On realistic speedup and possible homomorphic operations of Somewhat Homomorphic Encryption Schemes in hardware.

Vincent MIGLIORE

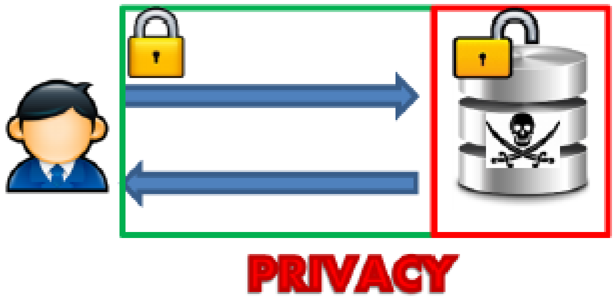
June $22^{nd}, 2016$

DGA Lab-STICC UMR IRISA

outline

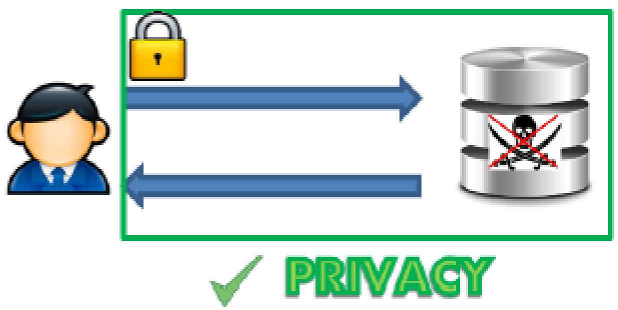
- Presentation of SHE
- How to attack SHE
- Impact on hardware accelerators

