To appear in Theoretical Computer Science.

A Simple Transitive Signature Scheme for Directed Trees

Gregory Neven

Department of Electrical Engineering, Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, B-3001 Heverlee-Leuven, Belgium
Gregory.Neven@esat.kuleuven.be
http://www.neven.org

Abstract

Transitive signatures allow a signer to authenticate edges in a graph in such a way that anyone, given the public key and two signatures on adjacent edges (i, j) and (j, k), can compute a third signature on edge (i, k). A number of schemes have been proposed for undirected graphs, but the case of directed graphs remains an open problem. At CT-RSA 2007, Yi presented a scheme for directed trees based on RSA and a standard signature scheme. We present a new, conceptually simple, and generic construction from standard signatures only. Apart from not relying on any RSA-related security assumptions, our scheme outperforms that of Yi in both computation time and (worst-case) signature length. Our results indicate that the setting envisaged by Yi is much simpler than the general one of directed transitive signatures, which remains an open problem.

Keywords: Cryptography, transitive signatures, provable security.

1 Introduction

Transitive signatures. Imagine a signer who wants to authenticate a dynamically growing graph by issuing signatures on edges, one edge at a time. When a third party wants to show the existence of a path between two nodes in the graph thus constructed by the signer, he could simply show the individual signatures on all the edges in the path. However, it would be nice if the third party could somehow "compress" the signatures into a single one for the direct edge between the endpoints, thereby possibly saving on bandwidth and computation time.

This is exactly the problem setting of *transitive signature schemes* as introduced by Micali and Rivest [MR02]. Possible applications of transitive signatures include administrative domains (where an edge between i and j indicates that i and j are in the same domain), military chains of command (where an edge from i to j indicates that i commands j), and secure routing [ACdMT05]. Apart from introducing the concept, Micali and Rivest [MR02] also presented the first transitive signature scheme based on discrete logarithms and RSA. Later, Bellare and Neven [BN02, BN05] presented more schemes based on factoring, discrete logarithms, and pairings.

All these schemes are for undirected graphs only however, and the problem of a non-trivial transitive signature scheme for the directed case has been open since the seminal work by Micali and Rivest. In fact, Hohenberger [Hoh03] even provided evidence that such schemes may be very hard to construct, because they would imply a new mathematical structure called Abelian trapdoor groups with infeasible inversion, of which currently no instances are known.

At CT-RSA 2007, Yi [Yi07] proposed a directed transitive scheme for the special case that the graph being authenticated is a tree. The security of the scheme is based on an RSA-related assumption and the security of an underlying standard signature scheme. The signatures are not constant in size, however: they grow linearly with the number of composed edges. Yi argues however that this growth is limited: the signature on a path of m edges in a tree of n nodes includes two standard signatures, two RSA group elements, and an $m \cdot \log(n \log n)$ bit integer.

Our contributions. In this paper, we present a conceptually simple and generic construction of a transitive signature scheme for directed trees from any standard signature scheme that is more efficient than that of Yi and that does not rely on any RSA-related assumptions. Signature verification in our scheme is computationally cheaper since it avoids all costs related to RSA exponentiations. Edge signatures in our scheme contain two standard signatures and, in the worst case that the tree is a linear chain, a bit string of $n \cdot \log n$ bits. This is two RSA group elements and $n \log \log n$ bits shorter than the worst-case signature size of Yi.

We do not think the main contribution of this paper lies in the scheme itself though, but rather in the point that the setting envisaged by Yi of directed trees with linear-size signatures is a much simpler one than the original one intended by Micali and Rivest of arbitrary directed graphs with constant-size signatures. In fact, our results imply that the former can be built from any one-way function [Rom90], while the latter have been shown to require a mathematical structure that is at least as strong as trapdoor permutations [Hoh03]. We do not think our results make any advances towards general directed transitive signature schemes, so this should still be considered an open problem.

Related work. Transitive signatures are a special instance of homomorphic signatures, first introduced in a series of talks by Rivest [Riv00] and formalized by Johnson et al. [JMSW02]. Other instances of homomorphic signatures include prefix aggregation signatures [CRR02], redactable signatures [JMSW02], set-based signatures [HM02], and verifiably encrypted signatures [BGLS03].

