1 DSA και ECDSA

Οι αλγόριθμοι DSA (Digital Signature Algorithm) και ECDSA (Elliptic Cure Digital Signature Algorithm) χρησιμοποιούνται για την κατασκευή ψηφιακών υπογραφών σε ομάδες διακριτού λογαριθμου. Σε άλλες κατασκευές που έχουμε δει στο μάθημα, όπως το κρυπτοσύστημα ElGamal ή οι δεσμεύσεις Pedersen, μπορούμε να δώσουμε μια ενιαία κατασκευή η οποία βασίζεται μόνο στη δομή της ομάδας (λ.χ. μια κυκλική ομάδα με τάξη πρώτο) και όχι την συγκεκριμμένη υλοποιησή της (πχ μια ελλιπτική καμπύλη). Στους αλγοριθμους που προέρχονται από τον DSA όμως, ο αλγόριθμος αλλάζει ανάλογα με το είδος της ομάδας στο οποίο χρησιμοποιείται.

Αν και το χαρακτηριστικό αυτό δυσχεραίνει την θεωρητική τους ανάλυση, έχουν χρησιμοποιηθει ευρύτατα στην πράξη. Επίσης όμως, έχουν υπάρξει αρκετές απλές επιθεσεις σε υλοποιήσεις του ECDSA, με πιο πρόσφατη αυτή στην υλοποίηση της Java το 2022. Με αυτή την αφορμή θα κάνουμε μια σύντομη επισκόπηση του συστήματος και μερικών απλών επιθέσεων σε αυτό. Για να διευκολυνθούμε στην περιγραφή μας, θα χρησιμοποιήσουμε τη γενίκευση και τους συμβολισμούς των Katz & Lindell (Introduction to Modern Cryptography 2nd ed.).

2 Μια γενίκευση του ECDSA

Έστω μια ομάδα $\mathbb{G}, g, q$ όπου q πρώτος. Θεωρούμε ως χώρο μηνυμάτων το $\mathcal{M} := \{0,1\}^*$, και δύο βοηθητικές συναρτήσεις H, F όπου:

- $H : \mathcal{M} \to \mathbb{Z}_q$, η οποία θεωρούμε ότι είναι μια συνάρτηση κατακερματισμού ανθεκτική σε συγκρούσεις[1].
- $F : \mathbb{G} \to \mathbb{Z}_q$, η οποία διαφέρει ανάλογα με την κατασκευή της $\mathbb{G}$. Ένα κοινό χαρακτηριστικό είναι ότι είναι μη τετριμμένη, δηλαδή παίρνει αρκετές διαφορετικές τιμές στο $\mathbb{Z}_q$ (πχ τουλάχιστον το $\frac{1}{c}$ των πιθανών τιμών για ένα σταθερό c). Ρητά, *δεν* θεωρούμε ότι είναι μονόδρομη: σε πολλές περιπτώσεις είναι εύκολα αντιστρέψιμη.

Ορίζουμε το σχήμα υπογραφών ως εξής:

- $\mathsf{Gen}(1^\lambda)$ Θέτουμε $(\mathbb{G}, q, q) \leftarrow \mathrm{GGen}(1^\lambda)$, επιλέγουμε ένα ιδιωτικό κλειδί $x \leftarrow \mathbb{Z}_q$, υπολογίζουμε το δημόσιο κλειδί $h \leftarrow g^x$ και επιστρέφουμε $(sk := x, vk := h)$.
- $\mathsf{Sign}(sk, m)$ Επιλέγουμε ένα τυχαίο $k \leftarrow \mathbb{Z}_q^*$ και θέτουμε $r \leftarrow F(g^k)$. Υπολογίζουμε $s \leftarrow k^{-1}(H(m) + xr) \mod q$. Εάν έχουμε $r = 0$ ή $s = 0$ δοκιμάζουμε ξανά από την αρχή. Αλλιώς, επιστρέφουμε την υπογραφή $\sigma = (r, s)$.
- $\mathsf{Ver}(h, m, \sigma)$ Διαβάζουμε το σ ως $\sigma = (r, s)$. Ελέγχουμε ότι $r \neq 0 \mod q$, και $s \neq 0 \mod q$, και υπολογίζουμε $w \leftarrow s^{-1} \mod q$. Έπειτα ελέγχουμε αν ισχύει ότι:

$$r \stackrel{?}{=} F(g^{w \cdot H(m)} h^{w \cdot r})$$

Εάν οποιοσδήποτε έλεγχος αποτύχει, επιστρέφουμε 0, διαφορετικά επιστρέφουμε 1.

Ορθότητα Αρχικά ελέγχουμε ότι ο αλγόριθμος υπογραφής τερματίζει με συντρηπτική πιθανότητα, και ότι μια σωστά κατασκευασμένη υπογραφή θα γίνει αποδεκτή από το Ver.

