Digital Signature Scheme (DSS) = (KeyGen, Sign, Verify)

**ALGOs:**

- $SK, VK \leftarrow KeyGen(1^n)$
- $\sigma \leftarrow Sign(m, sk)$
- Accept/Reject $\leftarrow Verify(VK, m, \sigma)$

**PROPERTIES:**

- Correctness
- Security

**Correctness:** $\forall m \in M_k, (M_k$ message space parametrized by security parameter $k), \forall (sk, vk) \in KeyGen(1^n), \forall \sigma \in Sign(m, sk): Verify(VK, m, \sigma) = Accept.$

**Aside:** Can relax definition to allow failure with negligible probability.

**Security. Intuition:** $A$ gets $VK$ and can query Alice for messages of its choice (denoted by query set $Q$ of poly length) and wants to create a signature on a message $m$ that has not already been signed.

$\forall PPT A \exists s.t.
Pr\left[ (sk, vk) \leftarrow KeyGen(1^n); (Q, m, \sigma) \leftarrow A(VK); m \notin Q \land Verify(VK, m, \sigma) = Accept \right] = \nu(k).$

Formally, this is **EXISTENTIAL UNFORGEABILITY** under the **ADAPTIVE CHOSEN MESSAGE ATTACK (EUF-CMA)**.
1976: DH presents idea of DSSs
1977: RSA's signature scheme

\[(N, e), d \leftarrow \text{KeyGen}(1^n)\]
\[\sigma \equiv m^d \mod N \leftarrow \text{Sign}((N, e), d, m) \text{ for } m \in \mathbb{Z}_N^*\]
Accept if \[\sigma^e \equiv m \mod N \leftarrow \text{Verify}((N, e), m, \sigma)\]

**Abstraction:** \(pk, sk \) for a TDP \( f \)
\[\sigma = f_{pk}^{-1}(m)\]
\[m = f_{pk} (\sigma)\]

**Problem:** A can apply TDP

**Rivest:**

\[\text{pk} \rightarrow m_i \quad R \quad \sigma_i \rightarrow \text{soln}\]

\[\text{but R could have come up with this itself, Rivest argues it might be impossible}\]

1984: GMR presents construction overcoming problem
1989: Rompel shows OWFs exist \( \implies \) DSSs exist

In practice, people use full domain hash RSA (FDH-RSA), which is provably secure in the Random Oracle (RO) model (where A gets black-box access to the HF rather than a full description, and expects output that is uniformly random).
FDH-RSA. Let \( H_N \) be a hash function \( H_N : \{0,1\}^* \rightarrow \mathbb{Z}_n^* \)

KeyGen: \((N, e, d)\),

\( \text{Sign}(sk, m) : (H_N(m))^d = \sigma \)

\( \text{Verify}(pk, m, \sigma) : \text{Accept if } \sigma^e = H_N(m) \mod N \)

Modified Abstraction: \( \sigma = f_{pk}^{-1}(H(m)) \)

\( f_{pk}(\sigma) = H(m) \)

Security in the RO model:

\( (N, e), \quad x \in \mathbb{Z}_n^* \)

\( y_i \) where \( v_i = m_j \) was query to \( R \)

\( \sigma_i = \cdot \)

\( y(s.t. y^e = x \mod N) \)

\( \text{pf.} \) \( A \) sets \( VK = (N, e) \), invokes \( A \) on \( VK \).

\( A \) answers \( A \)'s queries to \( H_N \):

- for one of the queries chosen at random, \( v^* \), let \( H(v^*) = x \) (s.t. if \( A \) provides a correct forgery, it is the soln to the TDP).
- for the rest, pick \( y_i \) \( \leftarrow \mathbb{Z}_n^* \), \( H(v_i) = y_i \) \( \mod N \)

\( A \) answers \( A \)'s queries to the Signer:

- if \( m_j = v^* \) (the chosen query), output 1 (fail)
- else, return \( y_i \) (if not already queried, use \( H_N \))

In the wild, it is highly unlikely that \( A \) can issue a forgery on \( m, \sigma \) without applying \( H_N \). So if \( A \) correctly guesses \( v^* \), it wins with the same probability as \( A \).
The reduction correctly guesses which msg will be used for the forgery with probability \( \frac{1}{q} \), where \( q \) is the number of queries to \( H_n \).

**NIST**: DSA (Digital Signature Algorithm)
ECDSA (Elliptic Curve DSA)

neither have a pf of security (even in the RO model)

**Schnorr**: provably secure scheme under the DL assumption in the RO model

2 ingredients for a provably secure DSS construction

1. one-time secure signature scheme
2. transformation from one-time to many-time

\[ \begin{align*}
&\text{VK}_0 \xrightarrow{\text{sign}} [\text{VK}_1, m_1] \\
&\text{sign} \\
\downarrow \\
&\text{VK}_2, m_2 \\
\end{align*} \]

\[ \rightarrow \text{Any signature issued will have a "chain" of signatures back to Alice} \]

\[ \rightarrow \text{huge size blowup} \]

- Optimization: use a tree
  - depth \( k \): create left leaf, sign \( m \), next time use sibling
  - No longer proportional to \# of previously signed messages
  - Assumes msgs can be 2x as long as verification keys - not obvious to achieve, but can using CRHFs

**Merkle Tree**: generate \( \text{VK}s \) at leaves, put hashes of children on inner layers

\[ \begin{align*}
&\text{VK}_1 \xrightarrow{\text{H}} \text{VK}_2 \\
&\text{VK}_0 \xrightarrow{\text{VK}_1} \\
&\text{VK}_0 \xrightarrow{\text{VK}_0} \\
\end{align*} \]
1979: Lamport’s construction of onetime secure DSS

Let $f$ be a OWF.

KeyGen:

<table>
<thead>
<tr>
<th>$x_1^0$</th>
<th>$x_2^0$</th>
<th>\ldots</th>
<th>$x_k^0$</th>
<th>for $x_i \leftarrow \mathcal{E}_0.13^k$ (SK)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1^1$</td>
<td>$x_2^1$</td>
<td>\ldots</td>
<td>$x_k^1$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$f(x_1^0)$</th>
<th>$f(x_2^0)$</th>
<th>\ldots</th>
<th>$f(x_k^0)$</th>
<th>(VK)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x_1^1)$</td>
<td>$f(x_2^1)$</td>
<td>\ldots</td>
<td>$f(x_k^1)$</td>
<td></td>
</tr>
</tbody>
</table>

Sign: $\sigma$ on $k$-bit msg consists of $x_1^{m_1}, x_2^{m_2}, \ldots, x_k^{m_k}$ for $m_1, m_2, \ldots, m_k$ bits of $m$.

Verify: apply $f$ to each $x_i^{m_i}$, check that $f(x_i^{m_i})$ matches verification table.

Intuition: Reduction knows every entry except 1. It picks that entry for the challenge & hopes $A$ chooses to output a forgery that reveals the pre-image (and that it doesn’t have to use the challenge in the signature query).

On input $y = f(x)$ let $\text{SK}$ be known except $x_i^b$ unknown. $\text{VK}$ contains $y$ in that position, everywhere else correct. With $\text{pr} \frac{1}{2}$ $A$’s signing query won’t require $R$ to reveal the unknown value. With $\text{pr} \frac{1}{9}$ the forgery differs in bit $i$, revealing the preimage of $f$. 