Directed Acyclic Graphs, One-way Functions and Digital Signatures

(Extended Abstract)

Daniel Bleichenbacher and Ueli M. Maurer

Institute for Theoretical Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland
Email addresses: maurer@inf.ethz.ch, bleichen@inf.ethz.ch

Abstract. The goals of this paper are to formalize and investigate the general concept of a digital signature scheme, based on a general one-way function without trapdoor, for signing a predetermined number of messages. It generalizes and unifies previous work of Lamport, Winternitz, Merkle, Even et al. and Vaudenay. The structure of the computation yielding a public key from a secret key corresponds to a directed acyclic graph $\mathcal{G}$. A signature scheme for $\mathcal{G}$ can be defined as an antichain in the poset of minimal verifyable sets of vertices of $\mathcal{G}$ with the naturally defined computability relation as the order relation and where a set is verifyable if and only if the public key can be computed from the set. Several types of graphs are analyzed, results on the number of signatures of these schemes are presented (with and without restriction on the size of signatures), and several open research problems are proposed. In particular, a tree is shown which allows to sign 0.4162 bits per one-way function evaluation and it is proved that this is also an upper bound for all trees.

1. Introduction

Lamport [6] proposed a so-called one-time signature scheme based on a general one-way function (OWF), i.e., a function f that is easy to compute but computationally infeasible to invert, for suitable definitions of "easy" and "infeasible". Lamport's scheme for signing a single bit is set up by choosing as the secret key two strings x_0 and x_1 at random and revealing as the public key the pair $\langle f(x_0), f(x_1)\rangle$. The signature for bit b is x_b. For signing longer messages, several instances of this scheme can be used.

Motivated by Lamport's approach, many researchers have subsequently proposed more efficient one-time signature schemes, which are discussed in Section 2. The goals of this paper are to formalize the concept of a signature scheme based on any OWF for signing a predetermined number of messages, to discuss techniques for designing and analyzing such schemes and to present several results on the number and size of messages that can be signed with a given scheme. In contrast to Rompel's result [10] showing that a signature scheme can be obtained from any OWF, the emphasis of this paper is on efficiency and on a unified description of the general idea rather than on rigorously proving the security of schemes with respect to a certain intractability assumption.

Our motivations for considering the design of signature schemes based on OWFs are as follows. First, there is a severe limitation on the diversity of mathematical problems (such as factoring integers [9] or computing discrete logarithms in certain finite groups [11]) that can at present be used as the bases for a digital signature scheme. Therefore an alternative design approach with a much larger degree of freedom in choosing the underlying cryptographic function appears to be of interest. Second, for applications where few messages need to be signed, schemes based on an arbitrary one-way function have the potential of being computationally more efficient than presently-used number-theoretic schemes, but their disadvantage is that each public key can only be used for signing a predetermined number of messages. Second, even if these schemes turn out to be of limited interest as a normal digital signature scheme, they do have applications in various contexts such as on-line/off-line signatures [2]. Our third motivation is a theoretical one: the presented approach leads to interesting classes of challenging combinatorial problems that appear to be also of certain independent interest.

2. Review of previously proposed one-time signature schemes

The number (i.e., diversity) of messages that can be signed by the Lamport scheme with r public-key pairs is 2^r. Using the same secret key and public key, but allowing as signatures all subsets of cardinality r of the set of $2r$ public-key components, the number of messages can be improved to $\binom{2r}{r}$. These sets are compatible because computing a signature from a different signature requires the inversion of the OWF for at least one value. A well-known result by Sperner [12] states that $\binom{2r}{r}$ is the maximal number of compatible subsets for the Lamport scheme.

Note that the size of the secret key of such a scheme can be reduced significantly by generating all the secret-key components in a pseudo-random fashion from a single secret key S. Similarly, the public key can be reduced to a single value P by applying a one-way hash function to the list of public-key components.

A generalization of the Lamport scheme attributed by Merkle to Winternitz [7] is to apply the OWF to the secret key components iteratively a fixed number of times: the secret key consists of x_0 and x_1, the public key is $< f^t(x_0), f^t(x_1) >$ where f^t denotes the t-fold application of f, and the $t+1$ compatible signatures (for signing a messages that can take on at most $t+1$ different values) are the pairs $< f^{i-1}(x_0), f^{t-i+1}(x_1) >$ for $0 \leq i \leq t$. (It is assumed here that the range of f is a subset of the domain of f.) For $t = 1$ the Winternitz scheme coincides with the Lamport scheme for signing a single bit.

Meyer and Matyas [8] proposed as a further improvement to use more than two chains of function evaluations: they observed that $K!$ signatures can be obtained from a scheme with K chains of length K, by allowing as signatures all combinations of K vertices containing one node in each chain and one node at each level. This scheme was generalized by Even, Goldreich and Micali [2] and later by Vaudenay [13] who suggested that a large set of compatible signatures can be obtained by allowing all signatures consisting of one node in each chain such that the total sum of the levels of these vertices (within their chains) is constant. One of the results of this paper is the analysis of this scheme which we will refer to as the "rake scheme". In particular, we will show that the described strategy yields the maximal number of signatures.