Definition of homomorphic encryption



Data are computed after decryption

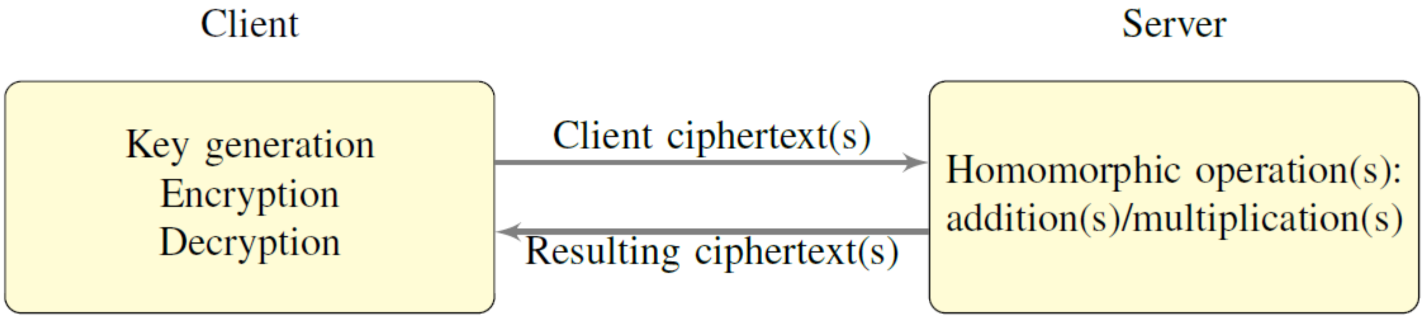
Standard cloud service



Data are computed in the cipher domain

Homomorphic encryption style cloud service

Definition of homomorphic encryption



Partially Homomorphic Encryption

+ or ×

2009 – Gentry Breakthrough

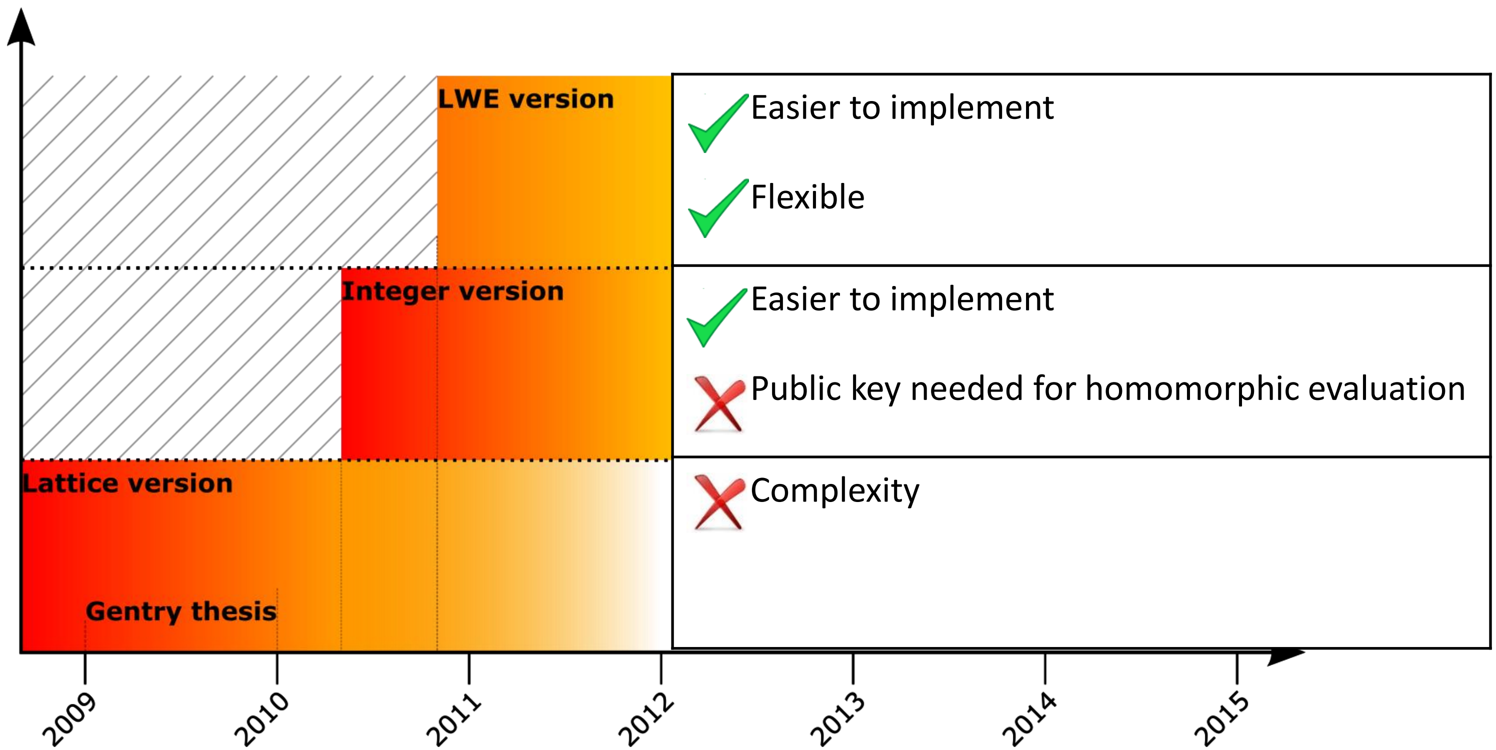
Fully Homomorphic Encryption (FHE)

+ and ×

Somewhat Homomorphic Encryption (SHE)

