# Optimal Tree-based One-time Digital Signature Schemes

Daniel Bleichenbacher and Ueli M. Maurer

Institute for Theoretical Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland
{bleichen,maurer}@inf.ethz.ch,

**Abstract.** A minimal cutset of a tree directed from the leaves to the root is a minimal set of vertices such that every path from a leaf to the root meets at least one of these vertices. An order relation on the set of minmal cutsets can be defined: $U \leq V$ if and only if every vertex of $U$ is on the path from some vertex in $V$ to the root. Motivated by the design of efficient cryptographic digital signature schemes, the problem of constructing trees with a large number of pairwise incomparable minimal cutsets or, equivalently, with a large antichain in the poset of minimal cutsets, is considered.

**Keywords.** Cryptography, digital signature schemes, trees, partially ordered sets.

## 1 Introduction

We consider trees directed from the leaves to the root where every vertex has at most two predecessors. In this paper, a cutset of such a tree $\mathcal{T}$ is defined as a set of vertices which contains at least one vertex of every path from a leaf to the root. A cutset is minimal when it contains exactly one vertex of every such path[1]. An order relation $\leq$ on the set of minimal cutsets can be defined as follows: we have $U \leq V$ for two minimal cutsets $U$ and $V$ if and only if every path from a vertex in $V$ contains a vertex of $U$ or, equivalently, if and only if $U$ is a cutset of the subtree of $\mathcal{T}$ obtained by pruning all branches stemming from a vertex in $V$.

This order relation defines a partially ordered set (poset) of minimal cutsets called the associated poset of the tree. For reasons motivated by a cryptographic application discussed in Section 2, we are interested in finding small trees with a large associated poset and, more importantly, trees whose associated poset contains a large antichain. More specifically, this paper investigates the problems of finding, for a given number $n$ of vertices, the tree with the largest associated poset and the tree with the largest antichain in the associated poset. The maximal achievable size of such a poset is denoted by $\nu(n)$ and the maximal achievable size of such an antichain is denoted by $\mu(n)$. Several results on these two functions are proved.

[1] Note that when $\mathcal{T}$ is interpreted as the graph of the transitive reduction of a poset, then our definition of a cutset corresponds to the standard definition of a cutset of a poset. However, in order to avoid any confusion, note that the associated poset of $\mathcal{T}$ defined in this paper is a completely different poset.

The outline of the paper is as follows. Section 2 provides the cryptographic motivation for considering the described problems, Section 3 summarizes the definitions, and in Section 4 the optimality results are derived. Numerical tables for $\nu(n)$ and $\mu(n)$ are given in the Appendix.

## 2 Motivation

Digital signature schemes are one of the most fundamental cryptographic mechanisms. Such a scheme allows a user, who has previously generated a secret key and the corresponding public key (which is made public), to generate a digital signature for a message. While anybody (e.g. a judge) knowing the user's public key can verify signatures, forging a user's signature, i.e., generating signatures for messages not previously signed by this user but verifyable with his public key, is computationally infeasible without knowledge of the secret key. Digital signature schemes are essential in many applications like document authentication in EDI or for public-key certification used to establish the emerging international public-key infrastructure.

One can distinguish between two types of digital signature schemes. The first type of scheme was proposed by Lamport [4] and generalized in [5], [6], [3], [8] and [1]. Once it is set up, it can only be used for signing a predetermined number of messages[2] from a certain message space. The second type of schemes, the first realization of which was the well-known RSA system [7], can be used an unlimited number of times.

A motivations for investigating and possibly using the first type of schemes, despite their limited number of uses, is the potentially higher efficiency compared to conventional schemes like RSA [7]. Another motivation is that these schemes can be based on an arbitrary cryptographic one-way function[3] (OWF), a very general cryptographic primitive.

The general idea of a one-time signature scheme is that the secret key is used as the input to a sequence of OWF evaluations which results in a sequence of intermediate results and finally in the public key. The one-wayness of the functions implies that it is infeasible to compute the secret key, or any intermediate result of the computation, from the public key.

[2] In this paper we only consider schemes for signing a single message, but these results can easily be generalized to obtain schemes that allow to sign a fixed number of messages [1].

[3] A one-way function $f$ is a function that is easy to compute but computationally infeasible to invert, for suitable definitions of "easy" and "infeasible". It is not difficult to define a function that appears to be one-way even to an expert in cryptography. However, not even the existence of one-way functions, for a suitable definition, has been proved. To be secure in the context of this paper, one-way functions with certain very special properties should be avoided. For instance, a one-way function $f(x, y)$ of two arguments should satisfy $f(x, y) = f(y, x)$ for $x \neq y$ only with negligible probability. Security proofs (based on the sole assumption that a function $f$ is one-way) are outside the scope of this paper, and hence so is the exact characterization of the properties required for a one-way function. The conjecture that a proposed function is one-way is virtually, but not exactly equivalent to the conjecture that it leads to a secure signature scheme.

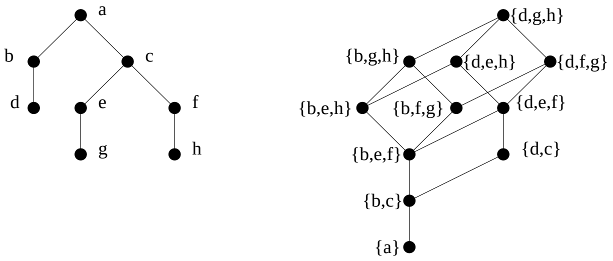


**Fig. 1.** A toy example of a tree $\mathcal{T}$ and its associated poset $(\mathcal{T}^*, \leq)$. The set $\{\{b, g, h\}, \{d, e, h\}, \{d, f, g\}\}$ is an antichain of maximal size 3, which is hence the width $w(\mathcal{T}^*)$. Another such antichain is $\{\{b, e, h\}, \{b, f, g\}, \{d, e, f\}\}$.