Στον αλγόριθμο υπογραφής, από υπόθεση, με πιθανότητα τουλάχιστον $\frac{1}{c} - 1\frac{}{q}$ το $F(g^k)$ θα είναι μη μηδενικό, αφού η F παίρνει τουλάχιστον $\frac{q}{c}$ τιμές, κάθε μία με τουλάχιστον μια αντίστροφη εικόνα. Έπειτα, αφού το k είναι αντιστρέψιμο, το $s = k^{-1}(H(m) + xr)$ μηδενίζει μόνο όταν μηδενίζει και η παράσταση $(H(m) + xr)$. Για δεδομένα x, m, όμως αυτή μηδενίζει για μία μόνο τιμή του r. Συνολικά, μια προσπάθεια

[1] Και άρα μονόδρομη αφού το $\mathcal{M}$ είναι πολύ μεγαλύτερο από το $\mathbb{Z}_q$

επιτυγχάνει με πιθανότητα τουλάχιστον $\frac{1}{c} - 2_{\overline{q}}$. Άρα, η πιθανότητα αποτυχίας γίνεται εκθετικά μικρή ως προς τον αριθμό των προσπαθειών.

Για την επαλήθευση της υπογραφής, αντικαθιστούμε το s με $k^{-1}(H(m)+xr)$, και έχουμε $w = \frac{k}{(H(m)+xr)}$. Έπειτα υπολογίζουμε $T = g^{w \cdot H(m)} h^{w \cdot r}$, ή ισοδύναμα $T = g^{w \cdot H(m)+xwr}$. Αντικαθιστώντας το w έχουμε $T = g^k$ και άρα ισχύει ότι $r = F(T)$. Επίσης, σημειώνουμε ότι από την κανονική συνάρτηση υπογραφής δεν επιστρέφονται ποτέ μηδενικές τιμές, οπότε όλοι οι έλεγχοι πετυχαίνουν.

3 Επιθέσεις στον ECDSA

3.1 Επανάχρηση του k

Κατα την παραγωγή υπογραφών, η τυχαία επιλογή του k είναι ιδιαίτερα σημαντική: εάν χρησιμοποιηθεί το ίδιο k για διαφορετικά μηνύματα m_1, m_2 τότε υπάρχει άμεση διαρροή του ιδιωτικού κλειδιού: Έστω δύο υπογραφές (r, s_1) και r, s_2 για δύο μηνύματα m_1, m_2, που προέρχονται από επιλογή του ίδιου k (οπότε και τα $r = F(g^k)$ συμπίπτουν).

Εύκολα υπολογίζουμε ότι σε αυτή την περίπτωση $s_1 - s_2 = k^{-1}(H(m_1) - H(m_2)) \mod q$. Αφού τα m_1, m_2 είναι γνωστά και οι εικόνες τους με μεγάλη πιθανότητα δεν είναι ίσες, υπολογίζουμε το k ως $k = \frac{H(m_1)-H(m_2)}{s_1-s_2}$.

Έπειτα γνωρίζοντας το k και το r μπορούμε να εξάγουμε το x από το s_1. Έχουμε $s_1 = k^{-1}(H(m) + xr)$, άρα $s_1^k = H(m) + xr$ οπότε έχουμε $x = \frac{s_1^k - H(m)}{r}$.

3.2 Λανθασμένες βελτιστοποιήσεις στον έλεγχο υπογραφών

Το 2022, έγινε γνωστή μια αδυναμία με το προσωνύμιο "Psychic Signatures" (CVE-2022-21449), κατα την οποία πρόσφατες εκδόσεις της γλώσσας Java έκαναν δεκτές υπογραφές της μορφής $(r, s) = (0, 0)$. Αυτό μοιάζει εκ πρώτης όψης αναπάντεχο, αφού στον ορισμό της Ver υπάρχει ρητός έλεγχος για μηδενικές τιμές. Επιπλέον, ακόμα και αν έχει παραληφθεί ο έλεγχος, μοιάζει αδύνατος ο υπολογισμός του $w \leftarrow s^{-1}$.

Μία πιθανή εξήγηση είναι ότι ακριβώς λόγω του αναμενόμενου σφάλματος στο $w \leftarrow s^{-1}$ εάν το s είναι μηδέν, ο έλεγχος $s \neq 0$ μοιάζει περιττός. Μια άλλη, φαινομενικά ανεξάρτητη σκέψη, είναι ότι για $s \neq 0$ ισχύει $s^{-1} = s^{q-2}$, αφού το s θα ανοίκει στην ομάδα Z_q^* με πράξη τον πολλαπλασιασμό και τάξη $q - 1$. Με αυτό τον τρόπο μπορούμε να αποφύγουμε την υλοποίηση του αλγορίθμου του Ευκλείδη και απλά να βασιστούμε στην ύψωση στοιχείου σε δύναμη.

Κάθε μία από τις δύο αυτές προτάσεις είναι απο μόνη της ασφαλής. Όμως, σε συνδιασμό, καταστρέφουν την ασφάλεια του συστήματος: Αν θέσουμε $s = 0$ και υπολογίσουμε το w ως $w \leftarrow s^{q-2}$, έχουμε $w = 0$. Το $T = g^{w \cdot H(m)+xr}$ υπολογίζεται ως $T = g^0$, και ο τελικός έλεγχος είναι $r \stackrel{?}{=} F(g^0)$. Άρα, εάν επιτρέπεται μηδενική εισοδος στο s και ο υπολογισμός του w γίνεται όπως παραπάνω, υπάρχει τιμή του r η οποία είναι αποδεκτή για κάθε μήνυμα. Στην πράξη, η τιμή αυτή είναι $r = 0$, το οποίο επίσης ρητά πρέπει να απορρίπτεται από την Ver. Εδώ, δεν είναι εύκολο να πιθανολογήσουμε γιατί απουσιάζει ο έλεγχος.