Techniques in privacy-preserving machine learning

Hao Chen, Microsoft Research

Private AI Bootcamp

12.2.2019

Introduction

- Cryptography and privacy research group at Microsoft research
  - Researching new cryptographic methods and applications.
  - Areas of focus:
    - Homomorphic Encryption
    - Post-quantum cryptography
    - Multiparty computation
    - Protocols and primitives (zkproof, oram, abe, verifiable computation, …)
    - Compiler and hardware acceleration of cryptographic primitives
  - Members:
    - Kristin Lauter
    - Wei Dai
    - Yongsoo Song
    - Kim Laine
    - Hao Chen
    - Melissa Chase
    - Esha Ghosh

Beginning

- Why we should work on HE
  - For theoriests -- relatively new research area, lots of open problems
    - Non-lattice based constructions?
  - For practioners -- useful as a building block, not just an end-to-end solution
    - For example, 1-level HE is very fast now.

Overview

- Recap on privacy problems
  - Types of leakage
  - Alternative Technologies
- HE-specific techniques for PPML
  - Example1: linear functions (matrix-vector product)
  - Example2: nonlinear functions (approximation + polynomial evaluation)
- Additional remarks
  - Cost model
  - Circuit privacy
  - Threat model

Part 1: A tour of HE in PPML

Type of leakage in ML

- Explicit leakage
  - Leakage of data in inference and training
    - Training data, testing data, model

- Implicit leakage
  - Leakage of training data through trained model
  - Leakage of data / model through result of query

Privacy enhancing technologies in ML

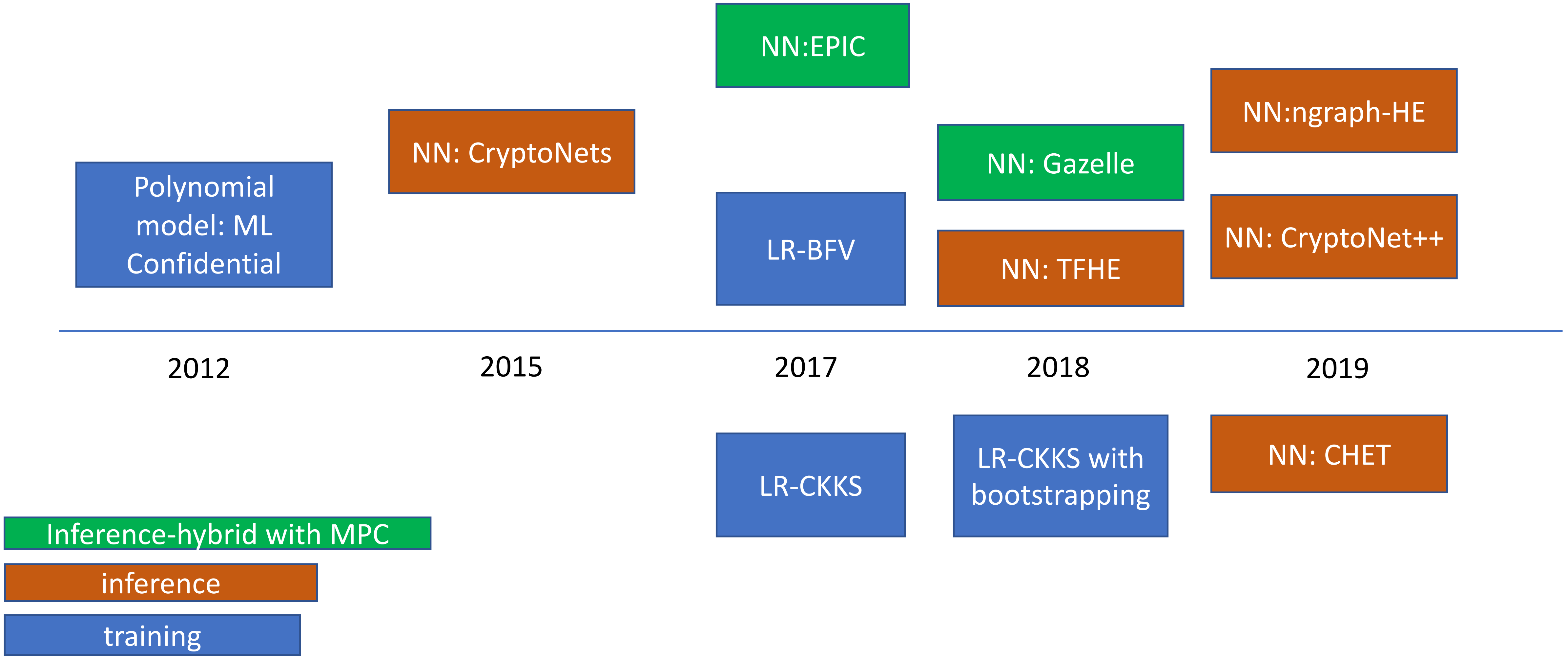
- Differential privacy [hardness to identify individual data entry]
  - Only a meaningful notion for a large database and “coarse” computation
  - Add noise in a controlled and structured way
  - For training
  - For answering queries (one-sided privacy)
- Federated learning
  - Turns explicit leakage [data] into implicit ones [aggregated gradients]
- Cryptographic techniques
  - Secure multiparty computation (MPC)
  - Homomorphic Encryption (HE)
  - Eliminate explicit leakage issue, for all participating parties