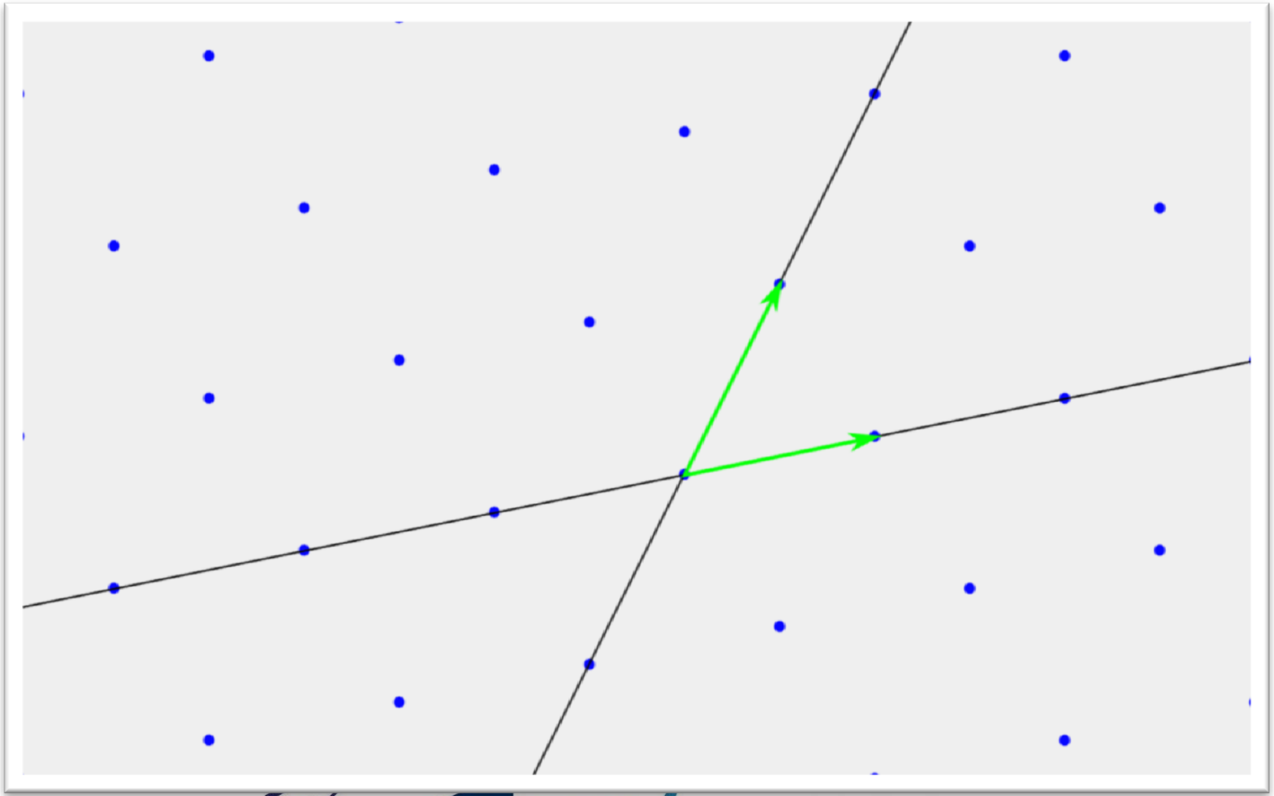
+ and Bounded number of ×

FHE/SHE


LWE version
Integer version
Lattice version
Gentry thesis
2009
2010
2011
2012
2013
2014
2015
Easier to implement
Flexible
Easier to implement
Public key needed for homomorphic evaluation
Complexity

