Transitive Signatures Based on Bilinear Maps[⋆]

Changshe Ma, Kefei Chen, Shengli Liu and Dong Zheng

Department of Computer Science and Engineering,
Shanghai Jiaotong University, China

Abstract. The notion of transitive signature, firstly introduced by Micali and Rivest, is a way to digitally sign the vertices and edges of a dynamically growing, transitively closed graph. All the previous proposed transitive signature schemes were constructed from discrete logarithm, factoring, or RSA assumption. In this paper, we introduce two alternative realizations of transitive signature based on bilinear maps. The proposed transitive signature schemes possess the following properties: (i) they are provably secure against adaptive chosen-message attacks in the random oracle model; (ii) there are no need for node certificates in our transitive signature schemes, so the signature algebra is compact; (iii) if using Weil pairing, our signature schemes are more efficient than all previous proposed schemes.

1 Introduction

1.1 Motivations

In a Public Key Infrasture (for short PKI) [8] system of depth n, there are many CAs, and each user is given a chain of n certificates. Suppose two users A and B want to carry out an authenticated and private communication, but they are not in the same domain and authenticated by different CAs. So A must find a pass of certificates from him to B. The length of the pass is linear in the number of CA nodes from A to B. Can this pass be compressed, or in another word, can the length of the pass be shortened to the length of one signature? As another example, in distributed networks [15], an object T could never meet with a subject S, therefore S may not hold any prior evaluation of trustworthiness of T. To get permit to access S, T should be somewhat trusted by S. How can S evaluate the trustworthiness of T accurately and efficiently?

A transitive signature scheme can help us to solve these problems perfectly. The concept of transitive signature, first introduced by Micali and Rivest in [12] and subsequently formalized by Bellare and Neven in [4], is a way to build an authenticated dynamically growing transitively closed graph G, edge by edge, such that:

- Transitivity. Given the signatures of two edges (i, j) and (j, k) of the graph G, it is computationally feasible for anyone to derive a valid digital signature of edge (i, k).

[⋆] This work was partially supported under NFSC 60273049,60303026 and 60473020

Ma C., Chen K., Liu S. and Zheng D. (2005).
Transitive Signatures Based on Bilinear Maps.
In *Proceedings of the 3rd International Workshop on Security in Information Systems*, pages 48-56
DOI: 10.5220/0002542700480056

- Unforgebility. It is computationally infeasible for any adversaries to forge a valid digital signature of any edges that is not in the transitive closure $\tilde{G}$ of the graph G, even if the adversary can request the legitimate signer to digitally sign any number of vertices and edges of his choice in an adaptive fashion.

1.2 Our Contributions

All the previous proposed transitive signature schemes were constructed from discrete logarithm, factoring, or RSA assumption. It is standard practice in cryptography to seek new and alternative realizations of primitives of potential interest, both to provide firmer theoretical foundations for the existence of the primitive by basing it on alternative conjectured hard problems and to obtain performance improvements. In this paper, we provide two novel realizations of the transitive signature scheme BMTS-1 and BMTS-2 from bilinear maps to accomplish both of these objectives. These signature schemes work in any groups where the Decision Diffie-Hellman problem (DDH) is easy, but the Computational Diffie-Hellman problem (CDH) is hard. Such groups are referred as gap groups [10]. Our transitive signature schemes BMTS-1 and BMTS-2 possess the following properties: Firstly, our transitive signature schemes are constructed without node certificates, so the signature algebra is compact. Secondly, our schemes are provably secure under adaptive chosen-message attack in the random oracle model. Furthermore, if using Weil pairing over supersingular Elliptic curves, such as the signature in [3], our transitive signature schemes are more efficient than all previous proposed schemes (as showed by Figure 1 in section 5).

1.3 Related Works

The transitive signature scheme MTRS presented in [12] is provably secure under adaptive chosen-message attack assuming that the discrete logarithm problem is hard over prime order group $\mathcal{Z}_p^*$. In [4] Bellare and Neven proposed four transitive signature schemes which are all provable security. Johnson et al introduced the notation of homomorphic signature [9] and described several schemes that are homomorphic with respect to useful binary operations. Context Extraction Signatures, introduced early by [13], falls in the framework of [9]. A signature scheme that is homomorphic with respect to the prefix operation is presented by Chari, Rabin and Rivest [7].

