From: Fre <frederik.vercauteren@gmail.com>
Sent: Thursday, January 11, 2018 3:24 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: Giophantus

Dear designers of Giophantus, dear all,

Below we describe a very easy distinguishing attack on the Giophantus encryption scheme that breaks the claimed IND-CPA security (in less time than one decryption). The attack simply exploits the fact that for the chosen base ring, evaluation at 1 is a ring homomorphism to Z_q. The attack is well known in other settings such as NTRU or RLWE.

The good news is that the attack can be thwarted by switching to a different base ring R_q.

The scheme can be summarized as follows:

- Define the polynomial ring R_q = Z_q[t]/(t^n-1) where q = 2^31-1 and for the highest security level we have n = 2267.

- The public key defines a plane over R_q:

X(x,y) = f10*x + f01*y + f00 with f10, f01, f00 in R_q

- The private key is a "small" point in R_q^2 on this plane, i.e.

priv key = (ux(t), uy(t)) in R_q^2 and X(ux(t), uy(t)) = 0

and all coefficients are smaller than a parameter l, which is equal to 4 in practice.

- Encryption first takes a message M of length 2*n bits and creates an element in R_q

m(t) = m0 + m1*t + ... m_(n-1)*t^(n-1)

with the mi in [0..4[corresponding to 2 consecutive bits.

- Encryption is then defined as

Enc(m(t)) = m(t) + X(x,y)*r(x,y) + l*e(x,y)

where r(x,y) is a random polynomial of total degree 1, and e(x,y) a degree 2 polynomial with small coefficients (for the chosen parameters, infinity norm is < 4).

The attack then proceeds as follows:

- Evaluation at t = 1 defines a ring homomorphism from R_q to Z_q

- If we apply this to the public key, we obtain a plane over Z_q of the form

X1(x,y) = f10(1)*x + f01(1)*y + f00(1) mod q

Example: for the first case in the KAT file PQCencryptKAT_1134.rsp for the highest security level (256 bits) we obtain after evaluation in 1 the equation

X1(x,y) := 683072552*x + 288881024*y + 1624678229 mod (2^31-1).

- The small solution ux(t), uy(t) evaluates to a small solution in Z_q^2, which can be easily found by running through all possibilities for ux(1) (which is smaller than 4*n) and solving for uy(1) and verifying that uy(1) is also small.

Example: continuing the example above, we easily find the solution (ux(1), uy(1)) := (3355, 3383), which indeed has infinity norm smaller than 4*n = 9068.

- We can now evaluate the coefficients of the ciphertext c(x,y) at 1, and evaluate in the above small solution (ux(1), uy(1)) to recover m(1) mod l, in this case m(1) mod 4.

- Since it is possible to compute m(1) mod 4, it is possible to distinguish between the encryption of 2 chosen messages m1(t) and m2(t) such that m1(1) != m2(1) mod 4, which breaks the IND-CPA claim.

- The attack can be thwarted by using a better R_q such as for RLWE.

Best regards,

Wouter Castryck and Frederik Vercauteren

From: Jacob Alperin-Sheriff <jacobmas@gmail.com>
Sent: Thursday, January 11, 2018 5:14 PM
To: Fre
Cc: pqc-comments; pqc-forum
Subject: Re: [pqc-forum] OFFICIAL COMMENT: Giophantus

(Do you have a script? If not, we can [eventually] write one).

On Thu, Jan 11, 2018 at 3:24 PM, Fre <frederik.vercauteren@gmail.com> wrote:

> Dear designers of Giophantus, dear all,
>
> Below we describe a very easy distinguishing attack on the Giophantus encryption
> scheme that breaks the claimed IND-CPA security (in less time than one decryption).
> The attack simply exploits the fact that for the chosen base ring, evaluation at 1 is a
> ring homomorphism to Z_q. The attack is well known in other settings such as NTRU or RLWE.
>
> The good news is that the attack can be thwarted by switching to a different base ring R_q.
>
> The scheme can be summarized as follows:
>
> - Define the polynomial ring R_q = Z_q[t]/(t^n-1) where q = 2^31-1 and for the
> highest security level we have n = 2267.
>
> - The public key defines a plane over R_q:
>
> X(x,y) = f10*x + f01*y + f00 with f10, f01, f00 in R_q
>
> - The private key is a "small" point in R_q^2 on this plane, i.e.
>
> priv key = (ux(t), uy(t)) in R_q^2 and X(ux(t), uy(t)) = 0
>
> and all coefficients are smaller than a parameter l, which is equal to 4 in practice.
>
> - Encryption first takes a message M of length 2*n bits and creates an element in R_q
>
> m(t) = m0 + m1*t + ... m_(n-1)*t^(n-1)
>
> with the mi in [0..4[corresponding to 2 consecutive bits.
>
> - Encryption is then defined as
>
> Enc(m(t)) = m(t) + X(x,y)*r(x,y) + l*e(x,y)
>
> where r(x,y) is a random polynomial of total degree 1, and e(x,y) a degree 2 polynomial
> with small coefficients (for the chosen parameters, infinity norm is < 4).
>
> The attack then proceeds as follows:

From: koichiro.akiyama@toshiba.co.jp
Sent: Friday, January 12, 2018 6:48 AM
To: frederik.vercauteren@gmail.com; pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: RE: [pqc-forum] OFFICIAL COMMENT: Giophantus

Dear Frederik Vercauteren and Wouter Castryck,

Thank you for pointing out the attack against IND-CPA for Giophantus.

And we appreciate that you suggested an countermeasure switching R_q to that of RLWE employed as well. According to your comment, we are going to consider some secure and suitable R_q setting.

This is just note to thank you.

Koichiro Akiyama

From: Fre [mailto:frederik.vercauteren@gmail.com]
Sent: Friday, January 12, 2018 5:24 AM
To: pqc-comments@nist.gov
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] OFFICIAL COMMENT: Giophantus

Dear designers of Giophantus, dear all,

Below we describe a very easy distinguishing attack on the Giophantus encryption scheme that breaks the claimed IND-CPA security (in less time than one decryption). The attack simply exploits the fact that for the chosen base ring, evaluation at 1 is a ring homomorphism to Z_q. The attack is well known in other settings such as NTRU or RLWE.

The good news is that the attack can be thwarted by switching to a different base ring R_q.

The scheme can be summarized as follows:

- Define the polynomial ring R_q = Z_q[t]/(t^n-1) where q = 2^31-1 and for the highest security level we have n = 2267.

- The public key defines a plane over R_q:

X(x,y) = f10*x + f01*y + f00 with f10, f01, f00 in R_q

- The private key is a "small" point in R_q^2 on this plane, i.e.

priv key = (ux(t), uy(t)) in R_q^2 and X(ux(t), uy(t)) = 0