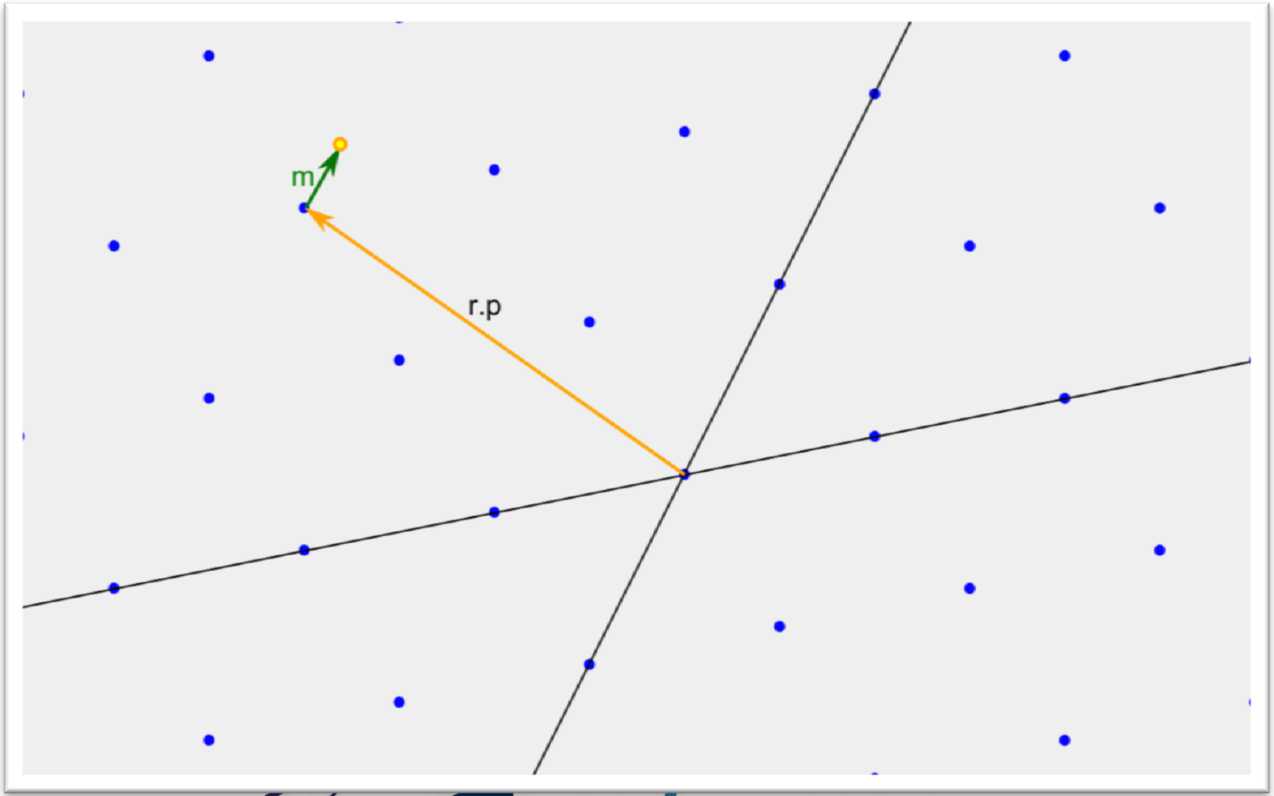
Attacks on Homomorphic Encryption Schemes

- FHE/SHE schemes are based on hard lattice problems:



