Breaking a Predictable ECDSA Nonce

Jonas Nick

February 4, 2015

Let n, G be the parameters of secp256k1, where n is the curve order and G is the base point. Let d be the private key $\in [1, n-1]$, z be the hash of the message, (r, s) the signature corresponding to the private key d where $r, s \in [0, n-1]$, and k the corresponding nonce.

Then it holds that $s = k^{-1}(z + rd) \mod n$. The github.com/obscuren/secp256k1-go package chooses k to be $z \oplus d$ (xor). At first glance this seems to be ok, since k is unique for each message and it is unpredictable. An attacker can not directly influence z because it is the outcome of a hash function. However, if an attacker obtains multiple signatures, the reuse of d becomes a problem because k becomes in fact predictable.

The problem can be reformulated as a linear system:

$$\alpha = \sum_i d_i 2^i \beta_i \tag{1}$$

where $\alpha = (s-1)z$ and $\beta_i = (r + (2z_i - 1)s)$ and d_i, z_i are the i-th bit in the binary representation of d and z. Thus, the attacker collects 256 signatures and solves the linear system for d. In other words, each signature leaks one bit of the private key.

1 Proof

Note that $a \oplus b = a + b - 2(a \wedge b)$.

$$\begin{aligned}
s &= k^{-1}(z + rd) \\
&= (d \oplus z)^{-1}(z + rd) \\
&= (d + z - 2(d \wedge b))^{-1}(z + rd) \\
\iff ds + zs - 2s(d \wedge z) &= z + rd \\
\iff (s-1)z &= 2s(d \wedge z)(s - r)d \\
&= \sum_i 2^i d_i z_i 2s + \sum_i 2^i d_i (r - s) \\
&= \sum_i d_i 2^i (r + (2z_i - 1)s)
\end{aligned}$$

q.e.d