On Directed Transitive Signature

Jia Xu

Department of Computer Science, National University of Singapore
xujia@comp.nus.edu.sg

Abstract. In early 2000's, Rivest [Riv00,MR02] and Micali [MR02] introduced the notion of *transitive signature*, which allows a third party to generate a valid signature for a composed edge (v_i, v_k), from the signatures for two edges (v_i, v_j) and (v_j, v_k), and using the public key only. Since then, a number of works, including [MR02,BN02,Hoh03,SFSM05,BN05], have been devoted on transitive signatures. Most of them address the undirected transitive signature problem, and the directed transitive signature is still an open problem. S. Hohenberger [Hoh03] even showed that a directed transitive signature implies a complex mathmatical group, whose existence is still unkown. Recently, a few directed transitive signature schemes [Yi07,Nev08] on directed trees are proposed. The drawbacks of these schemes include: the size of composed signature increases linearly with the number of recursive applications of composition and the creating history of composed edge is not hidden properly. This paper presents $\mathcal{DTTS}$—a *Directed*-Tree-Transitive Signature scheme, to address these issues. Like previous works [Yi07,Nev08], $\mathcal{DTTS}$ is designed only for directed trees, however, it features with constant (composed) signature size and privacy preserving property. G. Neven [Nev08] pointed out constant signature size is an essential requirement of the original directed transitive signature problem raised by Rivest and Micali. In this sense, our scheme $\mathcal{DTTS}$ is the *first* transitive signature scheme on a directed tree. We also prove that $\mathcal{DTTS}$ is transitively unforgeable under adaptive chosen message attack in the standard model.

1 Introduction

In 2000, Rivest [Riv00] introduced the notion of homomorphic signatures (formalized in [JMSW02,ACdMT05] etc.) and proposed an open problem on the existence of directed transitive signatures. Later, Micali and Rivest [MR02] proposed the first undirected transitive signature scheme, and raised the directed transitive signature as open problem again and officially. A transitive signature scheme aims to authenticate the transitive closure of a dynamically growing graph [Yi07]. The scheme works in this way: a signer has a pair of public/private signing key, and is able to sign a new vertex or edge when it is generated at any time. Unlike standard digitial signature, the transitive signature scheme supports a transitive property. That is, given the signatures $\sigma_{i,j}$ and $\sigma_{j,k}$ of edges (v_i, v_j) and (v_j, v_k) respectively, anyone can produce a signature $\sigma_{i,k}$ for composed edge (v_i, v_k) using the public key only, where v_i, v_j, and v_k are vertices, and $(v_i, v_j), (v_j, v_k)$ are edges in a graph. If the graph is undirected, such scheme is called *undirected transitive signature* scheme; if the graph is directed, it is called *directed transitive signature* scheme.

Since Rivest's talk in 2000, a number of undirected transitive signature schemes [MR02,BN02,SFSM05,BN05,WCZ+07] have been proposed. However, the directed transitive signature is still an open problem [Hoh03,Nev08], although some *plausible* directed transitive signature schemes [KT03,Yi07,Nev08] on restricted directed graphs, like directed tree, have been proposed. Y. Xun et al. [YTO04] pointed out that Kuwakado-Tanaka transitive signature scheme [KT03] on directed trees is not secure under chosen message attack by proposing a forgery attack. Y. Xun [Yi07] also proposed a transitive signature scheme $\mathcal{RSADTS}$ on directed trees , but the (composed) signature size is not constant. G. Neven [Nev08] pointed out that it would be much easier to construct a directed transitive signature scheme (on directed tree) if the signature size is allowed to grow linearly, and gave a simple scheme as a demonstration. So far, to our knowledge, there is no known transitive signature scheme on directed trees, which is provably secure and has constant signature size. Table 1 and Table 2 compare various transitive signature schemes appeared in literatures with $\mathcal{DTTS}$ and $\mathcal{AOP}$-$\mathcal{DTS}$ proposed in this paper, from different aspects.

Scheme	**Signing cost**	**Verification cost**	**Composition cost**	**Signature size**	**Composed Signature size**	**Supported Graph**
$\mathcal{DLTS}$ [MR02]	2 stand. sigs. 2 exp. in $\mathbf{G}$	2 stand. verifs 1 exp. in $\mathbf{G}$	2 adds in $\mathbb{Z}_q$	2 stand. sigs 2 points in $\mathbf{G}$ 2 points in $\mathbb{Z}_q$	constant	undirected graph
$\mathcal{RSATS}$-1 [MR02]	2 stand. sigs. 2 RSA encs	2 stand. verifs 1 RSA enc.	$O(\|n\|^2)$ ops	2 stand. sigs. 3 points in $\mathbb{Z}_n^*$	constant	undirected graph
$\mathcal{F}act\mathcal{TS}$-1 [BN05]	2 stand. sigs $O(\|n\|^2)$ ops	2 stand. verifs $O(\|n\|^2)$ ops	$O(\|n\|^2)$ ops	2 stand. sigs 3 points in $\mathbb{Z}_n^*$	constant	undirected graph
$\mathcal{G}ap\mathcal{TS}$-1 [BN05]	2 stand. sigs 2 exp. in $\hat{\mathbb{G}}$	2 stand. verifs 1 S_{ddh}	$O(\|n\|^2)$ ops	2 stand. sigs. 3 points in $\hat{\mathbb{G}}$	constant	undirected graph
$\mathcal{RSADTS}$ [Yi07]	2 stand. sigs 1 exp. in $\langle\mathcal{G}\rangle$	2 stand. verifs 1 exp. in $\langle\mathcal{G}\rangle$	$\leq \|M\|$ ops	2 stand. sigs 2 points in $\langle\mathcal{G}\rangle$ 1 label $\delta_{i,j} \leq M$	increase	directed tree
$\mathcal{DTTS}$	$\leq$ 2 stand. sigs 2 exp. in $\mathbb{Z}_n^*$	2 stand. verifs 2 exp. in $\mathbb{Z}_n^*$	1 exp. in $\mathbb{Z}_n^*$	2 stand. sigs. 3† points in $\mathbb{Z}_n^*$	constant	directed tree (Arborescence)
$\mathcal{AOP}$-$\mathcal{DTS}$	$O(\|V\|^2)$	1 stand. verif	$O(\|V\|^2)$	1 stand. sig	constant	generic directed graph

Table 1. Performance comparision among transitive signature schemes [BN05,Yi07]. †: The left labels in a signature can be reduced using a hash function (Section 3.3).