Recently, the bilinear maps have initiated some completely new fields in cryptography, making it possible to realize some cryptographic primitives unknown or impractical [1–3, 11]. In [2], Boneh introduced a new kind of digital signature named aggregate signature which support aggregation.

The rest of the paper is organized as follows. In § 2, we give some notations and definitions . In § 3, we describe the model of transitive signature schemes. In § 4, we present two transitive signature schemes. In § 5, we provide security proofs and efficiency analysis. In § 6, we draw a conclusion.

2 Notations and Definitions

In this section, we give some notations and definitions that will be used in the paper.

Throughout this paper, we let $a \longleftarrow A$ denote a is selected from the set A, $a \xleftarrow{R} A$ denote a is selected randomly and uniformly from the set A. Let $a \longleftarrow b$ (b is an element) denote a is assigned with the value of b. Let $a \longleftarrow \mathcal{A}(a_1, a_2, \cdots)$ denote a is assigned with the value of the output of algorithm $\mathcal{A}$ on inputs $a_1, a_2, \cdots$. Let G_1, G_2 and G_T be three cyclic groups of prime order p, g_1 and g_2 be the generators of G_1 and G_2 separately, ψ be a computable isomorphism from G_2 to G_1 with $\psi(g_2) = g_1$.

Definition 2.1 A bilinear map is a map $e : G_1 \times G_2 \longrightarrow G_T$ with the following properties:

Bilinear: for all $u \in G_1, v \in G_2$ and $a, b \in \mathcal{Z}_p, e(u^a, v^b) = e(u, v)^{ab}$;

Non-degenerate: $e(g_1, g_2) \neq 1$.

Computational Co-Diffie-Hellman (co-CDH). Given $g_2, g_2^a \in G_2$ and $h \in G_1$ compute $h^a \in G_1$. An algorithm $\mathcal{A}$ has advantage ε in solving co-CDH problem if

$$\texttt{Adv}\ \text{co-CDH}_{\mathcal{A}}^{G_1,G_2} = \texttt{Pr}[\mathcal{A}(g_2, g_2^a, h) = h^a] \geq \varepsilon.$$

The probability is taken over the choice of a, h, and $\mathcal{A}$'s coin tosses. An algorithm $\mathcal{A}$ (t, ε)-breaks Computational co-Diffie-Hellman on G_2 and G_1 if $\mathcal{A}$ runs in time at most t, and $\texttt{Adv co} - \texttt{CDH}_{\mathcal{A}}^{G_1,G_2}$ is at least ε. When $G_1 = G_2$ and $g_1 = g_2$, these problem reduce to the standard CDH [10].

Definition 2.2. Two groups (G_1, G_2) are a (t, ε)-bilinear group pair for co-Diffie-Hellman if there exists a bilinear map $e : G_1 \times G_2 \longrightarrow G_T$ and no algorithm (t, ε)-breaks Computational co-Diffie-Hellman on them.

q-co-Weak Computational Diffie-Hellman Problem (q-co-WCDH). The q-co-WCDH problem is defined as follows: given $q + 2$-tuple $(g_1, g_2, g_2^x, ..., g_2^{x^q})$ as input where $\psi(g_2) = g_1$, output $g_1^{\frac{1}{x}} \in G_1$. An algorithm $\mathcal{A}$ has advantage ε in solving q-co-WCDH problem if

$$\texttt{Adv}\ q\text{-co-WCDH}_{\mathcal{A}}^{G_1,G_2} = \texttt{Pr}[\mathcal{A}(g_1, g_2, g_2^x, ..., g_2^{x^q}) = g_1^{\frac{1}{x}}] \geq \varepsilon.$$

The probability is taken over the choice of g_1, g_2 and $x \in \mathcal{Z}_p^*$, and $\mathcal{A}$'s coin tosses.

3 Transitive Signature

All graphs in this paper are undirected. A graph $G = (V, E)$ is said to be transitively closed if all nodes $i, j, k \in V$ such that $(i, j) \in E$ and $(j, k) \in E$, it also holds that $(i, k) \in E$: or in other words, edge $(i, j) \in E$ whenever there is a path from i to j in G. For a graph $G = (V, E)$, its transitive closure is the graph $\tilde{G} = (V, \tilde{E})$, where $(i, j) \in E$ iff there is a path from i to j in G. Note that the transitive closure of any graph G is a transitively closed graph.