The described schemes can only be used to sign a single message. Merkle [7] proposed the so-called tree-authentication scheme for signing several messages consecutively with a single public key P. The basic idea is that P allows to selectively authenticate any one of a list $p_1, \ldots, p_{2^d}$ of 2^d public keys, each one of which, once authenticated, can be used to sign a single message. The term authenticate (as opposed to sign) is used here because the public keys to be authenticated must be known at the time of setting up the scheme. The signature for the ith message hence consists of three parts: the corresponding public key p_i, a string of d values authenticating p_i from P, and a signature authenticating the message from p_i. Merkle's authentication tree will be discussed briefly in Section 6 where it is shown that this scheme for signing several messages is not optimal in general.

3. Preliminaries

In this 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,

This section summarizes some well-known definitions and results on partially ordered sets (poset). 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. An element x *covers* a distinct element y, denoted $y \prec x$, if and only if $y \leq x$ and $y \leq z \leq x$ implies $z = y$ or $z = x$. 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. A chain $\mathcal{C}$ is *maximal* if there exists no chain $\mathcal{C}' \neq \mathcal{C}$ with $\mathcal{C} \subset \mathcal{C}'$, and it is called *maximal-sized* if $|\mathcal{C}'| \leq |\mathcal{C}|$ for all chains $\mathcal{C}'$ of $\mathcal{Z}$. An antichain A is *maximal* if there exists no antichain $A' \neq A$ such that $A \subset A'$ and it is called *maximal-sized* if $|A'| \leq |A|$ for all antichains A' of $\mathcal{Z}$. The *width* of a poset $\mathcal{Z}$, denoted $w(\mathcal{Z})$, is the cardinality of a maximal-sized antichains. A subset $C \subseteq Z$ is called a *cutset* if it meets every maximal chain of $\mathcal{Z}$. A cutset C is *minimal* if no subset of C is a cutset.

Definition 1. 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 $r(x) < r(y)$. A representation function r is called a *rank function* if $r(m) = 0$ for all minimal elements m of $\mathcal{Z}$ and if $x \prec y$ implies $r(y) = r(x) + 1$.

Note that if a rank function for a poset exists then it is unique.

4. One-time signature schemes based on directed acyclic graphs

4.1. The graph of a one-time signature scheme

Let B be a suitable large set (e.g., the set of 64, 96 or 128-bit strings) and let $f_1, f_2, \ldots$ with $f_i : B^i \longrightarrow B$ be a list of one-way functions, where f_i takes as input a list of i values in B and produces as output a single value in B. Consider a scenario in which a secret key S consisting of u values $s_1, \ldots, s_u \in B$ is chosen at random, and a sequence of values $s_{u+1}, s_{u+2}, \ldots, s_t$ is computed from $s_1, \ldots, s_u$ by applications of the one-way functions f_i. More precisely, for $u + 1 \leq j \leq t$, s_j is the result of applying an appropriate OWF to a subset $U(s_j)$ (of appropriate size) of $\{s_1, \ldots, s_{j-1}\}$, where the order of the arguments is assumed to be fixed but is irrelevant for the further discussion. Some of these computed values will not be used as input to a OWF and are published as the public key P. Signatures consist of appropriately chosen subsets of $\{s_1, \ldots, s_t\}$.

In the following we need to distinguish between the structure of the described computation for setting up a digital signature scheme and the particular values resulting for a particular choice of the secret key. Consider a graph $\mathcal{G} = (V, E)$ with vertex set $V = \{v_1, \ldots, v_t\}$, where v_i corresponds to the value s_i, and with edge set E containing the edge (v_i, v_j) if and only if s_i is an input to the OWF resulting in s_j. Hence the value corresponding to v_j can be computed from the values corresponding to the predecessors of v_j, and it functionally depends on the value s_k (corresponding to v_k) if and only if there exists a directed path from v_k to v_j.

In such a graph the secret key set and the public key set correspond (without loss of generality, see final paper) to the sets of vertices with in-degree 0 and out-degree 0, respectively. The graph $\mathcal{G}$ is assumed to be known publicly and can be used by all users, but the values corresponding to the vertices for a user's particular secret key are kept secret by the user, except those values corresponding to the public key. A signature scheme assigns a signature pattern, i.e. an appropriate subset of vertices, to every message in the message space. 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. The set of signature patterns must satisfy certain conditions discussed below.

Remarks.
(1) Of course, the OWFs used for evaluating different vertices can be different, as long as a function together with the order of the arguments is uniquely specified for each

vertex.
(2) As mentioned before, the secret key components can be generated in a pseudo-random manner from a single secret key. We will use the convention that when two vertices in $\mathcal{G}$ have the same set of predecessors, then the two OWFs used in the corresponding computation steps are different and unrelated. We can hence extend $\mathcal{G}$ by introducing an extra vertex s_0 (the real secret key) and edges form s_0 to the vertices $s_1, \ldots, s_u$. Similarly, one can without much loss of generality restrict the discussion to graphs with only one public-key component because a list of public-key components could be hashed using a secure cryptographic hash function.
(3) Messages that are too long to be signed by a given scheme can be compressed by a one-way hash function prior to signing. Hence the message space of a signature scheme can be chosen to coincide with the range of a secure cryptographic hash function, for instance the set of 128-bit strings.

