Breaking a Predictable ECDSA Nonce

Jonas Nick

January 21, 2015

Let n, G be the parameters of secp256k1, where n is the curve order and G is the base point. Let d_A 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_A where $r, s \in [0, n-1]$, and k the corresponding nonce.

Then it holds that $s = k^{-1}(z+rd_A) \mod n$. The github.com/obscuren/secp256k1-go package chooses k to be $z \oplus d_A$ (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_A becomes a problem because k becomes in fact predictable. Note that $a \oplus b = a + b - 2(a \wedge b)$.

$$\begin{aligned}
s &= k^{-1}(z + rd_A) \\
&= (d_A \oplus z)^{-1}(z + rd_A) \\
&= (d_A + z - 2(d_A \wedge b))(z + rd_A) \\
\iff d_A s + zs - 2s(d_A \wedge z) &= z + rd_A \\
\iff (s - r)d_A &= (1 - s)z + 2s(d_A \wedge z) \\
\iff d_A &= ((1 - s)z + 2s(d_A \wedge z))(s - r)^{-1}
\end{aligned}$$

Assume that the attacker obtains a second signature (r', s') over z'. Then it holds that

$$\begin{aligned}
d_A - d_A &= ((1 - s)z + 2s(d_A \wedge z))(s - r)^{-1} - ((1 - s')z' + 2s'(d_A \wedge z'))(s' - r')^{-1} \\
\iff 0 &= (1 - s)z(s - r)^{-1} + 2s(d_A \wedge z)(s - r)^{-1} - (1 - s')z'(s' - r')^{-1} - 2s'(d_A \wedge z')(s' - r')^{-1}
\end{aligned}$$

$$\iff (1-s')z'(s'-r')^{-1} - (1-s)z(s-r)^{-1} = 2s(s-r)^{-1}(z \wedge d_A) - 2s'(s'-r')^{-1}(z' \wedge d_A))$$

Let

$$\begin{aligned}
\alpha &= (1 - s')z'(s' - r')^{-1} - (1 - s)z(s - r)^{-1}, \\
\beta &= 2s(s - r)^{-1}, \\
\gamma &= 2s'(s' - r')^{-1}
\end{aligned}$$

such that

$$\begin{aligned}
\alpha &= \beta(d_A \wedge z) - \gamma(d_A \wedge z) \\
&= \sum_i \beta_i 2^i \sum_j d_{A_j} z_j 2^j - \sum_i \gamma'_i 2^i \sum_j d_{A_j} z'_j 2^j \\
&= \sum_j d_{A_j} 2^j \sum_i (\beta_i z_j - \gamma'_i z'_j) 2^i \\
&= \sum_j d_{A_j} 2^j (z_j \beta - z'_j \gamma')
\end{aligned}$$

where δ_i is the i-th bit of some δ. Now the attacker collects 512 signatures, forms 256 such signature pairs and solves the linear system for d_A with $2^j(z_j k - z'_j k')$ as coefficient.