Definition 3.1. A transitive signature scheme $TS = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ is specified by four polynomial-time algorithms described as follows:

TKG the key generation algorithm, takes input 1^k and returns a pair (tpk, tsk) consisting of a public key and the matching private key.

TSign the signing algorithm, takes input the private key and nodes $i, j \in V$, and returns a the original signature of edge (i, j) relative to tpk.

TVf the verification algorithm, given tpk, nodes $i, j \in V$, and a candidate signature σ, verifies if σ is a valid signature of edge (i, j), returns 1 if so, otherwise returns 0.

Comp the composition algorithm, given tpk, nodes $i, j, k \in V$, where $i < j < k$, the values σ_1(the valid signature of edge (i, j)) and σ_2(the valid signature of edge (j, k)), then computes a value σ according the above given data. if σ is a valid signature of edge (i, k), returns 1, otherwise returns a symbol $\perp$ to indicate failure.

4 Proposed Transitive Signature Schemes

In this section we describe two bilinear transitive signature schemes via bilinear maps. Initially, the systems need run a setup procedure to generate and publish the following system parameters:

- a undirected graph $G = (V, E)$,
- three cyclic groups G_1, G_2, G_T of prime order p,
- generators g_1 and g_2 of G_1 and G_2 separately,
- a computable isomorphism ψ from G_2 to G_1 with $\psi(g_2) = g_1$,
- a computable bilinear map $e : G_1 \times G_2 \longrightarrow G_T$, and

4.1 The Scheme BMTS-1

Let $H : \{0, 1\}^* \longrightarrow G_1$ be a full domain hash function[6]. Then, our transitive signature scheme BMTS-1=(TKG, TSign, TVf, Comp) is defined as follows:

TKG given 1^k, pick random $x \xleftarrow{R} \mathcal{Z}_p$,and compute $v \longleftarrow g_2^x$. It outputs (tpk, tsk) $\longleftarrow (v, x)$.

TSign it maintains the state V which initially is empty. Suppose $Node$ is the set of integers indexing all the nodes in graph G, then $V \subset Node$ represents queried nodes. When asked to produce a signature on edge (i, j), it does as follows:

TVf on input $tpk = v$, and a transitive signature σ, it does as follows:

```
The $TSign$ algorithm:
    If $j < i$ Then swap $i, j$
    If $i \notin V$ Then $V \longleftarrow V \cup \{i\}$
    If $j \notin V$ Then $V \longleftarrow V \cup \{j\}$
    $\delta = (H(i)H(j)^{-1})^{tsk}$
    return $\sigma = (i, j, \delta)$
```

```
The $TVf$ algorithm:
parse $\sigma$ as $i, j, \delta$
If $i \notin V \vee j \notin V$ Then return 0
Else if $e(\delta, g_2) = e(H(i)H(j)^{-1}, tpk)$
    Then return 1
Else return 0
```

Comp Given two valid transitive signatures $\sigma_1 = (i, j, \delta_1)$ and $\sigma_2 = (j, k, \delta_2)$, suppose $i < j < k$, if not so, we can swap the sequence of i, j, k. Calculate $\delta = \delta_1\delta_2$ and output δ as the transitive composition.

4.2 The Scheme BMTS-2

The above proposed scheme needs a **MapToPoint**[3] hash function $H : (0, 1)^* \longrightarrow G_1$ which is probabilistic algorithm described in [3]. Now we introduce another transitive signature scheme BMTS-2 which is more efficient than BMTS-1 as it uses a regular cryptographic hash function $H_1 : (0,1)^* \longrightarrow (0,1)^l$ rather than a **MapToPoint** hash function. This transitive signature is constructed by applying another short signature scheme[14]. The transitive signature scheme BMTS-2 is defined as follows.

TKG given 1^k, pick random $x \xleftarrow{R} \mathcal{Z}_p$,and compute $v \longleftarrow g_2^x$. It outputs $(tpk, tsk) \longleftarrow (v, x)$.