4.2. The associated poset of a directed acyclic graph

This section gives a formal definition of a one-time signature scheme based on a directed acyclic graph (DAG) $\mathcal{G} = (V, E)$. The *secret key pattern* $S(\mathcal{G}) \subset V$ and the *public key pattern* $P(\mathcal{G}) \subset V$ are defined as the sets of vertices with in-degree 0 and out-degree 0, respectively. Let X be a subset of V. A vertex v is defined recursively to be *computable* from X if either $v \in X$ or if v has at least one predecessor and all predecessors are computable from X. A set Y is computable from X if every element of Y is computable from X. Note that V and hence every subset of V is computable from the secret key $S(\mathcal{G})$.

A set $X \in V$ is called *verifyable* (with respect to the public key) if $P(\mathcal{G})$ is computable from X. Note that a set X is verifyable if and only if every maximal path (in the sense that it cannot be extended to a longer path or, equivalently, a path from a vertex in $S(\mathcal{G})$ to a vertex in $P(\mathcal{G})$) contains at least one element in X. A verifyable set X is *minimal* if no subset of X is verifyable. Two minimal verifyable sets (MVS) X and Y are *compatible* if neither X is computable from Y nor Y is computable from X. A set of MVSs is compatible if they are pairwise compatible.

The computability relation on the set of MVSs of a graph is transitive, antisymmetric and reflexive, and hence the set of MVSs of a graph $\mathcal{G}$, denoted $\mathcal{W}(\mathcal{G})$, forms a poset $(\mathcal{W}(\mathcal{G}), \leq)$ with computability as the order relation, i.e., we have $X \leq Y$ for $X, Y \in \mathcal{W}(\mathcal{G})$ if and only if X is computable from Y. Note that two MVSs of $\mathcal{G}$ are compatible if and only if they are incomparable in $(\mathcal{W}(\mathcal{G}), \leq)$.

Definition 2. The *associated poset* of DAG $\mathcal{G}$ is the poset $(\mathcal{W}(\mathcal{G}), \leq)$ of minimal verifyable subsets of $\mathcal{G}$.

In order to remove a possible source of confusion it should be pointed out that a DAG in which every edge (x, y) is the only path from x to y has itself the structure of a poset and $\mathcal{W}(\mathcal{G})$ is the poset of cutsets in this poset. However, we have avoided the term "cutset" for the signature patterns because this term has a different meaning for graphs.

Definition 3. A *one-time signature scheme* $\mathcal{A}$ for an acyclic directed graph $\mathcal{G} = (V, E)$ is an antichain of the associated poset $\mathcal{W}(\mathcal{G})$.

Note that we are interested only in the set of signature patterns for a given message space (whose cardinality must not exceed the size of the antichain), but not in the particular mapping that assigns signature patterns to messages.

The important parameters of a one-time signature scheme $\mathcal{A}$ for a graph $\mathcal{G} = (V, E)$ are the number $|V|$ of vertices (which is equal to the sum of the size of the secret key and the number function evaluations required for computing a public key from a secret key), the number $|\mathcal{A}|$ of signatures which must be at least equal to the size of the message space, and the maximal size of signatures, $\max_{X\in\mathcal{A}} |X|$. The following interesting problems almost motivate themselves. First, for a given graph to find a large (ideally a maximal-sized) antichain in the associated poset. Note that $w(\mathcal{W}(\mathcal{G}))$ denotes the maximal size of such an antichain. Second, for a given size of the message space to find a graph with few (ideally the minimal number of) vertices allowing the construction of a one-time signature scheme. Third, both problems should be treated with a constraint on the maximal size of signatures.

4.3. Representation functions and generating functions

It follows from the definition of a representation function r of a poset $\mathcal{Z} = (Z, \leq)$ that $r(x) = r(y)$ implies that x and y are incomparable. Hence for any representation function r of the associated poset $(\mathcal{W}(\mathcal{G}), \leq)$ of a given DAG $\mathcal{G}$ and for any integer k, the set $\mathcal{D}(\mathcal{G}, r, k)$ defined by

$$\mathcal{D}(\mathcal{G}, r, k) = \{U \in \mathcal{W}(\mathcal{G}) : r(U) = k\}$$

is a one-time signature scheme. Let $\Psi_{\mathcal{G},r}(x)$ be the generating function of the cardinalities of the sets $\mathcal{D}(\mathcal{G}, r, k)$, i.e., let

$$\Psi_{\mathcal{G},r}(x) = \sum_{U\in\mathcal{W}(\mathcal{G})} x^{r(U)} = \sum_k |\mathcal{D}(\mathcal{G}, r, k)| \cdot x^k,$$

and let $\beta(\mathcal{G}, r)$ be the maximal cardinality among these sets, i.e., let

$$\beta(\mathcal{G}, r) = \max_k(|\mathcal{D}(\mathcal{G}, r, k)|)$$

be the largest coefficient of $\Psi_{\mathcal{G},r}(x)$.