A signature for a given message consists of a subset of the intermediate results of this computation, where the message to be signed determines which particular subset is revealed as the corresponding signature. There exist two important requirements on these signatures. First, every signature must be verifyable, i.e., the public key must be computable from it. Second, in order to prevent forgery of signatures, the set of signatures (for the messages in the message space) must be compatible in the sense that no signature can be computed from the signature for a different message, without inverting a one-way function.

Let $B$ be a suitable large set (e.g., the set of 64, 96 or 128-bit strings), a subset of which is the range of the OWFs. In a reasonable implementation [1], the input to each OWF evaluation consists of one or two elements of $B$, the secret key consists of one or a list of elements of $B$, and the public key consists of only one element of $B$.

The structure of the computation leading from the secret key components to the public key can be represented as a directed acyclic graph $\mathcal{G} = (V, E)$ with vertex set $V$ and edge set $E$, where the vertices correspond to the secret key, the intermediate results, and the public key, and where a directed edge $(v_i, v_j)$ in $E$ indicates that $v_i$ is an input to the OWF computation resulting in $v_j$. The value corresponding to a vertex can only be computed when all inputs are known.

The graph $\mathcal{G}$ characterizing a one-time signature scheme is assumed to be known publicly, as is the mapping from messages to subsets of vertices (signature patterns). They can be used by all users. A user's signature for a given message consists of the values (for that user's secret key) corresponding to the vertices in the signature pattern for that message, when the computation according to $\mathcal{G}$ is performed for that user's secret key. A toy example of a signature scheme for a message space of size 3 is shown in Figure 1.

From an implementation point of view, an attractive class of graphs are trees with in-degree at most two [1]. In this paper we are interested primarily in the design of efficient signature schemes based on such trees, i.e. we consider the problem of maximizing the message space for a given number of vertices. While in a practical application the message space should be quite large, (e.g. the range of a cryptographically secure hash function, for instance the set of 128-bit strings), numerical values for some of our optimality results can feasibly be obtained only for small trees. However, large efficient schemes can be constructed from small trees, and hence optimality results for small trees lead to more efficient practical schemes.

## 3 Definitions and preliminaries

Throughout the paper, vertices and sets of vertices of a graph are denoted by small and capital letters, respectively, and graphs, posets, as well as sets of sets of vertices are denoted by calligraphic letters. The focus of this paper is on trees and therefore most of the definitions are given for trees only, although they can be generalized to arbitrary acyclic graphs. For the general definition and results for general graphs that are not trees we refer to [1] and [2].

Let $\mathcal{C}_m$ denote the directed graph consisting of a single path connecting $m$ vertices, i.e., a chain of length $m$. For two trees $\mathcal{T}_1$ and $\mathcal{T}_2$ let $[\mathcal{T}_1\mathcal{T}_2]$ denote the tree consisting of a new root and $\mathcal{T}_1$ and $\mathcal{T}_2$ as subtrees. The tree obtained from a tree $\mathcal{T}$ by introducing a new root and a single edge from the old to the new root, is denoted by $[\mathcal{T}]$.

A poset is defined as a set with an antisymmetric, transitive and reflexive order relation, denoted $\leq$. Two elements $x$ and $y$ of a poset $\mathcal{Z} = (Z, \leq)$ are *comparable* if and only if $x \leq y$ or $y \leq x$ and they are *incomparable* otherwise. A subset $U \subseteq Z$ is called a *chain* if every pair of elements of $U$ is comparable, and it is called an *antichain* if every pair of elements of $U$ is incomparable. The *width* of a poset $\mathcal{Z}$, denoted $w(\mathcal{Z})$, is the maximal cardinality of an antichain.

We now define a *one-time signature scheme* based on a tree. Let $\mathcal{T}$ be a tree where $T$ is the set of vertices, $L \subset T$ is the set of leaves and $p \in T$ is the root. The edges of the tree are directed from the leaves to the root. In the context of signature schemes, $L$ and $p$ correspond to the secret key and the public key, respectively. A *cutset* of $\mathcal{T}$ is a set of vertices which contains at least one vertex of every path from a leaf to the root, and a cutset is *minimal* when it contains exactly one vertex of every such path.

The set of minimal cutsets is denoted by $\mathcal{T}^*$. An order relation $\leq$ on $\mathcal{T}^*$ can be defined as described in the introduction. This order relation defines a partially ordered set (poset) of minimal cutsets called the *associated poset* of the tree and denoted by $(\mathcal{T}^*, \leq)$. For the sake of simplicity, the width of this poset will often be denoted by $w(\mathcal{T}^*)$ instead of the more precise notation $w((\mathcal{T}^*, \leq))$. Figure 1 shows an example of a tree and its associated poset.

In our context, each vertex of the tree represents the evaluation of a cryptographic one-way function. Hence the value corresponding to a vertex (or simply the vertex) is computable if and only if all predecessors of this vertex are known. This naturally (and recursively) defines the set of vertices that are computable from a given set of vertices. Note that the root (i.e. the public key) is (efficiently) computable from every cutset. For two minimal cutsets $U$ and $V$, $U$ is computable from $V$ if and only if $U \leq V$, but $V$ is not feasibly computable from $U$ unless $U = V$. The term computable and the symbol $\leq$ are in the sequel used as synonyms.

