for anyone to be convinced the validity of the signature in some situations. It is sufficient for a designated recipient, who will try to preserve the signer's privacy if the signer follow some specified rules, to verify the signature. Limited verifier signature was introduced to solve this problem. If the signer violated the rules, the designated recipient (namely, limited verifier) can provide a proof to convince a judge that the signer indeed generated the signature for the message. Also, the limited verifier can also convert the signature into an ordinary one for public verification when necessary. In this paper, we firstly present the precise definition and clear security notions for (convertible) limited verifier signature, and then propose two new (convertible) limited verifier signature schemes from bilinear pairings. Moreover, we proved that our schemes achieved the desired security notions in the random oracle.

In our schemes, the confirmation protocol does not need the interactive communication and the conversion does not need the cooperation of the original signer. Therefore, they are much efficient than previous scheme.

Acknowledgement. The authors are grateful to the anonymous reviewers for their valuable suggestions and comments to this paper. This work was supported by a grant No.R12-2003-004-01004-0 from the Ministry of Science and Technology, Korea.

References

1. S. Araki, S. Uehara, and K. Imamura, *The limited verifier signature and its application*, IEICE Trans. Fundamentals, vol.E82-A, No.1, pp. 63-68, 1999.
2. P.S.L.M. Barreto, H.Y. Kim, B.Lynn, and M.Scott, *Efficient algorithms for pairings-based cryptosystems*, Advances in Cryptology-Crypto 2002, LNCS 2442, pp.354-368, Springer-Verlag, 2002.
3. D. Boneh and M. Franklin, *Identity-based encryption from the Weil pairings*, Advances in Cryptology-Crypto 2001, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
4. D. Boneh, B. Lynn, and H. Shacham, *Short signatures from the Weil pairings*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.514-532, Springer-Verlag, 2001.
5. D. Boyar, D. Chaum, and D. Damgård, *Convertible undeniable signatures*, Advances in Cryptology-Crypto 1990, LNCS 537, pp.183-195, Springer-Verlag, 1991.
6. X. Boyen, *Multipurpose identity-based signcryption: a Swiss army knife for identity-based cryptography*, Advances in Cryptology-Crypto 2003, LNCS 2729, pp.382-398, Springer-Verlag, 2003.
7. J.C. Cha and J.H. Cheon, *An identity-based signature from gap Diffie-Hellman groups*, Public Key Cryptography-PKC 2003, LNCS 2567, pp.18-30, Springer-Verlag, 2003.
8. D. Chaum, *Designated confirmer signatures*, Advances in Cryptology-Eurocrypt 1994, LNCS 950, pp.86-91, Springer-Verlag, 1994.
9. D. Chaum, *Zero-knowledge undeniable signatures*, Advances in Cryptology-Eurocrypt 1990, LNCS 473, pp.458-464, Springer-Verlag, 1991.
10. D. Chaum and H. van Antwerpen, *Undeniable signatures*, Advances in Cryptology-Crypto 1989, LNCS 435, pp.212-216, Springer-Verlag, 1989.
11. D. Chaum and T.P. Pedersen, *Wallet databases with observers*, Advances in Cryptology-Crypto 1992, LNCS 740, pp.89-105, Springer-Verlag, 1993.
12. Y. Desmedt, C. Goutier, and S. Bengio, *Special uses and abuses of the Fiat-Shamir passport protocol*, Advances in Cryptology-Crypto 1987, LNCS 293, pp.21-39, Springer-Verlag, 1988.

13. Y. Desmedt and M. Yung, *Weaknesses of undenaiable signature schemes*, Advances in Cryptology-Eurpcrypt 1991, LNCS 547, pp.205-220, Springer-Verlag, 1992.
14. S. D. Galbraith, K. Harrison, and D. Soldera, *Implementing the Tate pairings*, ANTS 2002, LNCS 2369, pp.324-337, Springer-Verlag, 2002.
15. S. Galbraith, W. Mao, and K. G. Paterson, *RSA-based undeniable signatures for general moduli*, Advances in CT-RSA 2002, LNCS 2271, pp.200-217, Springer-Verlag, 2002.
16. S. Galbraith and W. Mao, *Invisibility and anonymity of undeniable and confirmer signatures*, Advances in CT-RSA 2003, LNCS 2612, pp.80-97, Springer-Verlag, 2003.
17. S. Gennaro, H. Krawczyk, and T. Rabin, *RSA-based undeniable signatures*, Advances in Cryptology-Crypto 1997, LNCS 1294, pp.132-149, Springer-Verlag, 1997.
18. C. Gentry and A. Silverberg, *Hierarchical ID-Based Cryptography*, Advances in Cryptology-Asiacrypt 2002, LNCS 2501, pp.548–566, Springer-Verlag, 2002.
19. F. Hess, *Efficient identity based signature schemes based on pairingss*, Proc. 9th Workshop on Selected Areas in Cryptography-SAC 2002, LNCS 2595, Springer-Verlag, pp.310-324, 2002.
20. M. Jakobsson, *Blackmailing using undeniable signatures*, Advances in Cryptology-Eurocrypt 1994, LNCS 950, pp.425-427, Springer-Verlag, 1994.
21. M. Jakobsson, K. Sako, and R. Impagliazzo, *Designated verifier proofs and their applications*, Advances in Cryptology-Eurocrypt 1996, LNCS 1070, pp.143-154, Springer-Verlag, 1996.
22. B. Libert and J. Quisquater, *ID-based undeniable signatures*, Advances in CT-RSA 2004, LNCS 2694, pp.112-125, Springer-Verlag, 2004.
23. D. Pointcheval and J. Stern, *Security arguments for digital signatures and blind signatures*, Journal of Cryptography, Vol.13, No.3, pp.361-396, Springer-Verlag, 2000.
24. T. Wu and C. Hsu, *Convertible authenticated encryption scheme*, The Journal of Systems and Software, Vol.62, No.3, pp.205-209, 2002.
25. J. Shin, K. Lee, and K. Shim, *New DSA-verifiable signcryption schemes*, ICISC 2002, LNCS 2587, pp.35-47, Springer-Verlag, 2003.
26. R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk, *Universal designated-verifier signatures*, Advances in Cryptology-Asiacrypt 2003, LNCS 2894, pp.523-542, Springer-Verlag, 2003.
27. R. Steinfeld, H. Wang, and J. Pieprzyk, *Efficient extension of standard Schnorr/RSA signatures into universal designated-verifier signatures*, Public Key Cryptography-PKC 2004, LNCS 2947, pp.86-100, Springer-Verlag, 2004.
28. F. Zhang and K. Kim, *A universal forgery on Araki et al.'s convertible limited verifier signature scheme*, IEICE Trans. Fundamentals, vol.E86-A, No.2, pp. 515-516, 2003.
29. Y. Zheng, *Digital signcryption or how to achieve cost (signature & encryption) $<<$ cost (signature)+ cost (encryption)*, Advances in Cryptology-Crypto 1997, LNCS 1294, pp.165-179, Springer-Verlag, 1997.