In order to find good signature schemes for a given graph, we need to find a good representation function, that is one with a large maximal coefficient. For $U \in \mathcal{W}(\mathcal{G})$ for a given DAG $\mathcal{G}$ let $C_{\mathcal{G}}(U)$ be the set of vertices of $\mathcal{G}$ that are computable from U but are not contained in U:

$$C_{\mathcal{G}}(U) = \{v : v \notin U \text{ and } v \text{ is computable from } U\}.$$

Let $c_{\mathcal{G}} : \mathcal{W}(\mathcal{G}) \rightarrow \mathbf{N}$ be the function defined by

$$c_{\mathcal{G}}(U) = |C_{\mathcal{G}}(U)|.$$

The following theorem is proved in the Appendix.

Theorem 1. *For any DAG $\mathcal{G}$, the function $c_{\mathcal{G}}$ is a representation function of the associated poset $\mathcal{W}(\mathcal{G})$ of $\mathcal{G}$.*

For many graphs $\mathcal{G}$, $c_{\mathcal{G}}$ is an optimal representation function in the sense that $\beta(\mathcal{G}, c_{\mathcal{G}})$ is equal to the maximal number $w(\mathcal{W}(\mathcal{G}))$ of signatures patterns. However, this is not true in general as the counter example shown in Figure 2 demonstrates. For this tree $\mathcal{T}$ we have $\beta(\mathcal{T}, c_{\mathcal{T}}) = 27$ but $w(\mathcal{W}(\mathcal{T})) = 28$ because $\beta(\mathcal{G}, r) = 28$ for the representation function defined by

$$r(U) := \begin{cases} c_{\mathcal{T}}(U) + 2 & \text{if } s \in U \\ c_{\mathcal{T}}(U) & otherwise, \end{cases}$$

where s is a distinguished vertex in Figure 2.

Let $\mathcal{C}_k$ denote the graph consisting of a single path connecting k vertices, which we will call a chain of length k. Its generating function is given by

$$\Psi_{\mathcal{W}(\mathcal{C}_k), c_{\mathcal{C}_k}}(x) = \sum_{i=0}^{k-1} x^i = \frac{1 - x^k}{1 - x}. \qquad (1)$$

Let $\mathcal{G}_1, \ldots, \mathcal{G}_k$ be k DAGs, let $\mathcal{F}$ be the graph consisting of unconnected copies of $\mathcal{G}_1, \ldots, \mathcal{G}_k$ and let $\mathcal{G} = [\mathcal{G}_1, \ldots, \mathcal{G}_k]$ be the DAG obtained from $\mathcal{F}$ by combining $\mathcal{G}_1, \ldots, \mathcal{G}_k$ by introducing a new vertex v as well as edges from the k public keys of $\mathcal{G}_1, \ldots, \mathcal{G}_k$ to v. Let $r_{\mathcal{G}_1}, \ldots, r_{\mathcal{G}_k}$ be some representation functions for $\mathcal{G}_1, \ldots, \mathcal{G}_k$, respectively. Then the MVSs of $\mathcal{F}$ are all sets $S = \bigcup_{i=1}^{k} S_i$ where S_i is a MVS of $\mathcal{G}_i$, for $1 \leq i \leq k$. The function

$$r_{\mathcal{F}}(\bigcup_{i=1}^{k} S_i) = \sum_{i=1}^{k} r_{\mathcal{G}_i}(S_i)$$

is a representation function for $\mathcal{F}$ and the corresponding generating function is

$$\Psi_{\mathcal{W}(\mathcal{F}), r_{\mathcal{F}}}(x) = \prod_{i=1}^{k} \Psi_{\mathcal{W}(\mathcal{G}_i), r_{\mathcal{G}_i}}(x) \qquad (2)$$

as will be shown in the final paper. Similarly, the MVSs of $\mathcal{G}$ are those of $\mathcal{F}$ together with $\{v\}$, i.e., $\mathcal{W}(\mathcal{G}) = \mathcal{W}(\mathcal{F}) \cup \{\{v\}\}$ and $r_{\mathcal{G}}$ defined by $r_{\mathcal{G}}(\{v\}) = 0$, $r_{\mathcal{G}}(S) = r_{\mathcal{F}}(S) + 1$ for $S \in \mathcal{W}(\mathcal{F})$ is a representation function for $\mathcal{G}$ and $\Psi_{\mathcal{W}(\mathcal{G}), r_{\mathcal{G}}}(x) = 1 + x\Psi_{\mathcal{W}(\mathcal{F}), r_{\mathcal{F}}}(x) = 1 + x\prod_{i=1}^{k} \Psi_{\mathcal{W}(\mathcal{G}_i), r_{\mathcal{G}_i}}(x)$ is the corresponding generating function. Note that recursive application of this equation and of (1) allows one to compute the generating function $\Psi_{\mathcal{W}(\mathcal{T}), c_{\mathcal{T}}}(x)$ for arbitrary trees $\mathcal{T}$.

