On the Efficiency of One-time Digital Signatures

Daniel Bleichenbacher[1] and Ueli Maurer[2]

[1] Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

[2] Department of Computer Science
Swiss Federal Institute of Technology (ETH Zurich)
CH-8092 Zürich, Switzerland

Abstract. Digital signature schemes based on a general one-way function without trapdoor offer two potential advantages over digital signature schemes based on trapdoor one-way functions such as the RSA system: higher efficiency and much more freedom in choosing a cryptographic function to base the security on. Such a scheme is characterized by a directed acyclic computation graph and an antichain in a certain partially ordered set defined by the graph. Several results on the achievable efficiency of such schemes are proved, where the efficiency of a scheme is defined as the ratio of the size of messages that can be signed and the number of one-way function evaluations needed for setting up the system. For instance, the maximal achievable efficiency for trees is shown to be equal to a constant $\gamma \approx 0.4161426$ and a family of general graphs with substantially greater efficiency 0.476 is demonstrated. This construction appears to be close to optimal.

Key words. Cryptography, Digital signature, One-way function, Directed acyclic graph, Partially ordered set.

1 Introduction

One can distinguish between three types of digital signature schemes. The first type of scheme was proposed by Lamport [7] and generalized in [8], [9], [5], [14] and [1]. Once it is set up, it can only be used for signing a predetermined number (e.g. one) of messages from a certain message space. The second type of schemes, the first realization of which was the RSA system [11], can be used an unlimited number of times. In contrast to the first type of scheme, the second requires strong mathematical structure in the underlying one-way function. A third type of scheme was proposed by Rompel [12] based on work by Naor and Yung [10]. The security of these schemes can provably be based on an arbitrary one-way function, but they are inefficient. The purpose of this paper is to discuss the design and analysis of schemes of the first type, where the emphasis is on efficiency and freedom in the choice of the cryptographic function on which the system is based. In contrast to Rompel's work, our goal is not to prove rigorously that the security is equivalent to the security of the one-way function(s).

There are two different motivations for investigating and possibly using the first type of schemes despite their limited number of uses. First, they can be based on virtually every cryptographic one-way function[3] (OWF), a very general cryptographic primitive, whereas the few schemes of the second type proposed so far are based on OWFs with a very strong mathematical structure. The diversity of conjectured difficult problems (such as the integer factoring problem [11] or the discrete logarithm problem in certain finite groups [13]) on which their security can be based is thus severely limited. While such mathematical structure is appealing to the designer and the users of a system, it could for an adversary just as well be the key to breaking the system if he is able to exploit the structure in a way not foreseen by the designer. Second, the first type of scheme is potentially more efficient because a general OWF, which for this purpose not even needs to be collision-free, can be realized much more efficiently than OWFs with appropriate structure. Moreover, these schemes have applications in efficiency-critical smartcard applications [6], in on-line/off-line signatures [5] and in the signature schemes of [3].

The general concept of a digital signature schemes of the first type was formalized in [1]. The purpose of this paper is to discuss constructions for such schemes and to prove several results on the achievable efficiency, in particular for computation graphs that are trees. The outline of the paper is as follows. To make the paper reasonably self-contained, the basic ideas underlying [1] are briefly discussed in Section 2, and the definitions are summarized in Section 3. In Section 4 several types of graphs and constructions are analyzed and lower and upper bound results on their efficiency are derived. The special case of trees is discussed in Section 5 and the best known general graph construction is presented in Section 6.

2 One-time Digital Signature Schemes

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.

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

[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. 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 $f(x, y)$ with two arguments should satisfy $f(x, y) = f(y, x)$ for $x \neq y$ only with negligible probability. It is an open problem to characterize when a function is secure in our context.

requirements on these signatures. First, every signature must be verifiable, 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) which is the range of the OWFs. The input to each OWF evaluation consists of one or several elements of B. The secret key consists of one or a list of elements of B. Without loss of essential generality only schemes are considered for which 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 (see Figure 1, left side).

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), and 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 is shown in Figure 1.

In this paper we are interested in the design of efficient signature schemes based on graphs, where the size of the message space should be maximized while the size of the graph should be minimized. Because messages to be signed can first be hashed by a collision-free hash function to a short string (e.g., of 128 bits), it is sufficient that the message space of our schemes corresponds to the range of such a hash function (e.g., has size 2^{128}).

3 Definitions and Preliminaries

This section summarizes the relevant definitions from [1] and introduces the concept of efficiency. 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.

