Cryptosystems and LLL

.

Thomas Baignères

EPFL - LASEC

thomas.baigneres@epfl.ch

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Contents

1. A survey on lattices
2. GPG and ElGamal Signatures
3. The attack against GPG-ElGamal signatures
4. Implementation of RSA in GPG

$\rightsquigarrow$ Based on Phong Nguyen's PhD Thesis and Eurocrypt'04 article

LASEC

Definition of a lattice

Let $\mathbf{f}_1, \ldots, \mathbf{f}_n$ be linearly independent vectors of $\mathbb{R}^n$

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} u_i \mathbf{f}_i \;\middle|\; u_i \in \mathbb{Z} \right\}$$

is a (full-ranked) *lattice*. The $\mathbf{f}_i$'s are a *basis* of $\mathcal{L}$.

If the $\mathbf{f}_i$'s are considered like rows of the $n \times n$ matrix

$$\mathsf{F} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \end{pmatrix}$$

then

$$\mathcal{L} = \{ \mathbf{u}\mathsf{F} \mid \mathbf{u} \in \mathbb{Z}^n \} \ .$$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Determinant of a lattice

The *determinant* of a lattice $\mathcal{L}$ is

$$\det(\mathcal{L}) = |\det(\mathsf{F})|$$

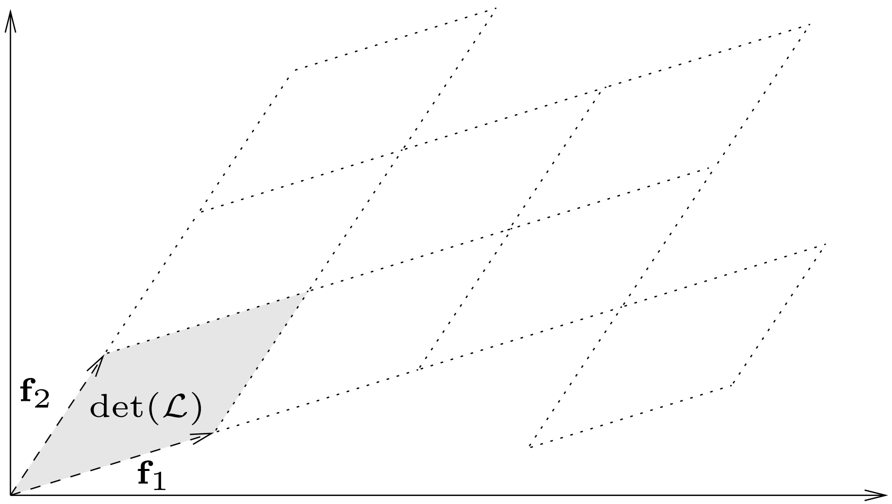
It is well defi ned. If F and G are two basis of $\mathcal{L}$, there exists some unimodular matrix P s.t.

$$\mathsf{F} = \mathsf{P} \times \mathsf{G} \;\Rightarrow\; \det(\mathsf{F}) = \det(\mathsf{P}) \cdot \det(\mathsf{G}) = \pm \det(\mathsf{G})$$

The determinant is independent of the basis choice.

It has a simple geometrical interpretation . . .

Geometrical interpretation of the determinant



In dimension 2 $\rightsquigarrow$ *area* of the parallelogram defined by $\mathbf{f}_1, \mathbf{f}_2$.

In dimension n $\rightsquigarrow$ *volume* of the parallelepiped defined by the $\mathbf{f}_i$

$\Rightarrow$ Hadamard inequality:

$$\det(\mathcal{L}) \leq \prod_{i=1}^{n} \| \mathbf{f}_i \|$$

Typical distance in $\mathcal{L} \longrightarrow \det(\mathcal{L})^{\frac{1}{n}}$

Shortest Vector Problem (SVP)

The *Shortest Vector Problem* (SVP) is to find a smallest non-zero vector in $\mathcal{L}$, i.e.

$$\mathbf{u} \in \mathcal{L} \setminus \{\mathbf{0}\} \quad \text{s.t.} \quad \|\,\mathbf{u}\,\| \leq \|\,\mathbf{v}\,\| \quad \forall \mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}$$

It is proved [Ajtai98] that SVP is NP-hard (under randomized reduction).
$\rightsquigarrow$ Can we *approximate* SVP ?