5. Results on optimal graphs and signature schemes

A reasonable implementation of a list of OWFs $f_1, f_2, f_3 \ldots$ with one, two, three, etc. arguments is by implementing a OWF f_2 with two arguments and implementing the function f_1 with one argument as $f_1(x) = f_2(x, x)$ and the functions f_i for $i \geq 3$ as $f_i(x_1, \ldots, x_i) = f_2(f_{i-1}(x_1, \ldots, x_{i-1}), x_i)$. The function f_2 can for instance be implemented by applying DES in an appropriate mode, but much more efficient implementations of good candidate OWFs are possible.

In the described implementation based on a function f_2, the graph could be considered to consist only of vertices with fan-in 1 or 2. In the sequel we discuss the problem of maximizing the number of signature patterns for a given number n of vertices under this fan-in restriction. Let $\nu(n)$ be the maximal number of MVSs obtainable for a graph with n vertices and let $\mu(n)$ be the maximal number of compatible MVSs for a graph with n vertices, i.e.,

$$\begin{aligned} \nu(n) &= \max\{|\mathcal{W}(\mathcal{G})| : \mathcal{G} = (V, E) \text{ with } |V| = n\} \\ \mu(n) &= \max\{w(\mathcal{W}(\mathcal{G})) : \mathcal{G} = (V, E) \text{ with } |V| = n\}, \end{aligned}$$

where $\mathcal{G}$ has fan-in at most 2 and public key of size 1. Likewise we define $\nu^*(n)$ to be the maximal number of MVSs obtainable for a tree with n vertices and $\mu^*(n)$ to be the maximal number of compatible MVSs for a tree with n vertices.

In this section we derive concrete and asymptotic results on $\mu(n)$ The size of signatures is also an important efficiency parameter and schemes requiring only short signatures will be discussed in Section 6.

For a DAG $\mathcal{G} = (V, E)$ we define $\mathcal{R}_{\mathcal{G},l}$ to be the forest consisting of l identical graphs $\mathcal{G}$. As pointed out in the previous section, the poset MVSs of $\mathcal{R}_{\mathcal{G},l}$ consists of all l-tuples $(S_1, \ldots, S_l)$ for which S_i is a MVS of the i-th copy of $\mathcal{G}$.

Let $r_\mathcal{G}$ be any representation function of $\mathcal{W}(\mathcal{G})$ such that there exist $S_1, S_2 \in \mathcal{W}(\mathcal{G})$ where $r_\mathcal{G}(S_1) - r_\mathcal{G}(S_2) = 1$. We define the representation function r of $\mathcal{W}(\mathcal{R}_{\mathcal{G},l})$ by $r(S) = \sum_{i=1}^{l} r_\mathcal{G}(S_i)$ for an MVS $S = (S_1, \ldots, S_l) \in \mathcal{W}(\mathcal{R}_{\mathcal{G},l})$.

Theorem 2. *For the representation function r defined above we have*

$$\lim_{l\to\infty} \beta(\mathcal{R}_{\mathcal{G},l}, r)\frac{\sqrt{l}}{m^l} = \frac{1}{\sigma\sqrt{2\pi}}$$

where σ is the standard deviation of $r_\mathcal{G}(S)$ if S is chosen uniformly from $\mathcal{W}(\mathcal{G})$.

Proof. Let Y be the random variable defined by $Y = (r_\mathcal{G}(S) - E[r_\mathcal{G}(S)])/\sigma$ where S is chosen uniformly from $\mathcal{W}(\mathcal{G})$. The distribution of Y is a lattice distribution with span $1/\sigma$, $E[Y] = 0$ and $E[Y^2] = 1$. Now we can apply theorem 3 of [3, p.490] to complete the proof (see final paper for details). □

It schould be mentioned that $\beta(\mathcal{R}_{\mathcal{G},l}, r) = O(m^l/\sqrt{l})$ is satisfied for any choice of $r_\mathcal{G}$. Theorem 2 implies the following result which will be proved in the final paper.

Corollary 3.

$$\lim_{n\to\infty} \frac{\log_2 \mu(n)}{n} \geq \max_m \frac{\log_2 \nu(m)}{m}$$

The DAG $\mathcal{G}_{16}$ shown in figure 4 has 16 vertices and its associated poset has 164 vertices. $\mathcal{R}_{\mathcal{G}_{16},l}$ has therefore $O(164^l/\sqrt{l})$ signatures. In order to combine l copies of the graph in a tree with fan-in 2 we need a tree of $l-1$ additional vertices. Hence we can asymptotically sign $(\log_2(164)/17) \cdot n = 0.4327n$ bits with n vertices. Theorem 4 below shows that this number cannot be achieved by trees.