Let $\mathcal{C}_m$ denote the DAG consisting of a single path connecting m vertices, i.e., a chain of length m. For k DAGs $\mathcal{G}_1, \ldots, \mathcal{G}_k$, let $\mathcal{G}_1 \cdots \mathcal{G}_k$ denote the graph consisting of unconnected copies of $\mathcal{G}_1, \ldots, \mathcal{G}_k$. If each of the graphs $\mathcal{G}_1, \ldots, \mathcal{G}_k$ has only one vertex of out-degree 0 (corresponding to the public key in our context), let $\mathcal{G} = [\mathcal{G}_1 \cdots \mathcal{G}_k]$ be the DAG obtained from $\mathcal{G}_1 \cdots \mathcal{G}_k$ by introducing a new vertex v and directed edges from these k distinguished vertices to v.

We now define a one-time signature scheme based on a 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

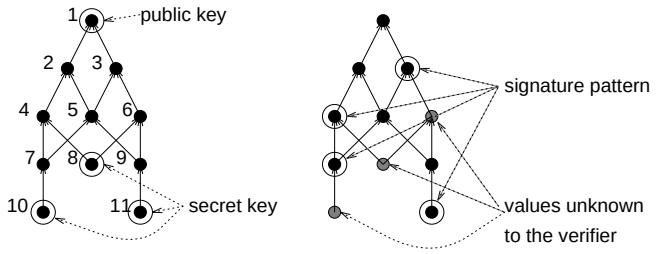

Fig. 1. A toy example of a one-time signature scheme. The secret key consists of the 3 vertices: 8, 10 and 11. One signature pattern (the set $\{3, 4, 7, 11\}$) is indicated on the right-hand side. The associated poset of this graph contains 29 signature patterns, but the maximal number of compatible signature patterns is 9. One maximal antichain consists of the sets $\{2, 5, 8, 11\}$, $\{2, 6, 7, 11\}$, $\{2, 6, 9, 10\}$, $\{3, 4, 7, 11\}$, $\{3, 4, 9, 10\}$, $\{3, 5, 8, 10\}$, $\{4, 5, 8, 9\}$, $\{4, 6, 7, 9\}$, and $\{5, 6, 7, 8\}$. All these signature patterns have size 4, but in general they have different sizes. The efficiency of this scheme is $(\log_2 9)/12 = 0.264$, which is better than Lamport's scheme with efficiency $1/6$.

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 of vertices $X \subseteq V$ is called *verifiable* (with respect to the public key) if $P(\mathcal{G})$ is computable from X. Note that a set X is verifiable 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 verifiable set X is *minimal* if no subset of X is verifiable. Two minimal verifiable sets X and Y are *compatible* if neither X is computable from Y nor Y is computable from X. A set of minimal verifiable sets is compatible if they are pairwise compatible.

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

Definition 1. Minimal verifiable sets will in the following be called *signature patterns*. The *associated poset* of DAG $\mathcal{G}$, denoted $\mathcal{G}^*$, is the poset $(\mathcal{G}^*, \leq)$ of signature patterns of $\mathcal{G}$. A *one-time signature scheme* Γ for $\mathcal{G}$ is an antichain of the associated poset $\mathcal{G}^*$, and the maximal size of an anti-chain in $\mathcal{G}$ is denoted by $w(\mathcal{G})$.

A small example of a signature scheme is shown in Figure 1 and is discussed in the figure caption.

The important parameters of a one-time signature scheme Γ for a graph

$\mathcal{G} = (V, E)$ are the number $|V|$ of vertices which is equal to the number of function evaluations required for computing the public key from the secret key[4], the number $|\Gamma|$ of signatures (which is at least equal to the size of the message space), and the maximal size of signatures, $\max_{U \in \Gamma} |U|$.

This motivates the following problems. First, for a given graph $\mathcal{G}$ to find a large (ideally a maximal-sized) antichain in the associated poset. 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 an additional constraint on the maximal size of signatures.

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 a one-time signature scheme.

In order to find good signature schemes for a given graph, we need to find a good representation function. For $U \in \mathcal{G}^*$ 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 $c_{\mathcal{G}} : \mathcal{G}^* \to \mathbf{N}$ be the function defined by

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

The following theorem was stated in [1] without proof.

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

Proof. Let U_1 and U_2 be distinct signature patterns with $U_1 \leq U_2$. We must 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 $|C_{\mathcal{G}}(U_1)| < |C_{\mathcal{G}}(U_2)|$. □

A natural implementation of a one-way function with i arguments is to apply a one-way function with two arguments repeatedly $i - 1$ times, each time

[4] Here we have assumed that a secret key consisting of several components is generated from a single component by applying, for each component, a different one-way function to the secret key.

combining the previous result with a new argument. This computation can be represented as a binary tree. Without much loss of generality we therefore restrict the discussion in this paper to graphs with a maximal in-degree of 2, counting OWF evaluations with 1 or 2 arguments equaly. Furthermore, because a public key consisting of several components can be hashed to a single value, we restrict the discussion to graphs with a single vertex of out-degree 0 (whose value corresponds to the public key).