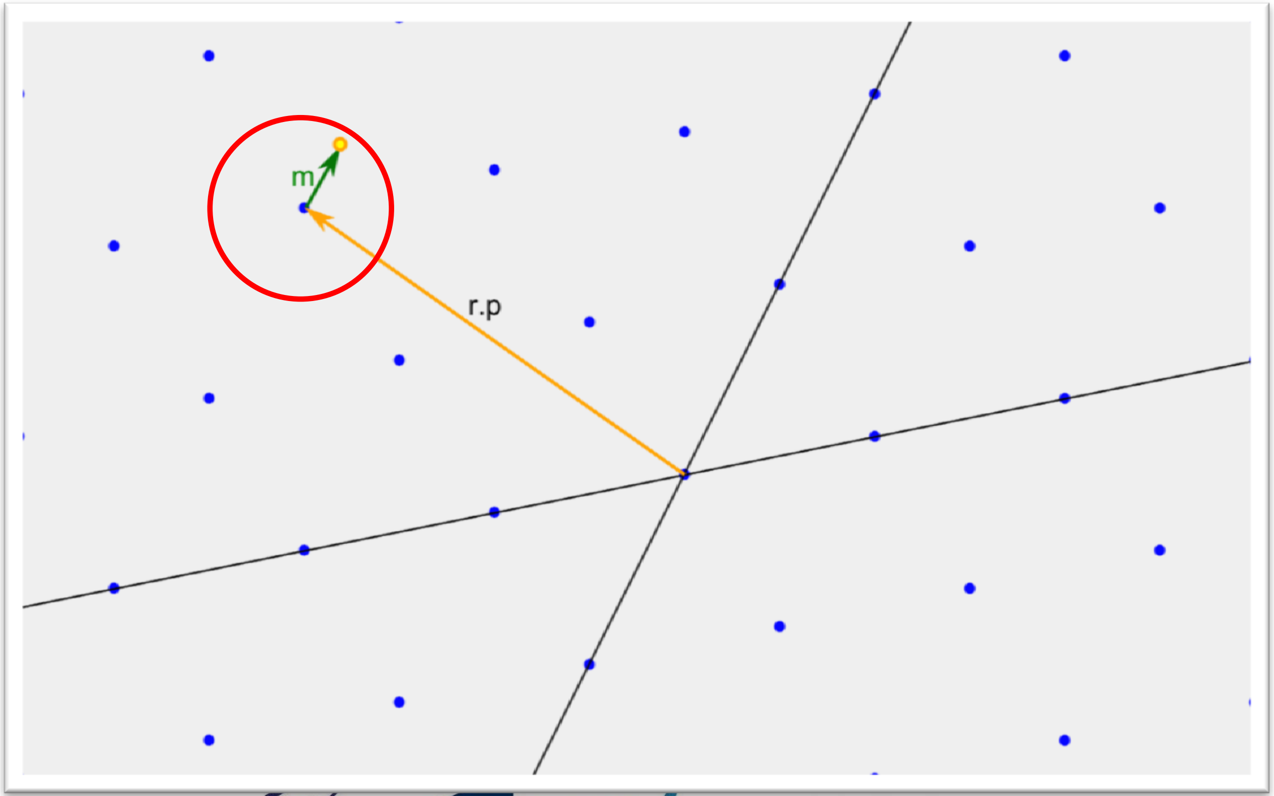
Attacks on Homomorphic Encryption Schemes

– Encryption : r.p + m



Attacks on Homomorphic Encryption Schemes

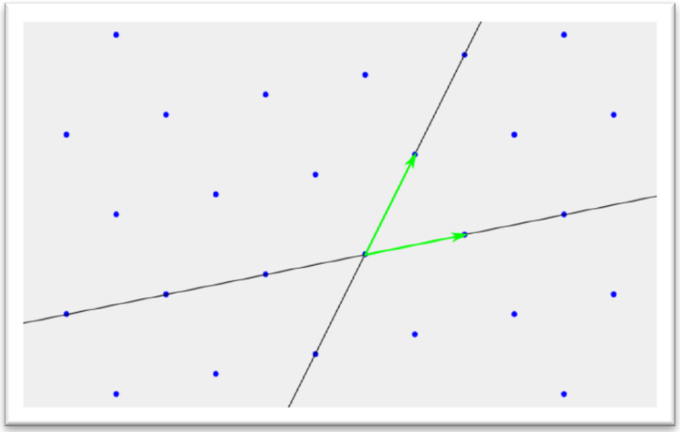
– Decryption :



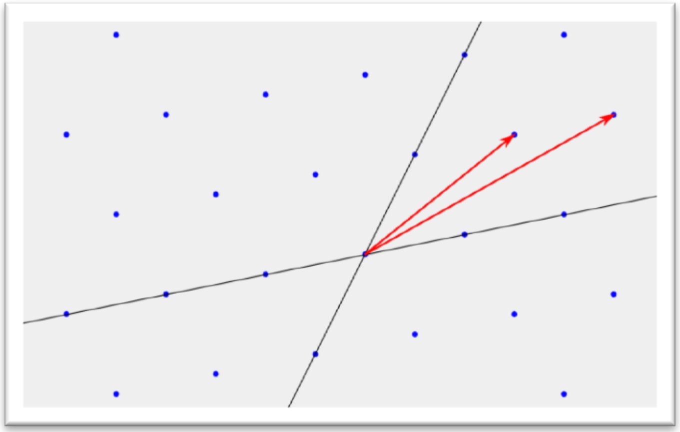
DGA Lab-STICC UMR IRISA

Attacks on Homomorphic Encryption Schemes

- Good base (secret key)