Find

$$\mathbf{u} \in \mathcal{L} \setminus \{\mathbf{0}\} \quad \text{s.t.} \quad \|\,\mathbf{u}\,\| \leq f(n)\,\|\,\mathbf{v}\,\| \quad \forall \mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}$$

LLL approximates SVP to within a factor $f(n) = 2^{\frac{n-1}{2}}$ in polynomial time.

LASEC

Closest Vector Problem (CVP)

Let $\mathbf{x} \in \mathbb{R}^n$ (not necessarily in $\mathcal{L}$).

The *Closest Vector Problem* (CVP) is to find $\mathbf{u} \in \mathcal{L}$ minimizing the distance between $\| \mathbf{x} - \mathbf{u} \|$, i.e.

$$\mathbf{u} \in \mathcal{L} \quad \text{s.t.} \quad \| \mathbf{x} - \mathbf{u} \| \leq \| \mathbf{x} - \mathbf{v} \| \quad \forall \mathbf{v} \in \mathcal{L}$$

It is proved [GMSS99] that SVP is not harder than CVP.

Approximating CVP is to find

$$\mathbf{u} \in \mathcal{L} \quad \text{s.t.} \quad \| \mathbf{x} - \mathbf{u} \| \leq f(n) \| \mathbf{x} - \mathbf{v} \| \quad \forall \mathbf{v} \in \mathcal{L}$$

The embedding method is an heuristic to reduce CVP to SVP...

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

The embedding method

$\mathcal{L}$ is a lattice of basis $\mathbf{f}_1, \ldots, \mathbf{f}_n$ (rows of F). CVP of $\mathbf{x} \in \mathbb{R}^n$?

Construct a lattice $\mathcal{L}'$ (of dimension $n+1$) of basis

$$\mathsf{F}' = \left(\begin{array}{c|c} \mathsf{F} & \mathbf{0} \\ \hline \mathbf{x} & 1 \end{array}\right)$$

As

$$\begin{cases} \dim(\mathcal{L}') & \approx \dim(\mathcal{L}) \\ \det(\mathcal{L}') & = \det(\mathcal{L}) \end{cases}$$

we consider that "*being short*" in $\mathcal{L}'$ also means "*being short*" in $\mathcal{L}$.

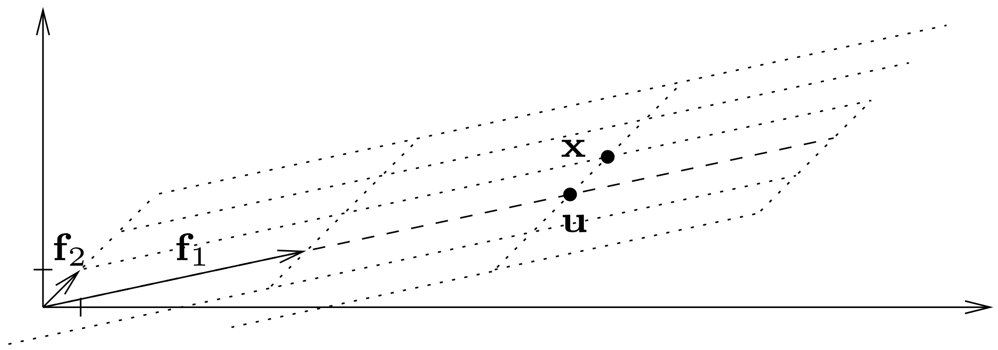
The embedding method (2)

The point

$$(-u_1, \ldots, -u_n, 1) \times \left(\begin{array}{c|c} \mathrm{F} & \mathbf{0} \\ \hline \mathbf{x} & 1 \end{array}\right) = (\mathbf{x} - \mathbf{u}, 1)$$

is supposed to a short vector of $\mathcal{L}' = \{\mathbf{u}\mathrm{F}' \mid \mathbf{u} \in \mathbb{Z}^n\}$.

$\Rightarrow$ solving SVP in $\mathcal{L}'$ (e.g. $\mathbf{f}_2$) solves CVP in $\mathcal{L}$ (e.g. $\mathbf{f}_2, \mathbf{x} \rightsquigarrow \mathbf{u}$).



GnuPG

- → GnuPG (GPG) is a full implementation of the OpenPGP standard.
- → Open-source effort supported by German government.
- → Provides encryption and signatures for securing email.
- → Supports DSA, RSA, AES, 3DES, Blowfi sh, Twofi sh, CAST5, MD5, SHA-1, RIPEMD-160, and TIGER.

