CryptoExperts
WE INNOVATE TO SECURE YOUR BUSINESS

Scale-Invariant Fully Homomorphic Encryption over the Integers

J.-S. Coron *T. Lepoint* M. Tibouchi

PKC 2014
Thursday, March 27th, 2014

FHE

$x_1, \dots, x_n$

$$\xrightarrow{\mathrm{Enc}(x_1), \dots, \mathrm{Enc}(x_n)}$$

$$\xleftarrow[\mathrm{Enc}(f(x_1, \dots, x_n))]{}$$

f

Homomorphic Encryption

$$f, \mathrm{Enc}(x_1), \dots, \mathrm{Enc}(x_n) \longrightarrow \mathrm{Enc}(f(x_1, \dots, x_n))$$

We assume w.l.o.g that x_i bits and f boolean circuit

FHE Schemes

FHE

Perform operations on plaintexts by manipulating only ciphertexts, and without knowing the private-key.

- Too many schemes existing to do an exhaustive list now...
- Main families: [Gen09], [vDGHV10], [BV11], [LTV12], [GSW13]

FHE Schemes

FHE

Perform operations on plaintexts by manipulating only ciphertexts, and without knowing the private-key.

- Too many schemes existing to do an exhaustive list now...
- Main families: [Gen09], [vDGHV10], [BV11], [LTV12], [GSW13]

$\Downarrow$

improved in a series of works
[CMNT11], [CNT12],
[CCKL**L**TY13]

CryptoExperts

FHE Schemes

FHE

Perform operations on plaintexts by manipulating only ciphertexts, and without knowing the private-key.

- Too many schemes existing to do an exhaustive list now...
- Main families: [Gen09], [vDGHV10], [BV11], [LTV12], [GSW13]

$\Downarrow$

improved in a series of works
[CMNT11], [CNT12],
[CCK**L**LTY13] $\Rightarrow$ Batch DGHV scheme based on the decisional AGCD problem

FHE Schemes

FHE

Perform operations on … y ciphertexts, and without knowing …

- Too many schem… list now…
- Main families: [G… V12], [GSW13]

i…

… DGHV scheme … the decisional … D problem

CRYPTOEXPERTS

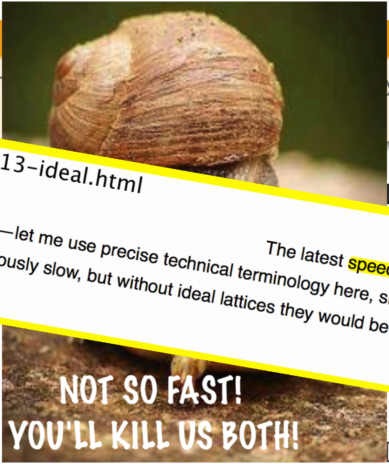
FHE Schemes

FHE

Perform operations on … y ciphertexts,
an… knowing …

blog.cr.yp.to/20140213-ideal.html

The latest *speed reports for fully homomorphic encryption are*—let me use precise technical terminology here, since I'm a big fan of careful benchmarking—ludicrously slow, but without ideal lattices they would be *utterly ludicrously slow*.



… list now…

… the decisional
…D problem

CryptoExperts

The DGHV Scheme [vDGHV10]

- Public $x_i = q_i \cdot p + 2r_i$ and error-free modulus $x_0 = q_0 \cdot p$
- Public encryption of $m \in \{0, 1\}$:

$$c = m + 2r' + \sum_{i \in S} x_i \bmod x_0$$

where p is the secret-key, S random subset and r' is a "big" random

The DGHV Scheme [vDGHV10]

- Public $x_i = q_i \cdot p + 2r_i$ and error-free modulus $x_0 = q_0 \cdot p$
- Public encryption of $m \in \{0, 1\}$:

$$c = m + 2\Big(r' + \sum_{i \in S} r_i\Big) + \Big(\sum_{i \in S} q_i\Big) \cdot p \bmod x_0$$

where p is the secret-key, S random subset and r' is a "big" random

The DGHV Scheme [vDGHV10]