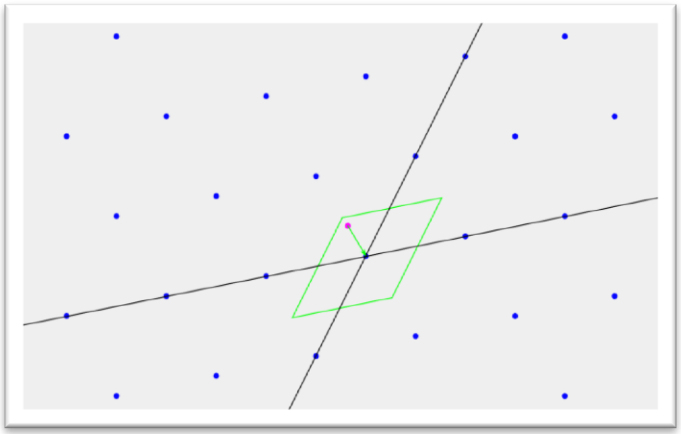


- « Bad » base : (public key)

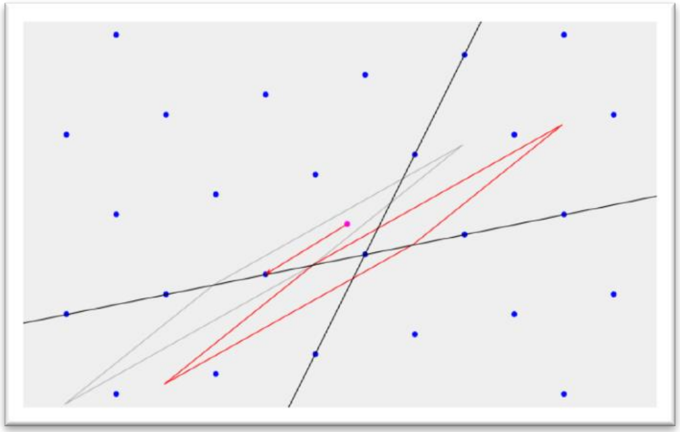


Attacks on Homomorphic Encryption Schemes

- Good base (secret key)



- « Bad » base : (public key)



Attacks on Homomorphic Encryption Schemes

- Standard resolution :
  - Find a short vector in the lattice
  - BKZ 2.0
    - Reduction : Improve basis quality
    - Enumeration : Find shortest vector (approximatively)
    - van de Pol and Smart:
      - Dimension x Enumeration x number of rounds

Hardware implementations

- Hardware implementations:
  - YASHE', based on two asumptions:
    - NTRU
    - DSPR

Hardware implementations

- Hardware implementations:
  - YASHE', based on two asumptions:
    - NTRU – Always secure
    - DSPR – Secured, but with more restrictive parameters (1)
      - But parameters not acceptable for homomorphic operations
  - Solution : Schemes not relying on DSPR:
    - Example : FV

(1) Martin Albrecht, Shi Bai and Leo Ducas, *A subfeld lattice attack on overstretched NTRU assumptions -Cryptanalysis of some FHE and Graded Encoding Schemes*, https://eprint.iacr.org/2016/127.pdf

Hardware implementations

	YASHE'	FV
Encryption	NTRU + DSPR	R-LWE
Ciphertext	1 polynomial	2 polynomials
Homomorphic addition	1 polynomial addition	2 polynomials addition
Homomorphic multiplication	1 polynomial multiplication 1 relinearization	4 polynomials multiplication 2 relinearizations