2 Definitions

Notation. If $k \in \mathbb{N}$ is a natural number, then 1^k is the bit string consisting of k ones. Let ε be the empty string. If S is a set, then $|S|$ is the number of elements in S. If $i_1, \ldots, i_n \in \mathbb{N}$ then $L = i_1 \| \ldots \| i_n$ is the binary encoding of an ordered list of natural numbers such that $i_1, \ldots, i_n$ are efficiently and uniquely reconstructed from L.

If A is a deterministic algorithm, then $y \leftarrow \mathsf{A}(x)$ denotes that y is assigned the output of A on input x. If A is randomized, then $y \stackrel{\$}{\leftarrow} \mathsf{A}(x)$ denotes that y is assigned the output of A when run with a fresh random tape. We implicitly assume that the running time of all algorithms is polynomial in a global security parameter $k \in \mathbb{N}$. A function $\nu : \mathbb{N} \to [0, 1]$ is said to be negligible in k if for all $c \in \mathbb{N}$ there exists a $k_c \in \mathbb{N}$ such that $\nu(k) < k^{-c}$ for all $k > k_c$.

Standard signatures. A standard signature scheme is a triple of algorithms $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$. The signer generates a key pair consisting of a public key spk and a matching private key ssk via $(spk, ssk) \stackrel{\$}{\leftarrow} \mathsf{SKg}(1^k)$. The signer computes a signature for a message M via $\sigma \stackrel{\$}{\leftarrow} \mathsf{SSign}(ssk, M)$. The verifier can check the validity of σ by checking that $\mathsf{SVf}(spk, \sigma, M)$ returns 1; if the signature is invalid, it will return 0.

The most common security notion for signatures, existential unforgeability under chosen-message attack (uf-cma) [GMR88], is defined through the following game with an adversary A. The adversary is given a fresh public key spk and access to a signing oracle $\mathsf{SSign}(ssk, \cdot)$. Eventually, it outputs a message M and a forged signature σ. The advantage $\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{A}}(k)$ is defined as the probability that A wins the game, meaning that $\mathsf{SVf}(spk, \sigma, M) = 1$ and A never queried M to the signing oracle. The scheme $\mathcal{SS}$ is said to be uf-cma secure if the advantage of any polynomial-time adversary A is negligible.

TRANSITIVE SIGNATURES. A directed graph $G = (V, E)$ is defined by a set of nodes $V \subset \mathbb{N}$ and a set of edges $E \subseteq V \times V$. If $(i, j) \in E$ then we say that i is a parent of j and that j is a child of i. A directed tree is a directed graph with one root node $r \in V$ such that r has no parent and all other nodes in $V \setminus \{r\}$ have exactly one parent. The transitive closure of G is the graph $\tilde{G} = (V, \tilde{E})$ such that $(i, j) \in \tilde{E}$ if E contains a a directed path from i to j.

A transitive signature scheme is a tuple of four polynomial-time algorithms $\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{Comp}, \mathsf{TVf})$. The signer generates a key pair via $(tpk, tsk) \xleftarrow{\$} \mathsf{TKg}(1^k)$, and authenticates an edge between nodes $i, j \in \mathbb{N}$ by issuing a signature $\tau \xleftarrow{\$} \mathsf{TSign}(tsk, i, j)$. The signing algorithm TSign may keep state between invocations. Given two signatures τ_1 and τ_2 on adjacent edges (i, j) and (j, k), anyone can compute a composed signature on edge (i, k) directly as $\tau_3 \xleftarrow{\$} \mathsf{Comp}(tpk, \tau_1, \tau_2)$. The validity of a (possibly composed) signature τ for edge (i, j) can be checked by testing whether $\mathsf{TVf}(tpk, \tau, i, j)$ returns 1.

