Digital Signatures based on General One-Way Functions

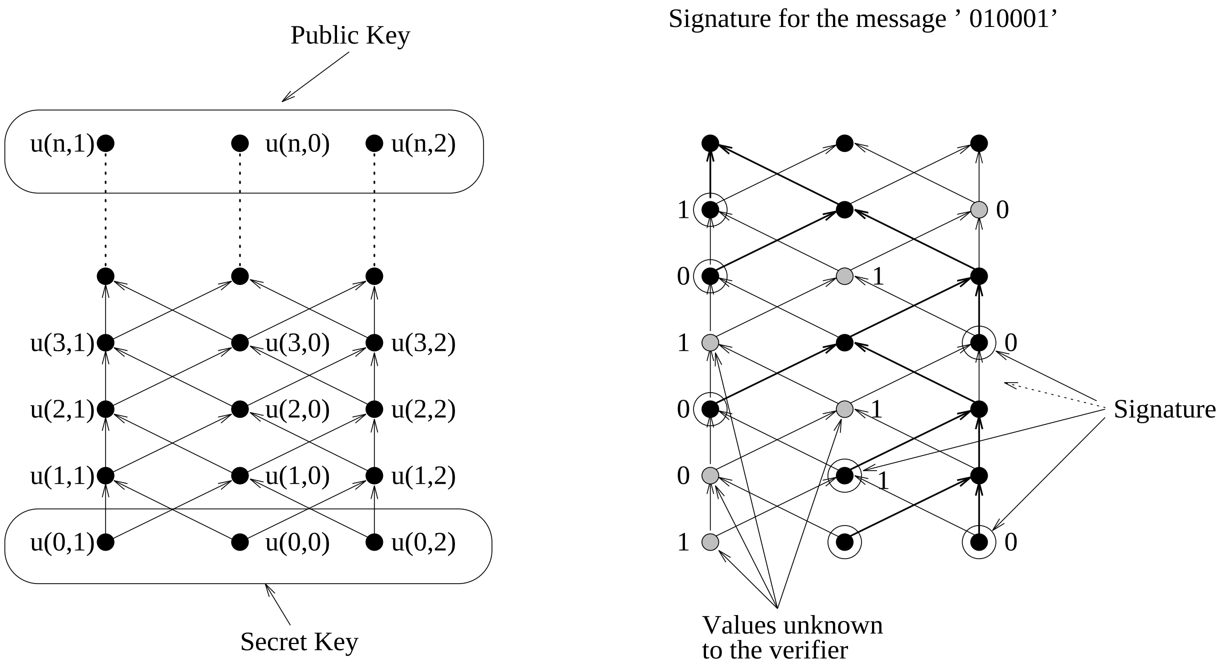
Goal: Design alternative, potentially very efficient digital signature schemes whose security relies on an arbitrary cryptographic principle rather than on one of a very narrow set of number-theoretic problems.

digital signature scheme allows a signer to digitally sign messages such that everyone knowing a public parameter corresponding to the signer, can verify the validity of the message, but such that nobody else can forge signatures.

one-way function is a function f such that given x one can compute $f(x)$, but given y it is hard to find an x such that $f(x) = y$. A one-way function has a **trapdoor** there exists a secret parameter that allows to compute $f^{-1}(y)$ easily.

Conventional digital signature schemes such as the RSA-scheme are based on trapdoor one-way functions which are relatively inefficient and whose security is based on specific number-theoretic conjectures such as "factoring large integers is difficult". In contrast, one-way functions without trapdoor can be based on arbitrary cryptographic principles and are much more efficient.

The following figure illustrates a possible design of such a digital signature scheme.



To set up a digital signature scheme the signer chooses a secret key consisting of three random values $u(0,0), u(0,1)$ and $u(0,2)$. The values of the remaining nodes of the graph are computed using a one-way function f by

$$u(k+1, i+j \bmod 3) = f(u(k,i), u(k,j))$$

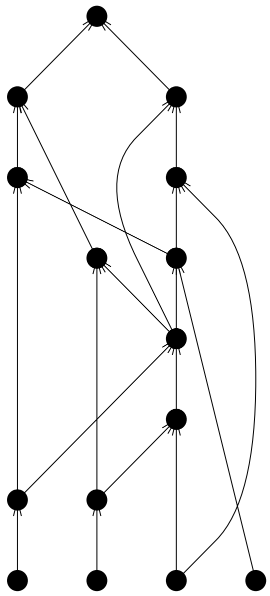
The *public key* consists of the values $u(n,0), u(n,1)$ and $u(n,2)$. and is published in an authenticated manner.

The signature for an n-bit message $M = (m_0, \ldots, m_{n-1})$ is the set of values $\{u(0,0)\} \cup \{u(i, r_i) : 0 \leq i \leq n\}$, where r_i depends on the message bits as follows:

$$\begin{aligned} r_0 &= 1 + m_0 \\ r_{k+1} &= 2r_k - m_k + m_{k+1} \end{aligned}$$

This leads to a digital signature scheme with the following properties:

- The signature can only be computed with knowledge of the secret key. It is computationally infeasible to generate a signature when one knows only the public key.
- The verifier of a signature can check signatures by recomputing part of the public key an comparing it with an authentic copy of the public key.
- The signatures for two different messages are different. Forging signatures is infeasible because no signature can be efficiently computed from the signature for a different message.



A graph with 15 vertices and message space 29 which is the best known result for this size

We have investigated small graphs with a single public-key component and a single secret-key component, a large message space and short signatures.

- A binary tree with n nodes supports a message space of size at most $2^{0.4162\,n}$, and this bound can be achieved by a recursive tree construction.
- The best known general graph with n vertices supports a message space of $O(2^{0.4749\,n})$.
- If the size of the signature is restricted to consist of at most m values then there exist graphs with n vertices and a message space of $O((\frac{n}{m})^{m-1})$.