Homomorphic Encryption in PPML

- HE allows secure inferencing and training in the semi-honest model

- Workflow (2 party case):
  - Inference:
    - Client -> Server: Enc(query)
    - Server: Eval(Inference, Enc(query), model) -> Enc(result)
    - Client: Dec(Enc(result)) - > result
  - Training
    - P1 -> P2: Enc(DB1)
    - P2: Eval(Training, Enc(DB1), DB2) -> Enc(model)
    - P1: Dec(Enc(model)) -> model

HE for PPML in the literature



LR = logistic regression

Example: logistic regression via HE

Repeated evaluation of the weight update function

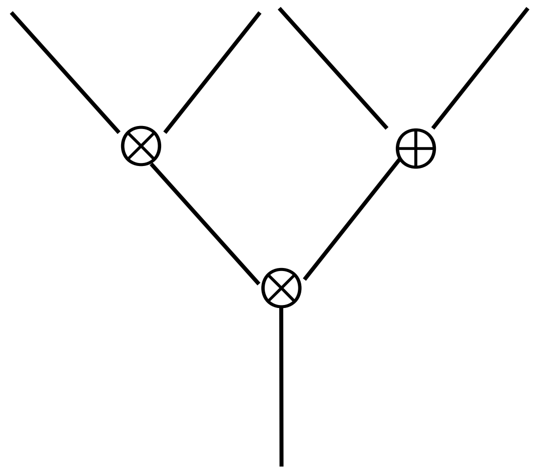
$$w = w - \alpha(\sigma(Xw) - y)X^T$$

$\alpha > 0$ is learning rate, $\sigma(x) = (1 + e^{-x})^{-1}$.

Requires homomorphic evaluations of sigmoid and matrix-vector products.

Recall: (multiplicative) depth of a circuit

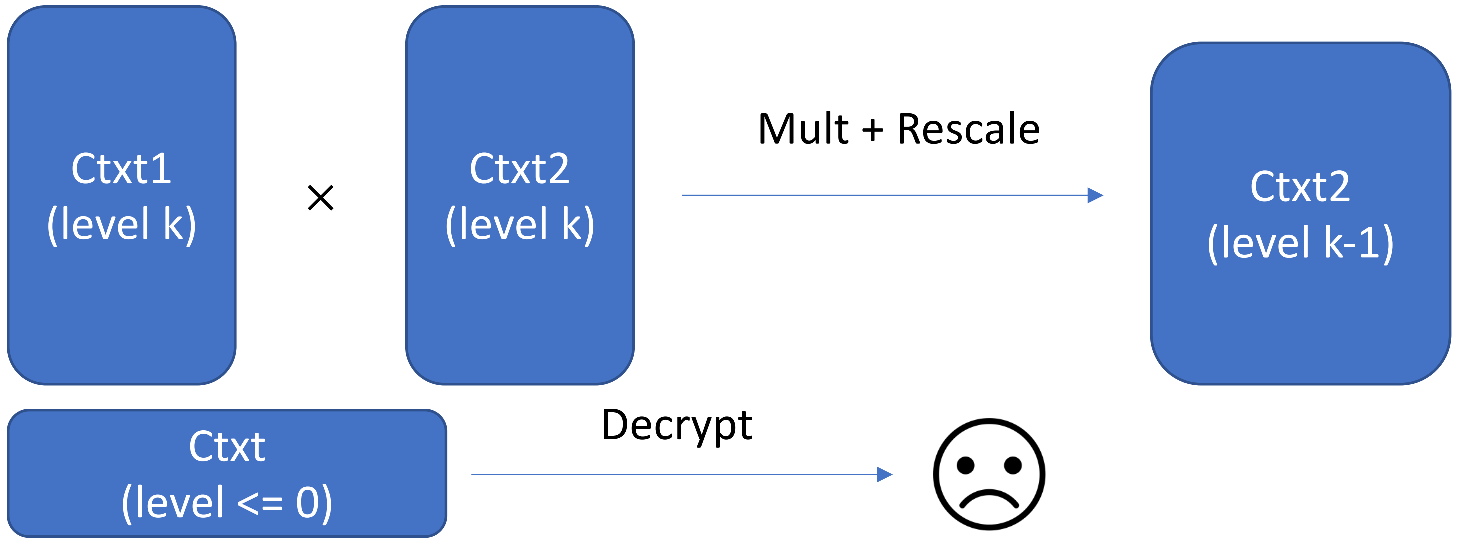
Depth = Maximum number of multiplications on a path



Depth = 2

Cost models of (leveled) HE

- Ciphertext modulus (q) grows linearly with the required depth (L)
- HE operation costs scales by $\mathrm{poly}\log(q)$
- For efficiency, it is crucial to reduce q, which boils down to reducing L.
- Reason (CKKS): mult+rescale consumes level of ciphertext.
- Ctxt needs to have positive level to be able to decrypt correctly.