Security of a transitive signature scheme is defined as follows. The adversary A is given as input a fresh public key tpk, and is given access to a signing oracle $\mathsf{TSign}(tsk, \cdot, \cdot)$ from which it can obtain signatures on directed edges of its choice. Eventually A outputs an edge (i, j) and a forged signature τ. Let $G = (V, E)$ be the graph defined by the signature queries of A, and let $\tilde{G} = (V, \tilde{E})$ be its transitive closure. The adversary wins the game if $\mathsf{TVf}(tpk, \tau, i, j) = 1$ and $(i, j) \notin \tilde{E}$. The advantage $\mathbf{Adv}^{\text{tuf-cma}}_{\mathcal{TS},\mathsf{A}}(k)$ is the probability that A wins this game. The scheme is said to be transitively unforgeable under chosen-message attack (tuf-cma) if this advantage is negligible for all polynomial-time A.

In a transitive signature scheme for directed trees, the adversary is limited to signature queries that preserve the tree structure of G.

3 Yi's Construction

We first recall that a trivial construction of directed transitive signatures exists from any standard signature scheme [MR02]. Namely, a basic transitive signature on an edge (i, j) is simply a standard signature on (i, j), and composition is done by concatenating signatures for the edges in the path. This scheme is usually excluded because of its growth in signature length (linear in the number of edges in the path) and because it reveals the creation history of the signature.

The scheme by Yi [Yi07] also has linear signature length, but its growth rate is slower than for the trivial scheme. The basic signature for an edge (i, j) is a tuple $(L_i, \sigma_i, L_j, \sigma_j, \delta_{i,j})$, where $L_i, L_j \in \mathbb{Z}_N^*$ for an RSA modulus $N = pq$, where σ_i and σ_j are standard signatures on $i \| L_i$ and $j \| L_j$, respectively, and where $\delta_{i,j}$ is a small prime number such that $L_i^{\delta_{i,j}} \equiv L_j \bmod N$. Composition of two signatures $(L_i, \sigma_i, L_j, \sigma_j, \delta_{i,j})$ and $(L_j, \sigma_j, L_k, \sigma_k, \delta_{j,k})$ is done by computing $\delta_{i,k} \leftarrow \delta_{i,j} \cdot \delta_{j,k}$ and returning the tuple $(L_i, \sigma_i, L_k, \sigma_k, \delta_{i,k})$. Note that the composed signature contains only two standard signatures, but that the size of $\delta_{i,k}$ increases with each composition. This increase can be limited by using in a tree of n nodes the smallest n prime numbers as values for $\delta_{i,j}$. The worst case in terms of signature length is if the tree is one long chain of n

edges and we want to sign the endpoints of this chain. The signature will contains two elements of $\mathbb{Z}_N^*$, two standard signatures, and one integer of approximate length $n \cdot \log(n \log n)$.

Contrarily to what is claimed in [Yi07], a composed signature does not hide its creation history. Namely, the factorization of $\delta_{i,k}$ (which is easy to compute since it is a product of small primes) reveals the values of $\delta_{i,j}$ and $\delta_{j,k}$ corresponding to the original edges (i, j) and (j, k). Since in a directed tree there is at most one path from one node to another however, one could argue that history-independence is not a very important feature here.

4 Our Construction

THE SCHEME. The trivial scheme mentioned in the previous section has the disadvantage of containing a linear number of standard signatures in a single transitive signature. If we only consider the special case of a directed tree that is constructed in a top-down fashion (meaning, the first node of the first signature issued is the root of the tree, and no incoming edges into this node are created later on), then there exists a fairly simple scheme that contains only a single standard signature. Namely, let the transitive signature of an edge (i, j) be a standard signature on the path from the root down to node j. Verification involves checking the standard signature and checking that i occurs in the path. For a tree of depth d with up to n nodes, the description of the path takes up to $d \log n$ bits. So in the worst case that the tree is a chain of n nodes, the length of a signature is $n \log n$ bits plus one standard signature. This is shorter signatures than the signatures produced by Yi's scheme, but has the disadvantage that the root of the tree cannot change over time. We now present an extension of this scheme that does allow the root to change, at the price of doubling the worst-case signature length.