Two minimal cutsets are *compatible* if and only if they are incomparable in the poset, i.e., if neither is computable from the other. For instance, in the example of Figure 1, the minimal cutset $\{b, c\}$ is computable from $\{b, e, f\}$ because the value of vertex $c$ can be obtained from the values of vertices $e$ and $f$. On the other hand, $\{b, e, f\}$ and $\{c, d\}$ are compatible. A set of minimal cutsets is compatible when they are pairwise compatible, i.e., if and only if it is an antichain in the associated poset of the tree. When a mapping from the message space to such an antichain is defined, the antichain can be used as a one-time signature scheme.

For a poset $\mathcal{Z} = (Z, \leq)$, a function $r : Z \to \mathbf{N}$ is called a *representation function* of $\mathcal{Z}$ if for all distinct $x, y \in Z$, $x \leq y$ implies $r(x) < r(y)$. Therefore $r(x) = r(y)$ implies that $x$ and $y$ are incomparable and hence for any representation function $r$ of the associated poset $(\mathcal{G}^*, \leq)$ of a given DAG $\mathcal{G}$ and for any integer $k$, the set

$$\{U \in \mathcal{G}^* : r(U) = k\}$$

is an antichain. A useful representation function for the associated poset of a tree can be defined as follows: For $U \in \mathcal{T}^*$ for a given tree $\mathcal{T}$, let $c_{\mathcal{T}}(U)$ be the number of vertices in $T$ computable from $U$ but not contained in $U$. It is easy to see that the function $c_{\mathcal{T}}$ is a representation function for the associated poset of $\mathcal{T}$. This result was proved in [2] for the case of general graphs for which it is less trivial. For the example of Figure 1, this representation function takes on the same value for cutsets depicted at the same level in the poset, ranging from 0 for $\{a\}$ to 5 for $\{d, g, h\}$.

## 4 Finding optimal trees

### 4.1 The associated poset of a tree

**Theorem 4.1** *The associated poset of a tree can be computed recursively by*

$$\begin{aligned} &(\mathcal{C}_n^*, \leq) \cong \mathcal{C}_n \\ \text{and} \qquad &([\mathcal{T}_1\mathcal{T}_2]^*, \leq) \cong ((\mathcal{T}_1^* \times \mathcal{T}_2^*) \cup \{x\}, \leq_T), \end{aligned}$$

*where $x$ is the root of $[\mathcal{T}_1\mathcal{T}_2]$, the order relation $\leq_T$ is defined by $\{x\} \leq U$ for all $U \in [\mathcal{T}_1\mathcal{T}_2]^*$ and by $(U, V) \leq_T (U', V')$ if and only if both $U \leq U'$ in $(\mathcal{T}_1^*, \leq)$ and $V \leq V'$ in $(\mathcal{T}_2^*, \leq)$.*

The proof is omitted because of space limitations.

### 4.2 Optimality criteria

The important parameters of a one-time signature scheme $\mathcal{A}$ for a tree $\mathcal{T}$ are the number $|T|$ of vertices (which is equal to the number of function evaluations required for computing the public key from the secret key[4], the number $|\mathcal{A}|$ of signatures (which is an upper bound on the size of the message space), and the maximal size of signatures, $\max_{U \in \mathcal{A}} |U|$.

[4] Here we have assumed, as would be the case in a reasonable implementation [1]), that a secret key consisting of the values at the leaves is generated from a single secret key by applying, for each leaf, an OWF to the secret key concatenated with a leaf index.

This motivates the following two problems. First, for a given tree $\mathcal{T}$ to find an antichain of maximal size in the associated poset. Second, for a given size of the message space (i.e., antichain) to find a tree with the minimal number of vertices allowing the construction of a one-time signature scheme. For reasons explained below we are also interested in the size of the associated poset.

**Definition 4.2** *For a given number $n$ of vertices, let $\nu(n)$ and $\mu(n)$ be the maximal achievable size of the associated poset, and the maximal achievable size of an antichain of the associated poset, respectively, of a tree with $n$ vertices with in-degree at most two.*

A simple relation between $\nu(n)$ and $\mu(n)$ is

$$\nu(n) \geq \mu(n) \geq \frac{\nu(n)}{n}. \tag{1}$$

The left inequality follows directly from the definition and the right inequality follows from the fact that the above defined representation function $c_{\mathcal{T}}$ for a tree $\mathcal{T}$ takes on at most $n$ different values. It follows from (1) that when one is interested only in the asymptotic behavior of $\mu(n)$ it suffices to investigate the asymptotic behavior of $\nu(n)$.

Let the function $\rho : \mathbf{Z}^2 \rightarrow \mathbf{Z}$ be defined recursively by $\rho(0, m) := m$ and $\rho(n+1, m) := \rho(n, m)^2 + 1$, and let $\gamma$ be the constant

$$\gamma = \lim_{n\rightarrow\infty} \frac{\log_2 \rho(n, 3)}{2^{n+2}} \approx 0.4161426.$$

Then we have

$$\lim_{n\rightarrow\infty} \frac{\log \nu(n)}{n} = \lim_{n\rightarrow\infty} \frac{\log \mu(n)}{n} = \gamma.$$

A weaker version of this result, where lim is replaced by lim sup, follows from Theorem 8 of [2]. The above stronger result can be proven by refining the arguments of the proof.

### 4.3 The maximal size of a poset for trees with $n$ vertices

The function $\nu(n)$ can be computed recursively as defined by the following theorem which also characterizes the trees with $n$ vertices and associated poset of maximal size. The values of $\nu(n)$ and $\mu(n)$ for small $n$ are summarized in the Appendix.