The efficiency of a signature scheme Γ for a graph $\mathcal{G}$ can be defined as the number of message bits, $\log_2 |\Gamma|$, that can be signed per vertex of the graph. However, the results on efficiency can be stated more nicely when the number of vertices is increased by one in the following definition.

Definition 2. The *efficiency* of a one-time signature scheme Γ for a graph $\mathcal{G}$ with n vertices, denoted $\eta(\Gamma)$, is defined by

$$\eta(\Gamma) = \frac{\log_2 |\Gamma|}{n+1}.$$

For example, the graph corresponding to Lamport's scheme for signing a k-bit message contains $6k-1$ vertices when all the public-key components are hashed in a binary tree to result in a single public-key component. Hence the efficiency of the Lamport scheme is $1/6$.

In the sequel we discuss the problem of maximizing the number of signature patterns for a given number n of vertices under the restriction of maximal in-degree 2. Let $\nu(n)$ be the maximal number of signature patterns for graphs with n vertices and let $\mu(n)$ be the maximal number of compatible signature patterns for graphs with n vertices, i.e., let

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

where vertices in $\mathcal{G}$ have fan-in at most 2 and $\mathcal{G}$ has a public key of size 1. The size of signatures is also an important efficiency parameter and schemes requiring only short signatures will be discussed in Section 4.3.

A simple relation between $\nu(n)$ and $\mu(n)$ is that for all $n \geq 1$,

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

The left inequality follows directly from the definition. To prove the right inequality, let $\mathcal{G}$ be a DAG with n vertices satisfying $|\mathcal{G}^*| = \nu(n)$. Since the range of $c_{\mathcal{G}}$ is a subset of $\{0,\ldots,n-1\}$ there exists an $i \in \{0,\ldots,n-1\}$ such that $|\{U \in \mathcal{G}^* : c_{\mathcal{G}}(U) = i\}| \geq \nu(n)/n$. According to Theorem 1, this set is a one-time signature scheme.

4 Efficient Constructions and Bounds on the Efficiency

In this section we investigate several constructions of one-time signature schemes, each of which leads to relations between the functions μ and ν.

4.1 Repetition of Graphs

The signature patterns of an unconnected collection $\mathcal{G}_1 \cdots \mathcal{G}_k$ of DAGs are the lists $[S_1, \ldots, S_k]$, where each S_i ranges over the signature patterns of $\mathcal{G}_i$. In other words $(\mathcal{G}_1 \cdots \mathcal{G}_k)^* = \mathcal{G}_1^* \times \cdots \times \mathcal{G}_k^*$ and hence $|(\mathcal{G}_1 \cdots \mathcal{G}_k)^*| = \prod_{i=1}^k |\mathcal{G}_i^*|$. When the $\mathcal{G}_i$ are graphs with $|\mathcal{G}_i| = n_i$ and $|\mathcal{G}_i^*| = \nu(n_i)$ for $1 \leq i \leq k$, the total number of signature patterns is $\prod_{i=1}^k \nu(n_i)$. Thus we have proved the following theorem, where the term $k-1$ is needed because according to our convention that graphs have only one vertex with out-degree 0, the k public key vertices of $\mathcal{G}_1, \ldots, \mathcal{G}_k$ must be combined by a binary tree with $k-1$ vertices.

Theorem 2. *For every list $n_1, \ldots, n_k$ of k positive integers,*

$$\nu(\sum_{i=1}^{k} n_i + k - 1) \;\geq\; \prod_{i=1}^{k} \nu(n_i). \tag{2}$$

In particular, $\nu((n+1)k - 1) \geq \nu(n)^k$.

4.2 Separate Representation Function Encoding

Generally it can be considerably easier to design a mapping from the message space to an arbitrary subset of the signature patterns of a graph $\mathcal{G}_1$ rather than to a subset of compatible signature patterns. The compatibility can be guaranteed by introducing a small additional graph $\mathcal{G}_2$. The graph $\mathcal{G}_2$ is used to compensate for the fact that the values of $c_{\mathcal{G}_1}$ vary over a wide range for all signature patterns of $\mathcal{G}_1$. (See the proof for a precise definition of the construction, a special case of which is actually used in smartcard applications [6].)