To any standard signature scheme $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$, we associate the following transitive signature scheme for directed trees $\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{Comp}, \mathsf{TVf})$:

- $\mathsf{TKg}(1^k)$: Identical to SKg, meaning that it generates a standard key pair $(spk, ssk) \xleftarrow{\$} \mathsf{SKg}(1^k)$ and sets $tpk \leftarrow spk$, $tsk \leftarrow ssk$.

- $\mathsf{TSign}(tsk, i, j)$: The signing algorithm maintains a state consisting of the root node r, the current tree $G = (V, E)$, and two tables $up[\cdot]$ and $down[\cdot]$. The signer modifies the current state distinguishing between the following cases:

  1. $V = \emptyset$:

     $$r \leftarrow i \; ; \; V \leftarrow V \cup \{i, j\} \; ; \; E \leftarrow E \cup \{(i, j)\}$$
     $$up[i] = down[i] = down[j] \leftarrow \varepsilon \; ; \; up[j] \leftarrow i$$

  2. $i \in V$ and $j \notin V$:

     $$V \leftarrow V \cup \{j\} \; ; \; E \leftarrow E \cup \{(i, j)\}$$
     $$up[j] \leftarrow up[i] \| i \; ; \; down[j] \leftarrow \varepsilon$$

  3. $i \notin V$ and $j = r$:

     $$r \leftarrow i \; ; \; V \leftarrow V \cup \{i\} \; ; \; E \leftarrow E \cup \{(i, j)\}$$
     $$up[i] \leftarrow \varepsilon \; ; \; down[i] \leftarrow j \| down[j]$$

  In all other cases the signer rejects because the query does not preserve the tree structure of the graph. The signer sets $C_i \leftarrow (i, down[i])$ and $C_j \leftarrow (j, up[j])$, and computes two standard signatures $\sigma_i \xleftarrow{\$} \mathsf{SSign}(tsk, C_i)$ and $\sigma_j \xleftarrow{\$} \mathsf{SSign}(tsk, C_j)$. The transitive signature on edge (i, j) is the tuple $\tau \leftarrow (C_i, \sigma_i, C_j, \sigma_j)$. We note that as an efficiency improvement, if one of $down[i]$ or $up[j]$ is empty, then (C_i, σ_i) or (C_j, σ_j) can be dropped from the transitive signature entirely.

- $\mathsf{Comp}(tpk, \tau_1, \tau_2)$: Parse τ_1 as $(C_i, \sigma_i, C_j, \sigma_j)$ and parse τ_2 as $(C_{j'}, \sigma'_{j'}, C_k, \sigma_k)$. If $j \neq j'$ or $\mathsf{SVf}(tpk, \tau_1, i, j) \neq 1$ or $\mathsf{SVf}(tpk, \tau_2, j, k) \neq 1$ then reject. Otherwise, return $\tau \leftarrow (C_i, \sigma_i, C_k, \sigma_k)$ as the composed signature.

- $\mathsf{TVf}(tpk, \tau)$: Parse τ as $(C_i, \sigma_i, C_j, \sigma_j)$, parse C_i as $(i, down)$, and parse C_j as (j, up). If $\mathsf{TVf}(tpk, C_i, \sigma_i) = 0$ or $\mathsf{TVf}(tpk, C_j, \sigma_j) = 0$ then return 0. If j occurs in $down$, if i occurs in up, or if there exists k that occurs in both $down$ and up, then return 1; else return 0.

Correctness. Correctness requires that legitimate signatures verify correctly, meaning that the verification algorithm returns 1 for signatures that were either obtained from the signer himself or through composition of legitimate signatures. Bellare and Neven [BN02] observed that subtleties arise in the formalization of this definition due to the statefulness of the signing algorithm; we use here the adaptation to directed trees from [Yi07]. Rather than recalling the formal definition and giving a full proof here, we provide some intuition into why correctness is satisfied. Let (i^*, j^*) be the first edge signature issued by the signer. Then one can see from the construction that all descendants j of i^* have $up[j]$ describing the path from i^* to j; that all ancestors i of i^* have $down[i]$ describing the path from i to i^*; and that all nodes j that are neither descendants nor ancestors of i^* have $up[j]$ describing the path from the closest common ancestor of i^* and j to j. To show the correctness of the verification algorithm, we distinguish between the following cases:

- Both i and j are descendants of i^*. In this case the table entry $up[j]$ describes the path from i^* down to j, which must include i if there exists a path from i to j.

