

# Hacking ECDSA based Digital Signature Algorithm

- **ECDSA** is newer and is based on DSA. It has the [same weaknesses](#) as DSA, but it is generally thought to be more secure, even at smaller key sizes. It uses the NIST curves (P256).
- **RSA** is well-regarded and supported everywhere. It is considered quite secure. Common key sizes go up to 4096 bits and as low as 1024. The key size is adjustable. [You should choose RSA](#).
- **DSA** is not in common use anymore, as [poor randomness](#) when *generating a signature* can leak the private key. In the past, it was guaranteed to work everywhere as per [RFC 4251](#), but this is no longer the case. DSA has been standardized as being only 1024 bits (in FIPS 186-2, though FIPS 186-3 has increased that limit). OpenSSH 7.0 and newer actually [disable](#) this algorithm.
- **Ed25519**, while not one you listed, is available on newer OpenSSH installations. It is similar to ECDSA but uses a [superior curve](#), and it does not have the same weaknesses when weak RNGs are used as DSA/ECDSA. It is generally considered to be the strongest mathematically.

The video contains very nice example.