Theorem 3. *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be graphs with n_1 and n_2 vertices, respectively, such that $|\mathcal{G}_2^*| \geq n_1$. Then the graph $[\mathcal{G}_1\mathcal{G}_2]$ has at least $|\mathcal{G}_1^*|$ compatible signature patterns, i.e., $w([\mathcal{G}_1\mathcal{G}_2]^*) \geq |\mathcal{G}_1^*|$. In particular, for all s and n satisfying $\nu(s) \geq n$ we have*

$$\mu(s + n + 1) \geq \nu(n).$$

Proof. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be DAGs with s and n vertices, respectively, satisfying $|\mathcal{G}_1^*| = \nu(s) \geq n$ and $|\mathcal{G}_2^*| = \nu(n)$. For every partially ordered set with t elements one can number these elements from 0 to $t-1$ such that their order is preserved. Hence there exists a representation function r_1 for $\mathcal{G}_1$ assigning the integers $0, \ldots, \nu(s) - 1$ to the signature patterns of $\mathcal{G}_1$. (Note that for instance the public key is assigned the value 0.) Let r be a representation function for the graph $\mathcal{G} = [\mathcal{G}_1\mathcal{G}_2]$ defined by $r(U) = c_{\mathcal{G}_2}(U_2) + r_1(U_1) + 1$ if the signature pattern S of $\mathcal{G}$ is defined by $U = U_1 \cup U_2$ where $U_1 \subset \mathcal{G}_1$ and $U_2 \subset \mathcal{G}_2$ are signature patterns of $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. Note that r is indeed a representation function because $r(U_1 \cup U_2) \leq r(U_1' \cup U_2')$ implies that either U_1' is not computable from U_1 or U_2' is not computable from U_2. $\square$

4.3 Schemes with Short Signatures

The size of a graph corresponding to a one-time signature scheme determines the computational effort for computing the public key from the secret key and is an important efficiency parameter. There are two additional requirements for making a scheme practical. First, as mentioned above, the mapping from the message space to the signature patterns must be simple and efficiently computable and second, signatures should be short. In this section we therefore discuss schemes with signature patterns consisting of at most l vertices. Let $\mu(n,l)$ and $\nu(n,l)$ be the maximal number of signature patterns of size at most l for a graph with n vertices, when the signature patterns are compatible, or not necessarily compatible, respectively.

Let $\mathcal{R}_{k,l} = \overbrace{\mathcal{C}_k \cdots \mathcal{C}_k}^{l \text{ times}}$ 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. 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.

The poset $\mathcal{R}^*_{k,l}$ of signature patterns of $\mathcal{R}_{k,l}$ consists of all l-tuples $(v_{1,a_1}, \ldots, v_{l,a_l})$ with $1 \leq a_i \leq k$. In the poset (not the graph) terminology, it is equal to the product of l chains of length k and has $|\mathcal{R}^*_{k,l}| = k^l$ elements. Interestingly, it has been shown [4] that a poset consisting of a product of chains has the Sperner property. This implies that the maximal number of signature patterns can be obtained by using the representation function $c_{\mathcal{R}_{k,l}}$ defined in Section 3. The proof of the following theorem is omitted because of space limitations. It shows that for a fixed l, $w(\mathcal{R}^*_{k,l})$ can be written as a polynomial in k of degree $l-1$ which is by a factor k smaller than the total number of signature patterns.

Theorem 4. *The number $w(\mathcal{R}^*_{k,l})$ of compatible signature patterns for the graph $\mathcal{R}_{k,l}$ satisfies*

$$w(\mathcal{R}^*_{k,l}) = \alpha_l k^{l-1} + O(k^{l-2}),$$

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* $\lim_{l\to\infty} \alpha_l \sqrt{l} = \sqrt{6/\pi}$.

We conjecture that the graph $\mathcal{R}_{k,l}$ is asymptotically optimal in the sense that

$$\lim_{l\to\infty} \mu(n,l)\sqrt{l}/(n/l)^{l-1} = \alpha_l. \tag{3}$$

However, there do exist graphs that are better than $\mathcal{R}_{k,l}$ in the coefficient of the second term k^{l-2}.

Rather than using a signature scheme for the graph $\mathcal{R}_{k,l}$, for which the mapping from the message space to the compatible signature patterns is not trivial, it is simpler to combine two rake graphs $\mathcal{G}_1 = \mathcal{R}_{k_1,l_1}$ and $\mathcal{G}_2 = \mathcal{R}_{k_2,l_2}$ by the construction of Section 3.2. The number $k_2^{l_2}$ of signature patterns of the second graph must be at least as large as the number $k_1 l_1$ of vertices of the first graphs. We therefore have

$$k_1 l_1 \leq k_2^{l_2} \implies \mu(k_1 + k_2 + 1, l_1 + l_2) \geq k_1^{l_1}.$$

Example. For instance, one can use $k_1 = 2^{10} = 1024$, $l_1 = 13$, $k_2 = 116$ and $l_2 = 2$. This scheme with signatures of size 15 allows to sign 130-bit messages which is compatible with the use of a cryptographically-secure hash function for hashing arbitrary messages to 128 bits prior to signing.