- Public $x_i = q_i \cdot p + 2r_i$ and error-free modulus $x_0 = q_0 \cdot p$
- Public encryption of $m \in \{0, 1\}$:

$$c = m + 2\left(r' + \sum_{i \in S} r_i\right) + \left(\sum_{i \in S} q_i\right) \cdot p \bmod x_0 = q_0 \cdot p$$

where p is the secret-key, S random subset and r' is a "big" random

  - LHL can be applied on the q_i's
  - LHL cannot be applied on the r_i's: so we use a drowning factor r'

CryptoExperts

The DGHV Scheme [vDGHV10]

- Public $x_i = q_i \cdot p + 2r_i$ and error-free modulus $x_0 = q_0 \cdot p$
- Public encryption of $m \in \{0, 1\}$:

$$c = m + 2r' + \sum_{i \in S} x_i \bmod x_0$$

where p is the secret-key, S random subset and r' is a "big" random

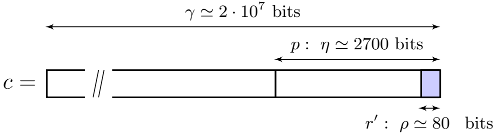
  - LHL can be applied on the q_i's
  - LHL cannot be applied on the r_i's: so we use a drowning factor r'
    - This did not generalized easily to batch DGHV...
    - Either intricate proof [C**L**T13, eprint 2013/036] or decisional AGCD problem (hard to distinguish $x_i = q_i p + r_i$ from random modulo x_0) [CCKL**L**TY13]

The DGHV Scheme [vDGHV10]

- Public $x_i = q_i \cdot p + 2r_i$ and error-free modulus $x_0 = q_0 \cdot p$
- Public encryption of $m \in \{0, 1\}$:

$$c = m + 2r' + \sum_{i \in S} x_i \bmod x_0$$

where p is the secret-key, S random subset and r' is a "big" random

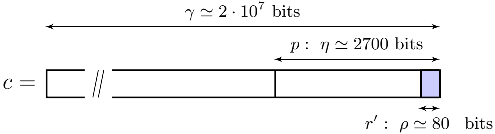


The DGHV Scheme [vDGHV10]

- Public $x_i = q_i \cdot p + 2r_i$ and error-free modulus $x_0 = q_0 \cdot p$
- Public encryption of $m \in \{0, 1\}$:

$$c = m + 2r' + \sum_{i \in S} x_i \bmod x_0$$

where p is the secret-key, S random subset and r' is a "big" random



- Decryption:

$$(c \bmod p) \bmod 2 = m$$

Homomorphic Properties

- Addition:

$$\begin{aligned} c_1 &= q_1 \cdot p + 2r_1 + m_1 \\ c_2 &= q_2 \cdot p + 2r_2 + m_2 \end{aligned} \Rightarrow c_1 + c_2 = q' \cdot p + 2r' + (m_1 + m_2)$$

Homomorphic Properties

- Addition:

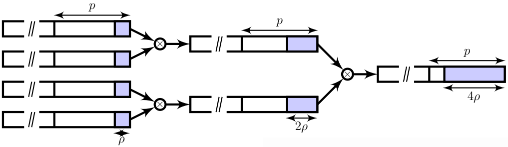
$$\begin{aligned} c_1 &= q_1 \cdot p + 2r_1 + m_1 \\ c_2 &= q_2 \cdot p + 2r_2 + m_2 \end{aligned} \quad \Rightarrow c_1 + c_2 = q' \cdot p + 2r' + (m_1 + m_2)$$