- Both i and j are ancestors of i^*. In this case $down[i]$ describes the path from i down to i^*, which must contain j if there exists a path from i to j.

- Node i is an ancestor of i^* and j is a descendant of i^*. In this case $down[i]$ and $up[j]$ have one element in common.

- Node i is an ancestor of i^* and j is neither an ancestor nor a descendant of i^*. Let k be the closest common ancestor of j and i^*. The entry $up[j]$ contains the path from k down to i. If there is a path from i to j, then i is also an ancestor of k and k occurs in $down[i]$. The lists $down[i]$ and $up[j]$ therefore have one element in common, namely k.

- Neither i nor j are ancestors or descendants of i^*. Let k be the closest common ancestor of i^* and i, and let k' be that of i^* and j. If there is a path from i to j, then it must hold that $k = k'$. Therefore i is on the path from k to j, so i occurs in $up[j]$.

From the above argumentation one can easily see that the verification algorithm returns 1 for all legitimate signatures as required.

Security. The following relates the security of our transitive signature scheme to that of the underlying standard signature scheme.

Theorem 4.1 *Let $\mathcal{TS}$ be the transitive signature scheme for directed trees associated to a standard signature scheme $\mathcal{SS}$ as described above. If $\mathcal{SS}$ is* uf-cma *secure, then $\mathcal{TS}$ is* tuf-cma *secure.*

Proof: Given an adversary A against the transitive signature scheme $\mathcal{TS}$, consider the following adversary B against $\mathcal{SS}$. On input spk, B runs A on input $tpk = spk$. If A queries for the signature on an edge (i, j), then B maintains state as the real TSign algorithm would, but uses its $\mathsf{SSign}(ssk, \cdot)$ oracle to generate the standard signatures.

When A outputs its forgery $\tau = (C_i, \sigma_i, C_j, \sigma_j)$ on edge (i, j), we distinguish between the following cases. If either i or j are new nodes that did not yet occur in V, then either C_i or C_j was not signed before, so B can output one of them as its own forgery. If both i and j are existing nodes in the graph and $\mathsf{TVf}(tpk, \tau, i, j) = 1$ even though no path exists from i to j, then B can extract a forgery for the standard signature scheme as follows. It first parses C_i as $(i, down)$ and C_j as (j, up). If j occurs in $down$, then certainly C_i was never signed before, so B outputs C_i, σ_i as its forgery. Likewise, if i occurs up, then C_j was never signed before and B outputs C_j, σ_j as its forgery. Finally, if there exists a k that occurs in both $down$ and up, then at least one of C_i or C_j was never signed before, so B can output the appropriate one as its own forgery. We have that

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{B}}(k) \;\geq\; \mathbf{Adv}^{\text{tuf-cma}}_{\mathcal{TS},\mathsf{A}}(k) \;,$$

from which the theorem follows. ▮

Efficiency. In terms of computation our scheme is strictly cheaper than that of Yi because it avoids all costs related to RSA operations. In particular, verification only takes two standard signature verifications, while Yi's scheme additionally requires a possibly expensive RSA exponentiation because the exponent $\delta_{i,j}$ grows linearly with the path length and cannot be reduced modulo $\varphi(N)$ since this value is unknown to the verifier.

As for signature length, we already mentioned that in a tree of n nodes Yi's scheme has a worst-case signature length of two elements of $\mathbb{Z}_N^*$, two standard signatures, and one integer of up to $n \cdot \log(n \log n)$ bits. The worst-case length for our scheme is two standard signatures and n integers of $\log n$ bits each, which is two RSA elements and $n \cdot \log \log n$ bits shorter than for Yi's scheme.