**Theorem 4.3** *For $n \leq 5$ the chain $\mathcal{C}_n$ of length $n$ is an optimal tree in the sense that $\nu(n) = |\mathcal{C}_n^*| = n$. For $n > 5$ all optimal trees are of the form $[T_1 T_2]$, where $T_1$ and $T_2$ are optimal trees. Hence no optimal tree can contain an edge from a vertex with in-degree 2 to a vertex with in-degree 1. Moreover, for $n > 5$, we have*

$$\nu(n) = 1 + \max_{1\leq i\leq n-2} \{\nu(i)\nu(n-i-1)\}.$$

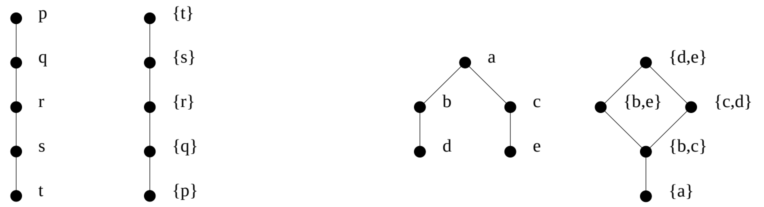


**Fig. 2.** This figure shows, from left to right, the graph $\mathcal{C}_5$, its associated poset $(\mathcal{C}_5^*, \leq)$, the graph $[\mathcal{C}_2\mathcal{C}_2]$ and its associated poset $([\mathcal{C}_2\mathcal{C}_2]^*, \leq)$.

*Proof.* By inspection of all trees with at most 6 vertices of in-degree at most two, it can be shown that $\mathcal{C}_n$ for $n \leq 5$ and $[\mathcal{C}_2\mathcal{C}_3]$ for $n = 6$ are optimal. It follows from $|[\mathcal{T}_1\mathcal{T}_2]^*| = 1 + |\mathcal{T}_1^*| \cdot |\mathcal{T}_2^*|$ that a tree of the form $[\mathcal{T}_1\mathcal{T}_2]$ is only optimal when $\mathcal{T}_1$ and $\mathcal{T}_2$ are optimal. It remains to prove that a tree with $n > 5$ vertices whose root has only one predecessor cannot be optimal. Such a tree has only one more minimal cutset than the tree resulting when the root is removed and hence has at most $\mu(n) + 1$ minimal cutsets. This also implies that $\mu(n + 1) \geq \mu(n) + 1$ which is used below. On the other hand, if $[\mathcal{T}_1\mathcal{T}_2]$ is an optimal tree with $n \geq 6$ vertices and $|\mathcal{T}_1| \leq |\mathcal{T}_2|$, which implies $|\mathcal{T}_2| > 1$, then the tree $[\mathcal{T}_1'\mathcal{T}_2]$, where $\mathcal{T}_1'$ results from $\mathcal{T}_1$ by adding a new root with an edge from the root of $\mathcal{T}_1$ to it, has at least $[\mathcal{T}_1\mathcal{T}_2]^* \geq |\mathcal{T}_1^* + 1| \cdot |\mathcal{T}_2^*| + 1 \geq \nu(n) + 2$. This completes the proof. □

### 4.4 Optimal trees

In contrast to $\nu(n)$, we do not know whether the function $\mu(n)$ can be computed recursively. Nevertheless finding a tree with $n$ vertices and $\mu(n)$ compatible minimal cutsets can be sped up considerably compared to an exhaustive search over all trees. Throughout this section, the term subtree is used only for subtrees whose leafs are also leafs in the tree. We will prove results of the following form: for any tree $\mathcal{T}$ containing a subtree $\mathcal{T}_1$, replacing $\mathcal{T}_1$ by a different subtree $\mathcal{T}_2$ with the same number of vertices results in a tree $\mathcal{T}'$ satisfying

$$w(\mathcal{T}'^*) \geq w(\mathcal{T}^*).$$

Hence no tree containing a subtree of the form $\mathcal{T}_1$ must be examined.
*Example:* Let $\mathcal{T}$ be any tree having $\mathcal{C}_5$ as a subtree and let $\mathcal{T}'$ be the tree resulting from $\mathcal{T}$ by replacing $\mathcal{C}_5$ by $[\mathcal{C}_2\mathcal{C}_2]$ (see Figure 2). Then $w(\mathcal{T}'^*) \geq w(\mathcal{T}^*)$ because for any antichain $\mathcal{A}$ in $(\mathcal{T}^*, \leq)$ there exists an antichain $\mathcal{A}'$ of minimal cutsets for $\mathcal{T}'$ with at least the same cardinality as $\mathcal{A}$. Let $\mathcal{A}' = \{\psi(U) : U \in \mathcal{A}\}$, where $\psi(U)$ is defined by

$$\psi(U) := \begin{cases} U \cup \{a\} \setminus \{p\} \text{ if } p \in U \\ U \cup \{b, c\} \setminus \{q\} \text{ if } q \in U \\ U \cup \{b, e\} \setminus \{r\} \text{ if } r \in U \\ U \cup \{c, d\} \setminus \{s\} \text{ if } s \in U \\ U \cup \{d, e\} \setminus \{t\} \text{ if } t \in U \\ U \text{ if } U \cap \{p, q, r, s, t\} = \emptyset \end{cases}$$

The fact that $\psi(U)$ is a minimal cutset for all $U \in \mathcal{A}$ and that $\mathcal{A}'$ is an antichain in $(\mathcal{T}'^*, \leq)$ follows from Theorem 4.6 proved below.