- Multiplication:

$$\begin{aligned} c_1 &= q_1 \cdot p + 2r_1 + m_1 \\ c_2 &= q_2 \cdot p + 2r_2 + m_2 \end{aligned} \quad \Rightarrow c_1 \cdot c_2 = q'' \cdot p + 2r'' + (m_1 \cdot m_2)$$

with

$$r'' = 2r_1 r_2 + r_1 m_2 + r_2 m_1$$



CryptoExperts

Scale Invariance

- How to avoid exponential growth?
  - **Modulus Switching** [BGV12]: multiply by q'/q and round; the noise goes down by a factor $\approx q'/q$
    Secret key $s \in \mathbb{Z}^n$, Ciphertext $c \in \mathbb{Z}_q^n$

$$\vec{c} \cdot \vec{s} = m + 2e + qI$$

Scale Invariance

- How to avoid exponential growth?
  - **Modulus Switching** [BGV12]: multiply by q'/q and round; the noise goes down by a factor $\approx q'/q$
    Secret key $s \in \mathbb{Z}^n$, Ciphertext $c \in \mathbb{Z}_q^n$

    $$\vec{c} \cdot \vec{s} = m + 2e + ql$$

  - **Scale-Invariance** [Bra12]: do not need to change modulus, but noise growth still linear
    Secret key $s \in \mathbb{Z}^n$, Ciphertext $c \in \mathbb{R}^n$

    $$\vec{c} \cdot \vec{s} = m + \epsilon + 2l$$

Scale Invariance

- How to avoid exponential growth?
  - **Modulus Switching** [BGV12]: multiply by q'/q and round; the noise goes down by a factor $\approx q'/q$
    Secret key $s \in \mathbb{Z}^n$, Ciphertext $c \in \mathbb{Z}_q^n$

    $$\vec{c} \cdot \vec{s} = m + 2e + ql$$

  - **Scale-Invariance** [Bra12]: do not need to change modulus, but noise growth still linear
    Secret key $s \in \mathbb{Z}^n$, Ciphertext $c \in \mathbb{R}^n$

    $$\vec{c} \cdot \vec{s} = m + \epsilon + 2l$$

- $\Rightarrow$ Leveled FHE: noise growth linear in mult. depth instead of exponential

Our Contributions

- **Equivalence** between Error-Free Decisional AGCD and Error-Free Computational AGCD
  - *Automatically simplifies* all previous DGHV schemes [vDGHV10,CMNT11,CNT12,CLT13a]
- Variant of DGHV and batch DGHV that is **scale invariant**
  - Noise growth linear in the multiplicative depth
  - but only one modulus: p^2 instead of p
- Homomorphic Evaluation of AES with a scale invariant scheme

Computational/Decisional AGCD

Error-Free Settings: For efficiency reason for FHE schemes, we work with an exact multiple

$$x_0 = q_0 \cdot p$$

of the secret key p.

- Computational $\mathrm{AGCD}_{\gamma,\eta,\rho}$: given x_0 and polynomially many $x_i = q_i \cdot p + r_i$, recover p
- Decisional $\mathrm{AGCD}_{\gamma,\eta,\rho}$: given x_0, polynomially many $x_i = q_i \cdot p + r_i$ and

$$z = q_z \cdot p + r_z + b \cdot u \bmod x_0$$

where $u \leftarrow [0, x_0)$, recover b

The (Error-Free) Computational and Decisional AGCD problems are equivalent

New (Batch) DGHV Scheme

- One-Slot Scheme
  - Public $x_i = q_i \cdot p + 2r_i$ and error-free modulus $x_0 = q_0 \cdot p$
  - Public encryption of $m \in \{0,1\}$:

$$c = m + \sum_{i \in S} x_i \bmod x_0$$

  - Decryption:

$$(c \bmod p) \bmod 2 = m$$

- Multi-Slots Scheme
  - Encryption of $\vec{m} = (m_i)$ is $q_i \cdot p_1 \times \cdots \times p_n + CRT_{p_i}(2r_i + m_i)$
  - Public $x_i = \mathrm{Enc}(0)$, error-free modulus $x_0 = q_0 \cdot p_1 \times \cdots \times p_n$ and elements $x'_i = \mathrm{Enc}(\vec{e}_i)$ (where $\vec{e}_i[j] = \delta_{i,j}$)
  - Public encryption of $\vec{m} \in \{0,1\}^n$:

$$c = \sum_{i=1}^{n} m_i \cdot x'_i + \sum_{i \in S} x_i \bmod x_0$$

Scale Invariant DGHV

- Main Ideas: work with secret p^2 and move bit message to MSB modulo p instead of LSB modulo p
- Type-I ciphertext:

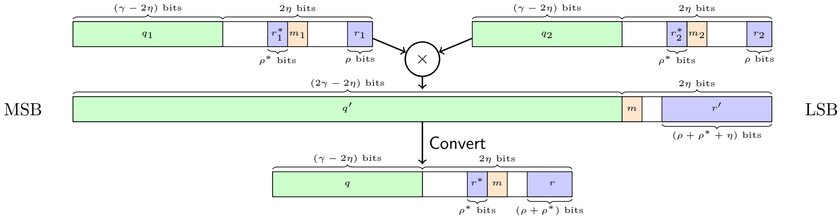
$$c = q \cdot p^2 + (2r^* + m) \cdot \frac{p-1}{2} + r$$

- Type II ciphertext (after multiplication of Type-I):

$$c' = q' \cdot p^2 + m \cdot \frac{p^2-1}{2} + r'$$

- Procedure convert: similar to modulus swiching [CNT12] from p^2 to p... but we somewhat remain with a secret p^2

Procedure Convert



Lemma

Let ρ' be such that $\rho' \geq \eta + \rho + \log_2(\eta\Theta)$. There exists a procedure Convert which converts a Type-II ciphertext with noise size ρ' into a Type-I ciphertext with noise $(\rho' - \eta + 5, \log_2 \Theta)$.

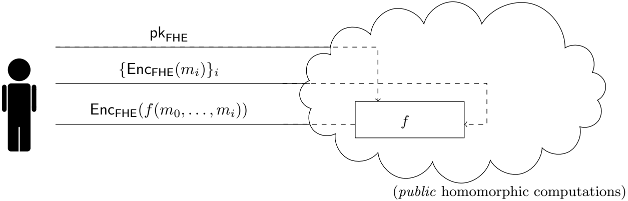
- Easy generalization to batching [CCKL**L**TY13]

CRYPTOEXPERTS

Description of the leveled FHE scheme

- Public $x_i = q_i \cdot p^2 + r_i$, error-free modulus $x_0 = q_0 \cdot p^2$ and
$$y = q_y \cdot p^2 + r_y + \frac{p-1}{2}$$
- Public encryption of $m \in \{0,1\}$:
$$c = m \cdot y + \sum_{i \in S} x_i \bmod x_0$$
- Decryption:
$$(2 \cdot c \bmod p) \bmod 2 = m$$
- Mult of c_1 and c_2:
$$c' = \mathrm{Convert}(2 c_1 c_2)$$

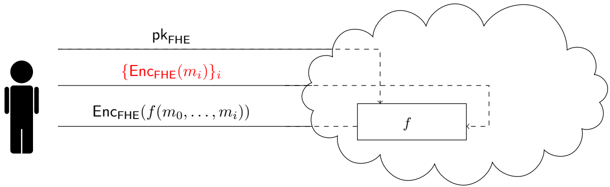
Homomorphic AES?



- Typical high-level FHE use-case

CryptoExperts

Homomorphic AES?

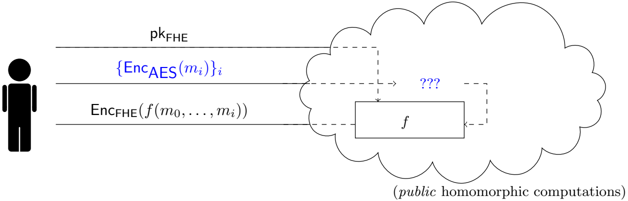


($\textit{public}$ homomorphic computations)

- Typical high-level FHE use-case
- ... wait a sec! The ciphertext expansion is HUGE (prohibitive)!
  - If m_i is a 4MB image, using [GHS12,CCKL**L**TY13], the user would have to send around 200/300GB of encrypted data

CRYPTOEXPERTS

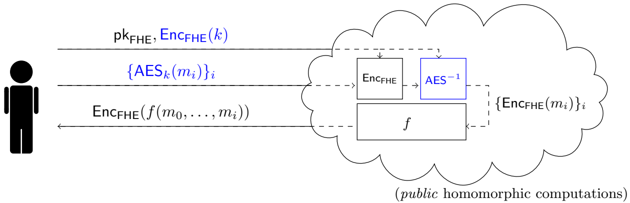
Homomorphic AES?