More generally, if in the construction of Section 3.2 the maximal size of signature patterns in $\mathcal{G}_1$ and $\mathcal{G}_2$ are l_1 and l_2, respectively, then the maximal size of signatures in the combined scheme is $l_1 + l_2$. The following corollary follows immediately.

Corollary 5. *For any l_1, k_1, l_2 and k_2 satisfying $\nu(n_2, l_2) \geq n_1$ we have*

$$\mu(n_1 + n_2 + 1, l_1 + l_2) \geq \nu(n_1, l_1).$$

5 Optimal Trees

In this section we only consider trees. The single node with out-degree 0 is called the root. Note that in contrast to most scenarios in computer science, our directed trees are directed from the leaves to the root. Let $\hat{\nu}(n)$ be the maximal number of signature patterns obtainable for a tree with n vertices and $\hat{\mu}(n)$ the maximal number of compatible signature patterns for a tree with n vertices. In analogy to the proof of (1) one can show that

$$\hat{\nu}(n) \geq \hat{\mu}(n) \geq \frac{\hat{\nu}(n)}{n} \tag{4}$$

Let A and B be two trees. Recall that $[AB]$ denotes the tree obtained from two A and B by introducing a new vertex v and connecting the roots of A and B to v. The following theorem from [2] characterizes the form of optimal trees.

Theorem 6. *For $n \leq 5$ the chain $\mathcal{C}_n$ of length n is an optimal tree in the sense that $\hat{\nu}(n) = |\mathcal{C}_n^*| = n$. For $n > 5$ all optimal trees are of the form $[AB]$, where A and B 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. For $n > 5$ we have*

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

We now consider a tree construction which connects the roots of 2^n identical trees with a full binary tree of depth n.

Definition 3. Let $\tau_n(\mathcal{T})$ for $n \geq 0$ be defined recursively by $\tau_0(\mathcal{T}) := \mathcal{T}$ and $\tau_{n+1}(\mathcal{T}) := [\tau_n(\mathcal{T})\tau_n(\mathcal{T})]$. Let further the function $\rho : \mathbf{Z}^2 \to \mathbf{Z}$ be defined by $\rho(0, m) := m$ and $\rho(n+1, m) := \rho(n, m)^2 + 1$, and let the tree efficiency constant γ be defined by

$$\gamma = \lim_{n \to \infty} \frac{\log_2 \rho(n, 3)}{2^{n+2}} \approx 0.41614263726.$$

Finally, let the function g be defined by $g(\mathcal{T}) := \log_2 |\mathcal{T}^*| / (|\mathcal{T}| + 1)$.

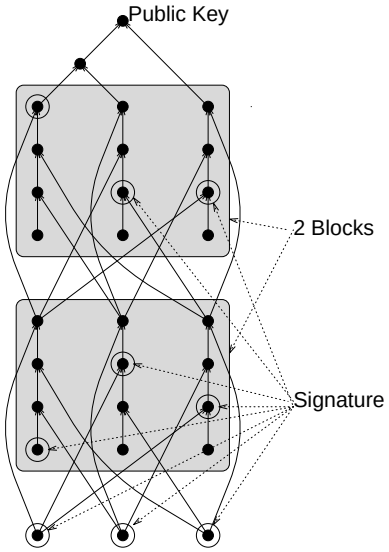

Fig. 2. The graph construction with the best known efficiency which converges asymptotically to 0.476 when the number of 12-vertex blocks (indicated by shaded areas) is increased. One particular signature pattern of size 9 is indicated.

Theorem 7. *The efficiency of every tree-based one-time signature scheme Γ is at most γ, i.e. $\eta(\Gamma) \leq \gamma$. Moreover,*

$$\hat{\mu}(n) \geq \frac{2^{\gamma n}}{2n}$$

and thus for every $\delta < \gamma$ there exists a tree-based one-time signature scheme Γ with $\eta(\Gamma) \geq \delta$.

The proof of Theorem 7 is given in the Appendix. Several results on the construction of optimal trees are proved in [2].