In $\mathcal{RSADTS}$, each edge (i,j) is associated with a random number $r_{i,j}$ as the label. Given two adjacent edges (i,j) and (j,k) and their signatures, anyone with public key can produce a signature for the composed edge (i,k), whose label is the integer product $r_{i,j} \times r_{j,k}$. If we apply the transitive property recursively, the length of the label of the newly composed edge increases linearly with the depth of the recursion. Furthermore, the integer multiplication reveals some information about the creating history of the newly composed edge: if the original random numbers chosen by the signer are small, then adversaries could factorize the integer product; otherwise the bit-length of the product may reveal significant information about the number of multiplications, which implies the length of the path used to create the composed edge.

The directed transitive signature scheme $\mathcal{DTTS}$ on directed tree proposed in this paper, is inspired by the relation between transitive signature and redactable signature (Chang et al. [CLX09]), and is different from previous schemes at least in these aspects: (1) It is provably secure under adaptive chosen message attack; (2) The length of signature of a composed edge is constant; (3) The creating hisotry of a composed edge is hidden properly; (4) The directed tree supported by $\mathcal{DTTS}$ is slightly more restricted (precisely, every vertex has at most one incoming edge) than that of $\mathcal{RSADTS}$ (See Section 2); (5) When the transitive property is applied recursively on a path, for example path $i_1 \to i_2 \to i_3 \to i_4$, the order of recursive applications is predetermined. That is, compose a signature for (i_1, i_3) first from signatures of edge (i_1, i_2) and edge (i_2, i_3), then compose a singature for (i_1, i_4) from signatures of edge (i_1, i_3) and edge (i_3, i_4). This is because, in $\mathcal{DTTS}$, Comp requires the second edge is original, i.e. signed directly by the orignal signer. Note that the last difference does not restrict the power of transitive property of $\mathcal{DTTS}$. Instead, this difference can be treated as a feature, and can be utilized to provide the signer with control on composition (See Section 3.3 for details).

Scheme $\mathcal{AOP}$-$\mathcal{DTS}$ authenticates all ordered pairs of vertices in a generic directed graph with a constant size signature. It can achieve whatever generic directed transitive signature can achieve, as long as the composition operation can access some state maintained by the signer. This scheme illustrates that generic directed transitive signature is feasible, if the problem setting is relaxed slightly.

1.1 Contributions of this paper

Directed transitive signature is a hard open problem. We attack this problem from different angles in different simplified but meaningful settings. The contributions of this paper include:

1. We present $\mathcal{DTTS}$ as the first directed transitive signature scheme on directed trees with constant signature size (Section 3.1).
2. We prove that $\mathcal{DTTS}$ is transitively unforgeable under adaptive chosen message attack in standard model and the creating history of composed signature is hidden properly (Section 3.2).

Scheme	**Assumptions for Provable Security**	**Privacy Preserving**	**How to grow?**	**Persistent Vertex?**
$\mathcal{DLTS}$ [MR02]	Security of standard signature scheme; Hardness of iscrete logarithm in prime order group	Perfect,Transparent	Arbitrarily	No
$\mathcal{RSATS}$-1 [MR02]	Security of standard signature scheme; RSA is secure against one-more-inversion attack	Perfect,Transparent	Arbitrarily	No
$\mathcal{F}act\mathcal{TS}$-1 [BN05]	Security of standard signature scheme; Hardness of factoring	Perfect,Transparent	Arbitrarily	No
$\mathcal{G}ap\mathcal{TS}$-1 [BN05]	Security of standard signature scheme; One-more gap Diffie-Hellman assuption	Perfect,Transparent	Arbitrarily	No
$\mathcal{RSADTS}$ [Yi07]	Security of standard signature scheme; RSA Inversion Problem in a Cyclic Group is hard	No (due to integer multiplication)	From a single source	No
$\mathcal{DTTS}$	Security of standard signature scheme; Strong RSA Problem is hard	Computational,Non-Transparent	From a single source	Yes
$\mathcal{AOP}$-$\mathcal{DTS}$	Security of the underlying redactable signature scheme	Perfect, Transparent	Arbitrarily	No

Table 2. All of these schemes are transitive unforgeable under adaptive chosen-message attack in standard model [BN05]. Section 3.3 introduces the concept of "persistent vertex".

3. We point out that the directed transitive signature on generic graph could be a feasible problem, if we relax the requirement of transitive signature such that composition operation (Comp) could access the state maintained by the signer (TSign). The scheme $\mathcal{AOP}$-$\mathcal{DTS}$ illustrates this idea (Section 4). We also prove that $\mathcal{AOP}$-$\mathcal{DTS}$ is transitively unforgeable and privacy preserving.

2 Definitions

Notations. Let $\mathbb{N} = \{1, 2, 3, 4, 5, \ldots\}$ be the set of integers. The notation $x \leftarrow a$ denotes that x is assigned a value a, and $x \xleftarrow{\$} S$ denotes that x is randomly selected from the set S. Let Prime be the set of all odd prime numbers.

Graph. Let $G = (V, E)$ be a simple directed graph with a set V of nodes (or vertices) v_i's and a set E of directed edges. In this paper, we focus on directed trees. Note that there exist different definitions of directed tree in the literature: (1)A directed tree is a directed graph that would be a (undirected) tree if ignoring the direction of edges; (2)A directed tree (or *Arborescence*) is a directed graph, where edges are all directed away from a particular vertex. The second definition is slightly more restricted than the first one. In this paper, we adopt the second definition for directed tree and the term "directed tree" refers to arborescence by default. Y. Xun [Yi07] adopted the first definition of directed tree.