- Typical high-level FHE use-case
- … wait a sec! The ciphertext expansion is HUGE (prohibitive)!
- What if we use hybrid encryption? [NLV11]
  - AES does not have ciphertext expansion

CryptoExperts

Homomorphic AES?



- Typical high-level FHE use-case
- ... wait a sec! The ciphertext expansion is HUGE (prohibitive)!
- What if we use hybrid encryption? [NLV11]
- Now we need to homomorphically evaluate AES^{-1}
  - Network communication from user to cloud essentially optimal
  - But now we need to efficiently evaluate AES^{-1} before f!!

CryptoExperts

Homomorphic AES using SIBDGHV

- Use the same framework as in [CCKL**L**TY13]
- State-wise AES implementation: 128 ciphertexts, one per bit of the AES state
- Batching used to perform several AES in parallel

Homomorphic AES using SIBDGHV

- Use the same framework as in [CCKL**L**TY13]
- State-wise AES implementation: 128 ciphertexts, one per bit of the AES state
- Batching used to perform several AES in parallel

Instance	λ	$\ell =$ # of enc. in parallel	AddRoundKey	SubBytes	ShiftRows	MixColumns	Total Time	**Time/AES block**
Toy	42	9	0.0s	1.5s	0.0s	0.0s	15.1s	**1.7s**
Small	52	35	0.1s	9.9s	0.0s	0.0s	1min 40s	**2.9s**
Medium	62	140	0.3s	80.5s	0.0s	0.1s	13min 29s	**5.8s**
Large	72	569	2.1s	21min	0.0s	0.6s	3h 35min	**23s**
Extra	80	1875	6.9s	10h 9min	0.1s	1.6s	102h	**195s**

Homomorphic AES using SIBDGHV

- Use the same framework as in [CCKL**L**TY13]
- State-wise AES implementation: 128 ciphertexts, one per bit of the AES state
- Batching used to perform several AES in parallel

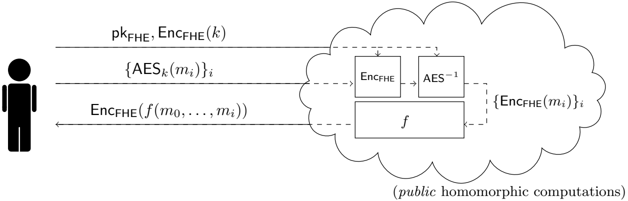
Instance	λ	$\ell = \#$ of enc. in parallel	AddRoundKey	SubBytes	ShiftRows	MixColumns	Total Time	Time/AES block
Toy	42	9	0.0s	1.5s	0.0s	0.0s	15.1s	**1.7s**
Small	52	35	0.1s	9.9s	0.0s	0.0s	1min 40s	**2.9s**
Medium	62	140	0.3s	80.5s	0.0s	0.1s	13min 29s	**5.8s**
Large	72	569	2.1s	21min	0.0s	0.6s	3h 35min	**23s**
Extra	80	1875	6.9s	10h 9min	0.1s	1.6s	102h	**195s**

- Compared to BDGHV ([CCKL**L**TY13])

Instance	λ	ℓ	# of enc. in parallel	AddRoundKey	SubBytes	ShiftRows	MixColumns	Total AES (in hours)	Relative time
Toy	42	10	10	0.06s	33s	0s	0.02s	**0.08**	29s
Small	52	37	37	0.06s	309s	0s	0.09s	**0.74**	1min 12s
Medium	62	138	138	4.5s	3299s	0s	0.44s	**7.86**	3min 25s
Large	72	531	531	27s	47656s	0.04s	2.8s	**113**	12min 46s