6 The Best Known Graph Construction

Figure 2 shows the one-time signature scheme with the currently best known efficiency. We consider graphs $\mathcal{H}_n$ consisting of n blocks of 12 vertices. Each block consists of 3 chains of length 4 and is connected to the next block in a periodic manner as illustrated in Figure 2, where $\mathcal{H}_2$ is shown as an example. The graphs $\mathcal{H}_3$, $\mathcal{H}_4, \ldots$ are similar to $\mathcal{H}_2$ but contain more blocks. The bottom layer of 3 vertices could be omitted and is shown only for reasons of symmetry.

A large (though not maximal) set of signature patterns for $\mathcal{H}_n$ can be described as follows. The bottom three vertices belong to each signature pattern. Each signature pattern contains one vertex of each of the three chains of each block. This would result in 64 combinations, but 13 of these must be excluded

because the resulting signature pattern would not be minimal. The reason is that in these cases, not all vertices at lower layers would be needed for the verification. Thus $\mathcal{H}_n$ has at least 51^n signature patterns.

The efficiency of the corresponding signature scheme Γ_n is lower bounded by

$$\eta(\Gamma_n) \geq \frac{\log_2(51^n/(12n+5))}{12n+6}$$

and, asymptotically, by

$$\lim_{n\to\infty} \eta(\Gamma_n) \geq \frac{\log_2 51}{12} \approx 0.473.$$

A more careful analysis involving Markov chains shows that the achievable efficiency for $\mathcal{H}_n$ converges to $\log_2(\omega)/12$ where ω is the maximal root of $x^3 - 56x^2 + 173x - 54$. This value is 0.476.

7 Concluding Remarks

We suggest as a challenging open problem to find one-time signature schemes with higher efficiency than that of Section 6, or to prove an upper bound on the efficiency of all such schemes. We conjecture that no scheme has efficiency greater than 1/2. Another open problem is to prove or disprove equation (3). Using Merkle's authentication tree [8] one can extend every one-time signature scheme to one that can be used a predetermined number of times rather than only once. However, this construction is known to be not optimal, and a further interesting problem is to design such schemes that are better than Merkle's construction applied to an optimal one-time signature scheme.

Appendix

The proof of Theorem 7 is divided into several steps, summarized in the following three lemmas. Recall that the functions ρ and g are defined in Definition 3.

Lemma 8. *For all $n \geq 0$,*

$$\rho(n, ab+1)^2 \leq \rho(n+1, a)\rho(n+1, b). \tag{5}$$

Moreover, for every fixed $m \in \mathbf{N}$, the function $\mathbf{N} \to \mathbf{N}$: $n \mapsto 2^{-n}\log_2 \rho(n,m)$ is monotonically increasing, and the function $\mathbf{N} \to \mathbf{N}$: $n \mapsto 2^{-n}\log_2(\rho(n,m)+1)$ is monotonically decreasing.

Proof. For $n = 0$ equation (5) follows from $\rho(1,a)\rho(1,b) - \rho(0,ab+1)^2 = (a^2+1)(b^2+1) - (ab+1)^2 = (a-b)^2 \geq 0$. The lemma follows by induction on n; assuming (5) is satisfied for $n-1$ implies that the difference of the right and the left side of (5) is positive:

$$\rho(n+1,a)\rho(n+1,b) - \rho(n,ab+1)^2 = (\rho(n,a)^2+1)(\rho(n,b)^2+1) \\ -(\rho(n-1,ab+1)^2+1)^2$$

$$\begin{aligned} &\geq (\rho(n,a)^2+1)(\rho(n,b)^2+1) \\ &\quad -(\rho(n,a)\rho(n,b)+1)^2 \\ &= (\rho(n,a)-\rho(n,b))^2 \ \geq \ 0. \end{aligned}$$

To prove the second part of the lemma we note that

$$2^{-n-1}\log_2(\rho(n+1,m)) > 2^{-n-1}\log_2(\rho(n,m)^2) = 2^{-n}\log_2(\rho(n,m)).$$

This implies that the first function is monotonically increasing. On the other hand,

$$\begin{aligned} 2^{-n}\log_2(\rho(n,m)+1) &= 2^{-n-1}\log_2((\rho(n,m)+1)^2) \\ &\geq 2^{-n-1}\log_2(\rho(n+1,m)+1) \end{aligned}$$

implies that the second function is monotonically decreasing. □

Lemma 9. *Let A and B be trees with $|A^*| = a$ and $|B^*| = b$. Then*

(i) $|\tau_n(A)| = 2^n(|A|+1)-1$.
(ii) $|\tau_n(A)^*| = \rho(n,a)$.
(iii) For all trees $\tau_n([AB])$ we have $g(\tau_n([AB])) \leq \max(g(\tau_{n+1}(A)), g(\tau_{n+1}(B)))$.
(iv) For every tree $\mathcal{T}$ there exist m and n such that $g(\tau_n(\mathcal{C}_m)) \geq g(\mathcal{T})$.

