DSA (Digital Signature Algorithm)

REPORTER：DONGXUE SHU

What Is DSA (Digital Signature Algorithm)?

- The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures.
- **Key generation,** Key generation has two phases. The first phase is a choice of algorithm parameters which may be shared between different users of the system, while the second phase computes public and private keys for a single user.

DSA

- **Per-user keys:**

1. Given a set of parameters, the second phase computes private and public keys for a single user:
2. Choose a secret key x by some random method, where $0 < x < q$.
3. Calculate the public key $y = g^x \bmod p$.
4. There exist efficient algorithms for computing the modular exponentiations $h^{(p-1)/q} \bmod p$ and $g^x \bmod p$, such as exponentiation by squaring.

DSA

- Let H be the hashing function and m the message
- Generate a random per-message value where $1 < k < q$
- Calculate $r = (g^k \bmod p) \bmod q$
- In the unlikely case that $r = 0$, start again with a different random k
- Calculate $s = k^{-1} (H (m) + x r) \bmod q$
- In the unlikely case that $s = 0$, start again with a different random k

DSA

- Sensitivity

1. With DSA, the entropy, secrecy, and uniqueness of the random signature value k are critical. It is so critical that violating any one of those three requirements can reveal the entire private key to an attacker.[11] Using the same value twice (even while keeping k secret), using a predictable value, or leaking even a few bits of k in each of several signatures, is enough to reveal the private key x.[12]
2. This issue affects both DSA and ECDSA – in December 2010, a group calling itself fail0verflow announced recovery of the ECDSA private key used by Sony to sign software for the PlayStation 3 game console. The attack was made possible because Sony failed to generate a new random k for each signature.[13]
3. This issue can be prevented by deriving k deterministically from the private key and the message hash, as described by RFC 6979. This ensures that k is different for each H(m) and unpredictable for attackers who do not know the private key x.
4. In addition, malicious implementations of DSA and ECDSA can be created where k is chosen in order to subliminally leak information via signatures. For example an offline private key could be leaked from a perfect offline device that only released innocent-looking signatures.[14]

Reject the signature if or is not satisfied.

DSA

- **Verifying**

  - Reject the signature if $0 < r < q$ or $0 < s < q$ is not satisfied.
  - Calculate $w = s^{-1} \bmod q$
  - Calculate $u_1 = H(m) \cdot w \bmod q$
  - Calculate $u_2 = r \cdot w \bmod q$
  - Calculate $v = (g^{u_1} y^{u_2} \bmod p) \bmod q$
  - The signature is invalid unless $v = r$

DSA is similar to the ElGamal signature scheme.