The DGHV public-key encryption scheme

Jean-Sébastien Coron

University of Luxembourg

1 Implementation of DGHV, symmetric-key version

In the symmetric-key version of the DGHV scheme [vDGHV10], there is only a private key p which is a random prime of η bits. A ciphertext is

$$c = q \cdot p + 2r + m$$

where q is a random integer of $\gamma - \eta$ bits, r is a random integer of ρ bits, and $m \in \{0, 1\}$ is the message. We can use $\gamma = 10^4$, $\eta = 100$, and $\rho = 40$.

1. Install the Sage library, available at `http://www.sagemath.org/`. Alternatively, you can use the Sage Cell Server at `https://sagecell.sagemath.org`. Please submit a `.ipynb` file.
2. Implement the key generation. You can use the function random_prime to generate a large prime p.

```
def keyGen(eta=100):
  return p
```

3. Implement the DGHV encryption. You can use the function ZZ.random_element to generate a random integer.

```
def encrypt(m,p,gam=10^4,eta=100,rho=40):
  return c
```

4. Implement the DGHV decryption, and check that decryption works. Check that decryption does not work when $\rho > \eta$.

```
def decrypt(c,p):
  return m

def checkDec():
  p=keyGen()
  for i in range(100):
    m=ZZ.random_element(2)
    assert(decrypt(encrypt(m,p),p)==m)
```

5. Implement the homomorphic addition and multiplication. Check that homomorphic addition works, by using the function checkAdd below.

```
def add(c1,c2):
  pass

def mult(c1,c2):
  pass

def checkAdd():
  p=keyGen()
  for i in range(100):
    m1=ZZ.random_element(2)
    m2=ZZ.random_element(2)
    assert(decrypt(add(encrypt(m1,p),encrypt(m2,p)),p)==mod(m1+m2,2))
```

6. Write a similar checkMult function to check that homomorphic multiplication works.

```
def checkMult():
  pass
```

7. Verify that homomorphic multiplication does not work anymore if we use $\rho = 60$ instead of $\rho = 40$, for $\eta = 100$.

References

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.