Proof.

(i) This follows by induction on n from $|\tau_0(A)| = |A|$ and $|\tau_{n+1}(A)| = 2|\tau_n(A)|+1$.

(ii) This follows by induction on n from $|\tau_0(A)^*| = a$ and $|\tau_{n+1}(A)^*| = |[\tau_n(A)\tau_n(A)]^*| = |\tau_n(A)^*|^2 + 1$.

(iii) Assuming the contrary and using (i) and (ii) would imply that

$$\frac{\log_2 \rho(n, ab+1)}{(|A|+|B|+2)2^n} > \frac{\log_2 \rho(n+1,a)}{(|A|+1)2^{n+1}}$$
$$\frac{\log_2 \rho(n, ab+1)}{(|A|+|B|+2)2^n} > \frac{\log_2 \rho(n+1,b)}{(|B|+1)2^{n+1}}$$

Multiplying these equations by $(|A|+1)2^{n+1}$ and $(|B|+1)2^{n+1}$, respectively, and adding them gives $2\log_2 \rho(n,ab+1) > \log_2 \rho(n+1,a) + \log_2 \rho(n+1,b)$, which is equivalent to $\rho(n,ab+1)^2 > \rho(n+1,a)\rho(n+1,b)$. This contradicts Lemma 8.

(iv) It suffices to consider only a tree $\mathcal{T}$ with a maximal number $|\mathcal{T}^*|$ of signature patterns. By Theorem 6 no such tree can contain an edge from a vertex with in-degree 2 to a vertex with in-degree 1. Therefore $\mathcal{T}$ is either fully symmetric in the sense that every subtree is a chain or has two identical subtrees, i.e., it is of the form $\tau_n(\mathcal{C}_m)$ for some m and n, or it is symmetric down to a certain level l and is asymmetric below. In the latter case, $\mathcal{T}$ is of the form $\tau_l([AB])$ for some $l > 0$ where A and B are different trees.

In the first case we are done. In the second case we can find a tree $\mathcal{T}_2$, by using (iii), such that $g(\mathcal{T}) \leq g(\mathcal{T}_2)$. By applying (iii) repeatedly we find a sequence of trees $\mathcal{T} = \tau_l([AB]), \mathcal{T}_2 = \tau_{l_2}([A_2B_2]), \mathcal{T}_3 = \tau_{l_3}([A_3B_3]), \ldots$ such that $g(\mathcal{T}) \leq g(\mathcal{T}_2) \leq g(\mathcal{T}_3) \leq \ldots$, where the sequence l, l_2, l_3 is strictly increasing and therefore this process must stop. Note that the depth of $\mathcal{T}_i$ cannot be greater than the depth of $\mathcal{T}$. Thus we can find some tree $\tau_n(\mathcal{C}_m)$ such that $g(\mathcal{T}) \leq g(\tau_n(\mathcal{C}_m))$. □

Lemma 10. *For all* $n \geq 2$

$$\hat{\nu}(n) \geq 2^{\gamma n - 1} \tag{6}$$

Proof. Let $r = (n-2) \bmod 4$ and let $I = \{i_1, i_2, \ldots, i_k\} \subset \mathbf{Z}$ be the set of positions in the binary representation of $(n-2-r)/4$ that are 1. i.e., $(n-2-r)/4 = \sum_{i \in I} 2^i$. A tree $\mathcal{T}$ with $w(\mathcal{T}^*) \geq 2^{\gamma n - 1}$ can be obtained by connecting the trees $\mathcal{C}_{r+2}$ and $\tau_{i_1}(\mathcal{C}_3), \tau_{i_2}(\mathcal{C}_3), \ldots$ in a binary tree, i.e.,

$$\mathcal{T} = [\mathcal{C}_{(r+2)}[\tau_{i_1}(\mathcal{C}_3)[\tau_{i_2}(\mathcal{C}_3) \ldots [\tau_{i_{k-1}}(\mathcal{C}_3)\tau_{i_k}(\mathcal{C}_3)]]]].$$