Let a sequence of trees $\mathcal{T}_0, \mathcal{T}_1, \ldots$ be defined recursively by $\mathcal{T}_0 = \mathcal{C}_3$ and $\mathcal{T}_{t+1} = [\mathcal{T}_t, \mathcal{T}_t]$. Then $\mathcal{T}_2$ has 15 vertices and its associated poset has 101 vertices. Let this tree be repeated l times with any top-layer consisting of $l-1$ vertices. With this scheme one can sign asymptotically $\log_2(101)/16 \approx 0.4161$ bits per vertex. Theorem 4, which is proved in the Appendix, shows that this is very close to the achievable optimum for trees.

Theorem 4. *Let $\mathcal{T}$ be a tree with n vertices. Then*

$$|\mathcal{W}(\mathcal{T})| \leq 2^{\gamma(n+1)} \textit{ where } \gamma = \log_2(685/216)/4 = 0.4162\ldots \tag{3}$$

In other words, no tree with n vertices allows to sign more than $\gamma(n+1)$ bits.

The constant γ can be reduced slightly by using more detailed arguments in the proof. However, this upper bound is almost tight as the previous example of recursively defined trees demonstrates. It can be verified numerically that the lower bound obtained for $\mathcal{T}_t$ approaches a constant which agrees with the best upper bound γ up to 8 decimal digits, but it remains to prove that there exists a constant which is both upper and lower bound.

The final paper will contain various results on optimal small graphs, i.e., exact values for $\mu(n)$ and $\mu^*(n)$. For all $n \geq 8$ we have investigated we have $\mu(n) > \mu^*(n)$. Table I in the Appendix summarizes some of these results.

6. Concrete graphs and implementations

The length of signatures is an important efficiency parameter for a signature scheme. Furthermore, there must exist an efficient algorithm for assigning a signature pattern to a given message. In this section we discuss schemes with signature patterns consisting of at most l vertices. Let $\mu(n,l)$ be the maximal size of a one-time signature scheme $\mathcal{A}$ with elements of size at most l for a graph with n vertices.

The most reasonable choice for a graph appears to be a forest of l equal-length chains with a top layer combining these chains to a single-component DAG.

Let $\mathcal{R}_{k,l}$ be the forest consisting of l chains of length k whose vertices will be denoted by $v_{i1}, \ldots, v_{ik}$ for the ith chain. According to (1) and (2) the generating function of $\mathcal{R}_{k,l}$ is given by

$$\Psi_{\mathcal{W}(\mathcal{R}_{k,l}), c_{\mathcal{R}_{k,l}}}(x) = \left(\frac{1-x^k}{1-x}\right)^l = \sum_{i\geq 0} x^i \sum_{j=0}^{\lfloor i/k \rfloor} (-1)^j \binom{l}{j} \binom{l+i-kj-1}{l-1}. \tag{4}$$

The last step will be proved in the final paper.

In a practical implementation of such a scheme, the public key consisting of the l top elements of the chains would of course be hashed cryptographically to a single public-key component, i.e., the chains would be connected to a rake-shaped tree. We therefore refer to the scheme based on this graph, which is discussed below, as the "rake scheme". Similarly, the secret key set consisting of the l bottom elements could be generated pseudo-randomly from a single secret key, thereby generating a symmetric hammock-shaped graph.

The poset of minimal verifyable sets of $\mathcal{R}_{k,l}$ consists of all l-tuples $(v_{1,a_1}, \ldots, v_{l,a_l})$ with $1 \leq a_i \leq k$. This associated poset $\mathcal{W}(\mathcal{R}_{k,l})$ is equal to the product of l chains (in the poset terminology) of length k. It has been shown that a poset consisting of a product of chains has the Sperner property [1] which implies that the maximal number of signature patterns can be obtained by using the representation function introduced in Section 4.3 because this function is a rank function. In other words, we have $w(\mathcal{W}(\mathcal{R}_{k,l})) = \beta(\mathcal{R}_{k,l}, c_{\mathcal{R}_{k,l}})$. It will be shown in the final paper that for a fixed l, $w(\mathcal{W}(\mathcal{R}_{k,l}))$ can be written as a polynomial in k of degree $l-1$.

Theorem 5. *We have* $w(\mathcal{W}(\mathcal{R}_{k,l})) = \alpha_l k^{l-1} + O(k^{l-2})$, *where* $\alpha_l = \frac{1}{(l-1)!} \sum_{j=0}^{\lfloor (l-1)/2 \rfloor} (-1)^j \binom{l}{j} (l/2 - j)^{l-1}$ *and where* $\lim_{l \to \infty} \alpha_l \cdot \sqrt{l} = \sqrt{6/\pi}$.

The described rake scheme $\mathcal{R}_{k,l}$ appears to be the simplest graph for implementing one-time signature schemes for a fixed signature lenth, an we conjecture that it is asymtotically optimal in the sense that $\lim_{l \to \infty} \mu(n,l)\sqrt{l}/(n/l)^{l-1} = \alpha_l$. However, there do exist graphs that beat the rake scheme in the coefficient of the second term k^{l-2}. As a realistic example, choosing the parameters $l = 14$ and $k = 2^{10}$ results in a digital signature scheme for a message space of size $4.9 \cdot 10^{38}$ and thus allows to sign arbitrary 128-bit messages.