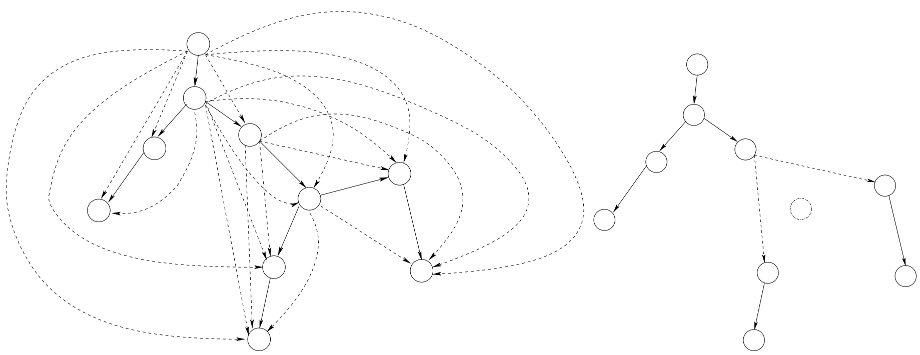
A *transitive closure* of a directed graph $G = (V, E)$, is a directed graph, denoted as $\widetilde{G} = (V, \widetilde{E})$, where $(v_i, v_j) \in \widetilde{E}$ if and only if there is a directed path from vertex v_i to vertex v_j in graph G.

Directed Transitive Signature Scheme. A directed transitive signature scheme $\mathcal{DTS} = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ is specified by four polynomial-time algorithms, and the functionality is as follows [BN05,Yi07]:

- The randomized *key generation* algorithm TKG takes as input 1^k, where k is the security parameter, and returns a pair of keys (tpk, tsk), where tpk is the public key and tsk is the private key.
- The *signing* algorithm TSign could be randomized or/and stateful. TSign takes the private key tsk, two vertices v_i and v_j, and returns a value called an *orignal signature* of the edge (v_i, v_j) relative to tsk. If stateful, it maintains a state which it updates upon each invocation.
- The deterministic *verification* algorithm TVf, given tpk, two vertices v_i, v_j and a candidate signature σ, returns either `TRUE` or `FALSE`. We say that σ is a *valid signature* of edge (v_i, v_j) relative to tsk, if the output is `TRUE`.
- The deterministic *composition* algorihtm Comp takes as input tpk, two directed edges (v_i, v_j) and (v_j, v_k) and two signatures $\sigma_{i,j}$ and $\sigma_{j,k}$, and returns either a *composed signature* $\sigma_{i,k}$ of the composed edge (v_i, v_k), or $\perp$ to indicate failure.

An edge e is called *original edge* if $e \in E$, or *composed edge* if $e \in \widetilde{E} - E$. All original edges are signed by the signer using TSign and tsk, and all composed edges could be indirectly signed by anyone using Comp and tpk.

Two different views of Transitive Signatures. Transitive signatures are originally designed to authenticate a transitively closed graph in an economic way, i.e. sign as least as possible number of vertices and edges to authenticate a transitively closed graph. Viewed from another angle, transitive signatures are actually redactable signatures on growing graph (Figure 1). The redaction operation can be implemented straightforwardly just using the composition operation Comp.



(a) Transitive Closure (b) Redaction

Fig. 1. This graph illustrates the two different views of transitive property. In Subfigure (a), composed edges represented by dashed lines are signed indirectly by applying composition operation Comp. In this graph of 10 vertices and 29 edges, 9 original edges are signed directly using TSign, and the signatures of the other 20 composed edges (dashed line) can be saved due to transitive property. In Subfigure (b), a vertex represented by the dashed circle is redacted from the graph, and the edges connecting its parent and children are created and signed by applying Comp.

Correctness, Security and Privacy. We slightly modify the definitons of correctness and security of (directed) transitive signature scheme in [BN05,Yi07] to adapt for $\mathcal{DTTS}$. We also formalize the definition of privacy of transitive signatures when viewed as redactable signatures.

Experiement 1 defines $\mathsf{Exp}^{\mathsf{Correct}}_{\mathcal{DTS},\mathcal{A}}$ for correctness of $\mathcal{DTS}$ and Experiment 2 defines $\mathsf{Exp}^{\mathsf{dtu-cma}}_{\mathcal{DTS},\mathcal{F}}$ for security of $\mathcal{DTS}$. $\mathsf{Exp}^{\mathsf{Correct}}_{\mathcal{DTS},\mathcal{A}}$ outputs $\texttt{TRUE}$, if all queries made by $\mathcal{A}$ are legitimate, and $\mathcal{A}$ can make a TSign query or Comp query which can cause TSign or Comp to generate an invalid signature. The experiment $\mathsf{Exp}^{\mathsf{dtu-cma}}_{\mathcal{DTS},\mathcal{F}}$ outputs 1 if and only if $\mathcal{F}$ succeeds in producing a forgery. The advantage of $\mathcal{F}$ in its adaptive chosen message attack on $\mathcal{DTS}$ is defined as

$$\mathsf{Adv}^{\mathsf{dtu-cma}}_{\mathcal{DTS},\mathcal{F}}(k) = \Pr\left[\mathsf{Exp}^{\mathsf{dtu-cma}}_{\mathcal{DTS},\mathcal{F}}(k) = 1\right]$$

where $k \in \mathbb{N}$ and the probability is taken over all random choices made in the experiment $\mathsf{Exp}^{\mathsf{dtu-cma}}_{\mathcal{DTS},\mathcal{F}}$. Experiment 3 defines $\mathsf{Exp}^{\mathsf{privacy}}_{\mathcal{DTS}}$, which is used to define privacy preserving property for transitive signatures when viewed as redactable signatures.

Definition 1 (Correctness). *A transitive signature scheme* $\mathcal{DTS} = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ *is* correct*, if for any (computationally unbounded) algorithm* $\mathcal{A}$ *and every* $k \in \mathbb{N}$*,*

$$\Pr\left[\mathsf{Exp}^{\mathsf{Correct}}_{\mathcal{DTS},\mathcal{A}} = \mathtt{TRUE}\right] = 0.$$

Experiment 1 $\mathsf{Exp}^{\mathsf{Correct}}_{\mathcal{DTS},\mathcal{A}}$ defines correctness of transitive signature scheme $\mathcal{DTS} = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ for directed tree.