The average signature length is harder to compare, if only because it is not clear how one would define an "average" tree. It is true that there are situations in which Yi's signatures are shorter than ours. Namely, in our scheme the signature on an edge (i, j) always contains the entire path from i^* to j. If j is only one hop away from i but far away from i^*, then this seems an overkill, and in fact in this case Yi's signature may be shorter than ours. On the other hand, our scheme saves the two RSA group elements from Yi's signatures. There is a break-even tree depth below which our signatures are always shorter, and above which some edges in the tree may have shorter signatures when using Yi's scheme. Some values of this break-even depth as a function of the number of nodes n are given in the table below for RSA moduli of 1024 and 2048 bits.

n	100	1000	10^6	10^9	10^{12}
d_{1024}	154	103	52	34	26
d_{2048}	308	206	103	69	52

We stress that Yi's signatures are certainly not guaranteed to be shorter for trees with higher depth than mentioned in the table above. Also, disregarding efficiency considerations, our scheme has the advantage of not relying on any other assumptions than the security of the underlying standard signature scheme.

5 Conclusion

We proposed a transitive signature scheme for directed trees that is conceptually simpler and more efficient than that of Yi [Yi07], and that does not rely on any other assumptions than the security of a standard signature scheme. Our scheme demonstrates that the setting envisaged by Yi of directed trees with linear-length signatures is considerably easier than the open problem mentioned by Micali and Rivest [MR02], namely that of constant-size transitive signatures for arbitrary directed graphs. The latter therefore remains an interesting open problem.

Acknowledgements

Gregory Neven is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO), and was supported in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government, and by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy).

References

[ACdMT05] Giuseppe Ateniese, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik. Sanitizable signatures. In *10th European Symposium on Research in Computer Security – ESORICS 2005*, volume 3679 of *Lecture Notes in Computer Science*, pages 159–177. Springer-Verlag, Berlin, Germany, 2005. (Cited on page 1.)

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany. (Cited on page 2.)

[BN02] Mihir Bellare and Gregory Neven. Transitive signatures based on factoring and RSA. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 397–414, Queenstown, New Zealand, December 1–5, 2002. Springer-Verlag, Berlin, Germany. (Cited on page 1, 5.)

[BN05] Mihir Bellare and Gregory Neven. Transitive signatures: new schemes and proofs. *IEEE Transactions on Information Theory*, 51(6):2133–2151, 2005. (Cited on page 1.)

[CRR02] Suresh Chari, Tal Rabin, and Ronald Rivest. An efficient signature scheme for route aggregation. Unpublished manuscript, 2002. (Cited on page 2.)

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988. (Cited on page 3.)

[HM02] Alejandro Hevia and Daniele Micciancio. The provable security of graph-based one-time signatures and extensions to algebraic signature schemes. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes* *in Computer Science*, pages 379–396, Queenstown, New Zealand, December 1–5, 2002. Springer-Verlag, Berlin, Germany. (Cited on page 2.)

[Hoh03] Susan Hohenberger. The cryptographic impact of groups with infeasible inversion. Master's thesis, Massachusetts Institute of Technology, 2003. (Cited on page 2.)

[JMSW02] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In Bart Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 244–262, San Jose, CA, USA, February 18–22, 2002. Springer-Verlag, Berlin, Germany. (Cited on page 2.)

[MR02] Silvio Micali and Ronald L. Rivest. Transitive signature schemes. In Bart Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 236–243, San Jose, CA, USA, February 18–22, 2002. Springer-Verlag, Berlin, Germany. (Cited on page 1, 3, 7.)

[Riv00] Ronald Rivest. Two signature schemes. Slides from talk given at Cambridge University, October 17, 2000., 2000. (Cited on page 2.)

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing*, pages 387–394, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press. (Cited on page 2.)

[Yi07] Xun Yi. Directed transitive signature scheme. In Masayuki Abe, editor, *Topics in Cryptology – CT-RSA 2007*, volume 4377 of *Lecture Notes in Computer Science*, pages 129–144, San Francisco, CA, USA, February 5–9, 2007. Springer-Verlag, Berlin, Germany. (Cited on page 2, 3, 4, 5, 7.)