Email encryption is compatible with provider-supplied functions

Trinabh Gupta*,†, Henrique Fingler*, Lorenzo Alvisi*,¶, and Michael Walfish†

*The University of Texas at Austin
†New York University
¶Cornell
If a mail server can access email, then …

… rogue employees can access email.
... **hackers** can access email.

... **law enforcement agencies** can access email.
So, why don’t email service providers deploy end-to-end email encryption?
End-to-end encryption is in conflict with service providers’ functions

“… we couldn't run our system if everything in it were encrypted because then we wouldn't know which ads to show you.”

“So this is a system that was designed around a particular business model.”

[Vint Cerf. Sixth Annual Meeting of the Internet Governance Forum. 2011]
We asked: can we build an email system that

a) supports **end-to-end email encryption**, 

b) supports **provider-supplied functions** consistent with existing commercial regime, and

c) has **low costs**?
Pretzel demonstrates:

Email encryption is compatible with provider-supplied functions.
Pretzel requirements:

- end-to-end email encryption
- basic functions: spam filtering, topic extraction
- low resource cost

“[we cannot have end-to-end encryption and AI] until someone figures out how to do homomorphic machine learning.”

[Thai Duong, an engineer who co-leads Google’s product security team. 2011]
Two-party secure computation (2PC) from 10,000 feet

- can handle arbitrary computations
Two-party secure computation (2PC) crypto protocols can enable encryption and functions but have huge resource (CPU, network, etc.) costs.
Pretzel:

reduces costs of 2PC by up to 100x, by refining 2PC for specific functions.
Rest of this talk

• Two example functions.

• Background on 2PC (Yao+GLLM) that can implement these functions.

• Refinement of 2PC.
Pretzel supports two functions: spam filtering and topic extraction.

**Topic extraction:**

Hi, I am looking to buy a Ford Sedan. Any suggestions?

Topic: Automobile

0% APR
## Linear classifiers
(for both spam filtering, topic extraction)

### Part 1: Add probabilities corresponding to words in email.
Example: networks: 0.7

<table>
<thead>
<tr>
<th>words in dictionary</th>
<th>networks</th>
<th>OS</th>
<th>security</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP</td>
<td>0.4</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>route</td>
<td>0.3</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>cloud</td>
<td>0.1</td>
<td>0.7</td>
<td>0.3</td>
</tr>
<tr>
<td>encrypt</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Part 2: Compare outputs from part 1.
Category is “networks”.

BGP may be used for routing.
Background on Yao+GLLM 2PC

Provider does the following:

- Encrypt
- Encrypt
- Encrypt
- Encrypt
- Encrypt

additively homomorphic encryption

model → encrypt

networks
- BGP
  - E(0.4)
- route
  - E(0.3)
- cloud
  - E(0.1)
- encrypt
  - E(0.2)

OS
- E(0.0)
- E(0.1)
- E(0.7)
- E(0.2)

security
- E(0.1)
- E(0.0)
- E(0.3)
- E(0.6)
Background on Yao+GLLM 2PC

Client does the following:

<table>
<thead>
<tr>
<th>networks</th>
<th>OS</th>
<th>security</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP</td>
<td>E(0.4)</td>
<td>E(0.0)</td>
</tr>
<tr>
<td>route</td>
<td>E(0.3)</td>
<td>E(0.1)</td>
</tr>
<tr>
<td>cloud</td>
<td>E(0.1)</td>
<td>E(0.7)</td>
</tr>
<tr>
<td>encrypt</td>
<td>E(0.2)</td>
<td>E(0.2)</td>
</tr>
</tbody>
</table>

Add encrypted probabilities using **additive homomorphism**.

Example:

networks: \( E(0.4) \circ E(0.3) = E(0.4 + 0.3) = E(0.7) \)

BGP may be used for routing.

words in email: \{BGP, routing\}

Client does the following:
Background on Yao+GLLM 2PC

Client and provider do the following:

- **Provider** sends the decryption key to the **client**.
- The decryption key is used to decrypt the data.
- The category is "networks".

Client sends encrypted data to the **provider**.