TSign $TSign$ maintains the state V which initially is empty. Suppose $Node$ is the set of integers indexing all the nodes in graph G, then $V \subset Node$ represents queried nodes. When asked to produce a signature on edge (i, j), it does as follows:

TVf on input $tpk = v$, and a transitive signature σ, it does as follows:

```
The $TSign$ algorithm:
    If $j < i$ Then swap $i, j$
    If $i \notin V$ Then $V \longleftarrow V \cup \{i\}$
    If $j \notin V$ Then $V \longleftarrow V \cup \{j\}$
    $\delta = g_1^{\frac{H_1(i)+tsk}{H_1(j)+tsk}}$
    return $\sigma = (i, j, \delta)$.
```

```
The $TVf$ algorithm:
parse $\sigma$ as $i, j, \delta$
If $i \notin V \vee j \notin V$ Then return 0
If $e(\delta, g_2^{H_1(j)} tpk) = e(g_1, g_2^{H_1(i)} tpk)$
    Then return 1
Else return 0
```

Comp Given two valid transitive signatures $\sigma_1 = (i, j, \delta_1)$ and $\sigma_2 = (j, k, \delta_2)$, suppose $i < j < k$, if not so, we can swap the sequence of i, j, k. Calculate $\delta = \delta_1 \delta_2$ and output δ as the transitive composition.

5 Security and Efficiency

5.1 Security Definition of Transitive Signature Scheme

Associated to the transitive signature scheme $TS = (TKG, TSign, TVf, Comp)$, the adversary $\mathcal{F}$ and security parameter $k \in \mathcal{N}$ is an experiment defined as follows:

```
**Experiment** :$Exp_{TS,\mathcal{F}}^{tu-cma}(k)$
$H \longleftarrow \Omega$
$(tpk, tsk) \longleftarrow TK(1^k)$
$(i', j', \sigma') \longleftarrow \mathcal{F}^{TSign(tsk,\cdot,\cdot)}(tpk)$
If $TVf(i', j', \sigma') = 1 \wedge i', j' \in V \wedge (i', j') \notin \tilde{E}$ Then return 1
Else return 0
```

This experiment begins by choosing the appropriate hash function H in the hash function family Ω and running TKG on input 1^k to get keys (tpk, tsk). It then runs $\mathcal{F}$, providing this adversary with input tpk and oracle access to the function $TSign(tsk, \cdot, \cdot)$. The oracle is assumed to maintain states or toss coins as needed. Eventually, $\mathcal{F}$ will output a triple (i', j', σ'). Let E be the set of all edges that $\mathcal{F}$ made oracle query i, j, and let V be the set of all vertices i such that i is adjacent to some edge in E and $\tilde{G} = (V, \tilde{E})$ be the transitive closure of G. The advantage of $\mathcal{F}$ in its attack on TS is defined for by

$$\mathtt{Adv}_{TS,\mathcal{F}}^{tu-cma} = \Pr(Exp_{TS,\mathcal{F}}^{tu-cma}(k) = 1)$$

Definition 5.1. Given the security parameter $k \in \mathcal{N}$, we say that a forger $\mathcal{F}$ $(t, q_H, q_s, \varepsilon, k)$-breaks the transitive signature scheme TS if the function $\mathtt{Adv}_{TS,\mathcal{F}}^{tu-cma}$ is at least $\varepsilon(k)$

for any adversary $\mathcal{F}$ which runs in time at most $t(k)$; makes at most $q_H(k)$ queries to the hash function and at most $q_S(k)$ queries to the signing oracle. A transitive signature scheme TS is $(t, q_H, q_S, \varepsilon, k)$-transitively unforgeable under adaptive chosen-message attack if no forger $(t, q_H, q_s, \varepsilon, k)$-breaks it.

5.2 Security Proof

We state the security of the transitive signature scheme BMTS-1 as following theorem.

Theorem 1 *If (G_1, G_2) is a (t', ε')-bilinear group pair for co-Diffie-Hellman, with each group of order p, with respective generators g_1 and g_2, with an isomorphism ψ computable from G_2 to G_1. Then the transitive signature scheme BMTS-1 is $(t, q_H, q_S, \varepsilon,$ $k)$-transitively unforgeable under adaptive chosen-message attack for all t and ε satisfying*

$$\varepsilon \geq e^2(q_S(k)+1)\cdot \varepsilon' \textit{and } t \leq t' - C_{G_1}(q_H(k)+3q_S(k)+4) - q_H(k) - 3q_S(k) - 5,$$

Where e is the base of natural logarithms, and exponentiation and inversion on G_1 take time C_{G_1}, $k \in \mathcal{N}$ is the security parameter.

proof. Assume that $\mathcal{F}$ is a forger algorithm that $(t, q_H, q_S, \varepsilon, k)$-breaks the signature scheme. We will use $\mathcal{F}$ as a subroutine to construct an algorithm that (t', ε')-breaks the co-CDH problem in (G_1, G_2).