7. Concluding remarks and outlook

The final paper will also describe an efficient algorithm for enumerating the signature patterns of $\mathcal{R}_{k,l}$, i.e. for assigning the signature pattern to a given message. Furthermore, schemes for signing a fixed number of messages will be discussed in the final paper, where it will be shown that Merkle's authentication tree is not optimal for this purpose.

In applications where the number of messages to be signed is limited (e.g. in certain public-key certification schemes), the rake scheme combined with Merkle's authentication tree appears to be a realistic alternative to conventional digital signature schemes. However, independently of possible applications, we believe that the presented general approach to signature schemes based on one-way functions is also of significant theoretical interest. It leads to a collection of interesting combinatorial research problems which will be described systematically in the final paper.

Acknowledgements

The authors would like to thank Martin Perewusnyk, Adi Shamir and Roger Wattenhofer for many inspiring discussions on the topic of this paper.

References

[1] N. de Brujin, C. A. van Ebbenhorst Tengebergen, and D. R. Kruyswijk, "On the set of divisors of a number," *Nieuw Arch. Wisk*, vol. 23, pp. 191–193, 1952.

[2] 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.

[3] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. II., second corrected printing, Wiley & Sons, 1966.

[4] J. R. Griggs, "The Sperner property," *Proc. Conference on Ordered Sets and their Application*, 1982.

[5] J. R. Griggs, "Maximum antichains in the product of chains," *Order*, vol. 1, pp. 21–24, 1984.

[6] L. Lamport, Constructing digital signatures from a one-way function, Technical Report SRI Intl. CSL 98, 1979.

[7] 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.

[8] C. Meyer and S. Matyas, *Cryptography – a new dimension in computer data security*, John Wiley & Sons, Inc., 1982.

[9] 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.

[10] J. Rompel, One-way functions are necessary and sufficient for secure signatures, *Proc. 22nd ACM Symp. on Theory of Computing (STOC)*, 1990, pp. 387-394.

[11] C.P. Schnorr, Efficient identification and signatures for smart cards, Advances in Cryptology – Crypto '89, Lecture Notes in Computer Science, vol. 435 (G. Brassard, ed.), Springer-Verlag 1990, pp. 239-252.

[12] E. Sperner, Ein Satz über Untermengen einer endlichen Menge, *Mathematische Zeitschrift*, vol. 27, pp. 544–548, 1928.

[13] 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.

Appendix

Proof of Theorem 1.

Let U_1 and U_2 be distinct MVSs with $U_1 \leq U_2$. We need to prove that $|C_{\mathcal{G}}(U_1)| < |C_{\mathcal{G}}(U_2)|$. Let v be any element in $C_{\mathcal{G}}(U_1)$. All predecessors of v are computable from U_1 by definition. Since U_1 is computable from U_2 any vertex that is computable from U_1 is computable from U_2. Therefore all predecessors of v are computable from U_2. If v were in U_2 then U_2 would not be minimal. Thus $v \in C_{\mathcal{G}}(U_2)$ and we have $C_{\mathcal{G}}(U_1) \subseteq C_{\mathcal{G}}(U_2)$. Moreover, U_1 is not a subset of U_2 because U_2 is minimal. Hence there exists a vertex $s \in U_1$ with $s \notin U_2$ which is computable from U_2 because U_1 is computable from U_2. Therefore $s \in C_{\mathcal{G}}(U_2)$ and $s \notin C_{\mathcal{G}}(U_1)$ and thus we have $C_{\mathcal{G}}(U_1) \neq C_{\mathcal{G}}(U_2)$. Hence $C_{\mathcal{G}}(U_1)$ is a proper subset of $C_{\mathcal{G}}(U_2)$ which implies that $|C_{\mathcal{G}}(U_1)| < |C_{\mathcal{G}}(U_2)|$. □

Proof of Theorem 4.

Let $f : x \mapsto x + (2x)^{-1} + (8x^3)^{-1}$. We have $f(x)f(y) \geq f(xy+1)$ for all $y \geq x \geq 1$. This has been verified for instance using MAPLE by replacing $x = z+1$ and $y = z+c+1$ and checking that $(f(x)f(y) - f(xy+1))64x^3y^3(xy+1)^3$ is a polynomial in z and c with no negative coefficients. Moreover, we have $f(x+1) \leq \frac{f(4)}{f(3)} f(x)$ for $x \geq 3$. Now define $g(\mathcal{T}) := \log_2(f(|\mathcal{W}(\mathcal{T})|))/(|\mathcal{T}|+1)$. We will show that for all trees