GnuPG Signatures

$\longrightarrow$ Standard mode: DSA (signature keys) + ElGamal (encryption keys).

$\longrightarrow$ Expert mode (1): ElGamal for both signature and encryption.

$\longrightarrow$ Expert mode (2): RSA for both signature and encryption.

Padding used by GnuPG

→ When RSA and ElGamal are used, the message is hashed, and the hash value is encoded as specified in PKCS# v1.5.

→ $\mathrm{0x00}||\mathrm{0x01}||\mathrm{0xFF}||\ldots||\mathrm{0xFF}||\mathrm{0x00}||\mathrm{H}(m)$.

ElGamal Signatures

→ Public parameters: a prime p and a generator g of $\mathbb{Z}_p^*$.

→ Private key: $x \in_{\mathsf{R}}]0, p-1[$.

→ Public key is $y = g^x \bmod p$.

→ Signature of m: take a random $k \in_{\mathsf{R}}]0, p-1[$ and compute

$$\begin{aligned} a &= g^k \bmod p \\ b &= (m - ax)k^{-1} \bmod (p-1) \end{aligned}$$

→ Signature is $\sigma = (a, b)$.

→ A signature is valid if the following congruence holds:

$y^a a^b \equiv g^m \pmod p$ since $y^a a^b \equiv g^{ax} g^{bk} \equiv g^{ax+bk} \equiv g^m \pmod p$

ElGamal Key Generation

→ First, a large prime p is generated pseudo-randomly, such that the factorization of $\frac{p-1}{2}$ is known.

→ All the factors of $\frac{p-1}{2}$ must have a bit length larger than a threshold q_{bit} depending of the bitlength of p.

→ q_{bit} is given by the so-called *Wiener's table*:

$\lvert p \rvert$	512	768	1024	1280	...
q_{bit}	119	145	165	183	...

→ Remember that the size of p is always larger than $4 \cdot q_{\text{bit}}$!

LASEC

ElGamal Key Generation (2)

→ Once q is selected, one fi nds a generator g of $\mathbb{Z}_p^*$ as follows:

→ If 3 is not a generator, then on tries 4, and so on.

→ g is likely to be small, but Bleichenbacher's forgery of ElGamal signatures does not seem to apply, because of the size of the factors of $\frac{p-1}{2}$.

→ The ElGamal private exponent x must be chosen uniformly at random on $0 < x < p - 1$, but, for *efficiency reasons*, it is chosen as $0 < x < \frac{3q_{\mathrm{bit}}}{2}$.

→ The ElGamal random nonce k must be chosen uniformly at random on $0 < k < p - 1$, but, for *efficiency reasons*, it is chosen as $0 < k < \frac{3q_{\mathrm{bit}}}{2}$.

Solving a congruence with a lattice

The attacker has access to a valid signature $\sigma = (a, b)$ of a message $m \in \mathbb{Z}_{p-1}$.

The following congruence should hold:

$$ax + bk \equiv m \pmod{p-1}$$

Unknowns: x and k (very small)

Solving the congruence $\leadsto$ solving a CVP instance in a lattice!

The lattice we need

Lemma: Let $(\alpha, \beta) \in \mathbb{Z}^2$ and $n \in \mathbb{N}$. Let

$$\begin{aligned} d &= \gcd(\alpha, n) \\ e &= \gcd(\alpha, \beta, n)\ . \end{aligned}$$

Let $\mathcal{L} = \{(u, v) \in \mathbb{Z}^2$ s.t. $\alpha u + \beta v \equiv 0 \pmod{n}\}$. Then

- $\mathcal{L}$ is a two dimensional lattice of $\mathbb{Z}^2$
- $\det(\mathcal{L}) = \frac{n}{e}$
- There exists $u \in \mathbb{Z}$ such that $\alpha u + (\beta/e)d \equiv 0 \pmod{n}$
- The vectors $(n/d, 0)$ and $(u, d/e)$ form a basis of $\mathcal{L}$

Nguyen's attack (1)

Let

$$\mathcal{L} = \{(u, v) \in \mathbb{Z}^2 \mid au + bv \equiv 0 \pmod{p-1}$$

$\mathcal{L}$ is a two-dimensional lattice.