Thoughts about Hom. Computations

Partly explicited in [**L**N14, eprint 2014/062]

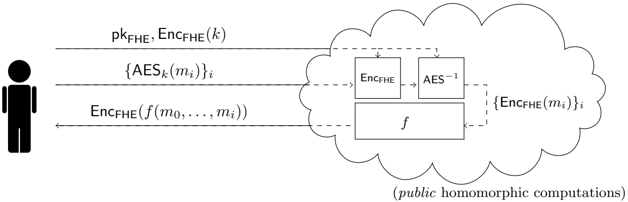


- Parameter selection: either room for f or need to bootstrap :-(

CryptoExperts

Thoughts about Hom. Computations

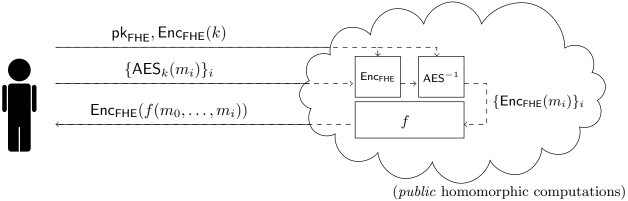
Partly explicited in [**L**N14, eprint 2014/062]



- Parameter selection: either room for f or need to bootstrap :-(
- Latency vs. throughput

Thoughts about Hom. Computations

Partly explicited in [**L**N14, eprint 2014/062]



- Parameter selection: either room for f or need to bootstrap :-(
- Latency vs. throughput
- Is AES such a good idea?

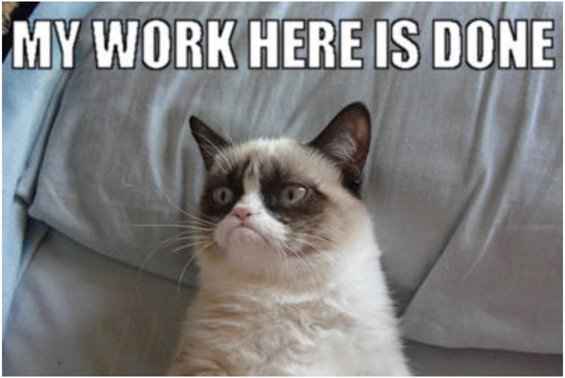
Conclusion

Conclusion

- Equivalence between Error-Free Decisional and Computational AGCD: automatic simplification of previous FHE schemes over the integers
- New leveled DGHV scheme that is scale invariant (no modulus switching)
- Timings one order of magnitude faster than [CCKL**L**TY13] and comparable to [GHS12] for homomorphic AES evaluation

- AGCD also used for Multilinear Maps [C**L**T13]: need more cryptanalysis on this problem
  - we hope that our pratical parameters practical parameters will spur on the cryptanalysis of AGCD

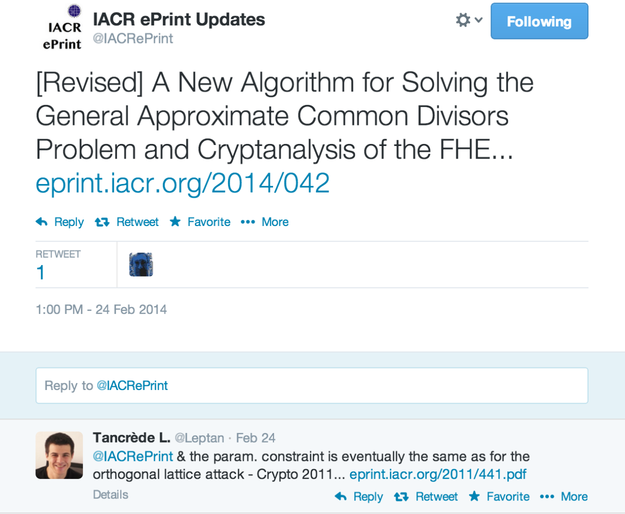
Questions? or...



Copyright Grumpy Cat

Thank you for your attention

Recent Attack on Eprint?



CryptoExperts