$(tpk, tsk) \leftarrow \mathsf{TKG}(1^k)$
$S \leftarrow \emptyset$; $Legit \leftarrow \mathtt{TRUE}$; $NotOK \leftarrow \mathtt{FALSE}$
Run $\mathcal{A}$ with its oracles until it halts, replying to its oracle queries as follows:
if $\mathcal{A}$ makes TSign query on (v_i, v_j) **then**
    if $v_i = v_j \ \lor\ (v_i, v_j) \in E$ **then**
        $Legit \leftarrow \mathtt{FALSE}$
    else
        Let σ be the output of TSign oracle
        $S \leftarrow S \cup \{(v_i, v_j, \sigma)\}$
        if $\mathsf{TVf}_{tpk}(v_i, v_j, \sigma) = \mathtt{FALSE}$ **then**
            $NotOK \leftarrow \mathtt{TRUE}$
if $\mathcal{A}$ makes Comp query on $v_i, v_j, v_k, \sigma_{i,j}, \sigma_{j,k}$ **then**
    if (v_j, v_k) is not an original edge $\lor$ v_i, v_j, v_k are not all distinct $\lor$ $(v_i, v_j, \sigma_{i,j}) \notin S$ $\lor$ $(v_j, v_k, \sigma_{j,k}) \notin S$ **then**
        $Legit \leftarrow \mathtt{FALSE}$
    else
        Let $\sigma_{i,k}$ be the output of Comp oracle
        if $\sigma_{i,k} = \perp$ **then**
            $Legit \leftarrow \mathtt{FALSE}$
        else
            $S \leftarrow S \cup \{(v_i, v_k, \sigma_{i,k})\}$
            if $\mathsf{TVf}_{tpk}(v_i, v_k, \sigma_{i,k}) = \mathtt{FALSE}$ **then**
                $NotOK \leftarrow \mathtt{TRUE}$
When $\mathcal{A}$ halts, output $(Legit \land NotOK)$ and halts

Definition 2 (Security). *A transitive signature scheme* $\mathcal{DTS} = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ *is* transitively unforgeable *under adaptive chosen message attack, if the function* $\mathsf{Adv}^{\mathsf{dtu-cma}}_{\mathcal{DTS},\mathcal{F}}(k)$ *is negligible in* k *for any adversary* $\mathcal{F}$ *whose running time is polynomial in* k*.*

Definition 3 (Privacy). *A transitive signature scheme* $\mathcal{DTS} = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ *is* non-transparent and computational privacy preserving *(respectively, transparent and computational privacy preserving), if for any* $\ell > 1$ *(respectively,* $\ell > 0$*),* X_ℓ *and* X_1 *(respectively,* X_0*) are computationally indistinguishable (w.r.t.* k*), where* X_1, X_ℓ *are defined as follow*

Experiment 2 $\mathsf{Exp}^{\mathsf{dtu-cma}}_{\mathcal{DTS},\mathcal{F}}$ defines security of transitive signature scheme $\mathcal{DTS}$ = $(\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ for directed tree.

$(tpk, tsk) \leftarrow \mathsf{TKG}(1^k)$
$S \leftarrow \emptyset$; $Legit \leftarrow$ `TRUE`
Run $\mathcal{F}$ with its oracles until it halts, replying to its oracle queries as follows:
if $\mathcal{F}$ makes TSign query on (v_i, v_j) **then**
    if $v_i = v_j \;\vee\; (v_i, v_j) \in E$ **then**
        $Legit \leftarrow$ `FALSE`
    else
        Let σ be the output of TSign oracle
        $S \leftarrow S \cup \{(v_i, v_j, \sigma)\}$
if $\mathcal{F}$ makes Comp query on $v_i, v_j, v_k, \sigma_{i,j}, \sigma_{j,k}$ **then**
    if (v_j, v_k) is not an original edge $\vee$ v_i, v_j, v_k are not all distinct $\vee$ $(v_i, v_j, \sigma_{i,j}) \notin S$ $\vee$ $(v_j, v_k, \sigma_{j,k}) \notin S$ **then**
        $Legit \leftarrow$ `FALSE`
    else
        Let $\sigma_{i,k}$ be the output of Comp oracle
        $S \leftarrow S \cup \{(v_i, v_k, \sigma_{i,k})\}$
Forger $\mathcal{F}$, with access to tpk and S, outputs (v', u', σ'): $(v', u', \sigma') \leftarrow \mathcal{F}(tpk, S)$.
Let $E \leftarrow \{(v_i, v_j) \mid \exists (v_i, v_j, \sigma) \in S\}$; $V = \{v \mid \exists u, (u, v) \in E \;\vee\; (v, u) \in E\}$
Let graph $G = (V, E)$ and its transitive closure $\widetilde{G} = (V, \widetilde{E})$
if $Legit =$ `FALSE` $\vee$ $(v', u') \in \widetilde{E}$ $\vee$ $\mathsf{TVf}(v', u', \sigma') =$ `FALSE` **then**
    return 0
else
    return 1

1. *Run* TKG *to generate public/private key:* $(tpk, tsk) \leftarrow \mathsf{TKG}(1^k)$.
2. *Randomly generate* v_0, v_1.
3. *For any* $c \geq 0$,
$$X_c \leftarrow \mathsf{Exp}^{\mathsf{privacy}}_{\mathcal{DTS}}(tpk, tsk, c, v_0, v_1)$$

Remark

1. $\mathcal{DTS}$ is *statistical privacy preserving*, if "computationally indistinguishable" is replaced with "statistically indistinguishable" in Definition 3.
2. $\mathcal{DTS}$ is *perfect privacy preserving*, if "computationally indistinguishable" is replaced with "identical" in Definition 3.
3. If $\mathcal{DTS}$ is transparent privacy preserving, then given an authenticated graph signed by $\mathcal{DTS}$, any advesary (computationlly bounded if $\mathcal{DTS}$ is computational privacy preserving) cannot distinguish orignal signatures from composed signatures. If $\mathcal{DTS}$ is non-transparent privacy preserving, then given an authenticated graph signed by $\mathcal{DTS}$, any advesary may be able to distinguish orignal singatures from composed signatures, but could not obtain any information about the creating history of a composed signature.

3 $\mathcal{DTTS}$: Transitive Signature on Directed Tree

3.1 The scheme

Let $\mathcal{SDS} = (\mathsf{SKG}, \mathsf{SSign}, \mathsf{SVf})$ be a standard signature scheme (For example, the signature scheme proposed by Goldwasser et al [GMR88]). We define the directed transitive signature scheme $\mathcal{DTTS} = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ as follows.

Experiment 3 $\mathsf{Exp}^{\mathsf{privacy}}_{\mathcal{DTS}}$ outputs a composed signature for edge (v_0, v_1) by composing a path of length $(\ell+1)$ recursively.