Hardware implementations

- Hardware implementations (FFT/NTT algorithm):
- $P \in Z_q[X]/f(X); \;\; f(X) = X^n \pm 1$
  - $f(X) = X^n - 1$: Standard FFT
    - For a polynomial multiplication of degree 2048, requires a FFT of size 4096
  - $f(X) = X^n + 1$: NWC
    - For a polynomial multiplication of degree 2048, requires a FFT of size 2048

Hardware implementations

- Hardware implementations (FFT/NTT algorithm):
- $P \in Z_q[X]/f(X);\ \ f(X) = X^n + 1$

	n	$log_2\, q$	Poly. mult	Relin	YASHE’ $\times$	FV $\times$ (est.)
(1)	4096	124	1.96 ms	4.79 ms	6.75 ms	15.46 ms
(1)	16384	512	27.88 ms	20.8 ms	48.68 ms	125.24 ms
(2)	32768	1228			121.678 ms	

(1) Thomas Pöppelmann, Michael Naehrig, Andrew Putnam and Adrian Macias, *Accelerating Homomorphic Evaluation on Reconfigurable Hardware*, *CHES 2015*

(2) Sujoy Sinha Roy, Kimmo Järvinen, Frederik Vercauteren, Vassil Dimitrov and Ingrid Verbauwhede, *Modular Hardware Architecture for Somewhat Homomorphic Function Evaluation* , *CHES 2015*

Hardware implementations

- Hardware implementations (FFT/NTT algorithm):
- $P \in Z_q[X]/f(X);\ \ f(X) = X^n + 1$

		Secured parameters for					
Accelerator parameters		Classical FFT			NWC		
$\boldsymbol{n}$	$\boldsymbol{log_2\, q}$	$\boldsymbol{n}$	$\boldsymbol{log_2\, q}$	L	$\boldsymbol{n}$	$\boldsymbol{log_2\, q}$	L
4096	124	~~4096~~ 1721	87	1	4096	~~124~~ 192	5
16384	512	~~16384~~ 8002	392	12	16384	~~512~~ 795	25

DGA Lab-STICC UMR IRISA

Hardware implementations

- Hardware implementations (FFT/NTT algorithm):
- $P \in Z_q[X]/f(X); \;\; f(X) = X^n + 1$

		Secured parameters for					
Accelerator parameters		Classical FFT			NWC		
n	$log_2\, q$	n	$log_2\, q$	L	n	$log_2\, q$	L
4096	124	~~4096~~ 1721	87	1	4096	~~124~~ 192	5
16384	512	~~16384~~ 8002	392	12	16384	~~512~~ 795	25

Integer multipliers too small

In practice

- For hardware accelerator with n = 4096 and log_2 = 124
  - For a L=1, better alternatives exist (BGN)
    - Not usefull in practice

In practice

- For hardware accelerator with n = 16384 and log_2 = 512
  - For L = 12, several algorithms can be performed :
    - 11 bits comparator :
      - 22 mults / 20 adds -> 2.8 s (1.1s for YASHE')
    - (1) Trivium :
      - 3,459 mults / 10,377 adds -> 435 s (169.32 s for YASHE')
    - (2) PIR : search on 512 items of 32 bits :
      - 4608 mults / 450560 adds: 661 s (266 s for YASHE')

(1) Christophe De Cannière and Bart Preneel, *Trivium,* *New Stream Cipher Designs – The eSTREAM Finalists 2008*

(2) Xun Yi, Mohammed Golam Kaosar, Russel Paulet and Elisa Bertino, *Single-Database Private Information Retrieval from Fully Homomorphic Encryption,* IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING 2013

DGA Lab-STICC UMR IRISA

Conclusion

– FFT implementations must be adapted to fit to new practical parameters
  - Smaller $\boldsymbol{log_2\ q}$ for classical FFT
  - Larger $\boldsymbol{log_2\ q}$ for NWC

– Even hardware accelerated, Homomorphic Encryption still time consuming.

DGA Lab-STICC UMR IRISA

Thanks for your attention !

Questions ?