- **Yao 2PC** is used for secure computation.
- The category is "networks".
- The encrypted data is denoted as \( E(0.7), E(0.1), E(0.1) \).
Provider does the following:

Provider sends encrypted model to the client.

Issue 1: encrypted model is large
Cost issues in Yao+GLLM 2PC

Client and provider do the following:

Issue 2: CPU and network costs of Yao part grow with the number of categories.
<table>
<thead>
<tr>
<th>Issues in Yao+GLLM</th>
<th>Pretzel’s refinements</th>
</tr>
</thead>
<tbody>
<tr>
<td>encrypted model is large</td>
<td>adapt packing from other domains</td>
</tr>
<tr>
<td>CPU and network costs of Yao part grow with the number of categories</td>
<td>decomposed classification</td>
</tr>
</tbody>
</table>
Pretzel uses packing to reduce client-side storage cost

- Packing can reduce the size of model by #elements packed
- **Caution**: Must preserve addition operation in cipherspace
<table>
<thead>
<tr>
<th>Issues in Yao+GLLM</th>
<th>Pretzel’s refinements</th>
</tr>
</thead>
<tbody>
<tr>
<td>encrypted model is large</td>
<td>adapt packing from other domains</td>
</tr>
<tr>
<td>CPU and network costs of Yao part grow with the number of categories</td>
<td>decomposed classification</td>
</tr>
</tbody>
</table>
Pretzel’s decomposed classification at a high level

**What we want:**

- \{network, OS, security, algo\} \rightarrow \{network\}
  - set of all topics
  - chosen topic

**step 1:** *performed at client using a public classifier*

- \{network, OS, security, algo\} \rightarrow \{network, algo\}
  - set of all topics
  - candidate topics

**step 2:** *performed using 2PC*

- \{network, algo\} \rightarrow \{network\}
  - chosen topic

**Category is “networks”**

- Provider
  - decryption key
  - \(E(0.7), E(0.1)\)

- Client
  - Category is “networks”

*only for candidate topics*
Outline

✓ Background on 2PC (Yao+GLLM).

✓ Design of Pretzel.

• Evaluation of Pretzel
Experiment method

Baselines:
• Non-private system
• Yao+GLLM (with Paillier cryptosystem and GLLM packing)

Functions:
• Spam filtering (5M features)
• Topic extraction (20K features, 2048 topics, 20 candidate topics)

Measure CPU time, network transfers, and storage space
Overheads for spam filtering (relative to status quo)

<table>
<thead>
<tr>
<th></th>
<th>Yao+GLLM</th>
<th>Pretzel</th>
</tr>
</thead>
<tbody>
<tr>
<td>provider-side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU time:</td>
<td>15.9x</td>
<td>2.7x</td>
</tr>
<tr>
<td>network transfers:</td>
<td>1.05x</td>
<td>1.26x</td>
</tr>
<tr>
<td>client-side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>storage:</td>
<td>1.3GB</td>
<td>183MB</td>
</tr>
</tbody>
</table>
## Overheads for topic extraction (relative to status quo)

<table>
<thead>
<tr>
<th></th>
<th>Yao+GLLM</th>
<th>Pretzel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>provider-side</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU time:</td>
<td>110x</td>
<td>1.8x</td>
</tr>
<tr>
<td><strong>network transfers:</strong></td>
<td>109x</td>
<td>5.4x</td>
</tr>
<tr>
<td><strong>client-side</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>storage:</td>
<td>288MB</td>
<td>720MB</td>
</tr>
</tbody>
</table>
Related work

• Improving performance of general purpose 2PC
  [SEC11, CCS12, NDSS12, S&P12, SEC12, S&P14, EUROCRYPT15]

• Secure dot-product 2PC protocols [CSFW01, ACSAC01, KDD02, AusDM07, PAKDD14, NSPW02, ICISC04, HICSS10, WiCOM10, CollaborateCom15]

• Privacy preserving data mining [CRYPTO00, SDM04, KDD05, ESORICS05, CCS15, ICDM03, VLDB Journal 08, SIAM05, Information Systems 09]
Take-away points from this talk

Pretzel:

- email encryption
- provider-supplied functions
- low cost

So, why don’t email service providers deploy end-to-end email encryption?