Input: $(tpk, tsk, \ell, v_0, v_1)$
Generate random vertex u_i, $0 < i < \ell+1$, and let $u_0 = v_0, u_{\ell+1} = v_1$.
Set the state of TSign to a random state.
for $j \leftarrow 0; j \leq \ell; j \leftarrow j+1$ **do**
    Make TSign query on (u_j, u_{j+1}) against tsk and obtain the signature $\sigma_{j,j+1}$
for $j \leftarrow 2; j \leq \ell+1; j \leftarrow j+1$ **do**
    Make Comp query on $u_0, u_{j-1}, u_j, \sigma_{0,j-1}, \sigma_{j-1,j}$ against tpk and obtain signature $\sigma_{0,j}$
return $\sigma_{0,\ell+1}$.

$\mathsf{TKG}(1^k)$. The key generation algorithm TKG taking 1^k as input, runs as follows:

1. Run $\mathsf{SKG}(1^k)$ to generate a key pair (spk, ssk).
2. Choose a RSA modulus $n = pq$, such that $p = 2p'+1, q = 2q'+1$, p, q, p' and q' are all prime, and $|p| = |q|$. Let $\lambda(n) = lcm(p-1, q-1)$.
3. Choose an element g from $\mathbb{Z}_n^*$, such that the multiplicative order of g modulo n is p'. Let $\langle g \rangle$ denote the subgroup of $\mathbb{Z}_n^*$ generated by g. Let $\mathcal{P}$ denote the set of all odd primes in $\mathbb{Z}_{p'}$, i.e. $\mathcal{P} = \mathbb{Z}_{p'} \cap \mathsf{Prime}$.
4. Output $tpk = (spk, n)$ as the public key and $tsk = (ssk, \lambda(n), p', g)$ as the private key.

$\mathsf{TSign}_{tsk}(v_i, v_j)$. The signing algorithm TSign maintains a state $(V, E, L, \Pi, \Delta, \Sigma)$:

- $V \subset \{0,1\}^*$ is a set of queried nodes;
- $E \subset V \times V$ is a set of directed edges;
- The function $L : V \rightarrow \mathcal{P} \times \mathbb{Z}_n^*$ assigns to each node $v \in V$ a public label $L(v)$, which consists of a prime (called left label, denoted as $L_{\mathcal{L}}(v)$) from $\mathcal{P}$ and an element (called right label, denoted as $L_{\mathcal{R}}(v)$) from $\mathbb{Z}_n^*$ $(L(v) \equiv (L_{\mathcal{L}}(v), L_{\mathcal{R}}(v)))$;
- The set Π records all prime numbers chosen in the signing process;
- The function $\Delta : E \rightarrow \mathbb{Z}_n^*$ assigns to each edge $(v_i, v_j) \in E$ a label $\delta_{i,j}$;
- The function $\Sigma : V \rightarrow \{0,1\}^*$ assigns to each node $v \in V$ a standard signature $\Sigma(v)$.

Initially, all of V, E and Π are empty sets. When invoked on input v_i, v_j $(v_i \neq v_j)$ and tsk, the signing algorithm TSign runs as follows:

1. Case 1: $v_i, v_j \notin V$, i.e. neither vertex v_i or vertex v_j is signed.
   (a) Choose r_i randomly from $\mathcal{P} - \Pi$: $r_i \xleftarrow{\$} \mathcal{P} - \Pi$. Update Π: $\Pi \leftarrow \Pi \cup \{r_i\}$.
   (b) The left label $L_{\mathcal{L}}(v_i)$ of v_i is: $L_{\mathcal{L}}(v_i) \leftarrow r_i$. The right label $L_{\mathcal{R}}(v_i)$ of v_i is: $L_{\mathcal{R}}(v_i) \leftarrow g^{r_i} \mod n$.
   (c) Choose r_j randomly from $\mathcal{P} - \Pi$: $r_j \xleftarrow{\$} \mathcal{P} - \Pi$. Update Π: $\Pi \leftarrow \Pi \cup \{r_j\}$.
   (d) The left label $L_{\mathcal{L}}(v_j)$ of v_j is: $L_{\mathcal{L}}(v_j) \leftarrow r_j$. The right label $L_{\mathcal{R}}(v_j)$ of v_j is: $L_{\mathcal{R}}(v_j) \leftarrow L_{\mathcal{R}}(v_i)^{r_j} \mod n$.
   (e) $\Sigma(v_i) \leftarrow \mathsf{SSign}_{ssk}(v_i, r_i, L_{\mathcal{R}}(v_i))$; $\Sigma(v_j) \leftarrow \mathsf{SSign}_{ssk}(v_j, r_j, L_{\mathcal{R}}(v_j))$.
   (f) The certificate of v_i is: $C(v_i) \leftarrow (v_i, r_i, L_{\mathcal{R}}(v_i), \Sigma(v_i))$. The certificate of v_j is: $C(v_j) \leftarrow (v_j, r_j, L_{\mathcal{R}}(v_j), \Sigma(v_j))$

(g) The label of the edge (v_i, v_j) is: $\Delta(v_i, v_j) \leftarrow g$.

2. Case 2: $v_i \in V, v_j \notin V$, i.e. vertex v_i is signed already but vertex v_j is not signed yet.
   (a) Let the certificate of v_i be $C(v_i) = (v_i, r_i, L_{\mathcal{R}}(v_i), \Sigma(v_i))$, where $r_i = L_{\mathcal{L}}(v_i)$.
   (b) Randomly choose r_j from $\mathcal{P} - \Pi$: $r_j \xleftarrow{\$} \mathcal{P} - \Pi$. Update Π: $\Pi \leftarrow \Pi \cup \{r_j\}$.
   (c) The left label $L_{\mathcal{L}}(v_j)$ of v_j is: $L_{\mathcal{L}}(v_j) \leftarrow r_j$. The right label of v_j is $L_{\mathcal{R}}(v_j) \leftarrow L_{\mathcal{R}}(v_i)^{r_j} \mod n$.
   (d) The certificate of vertex v_j is $C(v_j) \leftarrow (v_j, r_j, L_{\mathcal{R}}(v_j), \Sigma(v_j))$, where $\Sigma(v_j) \leftarrow \mathsf{SSign}_{ssk}(v_j, r_j, L_{\mathcal{R}}(v_j))$.
   (e) The label of the edge (v_i, v_j) is: $\Delta(v_i, v_j) \leftarrow L_{\mathcal{R}}(v_i)^{\frac{1}{r_i}} \mod n$.
3. Case 3: $v_i \notin V, v_j \in V$, i.e. vertex v_j is signed already but vertex v_i is not signed yet.
   (a) Let the certificate of v_j be $C(v_j) = (v_j, r_j, L_{\mathcal{R}}(v_j), \Sigma(v_j))$, where $r_j = L_{\mathcal{L}}(v_j)$.
   (b) Randomly choose r_i from $\mathcal{P} - \Pi$: $r_i \xleftarrow{\$} \mathcal{P} - \Pi$. Update Π: $\Pi \leftarrow \Pi \cup \{r_i\}$.
   (c) The left label $L_{\mathcal{L}}(v_i)$ of v_i is: $L_{\mathcal{L}}(v_i) \leftarrow r_i$. The right label of v_i is: $L_{\mathcal{R}}(v_i) \leftarrow L_{\mathcal{R}}(v_j)^{\frac{1}{r_j}} \mod n$.
   (d) The certificate of vertex v_i is: $C(v_i) \leftarrow (v_i, r_i, L_{\mathcal{R}}(v_i), \Sigma(v_i))$, where $\Sigma(v_i) \leftarrow \mathsf{SSign}_{ssk}(v_i, r_i, L_{\mathcal{R}}(v_i))$.
   (e) The label of the edge (v_i, v_j) is: $\Delta(v_i, v_j) \leftarrow L_{\mathcal{R}}(v_i)^{\frac{1}{r_i}} \mod n$.