Given a challenge (y, g_2, g_2^a), where $y \in G_1$ and $g_2 \in G_2$. Its goal is to output $y^a \in G_1$. Algorithm $\mathcal{A}$ simulates the challenger and does experiment $Exp_{TS,\mathcal{F}}^{tu-cma}$ with the forger $\mathcal{F}$ as follows.

Setup. $\mathcal{A}$ constructs $tpk \longleftarrow g_2^a$, and gives the forger $\mathcal{F}$ the generator g_2 and tpk.

Hash Query. At any time the forger $\mathcal{F}$ can query the random oracle H about the node i. To respond to these queries, $\mathcal{A}$ maintains a set V which contains all nodes queried by $\mathcal{F}$, and a list $\mathcal{L}$ of tuples $< i, h^{(i)}, b^{(i)}, r^{(i)} >$ as explained below. They are initially empty. When $\mathcal{F}$ queries the oracle at the node i, algorithm $\mathcal{A}$ responds as described in the following function $H\_Query(i)$.

Signatures Queries. $\mathcal{A}$ answers $\mathcal{F}$'s signature queries on edge (i, j) as described in the following function $TSign\_Query(i, j)$.

```
Function $H\_Query(i)$
    If $i \in V$ Then return $h^{(i)}$
    Else
        $V \longleftarrow V \cup \{i\}$
        $r^{(i)} \xleftarrow{R} \{0,1\}$
        $b^{(i)} \xleftarrow{R} \mathcal{Z}_p$
        $h^{(i)} \longleftarrow y^{r^{(i)}} \cdot \psi(g_2)^{b^{(i)}} \in G_1$
        $\mathcal{L} \longleftarrow \mathcal{L} \cup \{(i, h^{(i)}, b^{(i)}, r^{(i)})\}$
        return $h^{(i)}$
```

```
Function $TSign\_Query(i, j)$
    If $i \notin V$ Then $H\_Query(i)$
    If $j \notin V$ Then $H\_Query(j)$
    If $r^{(i)} \neq r^{(j)}$ Then abort
    Else if $i < j$
    Then return $\psi(g_2^a)^{b^{(i)}-b^{(j)}}$
        Else return $\psi(g_2^a)^{b^{(j)}-b^{(i)}}$
```

Note: In hash query function $H\_Query(i)$, Pr$[r^{(i)} = 1] = 1/(q_s(k)+1)$).

Output: Let $\mathcal{F}$'s forgery be (i', j', δ'). Let E be the set of edges for which $\mathcal{F}$ queried a signature and let $\tilde{G} = (V, \tilde{E})$ be the transitive closure of graph $G = (V, E)$. $\mathcal{A}$ performs the following series of checks, aborting if one of them is true.

If $\underbrace{TVf(tpk, i'j', \delta') \neq 1}_{B_2}$ Then abort

Elese if $\underbrace{\{i', j'\} \in \tilde{E}}_{B_3}$ Then abort

Else if $\underbrace{r^{(i')} - r^{(j')} = 0}_{B_4}$ Then abort.

If $\mathcal{A}$ does not abort, it calculates and outputs the required y^a as:

$$y^a \longleftarrow (\delta')^{r^{(i')} - r^{(j')}} \cdot \psi(g_2^a)^{(b^{(j')} - b^{(i')})(r^{(i')} - r^{(j')})}$$

This completes the description of algorithm $\mathcal{A}$. It remains to show that $\mathcal{A}$ solves the instance of the co-CDH problem in (G_1, G_2) with advantage at least ε'. To do so, we analyze the following event needed for $\mathcal{A}$ to succeed:

B_1: $\mathcal{A}$ does not abort as a result of any of $\mathcal{F}$'s signature queries.

