What is Homomorphic Encryption?

- $x$, $F$
- $F(x)$
- Encryption of $x$
- Encryption of $F(x)$

Eval($x$, $F$)
Motivational example: DNA test

- Many companies offer to determine ancestral origin via a DNA test
- Offering this service using HE would provide privacy for the client
<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1978</td>
<td>Proposed as open problem</td>
<td>[RAD78]</td>
</tr>
<tr>
<td>2009</td>
<td>Gentry’s blueprint and early schemes</td>
<td>[Gentry09]</td>
</tr>
<tr>
<td>2011</td>
<td>First literature on concrete applications (e.g. [NLV11])</td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td>Second generation schemes</td>
<td>[BGV12], [FV12], ...</td>
</tr>
<tr>
<td>2013</td>
<td>Third generation schemes</td>
<td>[GSW13], ...</td>
</tr>
<tr>
<td>2015</td>
<td>First open source libraries available</td>
<td></td>
</tr>
<tr>
<td>2017</td>
<td>Fast bootstrapping</td>
<td></td>
</tr>
<tr>
<td>2018</td>
<td>Standardisation effort begins</td>
<td></td>
</tr>
<tr>
<td>2018</td>
<td>Almost commercially viable?</td>
<td></td>
</tr>
</tbody>
</table>


Fully or somewhat homomorphic?

**Fully Homomorphic Encryption (FHE)**
A scheme that supports arbitrary computation on encrypted data

**Somewhat Homomorphic Encryption (SHE)**
A scheme that supports computing limited functions on encrypted data:
- Circuits of at most a fixed maximal depth
- Polynomials of at most a fixed maximal degree

[Gentry09] blueprint

FHE = SHE + bootstrapping

Recent developments: bootstrapping

How does bootstrapping work?

- Ciphertexts have inherent noise
- Noise grows during homomorphic operations
- If noise too large, decryption will fail
- **Bootstrapping refreshes a ciphertext into one with less noise**

Previous view: bootstrapping should be avoided

- Was very inefficient
- SHE with large parameters sufficient for some applications
Recent developments: bootstrapping

Recent highlights:

[CGGI16] Bootstrapping in less than 0.1s for GSW variant

[CHK17] Same evaluation faster than in [CLT14] without bootstrapping

[CH18] Lower depth bootstrapping for FV and BGV


Recent developments: encoding

- Typical plaintext space is $R_t = \mathbb{Z}_t[x]/(f(x))$, for $t$ an integer
- Raw data can be integer, rational, complex number, . . .
- Using an appropriate encoding is an important practical issue

Homomorphic addition of two integers
Recent developments: encoding

Recent highlights:

[BBB+17] Exploits large number of coefficients to allow smaller \( t \)

[CLPX18] Using \( t = x - b \) enables higher depth circuits

[CIV18] Generalised encoding framework + improved batching

Recent developments: approximate computation

**HEAAN** [CKKS17]

- HE for approximate computation can be achieved
- Main idea: treat ciphertext noise as part of error
- “the precision loss during evaluation is [...] at most one more bit compared to unencrypted approximate arithmetic such as floating-point operations”

**Recent highlights:**

[CHKKS18] Bootstrapping is both possible and more efficient


## Recent developments: libraries

<table>
<thead>
<tr>
<th>Name</th>
<th>Implements variant of</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>cuHE</td>
<td>LTV</td>
<td>github.com/vernamlab/cuHE</td>
</tr>
<tr>
<td>HElib</td>
<td>BGV</td>
<td>github.com/shaih/HElib</td>
</tr>
<tr>
<td>FV-NFLlib</td>
<td>FV</td>
<td>github.com/CryptoExperts/FV-NFLlib</td>
</tr>
<tr>
<td>Palisade</td>
<td>FV, BGV, LTV, SS</td>
<td>git.njit.edu/palisade/PALISADE</td>
</tr>
<tr>
<td>SEAL</td>
<td>FV</td>
<td>sealcrypto.org</td>
</tr>
<tr>
<td>TFHE</td>
<td>GSW</td>
<td>tfhe.github.io/tfhe</td>
</tr>
</tbody>
</table>

---

Challenges for deployment

1. Which scheme / implementation is best for a given application?
2. How can we ensure a comparison is fair?

Issues when comparing implementations:

- Fastest bootstrapping is for GSW variants
- Encoding work has focussed on schemes like FV and BGV
- Some libraries may be easier to use
- Different libraries have different default parameters

Challenges for deployment

3. How should we optimise for a given function?

Issues when implementing a given function:

- In implementations such as [CLP17], relinearization is optional.
- Determining when to relinearize is NP hard [Chen17].
- Similar hardness results for determining when is best to bootstrap [PV15], [BLMZ17].
Applications: areas considered in the literature

- Medical
- Genomics
- Machine Learning
- Statistics
- Smart cities
- Cyber physical systems
- Private information retrieval
- Database search
- Private set intersection
- Electronic voting
- ...

https://heat-project.eu/casestudies.html
https://homomorphicencryption.org/white_papers/applications_homomorphic_encryption_white_paper.pdf

## Applications: what can we achieve today?

<table>
<thead>
<tr>
<th>Application</th>
<th>Performance</th>
<th>Ctext size</th>
<th>Max depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search [AFS18]</td>
<td>800KB database in 1 min</td>
<td>10KB</td>
<td>11</td>
</tr>
<tr>
<td>Logistic regression [KSW+18]</td>
<td>256B dataset in 100 min</td>
<td>75MB</td>
<td>3</td>
</tr>
<tr>
<td>String matching [CCL+17]</td>
<td>1920KB dataset in 2s</td>
<td>250KB</td>
<td>5</td>
</tr>
<tr>
<td>Neural network [BMMP18]</td>
<td>100 neurons in 1.65s</td>
<td>8.2KB</td>
<td>-</td>
</tr>
</tbody>
</table>


Current and potential commercial applications

**Commercial use**

- Products based on HE offered by Duality, Enveil, Inpher, . . .
- Consultation on FHE offered by CryptoExperts, Galois, PQAT, . . .
- HE is on the research agenda of IBM, Microsoft, Thales, . . .

**Standardisation effort**

- Led by industry, government and academia consortium
- White papers\(^1\) on APIs, security and applications
- Draft standard for parameter selection\(^2\) based on [APS15] estimator\(^3\)

---

1. https://homomorphicencryption.org/white-papers/
3. https://bitbucket.org/malb/lwe-estimator

Conclusion

Developments: Bootstrapping, encoding, approximate computation, libraries

Challenges: 1. Which scheme is best for a given application? 2. How can we ensure a comparison is fair? 3. How should we optimise for a given function?

Applications: Many potential applications considered in literature, starting to see commercial applications

Thank you!

- rachel.player@lip6.fr
- @yayworthy