For all cases, update V and E: $V \leftarrow V \cup \{v_i, v_j\}, E \leftarrow E \cup \{(v_i, v_j)\}$, and output $(C(v_i), C(v_j), \Delta(v_i, v_j))$ as the signature of (v_i, v_j).

$\mathsf{TVf}_{tpk}(v_i, v_j, \sigma_{i,j})$. The verification algorithm TVf, when revoked on input tpk, nodes v_i, v_j and a candidate signature $\sigma_{i,j}$ on directed edge (v_i, v_j), runs as follows:

1. Parse $\sigma_{i,j}$ as $(C_i, C_j, \delta_{i,j})$. Parse C_i as $(v_i, r_i, L_{\mathcal{R},i}, \sigma_i))$ and parse C_j as $(v_j, r_j, L_{\mathcal{R},j}, \sigma_j)$.
2. If $\mathsf{SVf}_{spk}((v_i, r_i, L_{\mathcal{R},i}), \sigma_i) = \texttt{FALSE}$ or $\mathsf{SVf}_{spk}((v_j, r_j, L_{\mathcal{R},j}), \sigma_j) = \texttt{FALSE}$, then reject.
3. Accept if $\quad \delta_{i,j}^{r_i r_j} \equiv L_{\mathcal{R},j} \pmod n$.

$\mathsf{Comp}_{tpk}(v_i, v_j, v_k, \sigma_{i,j}, \sigma_{j,k})$. The composition algorithm Comp, when invoked on input tpk, nodes v_i, v_j, v_k, and two signatures $\sigma_{i,j}$ and $\sigma_{j,k}$, runs as follows:

1. Parse $\sigma_{i,j}$ as $(C_i, C_j, \delta_{i,j})$ and $\sigma_{j,k}$ as $(C'_j, C_k, \delta_{j,k})$.
2. If C_j and C'_j are different, output $\perp$ and abort.
3. Parse C_i, C_j, C_k as $(v_i, r_i, L_{\mathcal{R},i}, \sigma_i), (v_j, r_j, L_{\mathcal{R},j}, \sigma_j)$ and $(v_k, r_k, L_{\mathcal{R},k}, \sigma_k)$ respectively.
4. If $\mathsf{SVf}_{spk}((v_i, r_i, L_{\mathcal{R},i}), \sigma_i) = \texttt{FALSE}$ or $\mathsf{SVf}_{spk}((v_j, r_j, L_{\mathcal{R},j}), \sigma_j) = \texttt{FALSE}$ or $\mathsf{SVf}_{spk}((v_k, r_k, L_{\mathcal{R},k}), \sigma_k) = \texttt{FALSE}$, output $\perp$ and abort.

5. If $L_{\mathcal{R}}(v_j)^{r_k} \not\equiv L_{\mathcal{R}}(v_k) \mod n$, output $\perp$ and abort[1].
6. Compute $\delta_{i,k} \leftarrow \delta_{i,j}^{r_j} \mod n$.
7. Output $(C_i, C_k, \delta_{i,k})$ as the signature of edge (v_i, v_k).

Figure 2 shows the left and right labels associated with eavery vertex v_i.

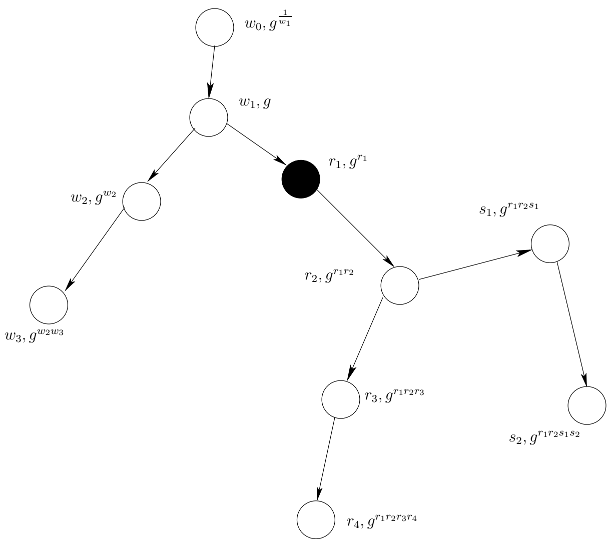


Fig. 2. This figure shows the left label $L_{\mathcal{L}}(v)$ and right label $L_{\mathcal{R}}(v)$ associated with every vertex v. Note this graph grows from the vertex represented by the dark circle.

Remarks.

1. $\mathcal{DTTS}$ assumes Case 1 of TSign will occur only once — when the very first edge is queried and signed. Except the first edge, any newly queried edge must have one adjacent node signed and the other unsigned yet. This implies that the graph grows from the first signed vertex.
2. As long as the graph $G = (V, E)$ is a tree, the case that $v_i, v_j \in V$, i.e. both v_i and v_j are queried before, should never occur during the execution of TSign.
3. When composing edges (v_i, v_j) and (v_j, v_k), Comp assumes that (v_j, v_k) is an original edge which is signed by the signer. This implies that the order of recursive applications of Comp on a path is predetermined. This feature allows the signer to have some control on the composition (See Section 3.3).