With $d = \gcd(a, p-1)$ and $e = \gcd(a, b, p-1)$, there exists $u \in \mathbb{Z}$ such that $au + (b/e)d \equiv 0 \pmod{p-1}$.

A basis of $\mathcal{L}$ is

$$\mathsf{B} = \begin{pmatrix} \frac{p-1}{d} & 0 \\ u & \frac{d}{e} \end{pmatrix}$$

$\det(\mathcal{L}) = \frac{p-1}{e} = \frac{p-1}{\gcd(a,b,p-1)} \approx p$ (by construction)

$\Rightarrow$ *Typical distance in the lattice* $\sqrt{\det(\mathcal{L})} \approx \sqrt{p}$

Nguyen's attack (2)

Find $(x', k') \in \mathbb{Z}^2$ such that $ax' + bk' \equiv m \pmod{p-1}$

For this:

Find $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{Z}$ such that

$$a\lambda_1 + b\lambda_2 + (p-1)\lambda_3 = e \quad \text{(with EEA)}$$

As $e \mid m$ (recall $ax + bk \equiv m \pmod{p-1}$),
multiplying λ_1, λ_2 by $\frac{m}{e}$ leads to x', k'

Nguyen's attack (3)

Let

$$\mathbf{l} = (x' - x, k' - k) \quad \text{(unknown vector } \in \mathcal{L}\text{)}$$

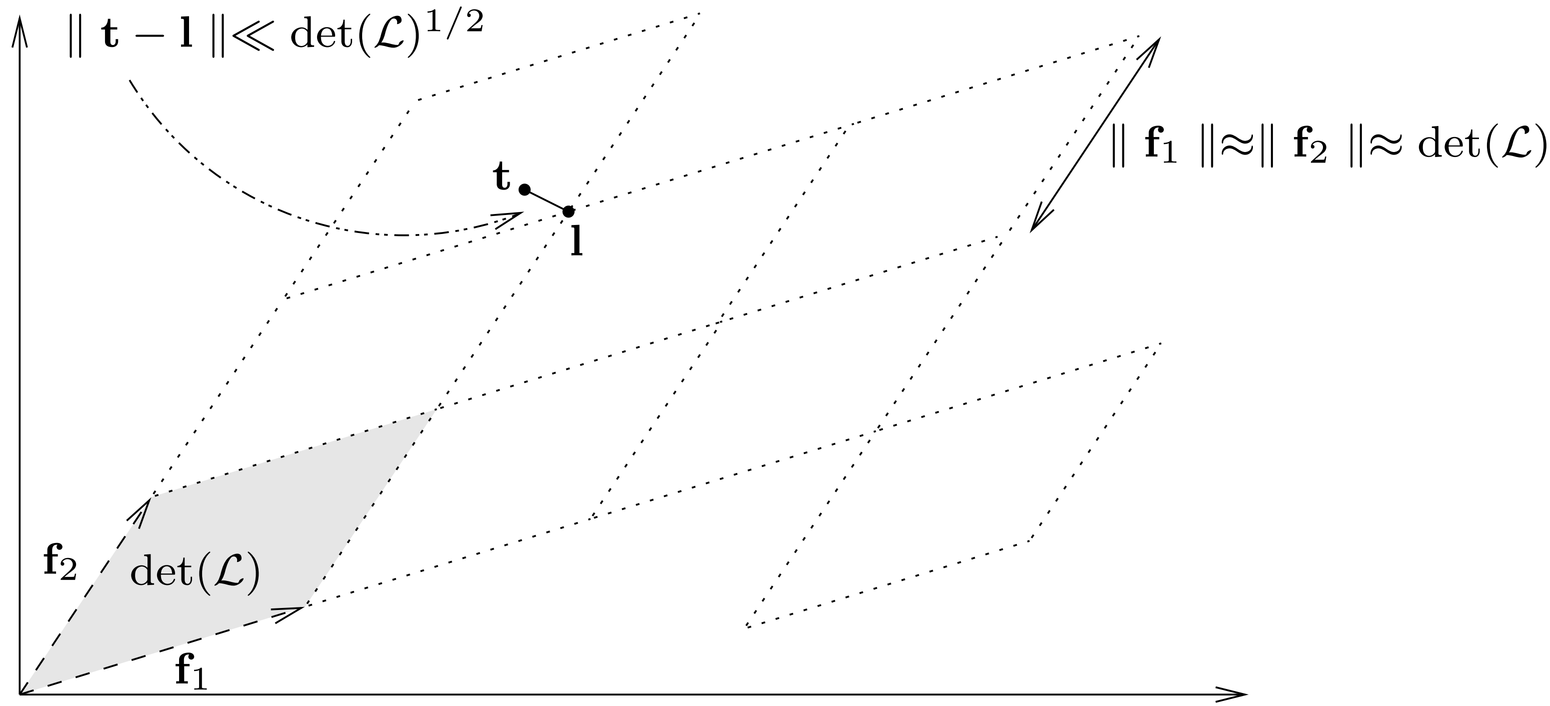
$$\mathbf{t} = (x' - 2^{3q_{\mathrm{bit}}/2}, k' - 2^{3q_{\mathrm{bit}}/2}) \quad \text{(known vector } \notin \mathcal{L}\text{)}$$