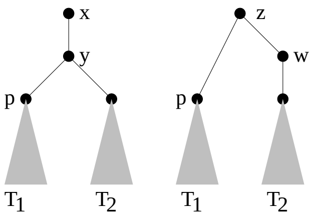
In the previous simple example, an explicit mapping from $\mathcal{A}$ to minimal cutsets in $\mathcal{T}'$ could be constructed which leads to an antichain of equal cardinality in $(\mathcal{T}'^*, \leq)$. The following lemmas and theorems are more general in that more than one minimal cutset in $\mathcal{A}'$ is constructed from a single minimal cutset in $\mathcal{A}$. For instance, assume in the above example that in $\mathcal{A}$ there are more minimal cutsets containing $r$ than minimal cutsets containing $s$. Then an antichain $\mathcal{A}''$ that is larger than $\mathcal{A}$ is obtained from $\mathcal{A}$ by using all $\psi(U)$ with $U \in \mathcal{A}$ and $s \notin U$ as well as all $U \cup \{c, d\} \setminus \{r\}$ with $U \in \mathcal{A}$ and $r \in U$. Thus two minimal cutsets of $\mathcal{A}''$ correspond to every minimal cutset of $\mathcal{A}$ containing $r$, and no minimal cutset of $\mathcal{A}''$ corresponds to a minimal cutset of $\mathcal{A}$ containing $s$.

**Lemma 1.** *Let $\mathcal{T}$ be a tree containing a subtree $[\mathcal{C}_1\mathcal{T}_1]$ and let $\mathcal{T}'$ be the tree resulting from $\mathcal{T}$ by replacing $[\mathcal{C}_1\mathcal{T}_1]$ by $[\mathcal{T}_1]$. Then the associated posets of $\mathcal{T}$ and $\mathcal{T}'$ are isomorphic.*

*Proof.* Let $v$ be the vertex of $\mathcal{C}_1$. Theorem 4.1 implies that $\phi : \mathcal{T}^* \to \mathcal{T}'^*$ defined by $\phi(U) = U \setminus \{v\}$ is an isomorphism.

**Lemma 2.** *Let $\mathcal{T}$ be a tree containing a subtree $[[\mathcal{T}_1\mathcal{T}_2]]$ and let $\mathcal{T}'$ be the tree resulting from $\mathcal{T}$ by replacing $[[\mathcal{T}_1\mathcal{T}_2]]$ by $[\mathcal{T}_1[\mathcal{T}_2]]$. Then the poset $(\mathcal{T}^*, \leq)$ is isomorphic to a subposet of $(\mathcal{T}'^*, \leq)$.*

*Proof.* Let the vertices of $[[\mathcal{T}_1\mathcal{T}_2]]$ and $[\mathcal{T}_1[\mathcal{T}_2]]$ be as shown in the following figure.



and let $\phi : \mathcal{T}_1^* \to \mathcal{T}_2^*$ be defined by $\phi(\{x\}) = \{z\}, \phi(\{y\}) = \{w\} \cup p$, and $\phi(U) = U$ otherwise. For all $U, V \in \mathcal{T}_1^*$ we have $U \leq V$ if and only if $\phi(U) \leq \phi(V)$. It follows that $(\mathcal{T}_1^*, \leq)$ is a subposet of $(\mathcal{T}_2^*, \leq)$. The lemma is now a consequence of Theorem 4.1.

**Theorem 4.4** *Let $\mathcal{T}$ and $\mathcal{T}'$ be trees such that $\mathcal{T}'$ can be obtained from $\mathcal{T}$ by replacing a subtree $\mathcal{T}_1$ by the subtree $\mathcal{T}_2$. Let $W$ be the set of vertices of $\mathcal{T}$ that are not part of $\mathcal{T}_1$ (or, equivalently, the vertices of $\mathcal{T}'$ that are not part of $\mathcal{T}_2$). Let $f : \mathcal{M} \to \mathcal{T}_1^*$ for $M \subseteq \mathcal{T}_2^*$ be a function which for all $U, V \in \mathcal{T}_2^*$ satisfies $U < V \Longrightarrow f(U) < f(V)$. For every antichain $\mathcal{A}$ in $(\mathcal{T}^*, \leq)$,*

$$\mathcal{A}' := \{X \cup U : U \in \mathcal{M} \text{ and } X \subseteq W \text{ and } X \cup f(U) \in \mathcal{A}\}$$

*is an antichain in $(\mathcal{T}'^*, \leq)$.*

*Proof.* First we note that all sets in $\mathcal{A}'$ are minimal cutsets. This follows immediately from that fact that the public key of $\mathcal{T}_2$ is computable from a minimal cutset in $\mathcal{A}'$ if and only if the public key of $\mathcal{T}_1$ is computable from the corresponding minimal cutset in $\mathcal{A}$.

Now assume that there exist two minimal cutsets $X \cup U$ and $Y \cup V$ in $\mathcal{A}'$ with $U, V \in \mathcal{T}_2$ and $X, Y \subseteq W$ satisfying $X \cup U \leq Y \cup V$ in $(\mathcal{T}'^*, \leq)$. It follows that $U \leq V$ in $({\mathcal{T}_2}^*, \leq)$ and $f(U) \leq f(V)$ in $(\mathcal{T}_1^*, \leq)$. This implies that $X \cup f(U) \leq Y \cup f(V)$ in $\mathcal{T}$. Because $\mathcal{A}$ is an antichain this inequality can only satisfied when $X \cup f(U) = Y \cup f(V)$. It follows from $U \leq V$ and the definition of $f$ that $f(U) = f(V)$ only if $U = V$. Hence $X \cup U = Y \cup V$.

**Corollary 4.5** *For the notation and variables defined in Theorem 4.4,*

$$\mathcal{A}'' := \mathcal{A}' \cup \{X \in \mathcal{A} : X \subseteq W\}$$

*is an antichain in* $(\mathcal{T}'^*, \leq)$.

*Proof.* In order to show that $\mathcal{A}''$ is an antichain we need to show only that any two minimal cutsets $X \subseteq W$ and $Y \cup U$ with $Y \subseteq W$ and $U \in \mathcal{T}_2^*$ are compatible. $Y \cup U$ is not computable from $X$, since no vertex of $U$ is computable from $X$. On the other hand, $Y \cup U \leq X$ in $(\mathcal{T}'^*, \leq)$ implies $Y \cup f(U) \leq X$ in $(\mathcal{T}^*, \leq)$, which contradicts the fact that $\mathcal{A}$ is an antichain.

A particular construction for applying Theorem 4.4 is provided by the following theorem. It is based on flows in the associated poset of a subtree to be replaced.

**Theorem 4.6** *Let $\mathcal{T}_1\mathcal{T}_2$ be trees and let $r_1, r_2$ be representation functions for $(\mathcal{T}_1^*, \leq)$ and $({\mathcal{T}_2}^*, \leq)$, respectively. For $U \in {\mathcal{T}_1}^*$, let $g(U)$ be defined by*

$$g(U) := |\{X \in \mathcal{T}_2^* : r_1(U) = r_2(X)\}|.$$

*If there exists a flow on $\mathcal{T}_1^*$ of value 1 from the top to the bottom such that the flow through each $U \in \mathcal{T}_1^*$ is a least $1/g(U)$, then replacing any subtree $\mathcal{T}_1$ by $\mathcal{T}_2$ will result in a tree whose associated poset has greater or equal width.*

*Proof.* Let $\mathcal{T}$ and $\mathcal{T}'$ be trees such that $\mathcal{T}'$ can be obtained from $\mathcal{T}$ by replacing a subtree $\mathcal{T}_1$ by $\mathcal{T}_2$ and let $\mathcal{A}$ be an antichain in $(\mathcal{T}^*, \leq)$. We will show that $w(\mathcal{T}'^*) \geq |\mathcal{A}|$. Let $W$ be the set of vertices contained in $\mathcal{T}$ but not contained in $\mathcal{T}_1$ and let $\mathcal{B} = \{U \in A : U \subseteq W\}$ be the set of minimal cutsets that have no vertex in common with $\mathcal{T}_1$. For $U \in \mathcal{T}_1^*$ define the cardinality of the set of minimal cutsets in $\mathcal{A}$ that contain $U$ by $h(U) := |\{X \in \mathcal{A} : U \subseteq X\}|$.

Assume that $\mathcal{K} = \{U_1, \ldots, U_n\}$ is a chain in $\mathcal{T}_1^*$. Let $R := \{r_1(U) : U \in \mathcal{K}\}$ and $\mathcal{M}_\mathcal{K} := \{X \in \mathcal{T}_2^* : r_2(X) \in R\}$. Now define $f_\mathcal{K} : \mathcal{M}_\mathcal{K} \to \mathcal{K}$ by $f_\mathcal{K}(X) = U$ if $r_2(X) = r_1(U)$. Note that $U_i \neq U_j$ implies $r_1(U_i) \neq r_1(U_j)$. This fact and the definitions above guarantee that $f_\mathcal{K}$ is well-defined. For all $X, Y \in \mathcal{T}_2^*$ where $X \leq Y$ we have $r_2(X) < r_2(Y)$ and $f_\mathcal{K}(X) \leq f_\mathcal{K}(Y)$ in $(\mathcal{T}_1^*, \leq)$. Thus $f_\mathcal{K}$ satisfies the conditions of Corollary 4.5. Now define $\mathcal{A}'_\mathcal{K}$ by

$$\mathcal{A}'_\mathcal{K} = \mathcal{B} \cup \{X \cup U : X \subseteq W \text{ and } U \in \mathcal{M}_\mathcal{K} \text{ and } X \cup f_\mathcal{K}(U) \in \mathcal{A}\}$$

| $r_1(U)$ | $1/g(U)$ |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 1/2 |
| 3 | 1/3 |
| 4 | 1/2 |
| 5 | 1 |

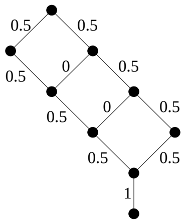


**Fig. 3.** A flow of value 1 on $[\mathcal{C}_2\mathcal{C}_4]^*$

It follows from Theorem 4.4 that $\mathcal{A}'_{\mathcal{K}}$ is an antichain in $\mathcal{T}'^*$. The cardinality of $\mathcal{A}'_{\mathcal{K}}$ can be computed by $|\mathcal{A}'_{\mathcal{K}}| = |\mathcal{B}| + \sum_{U \in \mathcal{K}} h(U)g(U)$ Therefore, for any chain $\mathcal{K}$ in $\mathcal{T}_1^*$, $w(\mathcal{T}'^*) \geq |\mathcal{B}| + \sum_{U \in \mathcal{K}} h(U)g(U)$.

Now we note that the flow on $\mathcal{T}_1^*$ can be represented as a sum of flows on chains $\mathcal{K}_1, \ldots, \mathcal{K}_m$ of values $\alpha_1, \ldots, \alpha_m$ which sum to one: $\sum_{i=1}^m \alpha_i = 1$. The condition that the flow on each minimal cutset $U \in \mathcal{T}_1^*$ is at least $1/g(U)$ is equivalent to the condition $\sum_{i: U \in \mathcal{K}_i} \alpha_i \geq 1/g(U)$. Thus we have

$$\begin{aligned} w(\mathcal{T}'^*) &= \sum_{i=1}^m \alpha_i w(\mathcal{T}'^*) \geq \sum_{i=1}^m \alpha_i \left( |\mathcal{B}| + \sum_{U \in \mathcal{K}_i} h(U)g(U) \right) \\ &= |\mathcal{B}| + \sum_{i=1}^m \alpha_i \sum_{U \in \mathcal{K}_i} h(U)g(U) = |\mathcal{B}| + \sum_{U \in \mathcal{T}_1^*} h(U)g(U) \sum_{i: U \in \mathcal{K}_i} \alpha_i \\ &\geq |\mathcal{B}| + \sum_{U \in \mathcal{T}_1^*} h(U)g(U) \frac{1}{g(U)} = |\mathcal{A}| \quad \square \end{aligned}$$

*Example:* Let $\mathcal{T}$ and $\mathcal{T}'$ be trees such that $\mathcal{T}'$ can be obtained from $\mathcal{T}$ by replacing a subtree $\mathcal{T}_1 = [\mathcal{C}_2\mathcal{C}_4]$ by the subtree $\mathcal{T}_2 = [\mathcal{C}_3\mathcal{C}_3]$. We will prove $w(\mathcal{T}'^*) \geq w(\mathcal{T}^*)$ by applying Theorem 4.6. For the choice $r_1 = c_{\mathcal{T}_1}$ and $r_2 = c_{\mathcal{T}_2}$ the required flow $1/g(U)$ through the vertices $U$ of $\mathcal{T}_1^*$ is shown in the table on the left side of Figure 3.

By using similar arguments many pairs of trees $(\mathcal{T}_1, \mathcal{T}_2)$ can be found such that replacing a subtree $\mathcal{T}_1$ by $\mathcal{T}_2$ in any tree $\mathcal{T}$ will result in a new tree $\mathcal{T}'$ whose poset has equal or greater width. The following table lists a few such pairs.

| size | $\mathcal{T}_1$ | $\mathcal{T}_2$ |
|---|---|---|
| 5 | $\mathcal{C}_5$ | $[\mathcal{C}_2\mathcal{C}_2]$ |
| 7 | $[\mathcal{C}_2\mathcal{C}_4]$ | $[\mathcal{C}_3\mathcal{C}_3]$ |
| 9 | $[\mathcal{C}_2[\mathcal{C}_2\mathcal{C}_3]]$ | $[\mathcal{C}_3[\mathcal{C}_2\mathcal{C}_2]]$ |
| 10 | $[\mathcal{C}_4[\mathcal{C}_2\mathcal{C}_2]]$ | $[\mathcal{C}_3[\mathcal{C}_2\mathcal{C}_3]]$ |
| 11 | $[\mathcal{C}_2[\mathcal{C}_3\mathcal{C}_4]]$<br>$[\mathcal{C}_4[\mathcal{C}_2\mathcal{C}_3]]$ | $[\mathcal{C}_3[\mathcal{C}_3\mathcal{C}_3]]$<br>$[\mathcal{C}_3[\mathcal{C}_3\mathcal{C}_3]]$ |

In all cases, $r_{\mathcal{T}_2}$ can be used as the representation function. The corresponding flows for the trees of size 9 and 10 in this table are shown in Figure 4.4.

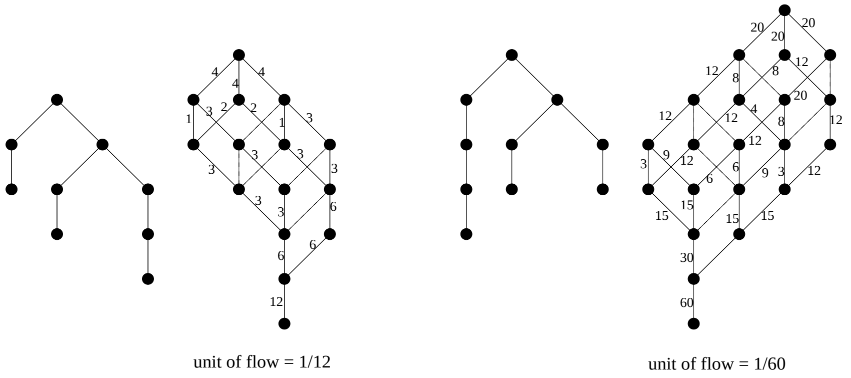


**Fig. 4.** Left: The tree $\mathcal{T}_1 = [\mathcal{C}_2[\mathcal{C}_2\mathcal{C}_3]]$ and its associated poset with a flow defined by Theorem 4.6 for proving that $\mathcal{T}_1$ can be replaced by $\mathcal{T}_2 = [\mathcal{C}_3[\mathcal{C}_2\mathcal{C}_2]]$. Right: The tree $\mathcal{T}_1 = [\mathcal{C}_4[\mathcal{C}_2\mathcal{C}_2]]$ and its associated poset with a flow defined by Theorem 4.6 for proving that $\mathcal{T}_1$ can be replaced by $\mathcal{T}_2 = [\mathcal{C}_3[\mathcal{C}_2\mathcal{C}_3]]$.

Some replacement rules cannot be proved by the flow argument of Theorem 4.6, and one must resort to the more general Theorem 4.4. Two further transformations are listed below.

| size | $\mathcal{T}_1$ | $\mathcal{T}_2$ |
|---|---|---|
| 11 | $[[\mathcal{C}_2\mathcal{C}_2][\mathcal{C}_2\mathcal{C}_2]]$<br>$[\mathcal{C}_2[\mathcal{C}_2[\mathcal{C}_2\mathcal{C}_2]]]$ | $[\mathcal{C}_3[\mathcal{C}_3\mathcal{C}_3]]$<br>$[\mathcal{C}_3[\mathcal{C}_3\mathcal{C}_3]]$ |

As a consequence of the above theorems, the search for a tree $\mathcal{T}$ with a given number $n$ of vertices that is optimal in the sense that $w(\mathcal{T}^*) = \mu(n)$, can be reduced dramatically by considering only trees whose subtrees belong to a small set of possible subtrees. In particular we have

**Corollary 4.7** *For every $n$ there exists a tree $\mathcal{T}$ with $n$ vertices and $w(\mathcal{T}^*) = \mu(n)$ such that every subtree of size $m \leq 11$ is contained in the following list:*

| size | subtrees | size | subtrees |
|---|---|---|---|
| 2 | $\mathcal{C}_2$ | 7 | $[\mathcal{C}_3\mathcal{C}_3]$ |
| 3 | $\mathcal{C}_3$ | 8 | $[\mathcal{C}_3\mathcal{C}_4]$, $[\mathcal{C}_2[\mathcal{C}_2\mathcal{C}_2]]$ |
| 4 | $\mathcal{C}_4$ | 9 | $[\mathcal{C}_4\mathcal{C}_4]$, $[\mathcal{C}_3[\mathcal{C}_2\mathcal{C}_2]]$ |
| 5 | $[\mathcal{C}_2\mathcal{C}_2]$ | 10 | $[\mathcal{C}_3[\mathcal{C}_2\mathcal{C}_3]]$, $[\mathcal{C}_2[\mathcal{C}_3\mathcal{C}_3]]$ |
| 6 | $[\mathcal{C}_2\mathcal{C}_3]$ | 11 | $[\mathcal{C}_3[\mathcal{C}_3\mathcal{C}_3]]$ |

*Proof.* This table consists of all trees with at most 11 vertices that contain no subtree appearing as $\mathcal{T}_1$ in the previous tables and no subtree of the form excluded by Lemmas 1 and 2. It remains to prove that an arbitrary sequential application of these replacement rules cannot be a cyclic process. This follows from the fact that, as can easily be verified, every replacement increases the cardinality of the poset except when $\mathcal{C}_5$ is replaced by $[\mathcal{C}_2\mathcal{C}_2]$. However, this latter replacement cannot lead to a cycle because the number of leaves is increased in this case.

## 5 Concluding Remarks

One can design highly efficient one-way functions if they are not required to have additional properties (like for instance a trapdoor). Therefore the proposed signature schemes are potentially more efficient than classical signature schemes like the RSA system. Furthermore, in contrast to specific assumptions like the difficulty of factoring large integers, they allow to base a system's security on very general cryptographic assumptions. Because the number of messages that can be signed is fixed when a public key is generated, these schemes have potential applications primarily in scenarios where only few messages need to be signed. However, this problem can be relaxed by signing a new public key as the last message in the life-time of a public key.

## 6 Appendix: Table of $\mu(n)$ and $\nu(n)$

| $n$ | $\mu(n)$ | $\nu(n)$ | $n$ | $\mu(n)$ | $\nu(n)$ | $n$ | $\mu(n)$ | $\nu(n)$ | $n$ | $\mu(n)$ | $\nu(n)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 9 | 4 | 17 | 17 | 29 | 171 | 25 | 246 | 1718 |
| 2 | 1 | 2 | 10 | 5 | 22 | 18 | 39 | 222 | 26 | 326 | 2228 |
| 3 | 1 | 3 | 11 | 7 | 31 | 19 | 53 | 311 | 27 | 448 | 3132 |
| 4 | 1 | 4 | 12 | 8 | 41 | 20 | 67 | 411 | 28 | 576 | 4142 |
| 5 | 2 | 5 | 13 | 11 | 53 | 21 | 85 | 534 | 29 | 732 | 5372 |
| 6 | 2 | 7 | 14 | 14 | 71 | 22 | 114 | 711 | 30 | 977 | 7172 |
| 7 | 3 | 10 | 15 | 19 | 101 | 23 | 156 | 1011 | | | |
| 8 | 3 | 13 | 16 | 23 | 131 | 24 | 195 | 1314 | | | |

## References

1. D. Bleichenbacher and U.M. Maurer, Directed acyclic graphs, one-way functions and digital signatures, *Advances in Cryptology - CRYPTO '94*, Y. Desmedt(Ed.), Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 839, pp. 75-82, 1994.
2. D. Bleichenbacher and U.M. Maurer, On the efficiency of one-time digital signatures, preprint.
3. S. Even, O. Goldreich and S. Micali, On-line/off-line digital signatures, *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science, vol. 435 (G. Brassard, ed.), Springer Verlag, 1990, pp. 263-275.
4. L. Lamport, Constructing digital signatures from a one-way function, Technical Report SRI Intl. CSL 98, 1979.
5. R. Merkle, A certified digital signature, *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science, vol. 435 (G. Brassard, ed.), Springer Verlag, 1990, pp. 218-238.
6. C. Meyer and S. Matyas, *Cryptography – a new dimension in computer data security*, John Wiley & Sons, Inc., 1982.
7. R.L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126, 1978.
8. S. Vaudenay, One-time identification with low memory, *Proc. of EUROCODE '92*, Lecture Notes in Computer Science, Springer Verlag. CISM Courses and Lectures, No. 339, International Centre for Mechanical Sciences, P. Camion, P. Charpin and S. Harari (eds.), Springer-Verlag, pp. 217–228.