[1] This means the Comp algorithm requres that the second edge (v_j, v_k) is an original edge, i.e. signed by the signer, instead of edge generated by composing a path.

4. There is a way to distinguish original edge, which is signed by the signer, from composed edge, which is signed by applying Comp. That is, $(v_i, v_j) \in \widetilde{E}$ is original, if $L_{\mathcal{R}}(v_i)^{r_j} \equiv L_{\mathcal{R}}(v_j) \mod n$; otherwise, it is composed.

3.2 Security and Privacy

Theorem 1. *$\mathcal{DTTS} = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ as defined in Section 3.1 is transitively unforgeable under adaptive chosen message attack, assuming the standard signature scheme $\mathcal{SDS} = (\mathsf{SKG}, \mathsf{SSign}, \mathsf{SVf})$ is unforgeable under adaptive chosen message attack and the Strong RSA problem is difficult.*

Assumption 1 *Let $n = pq, p = 2p' + 1$ and $q = 2q' + 1$, where p, q, p', q' are all prime, and $|p| = |q|$. Let $g \in \mathbb{Z}_n^*$ be an element with multiplicative order modulo n equal to p'. The following two random variables X and Y are computationally indistinguishable,*

– *Randomly and independently choose a, b from $Z_{p'} \cap \mathsf{Prime}$, $X \leftarrow g^{ab} \mod n$,*
– *Randomly and independently choose c, from $Z_{p'} \cap \mathsf{Prime}$, $Y \leftarrow g^{c} \mod n$.*

Note Assumption 1 is implied by Decisional Diffie-Hellman assumption in the cyclic sub-group of Z_n^*.

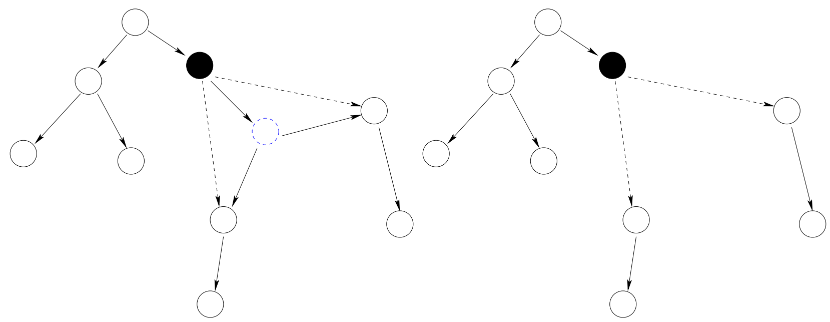
Theorem 2. *$\mathcal{DTTS} = (\mathsf{TKG}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ is non-transparent and computational privacy preserving, under Assumption 1.*

3.3 Variances

In this subsection, we give some variant schemes based on $\mathcal{DTTS}$ using different techniques. Note that these techniques can be combined together.

Control on Redaction In some applications, it could be very desirable to make some particular vertex *persistent*, so that no one, except the signer, can redact a persistent vertex from a signed graph. For example, in the hierarchy of chain of command, some particular person should never be crossed.

$\mathcal{DTTS}$ allows the signer to have control on which vertices are persistent and which are not (Figure 3). To add a non-persistent vertex, just follow the scheme described in Section 3.1. To add a persistent vertex v_i (for example, the vertex represented by the dark circle in Figure 3), the signer adds a dummy vertex u (for example, the vertex represented by the dashed circle in Figure 3(a)) as v_i's only child (so any child of v_i actually becomes the child of u), and then redacts this dummy vertex u using Comp algorithm.



(a) The signing process (b) The resulting graph

Fig. 3. This graph illustrates how to make a vertex (represented by the dark circle) persistent. In Subfigure (a), to make the vertex represented by the dark circle persistent, we introduce a dummy vertex, which is represented by the dashed circle. In Subfigure (b), dashed edges connecting the persistent vertex and its children are signed indirectly using Comp, so Comp cannot take these edges as the second input.

Reduce the signature size using hashing Similar as in Bellare et al. [BN05], we could reduce the signature size via hashing. Let $h(\cdot)$ be a division intractable hash function as defined in Gennaro et al. [GHR99]. By defining $L_{\mathcal{L}}(v_i) = h(v_i)$, we could remove r_i from the certification $C(v)$ of the vertex v. However, we cannot eliminate the right label of a vertex using the same technique. Indeed, the value of the right label of a vertex relies on the path from the very first signed vertex to itself. This makes $\mathcal{DTTS}$ a naturally stateful siginning algorithm. We cannot convert $\mathcal{DTTS}$ to a stateless signning algorithm using the technique introduced in Bellare et al. [BN05].

4 $\mathcal{AOP}$-$\mathcal{DTS}$: Authenticate all Ordered Pairs

In this section, we present a directed transitive signature scheme $\mathcal{AOP}$-$\mathcal{DTS}$ on generic directed tree, which allows the composition operation Comp to access some state variable (precisely, σ) maintainted by the signer TSign.

Let $G = (V, E)$ represent the directed graph, and $\widetilde{G} = (V, \widetilde{E})$ represent the transitive closure of G. Note G keeps changing, so does $\widetilde{G}$. Let $\mathcal{RSS} =$ (RKG, RSign, RVf, Redact, Union) be a redactable signature scheme on sets of objects, which supports the following two features

- Union: Given signatures of two sets S_1 and S_2, one can produce the signature for set $S_1 \cup S_2$ using public key only. Precisely, the output of $\mathsf{Union}(S_1, \sigma_1, S_2, \sigma_2)$ is a valid signature for the set $S_1 \cup S_2$.
- Set Difference (or Redaction): Given a signature of a set S, one can produce the signature for set $S - A$ for any set A using public key only. More precisely, the output of $\mathsf{Redact}(S, \sigma, A)$ is a valid signature of the set $S - A$.

Johnson et al. [JMSW02] gave an example of such redactable signature scheme (`Sig` in Section 5 of [JMSW02]).

Scheme $\mathcal{AOP}$-$\mathcal{DTS}$ works in this way: (1) Sign $\widetilde{E}$ using $\mathcal{RSS}$ to obtain the signature σ; (2) Once a new edge (v_i, v_j) is added, sign $\{(v_i, v_j)\}$ using $\mathcal{RSS}$, and update $V, E, \widetilde{E}$ and its signature σ; (3) From signature σ and graph G, anyone can produce a valid signature for any edge $e \in \widetilde{E}$. The details are as follows.