As $|x| \approx |k| \approx 3q_{\mathrm{bit}}/2$,

$$\| \mathbf{t} - \mathbf{l} \| \approx 2^{\frac{3q_{\mathrm{bit}}-1}{2}} \ll 2^{2q_{\mathrm{bit}}} < \sqrt{p} \approx \sqrt{\det(\mathcal{L})}$$

$\Rightarrow$ *Heuristic :* $\mathbf{l} \in \mathcal{L}$ *is the closest vector of* $\mathbf{t}$

Nguyen's attack (4)



Solving a CVP instance in $\mathcal{L}$ (e.g. with the embedded method) allows to recover $\mathbf{l} = (x' - x, k' - k)$ and thus x and k, i.e.

$\rightsquigarrow$ *the private key of the signer is recovered !*

LASEC

Yet another attack

Let K be a *large* integer let L' be the 4-dimensional lattice defi ned by

$$\mathbf{B}' = \begin{pmatrix} (p-1)K & 0 & 0 & 0 \\ -mK & 2^{3q_{\mathrm{bit}}/2} & 0 & 0 \\ bK & 0 & 1 & 0 \\ aK & 0 & 0 & 1 \end{pmatrix}.$$

As $ax + bk \equiv m \pmod{p-1}$, there exists $\lambda \in \mathbb{Z}$ s.t.

$$(p-1)\lambda - m + bk + ax = 0$$

so that

$$\begin{aligned} \mathbf{l}' &= (\lambda, 1, k, x)\mathbf{B}' = ((p-1)\lambda K - mK + bkK + axK, 2^{3q_{\mathrm{bit}}/2}, k, x) \\ &= (0, 2^{3q_{\mathrm{bit}}/2}, k, x) \in \mathcal{L}' \end{aligned}$$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Yet another attack (2)

Provided that K is large enough

$$\| \mathbf{l}' \| \ll \det(\mathcal{L}')^{1/4}$$

We make the assumption that $\mathbf{l}'$ is a *shortest vector* of $\mathcal{L}'$.

Solving an easy SVP instance in $\mathcal{L}'$ (e.g. with LLL) allows to recover $\mathbf{l}' = (0, 2^{3q_{\mathrm{bit}}/2}, k, x)$.

RSA Key Generation

→ GnuPG RSA key generation algorithm is flawed as well.

→ Once two primes p and q of size $k/2$ bits are generated such that $n = p \cdot q$ has a size of k bits, one generates a public exponent e.

→ If 41 is coprime with $(p-1) \cdot (q-1)$, then take $e = 41$; otherwise, try $e = 257$, $e = 65537$, $e = 65539$, $e = 65541$, until a proper e is found.

Biased Key Generation

$\longrightarrow$ Note that if $e \geq 65539$ (this occurs with small probability), then on knows a 30-bit factor of $\phi(n)$, namely $41 \times 257 \times 65537$!

$\longrightarrow$ Not a *real/practical* security problem, as one needs to know a factor of the size of $n^{\frac{1}{4}}$.

$\longrightarrow$ But... any information leakage about $\phi(n)$ is a bad idea !

$\longrightarrow$ One should fi rst choose e, and *then* generate p and q.

LASEC

Conclusion

→ The quality of the implementation of cryptography could be considerably improved !

→ OpenPGP (and GnuPG) should recommend *recent* standards !

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE