Exercise Set 1

Prof. Kenny Paterson
Jan Gilcher, Kien Tuong Truong

Problem 1 (Known Nonces). Show that an attacker who learns the nonce value k corresponding to an ECDSA signature (r, s) for a known message m can efficiently recover the secret signing key x. (Hint: recall that in ECDSA, $s = k^{-1}(h + xr) \bmod q$ where h is derived from the message.)

Problem 2 (Repeated Nonces). Suppose that a signer manages to prevent the aforementioned attack by making sure that the nonce is concealed from the attacker. However, due to a bug in the implementation, a nonce value gets repeated after every 10 signatures.

- Suppose an attacker sees two signatures (r_1, s_1) and (r_2, s_2). Argue that the attacker can trivially detect if a nonce k has been repeated across these two signatures.
- Now suppose that these two signatures (r_1, s_1) and (r_2, s_2) generated using the same nonce k correspond to messages m_1 and m_2, respectively, such that $m_1 \neq m_2$. Show that the attacker can recover the secret signing key x.

Problem 3 (Knapsack Cryptanalysis). In the lecture, you have seen basic definitions for lattices. In order to showcase their usefulness, we apply them to solve a subset of instances of the subset-sum problem. This comes up in the context of cryptanalysis, as there exists a scheme, proposed by Merkle and Hellman, which bases its security on the hardness of solving the subset-sum problem. The technique we show, due to Lagarias and Odlyzko, uses lattices to break the Merkle-Hellman cryptosystem.

Let $X \in \mathbb{N}$ and $\mathbf{a} = (a_1, ..., a_n)^\top$, where $a_i \in \{1, ..., X\}$. Let $\mathbf{x} \in \{0,1\}^n$ and let $s = \langle \mathbf{a}, \mathbf{x} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the dot product of two vectors. The *subset-sum problem* is: given an instance $(\mathbf{a}, s)$, find $\mathbf{x}$. Roughly speaking: given a set of values $a_1, ..., a_n$, find a subset of those values that sum to s (where s is created such that a solution will exist). The vector $\mathbf{x}$ decides which values are included in the sum.

If X is small, there exists a pseudo-polynomial algorithm to solve the subset sum (pseudo-polynomial meaning its complexity is linear in X, rather than in $\log(X)$). Thus, we will focus on instances where X is at least exponentially large. In particular, our algorithm will work for $X \geq 2^{n^2(1/2+c)}$, for some small $c > 1$.[1]

Assume that $s \geq (\Sigma_{i=1}^n a_i)/2$. If not, set $s' = (\Sigma_{i=1}^n a_i) - s$ and solve the instance $(\mathbf{a}, s')$ (in this case, our attack will yield a solution $\mathbf{x}_{\text{inv}}$, equal to $\mathbf{x}$ with all entries flipped).

Set $\beta = \left\lceil \sqrt{n \cdot 2^n} \right\rceil$ and define a lattice $\mathcal{L}$ using the basis B:

$$B = \begin{bmatrix} 1 & 0 & \cdots & 0 & -\beta a_1 \\ 0 & 1 & \cdots & 0 & -\beta a_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -\beta a_n \\ 0 & 0 & \cdots & 0 & \beta s \end{bmatrix}$$

- Prove that $\tilde{\mathbf{x}} = (x_1, ..., x_n, 0) \in \mathcal{L}$. In particular, find the vector of coefficients $\mathbf{y}$ that generates the vector $\tilde{\mathbf{x}}$ (i.e. $\tilde{\mathbf{x}} = \mathbf{y}B$). Find an upper bound on $\|\tilde{\mathbf{x}}\|$.

[1] These are called *low-density* instances of the knapsack problem.

Solution: Consider $\mathbf{y} = (x_1, ..., x_n, 1)$. A simple matrix-vector multiplication shows that $\tilde{\mathbf{x}} = \mathbf{y}B$. Intuitively: the first n elements are fixed due to the presence of the identity matrix in the top left. All that is left is to compensate for the last element, which we know must be 0. Fix $\mathbf{y} = (x_1, ..., x_n, y_{n+1})$ and compute the multiplication: we are left with $(x_1, ..., x_n, (-\beta a_1 \cdot x_1) + ... + (-\beta a_n \cdot x_n) + \beta s \cdot y_{n+1})$. Since $\sum_{i=1}^{n} a_i x_i = s$, then clearly $y_{n+1} = 1$. $\|\tilde{\mathbf{x}}\|$ is maximized when all $x_i = 1$. In this case, the norm is $\sqrt{n}$, so $\|\tilde{\mathbf{x}}\| \leq \sqrt{n}$.

- Prove that any vector with last coordinate different from 0 must have a length of at least $2^{n/2} \cdot \|\tilde{\mathbf{x}}\|$.

  Hint: the last coordinate of the vector is a linear combination of elements from the last column of the matrix.

Solution: Following the hint, any vector with last coordinate different from zero must have as last coordinate a multiple of β. We know that $\|\tilde{\mathbf{x}}\| \leq \sqrt{n}$ and that $\beta > \sqrt{n} \cdot 2^{n/2}$, thus $\beta > \|\tilde{\mathbf{x}}\| \cdot 2^{n/2}$.

- Prove that, with high probability, any vector $\tilde{\mathbf{z}}$ with last coordinate 0 and length $\|\mathbf{z}\| < \beta$ must be an integer multiple of $\tilde{\mathbf{x}}$

  Hint: fix an arbitrary vector $\tilde{\mathbf{z}}$ which is not an integer multiple of $\tilde{\mathbf{x}}$. Prove that such a vector has probability $1/X$ of belonging to $\mathcal{L}$. Then, apply a union bound over all possible values of $\tilde{\mathbf{z}}$.

Solution: Fix an arbitrary non-zero vector $\tilde{\mathbf{z}} = (\mathbf{z}, 0)$, where $\mathbf{z} = (z_1, ..., z_n)$, so that $\tilde{\mathbf{z}}$ is not an integer multiple of $\tilde{\mathbf{x}}$ and let $\mathbf{y} = (\mathbf{z}, z_{n+1})^T$ be the corresponding coefficient vector, for some $z_{n+1} \in \mathbb{Z}$.

If $\tilde{\mathbf{z}} \in \mathcal{L}$, then:

$$\sum_{i=1}^{n} a_i z_i = z_{n+1} \cdot s = z_{n+1} \cdot \sum_{i=1}^{n} a_i x_i$$

This means that:

$$\sum_{i=1}^{n} a_i (z_i - z_{n+1} x_i) = 0$$

Since $\tilde{\mathbf{z}}$ is not an integer multiple of $\tilde{\mathbf{x}}$, it must be the case that at least one of the z_i is not an integer multiple of the corresponding x_i, and thus it must be that $z_i - z_{n+1} x_i \neq 0$. Without loss of generality, assume that this holds for $i = 1$. In this case, we have that:

$$a_1 = \frac{-\sum_{i=1}^{n} a_i (z_i - z_{n+1} x_i)}{z_1 - z_{n+1} x_1}$$

Call E the event that the above happens, that is: E is the event that, for a_i chosen uniformly at random from $\{1, ..., X\}$, the equation holds for our fixed choice of $\mathbf{y}$. Then:

$$\Pr[\tilde{\mathbf{z}} \in \mathcal{L}] = \Pr[E] \leq \frac{1}{X}$$

Applying the union bound over all possible choices of $\mathbf{y}$ will yield the result. In order to estimate the possible choices, we have to consider two constraints:

- $\|\mathbf{z}\| < \beta$, thus all $|z_i| < \beta$ for $i = 1, ..., n$.
- $s \cdot |z_{n+1}| = |s \cdot z_{n+1}| = \left|\sum_{i=1}^{n} a_i \cdot z_i\right| \leq \|\mathbf{z}\| \sum_{i=1}^{n} a_i$ (apply Cauchy-Schwarz and the triangle inequality). Thus, $|z_{n+1}| \leq 2 \cdot \|\mathbf{z}\|$, given the assumption $s \geq \left(\sum_{i=1}^{n} a_i\right)/2$.

Thus, for each z_i (for $i = 1, ..., n$) we have $2\beta + 1$ choices, while for z_{n+1} we have $4\beta + 1$ choices.

An aggressive upper bounding then yields that there are:

$$(2\beta + 1)^n (4\beta + 1) \leq (5\beta)^{n+1} \leq 2^{3/2 \cdot n^2}$$

possible choices for $\mathbf{y}$. In our case we have that $X = 2^{n^2(1/2+c)}$, so the probability that $\tilde{\mathbf{z}} \in \mathcal{L}$ over all choices of $\tilde{\mathbf{z}}$ is, by union bound:

$$2^{n^2(1/2+1)} \cdot \frac{1}{X} = 2^{n^2(1/2+1) - n^2(1/2+c)} = 2^{n^2(1-c)} = 2^{-\Omega(n^2)}$$

which is negligibly small.

This exercise shows that, assuming you are given some algorithm that finds a short vector of the lattice, you can solve the subset-sum problem, since all non-useful vectors are either much longer than the wanted vector or they are negligibly rare. You will see the specifics of such an algorithm in the next lectures.

For context, the original Merkle-Hellman cryptosystem uses $\mathbf{x}$ as the message and sends the subset-sum instance $(\mathbf{a}, s)$ as the ciphertext. To enable decryption, instead of randomly sampling each a_i, the key generation algorithm choses them in a structured way that, however, makes each a_i look randomly distributed to anyone that does not have the secret key. This attack allows one to recover the message without knowing the key.