1. $\mathsf{KG}(1^k)$: Run $\mathsf{RKG}(1^k)$ to generate a key pair (pk, sk). Output (pk, sk).
2. $\mathsf{TSign}_{sk}(v_i, v_j)$: The signing algorithm TSign maintains a state $(V, E, \widetilde{E}, \sigma)$, where V is a set of quried vertices, $E \subset V \times V$ is a set of directed edges, $\widetilde{E}$ is the transitive closure of E, and σ is the signature of $\widetilde{E}$ under $\mathcal{RSS}$ w.r.t. sk.
   (a) Let A be an empty set. For any $u, v \in V$, if $(u, v_i) \in \widetilde{E}$, then add (u, v_j) into A; if $(v_j, v) \in \widetilde{E}$, then add (v_i, v) into A; if both $(u, v_i) \in \widetilde{E}$ and $(v_j, v) \in \widetilde{E}$, then add (u, v) into A.
   (b) Sign the set A: $\sigma_A \leftarrow \mathsf{RSign}_{sk}(A)$.
   (c) Update state: $\sigma \leftarrow \mathsf{Union}_{pk}(\widetilde{E}, \sigma, A, \sigma_A)$; $V \leftarrow V \cup \{v_i, v_j\}$; $E \leftarrow E \cup \{(v_i, v_j)\}$; $\widetilde{E} \leftarrow \widetilde{E} \cup A$.
   (d) The signature of edge (v_i, v_j) is: $\sigma_{i,j} \leftarrow \mathsf{RSign}_{sk}(\{(v_i, v_j)\})$.
3. $\mathsf{TVf}_{pk}(v_i, v_j, s)$: Return $\mathsf{RVf}_{pk}(\{(v_i, v_j)\}, s)$.
4. $\mathsf{Comp}_{pk}(v_i, v_j, \sigma, \widetilde{E})$: Here σ and $\widetilde{E}$ are state variables maintained by TSign.
   (a) If $(v_i, v_j) \notin \widetilde{E}$, output $\perp$ and abort.
   (b) $s \leftarrow \mathsf{Redact}_{pk}(\widetilde{E}, \sigma, \widetilde{E} - \{v_i, v_j\})$. Output s.

Note $\widetilde{E}$ can be generated from the graph G, which is public. So the only necessary state variable that Comp need access, is σ, which is the signature of the set $\widetilde{E}$ and of constant size.

Theorem 3. *$\mathcal{AOP}$-$\mathcal{DTS}$ is transitively unforgeable under adaptive chosen message attack, assuming $\mathcal{RSS}$ is unforgeable under adaptive chosen message attack.*

Theorem 4. *$\mathcal{AOP}$-$\mathcal{DTS}$ is transparent and perfect privacy preserving.*

5 Conclusion

In this paper, we gave the first directed transitive signature scheme $\mathcal{DTTS}$ on directed trees, which is inspired by the relationship between transitive signatures and redactable signatures. Unlike previous schemes, $\mathcal{DTTS}$ features with constant signature size and privacy preserving property. We also gave a directed transitive singature scheme $\mathcal{AOP}$-$\mathcal{DTS}$ on generic directed graph, in the simplified setting where composition operation Comp can access some state variable (of constant size) maintained by the signer TSign. We proved that both $\mathcal{DTTS}$ and $\mathcal{AOP}$-$\mathcal{DTS}$ are transitively unforgeable and privacy preserving under reasonable assumptions. In summary, we solved the open problem of directed transitive signature in different relaxed settings, although in general the directed transitive signature remains open problem.

References

ACdMT05. Giuseppe Ateniese, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik. Sanitizable Signatures. In *ESORICS*, pages 159–177, 2005.

BN02. Mihir Bellare and Gregory Neven. Transitive Signatures based on Factoring and RSA. In *ASIACRYPT '02: Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, pages 397–414, London, UK, 2002. Springer-Verlag.

BN05. Mihir Bellare and Gregory Neven. Transitive signatures: new schemes and proofs. *Information Theory, IEEE Transactions on*, 51(6):2133–2151, June 2005.

CLX09. Ee-Chien Chang, Chee Liang Lim, and Jia Xu. Short redactable signatures using random trees. In *CT-RSA*, pages 133–147, 2009.

GHR99. Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT*, pages 123++, 1999.

GMR88. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

Hoh03. Susan Rae Hohenberger. The cryptographic impact of groups with infeasible inversion. Master's thesis, MIT, 2003.

JMSW02. Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic Signature Schemes. In *CT-RSA '02: Proceedings of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology*, pages 244–262, London, UK, 2002. Springer-Verlag.

KT03. H. Kuwakado and H. Tanaka. Transitive signature scheme for directed trees. *IEICE Trans. Fundamentals*, 2003.

MR02. Silvio Micali and Ronald L. Rivest. Transitive Signature Schemes. In *CT-RSA '02: Proceedings of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology*, pages 236–243, London, UK, 2002. Springer-Verlag.

Nev08. Gregory Neven. Note: A simple transitive signature scheme for directed trees. *Theor. Comput. Sci.*, 396(1-3):277–282, 2008.

Riv00. Ronald Rivest. Two signature schemes, October 2000. Slides from talk given at Cambridge University.

SFSM05. Mahmoud Salmasizadeh Siamak Fayyaz Shahandashti and Javad Mohajeri. A Provably Secure Short Transitive Signature Scheme from Bilinear Group Pairs. *Security and Communication Networks*, 3352:60–76, 2005.

WCZ$^+$07. Licheng Wang, Zhenfu Cao, Shihui Zheng, Xiaofang Huang, and Yixian Yang. Transitive signatures from braid groups. In *INDOCRYPT*, pages 183–196, 2007.

Yi07. Xun Yi. Directed Transitive Signature Scheme. In *CT-RSA*, volume 4377 of *Lecture Notes in Computer Science*, pages 129–144. Springer Berlin / Heidelberg, 2007.

YTO04. X. Yi, C.H. Tan, and E. Okamoto. Security of Kuwakado-Tanaka Transitive Signature Scheme for Directed Trees. *IEICE Trans. on Fundamentals*, 2004.