Digital Signature Algorithm

I forged your digital signature

DSA: Digital Signature Algorithm

- The Digital Signature Algorithm is a US standard, proposed in 1991 by the NIST
- Along with the DSA, the hash function SHA-1 was also specified and made a standard
- The original DSA was based on the SHA-1 size of 160 bits
- The DSA was covered by a patent, attributed to David Kravitz, a former NSA employee
- Claus Schnorr also claimed one of his patents covers the DSA
- The DSA is a variant of the ElGamal signature algorithm

DSA Parameters

- Two size parameters: L and N
- L is the size of the first prime number p
- The original DSA defined L to be a multiple of 64 and in the range $512 \leq L \leq 1024$
- The security of the DSA is based on the difficulty of the DLP in the group $\mathcal{Z}_p^*$, therefore, $L \geq 1024$, perhaps 2048 or 3072 bits
- The second prime number is q, whose size is N
- The size of q should also match the hash function used
- In the original DSA, the hash function was SHA-1, and $N = 160$
- Considering the new hash functions, we may take $N = 256$ or higher

DSA Parameters

- The prime q divides $p - 1$
- The primes p and q are generated together, by first generating the prime q (which is of size N bits), and checking to see if $p = I \cdot q + 1$ is prime for different I values (which are of size $L - N$ bits), until a prime p is found
- Compute $g \neq 1$ whose multiplicative order is equal to q modulo p, that is, we want $g^q = 1 \pmod{p}$, which is computed as

$$g = h^{\frac{p-1}{q}} \pmod{p}$$

  such that h is an arbitrary integer in the range $1 < h < p - 1$
- The domain parameters are (p, q, g), the same for all users

DSA Key Generation

- Given the domain parameters are (p, q, g), User A selects a random integer $x < q$ as the private key
- The public key of the User A is y such that

$$y = g^x \pmod{p}$$

- User A publishes y in the director and keeps x secret

DSA Signing

- Domain parameters are (p, q, g) and the private key is x
- The hash function $h(\cdot)$ comes with the DSA and generates the hash values of size N bits, which is the size of the prime q
- Given the message m, User A generates a random number rm and computes the signature (s_1, s_2) using

$$\begin{aligned} s_1 &= (g^r \bmod p) \bmod q \\ s_2 &= (h(m) + x \cdot s_1) \cdot r^{-1} \bmod q \end{aligned}$$

- The size of s_1 and s_2 are N bits each
- Therefore, the signature size is $2N$ bits
- The size of the message is unlimited, due to hashing

DSA Verifying

- Domain parameters are (p, q, g) and the public key is y
- The verifier also has access to the hash function $h(\cdot)$
- The verifier receives the message and signature $[m, s_1, s_2]$ and performs the following operations:

$$
\begin{aligned}
w &= s_2^{-1} \bmod q \\
u_1 &= h(m) \cdot w \bmod q \\
u_2 &= s_1 \cdot w \bmod q \\
v &= ((g^{u_1} \cdot y^{u_2}) \bmod p) \bmod q
\end{aligned}
$$

If $v = s_1$, the signature is valid

DSA Correctness

The signer computes $s_2 = (h(m) + x \cdot s_1) \cdot r^{-1} \bmod q$, which gives

$$\begin{aligned} r &= s_2^{-1} \cdot (h(m) + x \cdot s_1) \bmod q \\ &= h(m) \cdot s_2^{-1} + x \cdot s_1 \cdot s_2^{-1} \bmod q \\ &= h(m) \cdot w + x \cdot s_1 \cdot w \bmod q \end{aligned}$$

We now calculate g^r using

$$\begin{aligned} g^r &= g^{h(m)w} \cdot g^{xs_1 w} \pmod{p} \\ &= g^{h(m)w} \cdot y^{s_1 w} \pmod{p} \\ &= g^{u_1} \cdot y^{u_2} \pmod{p} \end{aligned}$$

By taking modulo q both sides above, we find

$$\begin{aligned} s_1 &= g^r \pmod{q} \\ &= (g^{u_1} \cdot y^{u_2} \pmod{p}) \pmod{q} \\ &= v \end{aligned}$$

DSA Domain Parameters Example

- Select $q = 101$ and $p - 1 = 6 \cdot 101 = 606$, we find $p = 607$
- Select $h = 2$, and compute $g = h^{(p-1)/q} = 2^6 = 64$
- The multiplicative order of 64 mod 607 is equal to 101, indeed

$$\begin{aligned} 64^{101} &= 1 \pmod{607} \\ 64^{i} &\neq 1 \pmod{607} \quad \text{for} \quad 1 \leq i \leq 100 \end{aligned}$$

- The domain parameters: $(p, q, g) = (607, 101, 64)$
- The private key $x = 50 < q$, the public key

$$\begin{aligned} y &= g^{x} \pmod{q} \\ &= 64^{50} \pmod{607} \\ &= 76 \end{aligned}$$

DSA Signing Example

- For $h(m) = 10$ and $r = 75$, we generate the signature (s_1, s_2) using

$$
\begin{aligned}
s_1 &= (g^r \bmod p) \bmod q \\
&= (64^{75} \bmod 607) \bmod 101 \\
&= 44 \bmod 101 \\
&= 44 \\
s_2 &= (h(m) + x \cdot s_1) \cdot r^{-1} \bmod q \\
&= (10 + 50 \cdot 44) \cdot 75^{-1} \bmod 101 \\
&= 89 \cdot 66 \bmod 101 \\
&= 16
\end{aligned}
$$

The signature is $(s_1, s_1) = (44, 16)$

DSA Signing Example

- Given the parameters $(p, q, g, y) = (607, 101, 64, 76)$ and the message/signature $(h(m), s_1, s_2) = (10, 44, 16)$, the verifier performs:

$$
\begin{aligned}
w &= s_2^{-1} \bmod q \\
&= 16^{-1} \;=\; 19 \bmod 101 \\
u_1 &= h(m) \cdot w \bmod q \\
&= 10 \cdot 19 \;=\; 89 \bmod 101 \\
u_2 &= s_1 \cdot w \bmod q \\
&= 44 \cdot 19 \;=\; 28 \bmod 101 \\
v &= ((g^{u_1} \cdot y^{u_2}) \bmod p) \bmod q \\
&= ((64^{89} \cdot 76^{28} \bmod 607) \bmod 101 \\
&= (376 \cdot 549 \bmod 607) \bmod 101 \\
&= 44
\end{aligned}
$$

Since $v = s_1$, the signature is valid