We have $|\mathcal{T}| = r + 2 + \sum_{i \in I}(1 + |\tau_i(\mathcal{C}_3)|) = n$. It follows from Lemma 8 and from Lemma 9 (ii) that $|\tau_n(\mathcal{C}_3)| \geq 2^{\gamma 2^{n+2}} - 1$. Moreover we have

$$\begin{aligned} |\mathcal{T}^*| &\geq (r+2) \prod_{i \in I} |\tau_i(\mathcal{C}_3)| = (r+2) \prod_{i \in I} (2^{\gamma 2^{i+2}} - 1) \\ &\geq 2^{\gamma(r+2)} \prod_{i \in I} 2^{\gamma 2^{i+2}} \prod_{i \in I} \left(1 - \frac{1}{2^{\gamma 2^{i+2}}}\right) \geq 2^{\gamma n} \prod_{i \in I} \left(1 - \frac{1}{2^{\gamma 2^{i+2}}}\right), \end{aligned}$$

where we have used the fact that $r + 2 > 2^{\gamma(r+2)}$ for $0 \leq r \leq 3$. Let $\beta = 2^{-4\gamma}$. Then

$$\prod_{i \in I} \left(1 - \frac{1}{2^{\gamma 2^{i+2}}}\right) = \prod_{i \in I} \left(1 - \beta^{2^i}\right) \geq 1 - \sum_{j \geq 1} \beta^j = 1 - \frac{\beta}{1-\beta} > 0.5 \qquad \square$$

We can now prove Theorem 7. It follows from Lemma 9 (iv) that it is sufficient to prove $g(\mathcal{T}) < \gamma$ for trees of the form $\tau_n(\mathcal{C}_m)$. According to Lemma 8 we have to find $m \in \mathbf{N}$ which maximizes $\lim_{n \to \infty} \log_2(\rho(n,m))/2^n(m+1)$. This is the case for $m = 3$ as will be shown below. By Lemma 8 it is sufficient to show that for each $m \neq 3$ there exists some n such that $\log_2(\rho(n,m)+1)/((m+1)2^n) < \gamma$. For $m \geq 6$ we have $\log_2(\rho(1,m)+1)/(2(m+1)) < \log_2((m+1)^2)/2(m+1) = \log_2(m+1)/(m+1) < 0.41 < \gamma$ and for the remaining m it can be checked that $\log_2(\rho(2,m)+1)/4(m+1) < 0.41 < \gamma$. This shows that for every tree $\mathcal{T}$ we have

$$|\mathcal{T}^*| \leq 2^{\gamma(|\mathcal{T}|+1)}$$

and therefore $\eta(\Gamma) \leq \gamma$ for the one-time signature scheme Γ given by a maximal antichain of $\mathcal{T}^*$.

From equation (4) and Lemma 10 it follows that $\hat{\mu}(n) \geq \frac{2^{\gamma n}}{2n}$. Thus for all n satisfying $n \geq 2 + \log_2(n)/(\gamma - \delta)$ we have $\hat{\mu}(n) \geq 2^{\delta(n+1)}$. Hence for all $\delta < \gamma$ there exists a one-time signature scheme with efficiency δ. □

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, Optimal tree-based one-time digital signature schemes, *Proc. 13th Symp. on Theoretical Aspects of Computer Science (STACS'96)*, C. Puech and R. Reischuk (eds.), Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 1046, pp. 363–374, 1996.
3. J.N.E. Bos and D. Chaum, Provably unforgeable signatures, *Advances in Cryptology – CRYPTO '92*, E. Brickell (ed.), Lecture Notes in Computer Science, Berlin: Springer Verlag, vol. 740, pp. 1–14, 1993.
4. 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.
5. S. Even, O. Goldreich and S. Micali, On-line/off-line digital signatures, *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science, G. Brassard (ed.), Berlin: Springer Verlag, vol. 435, pp. 263–275, 1990.
6. N. Ferguson, personal communication, 1994.
7. L. Lamport, Constructing digital signatures from a one-way function, Technical Report SRI Intl. CSL 98, 1979.
8. R. Merkle, A certified digital signature, *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science, G. Brassard (ed.), Berlin: Springer Verlag, vol. 435, pp. 218–238, 1990.
9. C. Meyer and S. Matyas, *Cryptography – a new dimension in computer data security*, John Wiley & Sons, Inc., 1982.
10. M. Naor and M. Yung, Universal one-way hash functions and their cryptographic significance, *Proc. 21st ACM Symp. on Theory of Computing (STOC)*, pp. 33–43, 1989.
11. 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.
12. J. Rompel, One-way functions are necessary and sufficient for secure signatures, *Proc. 22nd ACM Symp. on Theory of Computing (STOC)*, pp. 387–394, 1990.
13. C.P. Schnorr, Efficient identification and signatures for smart cards, Advances in Cryptology – Crypto '89, Lecture Notes in Computer Science, G. Brassard (ed.), Berlin: Springer-Verlag, vol. 435, pp. 239–252, 1990.
14. 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.), Berlin: Springer-Verlag, pp. 217–228, 1992.