Consequently, the advantage of $\mathcal{A}$ is simply the probability of $\mathcal{A}$ not aborting during the experiment:

$$\begin{aligned}\texttt{Adv co} - \texttt{CDH}_{\mathcal{A}}^{G_1,G_2} &= \Pr[B_1 \wedge \bar{B_2} \wedge \bar{B_3} \wedge \bar{B_4}] \\ &= \Pr[\bar{B_4} | B_1 \wedge \bar{B_2} \wedge \bar{B_3}] \cdot \Pr[\bar{B_2} \wedge \bar{B_3} | B_1] \cdot \Pr[B_1] \qquad (1)\end{aligned}$$

In the following, we will give a lower bound for each of these terms.
(1) It is obviously that

$$\begin{aligned}\Pr[B_1] &= \Pr[\mathcal{A} \text{ asked for signatures only on edges } \{i, j\} \text{ with } r^{(i)} = r^{(j)}] \\ &\geq ((1 - 1/(q_S(k) + 1))^2 + 1/(q_S(k) + 1)^2)^m \\ &\geq (1 - 1/(q_S(k) + 1))^{2q_S(k)} \qquad (2)\end{aligned}$$

(2) The public key given to $\mathcal{F}$ is from the same distribution as public keys generated by algorithm TKG. Responses to hash queries are as in the real attack since each response is uniformly and independently in G_1. Since $\mathcal{A}$ did not abort as results of $\mathcal{F}$'s signature queries, all its responses to those queries are valid. Therefore, $\mathcal{A}$ will produce a valid and nontrivial transitive signature forgery with probability at least ε. Hence

$$\Pr[\bar{B_2} \wedge \bar{B_3} | B_1] \geq \varepsilon \qquad (3)$$

(3) Events B_1 and $\bar{B_2} \wedge \bar{B_3}$ have occurred, and $\mathcal{F}$ has generated a nontrivial signature forgery (i', j', δ'). If $\mathcal{F}$ asked for a signature under key tpk on some edges with one of their nodes is i', in which case the probability of $r^{(i')} = 0$ equals that of a hash query with $r^{(i)} = 0$, or it didn't, then $r^{(i')} = 0$ with probability $1 - 1/(q_S(k) + 1)$. So the probability of $r^{(i')} = 0$ is at least $1 - 1/(q_S(k) + 1)$. Also does node j'. Hence

$$\Pr[\bar{B_4} | B_1 \wedge \bar{B_2} \wedge \bar{B_3}] \geq 2/(q_S(k) + 1) \cdot (1 - 1/(q_S(k) + 1)) \qquad (4)$$

We use the bounds from equations (2), (3) and (4) in equation (1). Algorithm $\mathcal{A}$ produces the correct answer with probability at least $(1 - \frac{1}{q_S(k)+1})^{2(q_S(k)+1)-1} \cdot \frac{1}{q_S(k)+1} \cdot \varepsilon \geq \frac{\varepsilon}{e^2 \cdot (q_S(k)+1)} \geq \varepsilon'$ as required. Thus $\varepsilon \geq e^2(q_S(k)+1) \cdot \varepsilon'$. Obviously, algorithm $\mathcal{A}$'s running time is at most $t + C_{G_1}(q_H(k) + 3q_S(k) + 4) + q_H(k) + 3q_S(k) + 5 \leq t'$. This completes the proof of Theorem 1. □

The security of the transitive scheme BMTS-2 is described as the following theorem.

Theorem 2 *If (G_1, G_2) is a group pair for q-co-WCDH problem. Then the transitive signature scheme BMTS-2 is transitively unforgeable under adaptive chosen-message attacks in the random oracle model.*

proof. The proof of this theorem is the same as that of theorem 1. So it is unnecessary to give a full description. □

5.3 Efficiency

We will analyze the efficiency of BMTS-1 from the costs and signature size. Figure 1 gives us a detail comparison amongst transitive signature schemes. With regard to the costs, we are interested in the computational cost of signing an edge; the computational cost of verifying a candidate signature of an edge; the computational cost of composing two edge signatures to obtain another. As showed in figure 1, whether the costs of computation or the size of signature, our scheme is more efficient than all other schemes. If using Wail pairing, the advantage of our scheme can be obviously known.