$$g(\mathcal{T}) \leq \gamma \tag{5}$$

by induction over $|\mathcal{T}|$. There are four trees with at most 3 vertices which can easily be checked to satisfy equation (5). (The maximal value is $g(\mathcal{C}_3) = \gamma$.) Now assume that (5) is satisfied for all trees with at most n vertices. Let $\mathcal{T}$ be a tree with n vertices. We show that (5) is satisfied for the tree $[\mathcal{T}]$ that is constructed from $\mathcal{T}$ by adding one new root node. Note that $\gamma \geq \log_2(f(4)/f(3))$. Then we have $g([\mathcal{T}]) = \log_2(f(\mathcal{W}([\mathcal{T}])+1))/(n+2) \leq (\log_2(f(\mathcal{W}(\mathcal{T}))) + \log_2(f(4)/f(3)))/(n+2) \leq (\gamma(n+1) + \log_2(f(4)/f(3)))/(n+2) \leq \gamma(n+2)/(n+2) \leq \gamma$.

Now let $[\mathcal{T}, \mathcal{T}']$ be any tree with $n+1$ vertices that is composed by combining $\mathcal{T}$ and $\mathcal{T}'$. Note that $|\mathcal{T}| + |\mathcal{T}'| = n$. By the assumption of the induction we have $\log_2(f(|\mathcal{W}(\mathcal{T})|)) \leq \gamma(|\mathcal{T}|+1)$ and $\log_2(f(|\mathcal{W}(\mathcal{T}')|)) \leq \gamma(|\mathcal{T}'|+1)$. Thus it follows $g([\mathcal{T},\mathcal{T}']) = \log_2(f(|\mathcal{W}([\mathcal{T},\mathcal{T}'])|))/(|[\mathcal{T},\mathcal{T}']|+1) = \log_2(f(|\mathcal{W}(\mathcal{T})| \cdot |\mathcal{W}(\mathcal{T}')|+1))/(|\mathcal{T}| + |\mathcal{T}'|+2) \leq \log_2(f(|\mathcal{W}(\mathcal{T})|)) + \log_2(f(|\mathcal{W}(\mathcal{T}')|))/(|\mathcal{T}|+|\mathcal{T}'|+2) \leq (\gamma(|\mathcal{T}|+1)+\gamma(|\mathcal{T}'|+1))/(|\mathcal{T}|+|\mathcal{T}'|+2) = \gamma$.

To finish the proof we have to note that (5) implies the theorem. Indeed let $n = |\mathcal{T}|$ and note that $x \leq f(x)$. $\square$

The quantities $\mu(n)$ and $\mu^*(n)$ for small n

The following list summarizes the size of the optimal one-time signature scheme for trees and the size of the best one-time signature scheme we have found for general DAGs, which provides a lower bound on $\mu(n)$.

number n of vertices	$\mu^*(n)$	lower bound for $\mu(n)$	number n of vertices	$\mu^*(n)$	lower bound for $\mu(n)$
8	3	4	15	19	25
9	4	5	16	23	33
10	5	7	17	29	45
11	7	9	18	39	57
12	8	12	19	53	79
13	11	15	20	67	101
14	14	20	21	85	139

Figures

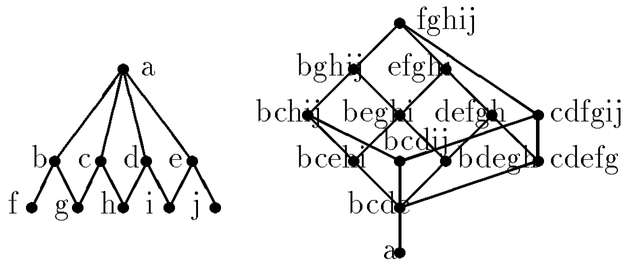


Figure 1: An example of a DAG (left) and its associated poset (right). This graph is special in that the poset does not have a rank function.

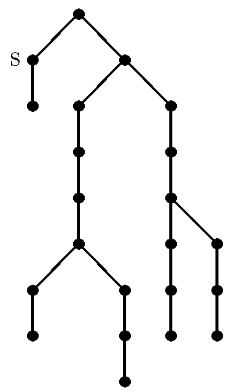


Figure 2: A tree $\mathcal{T}$ for which the function $c_{\mathcal{T}}$ is not an optimal representation function.

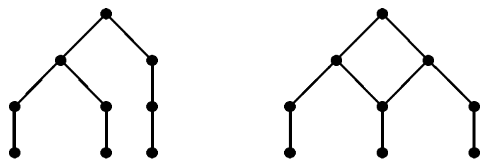
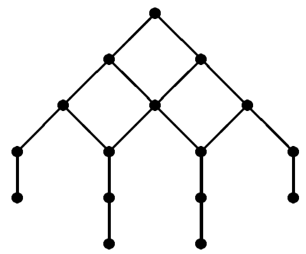


Figure 3: An optimal tree with 9 vertices and 4 compatible signature patterns, and a general DAG with 9 vertices and 5 compatible signature patterns.



$\{\mathcal{G}\}_{\infty/}$

Figure 4: A DAG with 16 vertices whose associated poset has 164 vertices