Scheme	Signing	Verification	Composition	Signature size
MRTS	2 sig. + 2 exp.	1 exp.	2 adds in $\mathcal{Z}_q$	2 sig. + 2 points + 2 points
FBRS-1	2 stand.sigs	$O(\lvert N\rvert^2)$ ops	$O(\lvert N\rvert^2)$ ops	2 sig. + 3 points
FBRS-2	4 sqr. in $\mathcal{Z}_N^*$	$O(\lvert N\rvert^2)$ ops	$O(\lvert N\rvert^2)$ ops	1point in $\mathcal{Z}_N^*$
BMTS-1	1 exp. in G_1	2 bms.	$O(\lvert p\rvert^2)$ ops	1 point in G_1
BMTS-2	1 exp. in G_1	2 bms. 2 exp.	$O(\lvert p\rvert^2)$ ops	1 point in G_1

Fig. 1. Comparisons amongst transitive signature schemes. Abbreviations used are: "exp." for an exponentiation in the group; "sqr." for a square root computation modulo N; "ops" for the number of elementary bit operations in big-O notation; "bm." for bilinear map computing.

6 Conclusion

Bilinear maps have many applications in cryptographic fields. In this paper, we introduced two efficient and provable secure transitive signature schemes from bilinear maps. The proposed transitive signature schemes possess several properties such as: (i) no need node certification, (ii) short signature, (iii) compact signature algebra. The previously presented transitive signature schemes cannot achieve all the above properties. We also prove that our schemes are transitively unforgeable under adaptive chosen-message attack in the random oracle model.

References

1. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. SIAM, J. Computing, 32(3), pp.583-615, 2003.
2. D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps. Advances in Cryptology–EUROCRYPT 2003, Lecture Notes in Computer. Science, Vol. 2656, pp.416-432, Springer, 2003.
3. D. Boneh, B. Lymn and H. Shacham. Short signatures from the Weil pairing. Prodeedings of Asiacrypt 2001, Vol. 2248, Lecture Notes in Computer Science, pp.514-532, Springer, 2001.
4. M. Bellare and G.. Neven. Transitive signatures based on factoring and RSA. Advances in Cryptology - ASIACRYPT'02, Lecture Noted in Computer Science Vol.2501, pp.391-414, Springer, 2002.
5. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. Proceeding s of the 1st Annual Conference on Computer and Communications Security, ACM. pp.62-73, 1993.
6. J.-S. Coron, On the exact security of full domain hash. Advances in Cryptology–CRYPTO 2000, Lecture Notes in Computer Science, Vol.1880, pp.229-235, Springer, Berlin, 2000.
7. S. Chari, T. Rabin and R. Rivest. An efficient signature scheme for route aggregation. http://theory.lcs.mit.edu/ rivest/publications.html, 2002.
8. R. Housley, M. Ford, W. Polk, D.solo. Internet X.509 Public Key Infrastructure: Certificate and CRL Profile. http://www.ietf.org/rfc.html, January 1999.
9. R. Johnson, D. Molnar, D. Song and D. Wagner. Homomorphic signature schemes. Topics in Cryptology - CT-RSA '02, Lecture Notes in Computer Science Vol.2271, pp.244-262, Springer, 2002.
10. A. Joux and K. Nguyen. Separating decision Diffie-Hellman form Diffie-Hellman in cryptographic groups. Cryptology ePrint Archive, Report 2001/003, 2001. http://eprint.iacr.org/.
11. Changshe Ma and Kefei Chen, Publicly verifiable authenticated encryption. Electronics Letters, vol 39, pp.281-282, 2003.
12. S. Micali and R. L. Rivest. Transitive signature schemes. Topics in Cryptology - CT-RSA '02, Lecture Notes in Computer Science Vol.2271 , pp.236-243, Springer, 2002.
13. R. Steinfeld, L. Bull and Y. Zheng. Content Extraction signatures. Information security and cryptology-ICI 2001, Lecture Notes in Computer SCience Vol.2288, pp.285-304, Spring-Verlag, 2002.
14. Fangguo Zhang, Rei Safavi-Naini, and Willy Susilo. An Efficient Signature Scheme from Bilinear Pairings and Its Application. Public Key Cryptography - PKC 2004, Lecture Notes in Computer Science, volume 2947, pages 277-290. Springer, 2004.
15. H.Zhu and B. Feng, Robert H. Deng. Computing of Trust in Distributed Networks. http://www.iacr.org, eprint.