Dealing with deep computations

- Training algorithms usually require a large-depth circuit
- Overhead of HE grows with depth

Option 1 : interactive solution

Use the sk owner (client) as decryption oracle.

Client will decrypt a “noisy” ciphertext. Then, it re-encrypts the message and sends it back. Now, further HE operation can be done.

✓ efficient

☹ interactive

More vulnerable to malicious attacks

Option 2 : bootstrapping

Server evaluates the decryption circuit homomorphically.

✓ non-interactive

☹ not yet efficient

Bootstrapping

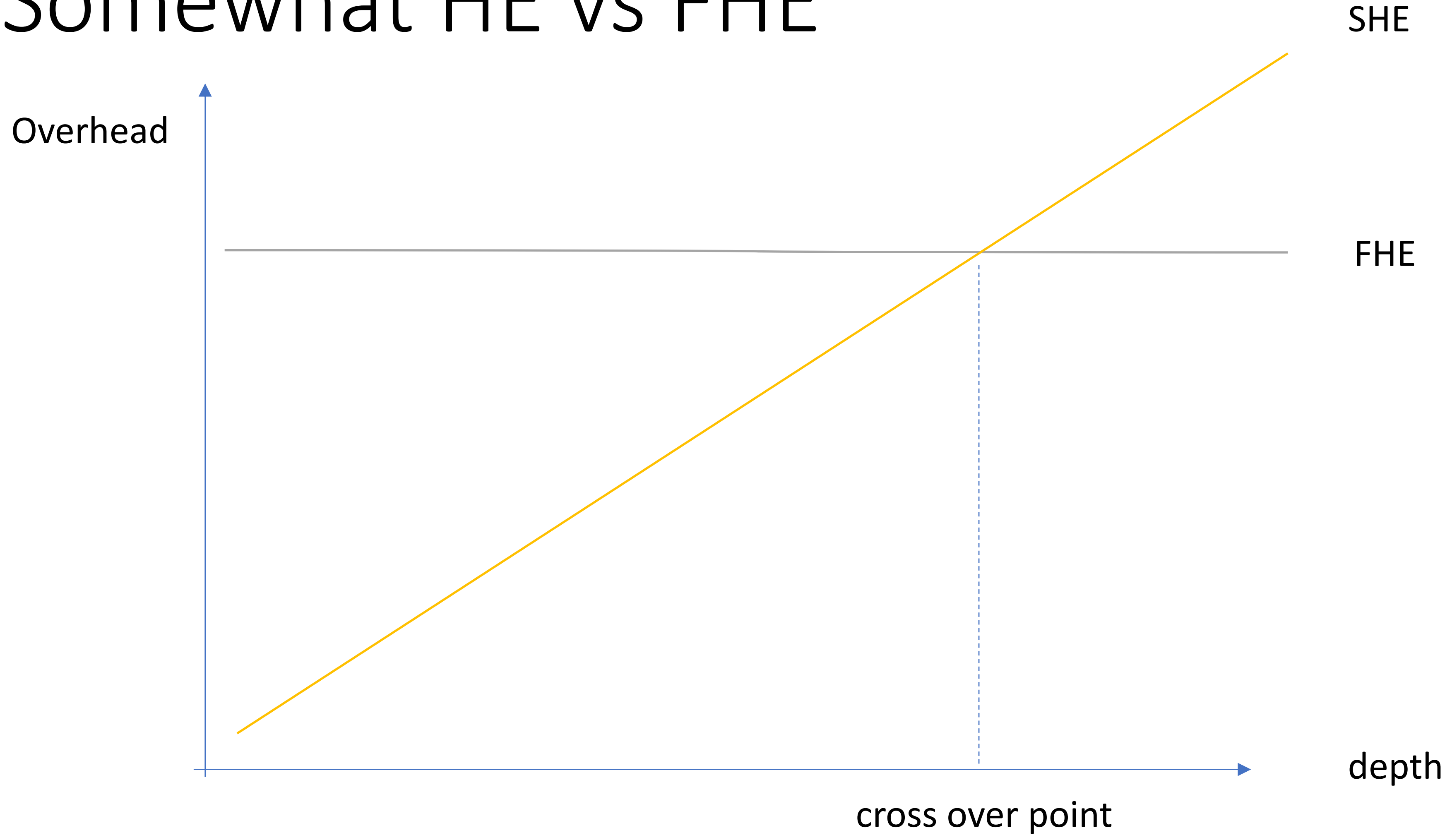
- Computing the decryption circuit homomorphically

Eval(Dec, ct, Enc(sk)) -> Enc(Dec(ct, sk)) -> Enc(m)

SHE

FHE

Somewhat HE vs FHE



Part 2: Algorithms for HE

Native HE operations are not sufficient for many ML applications. Thus, we need specifically designed algorithms.

We will go over two examples in detail:

- nonlinear functions
- matrix-vector products

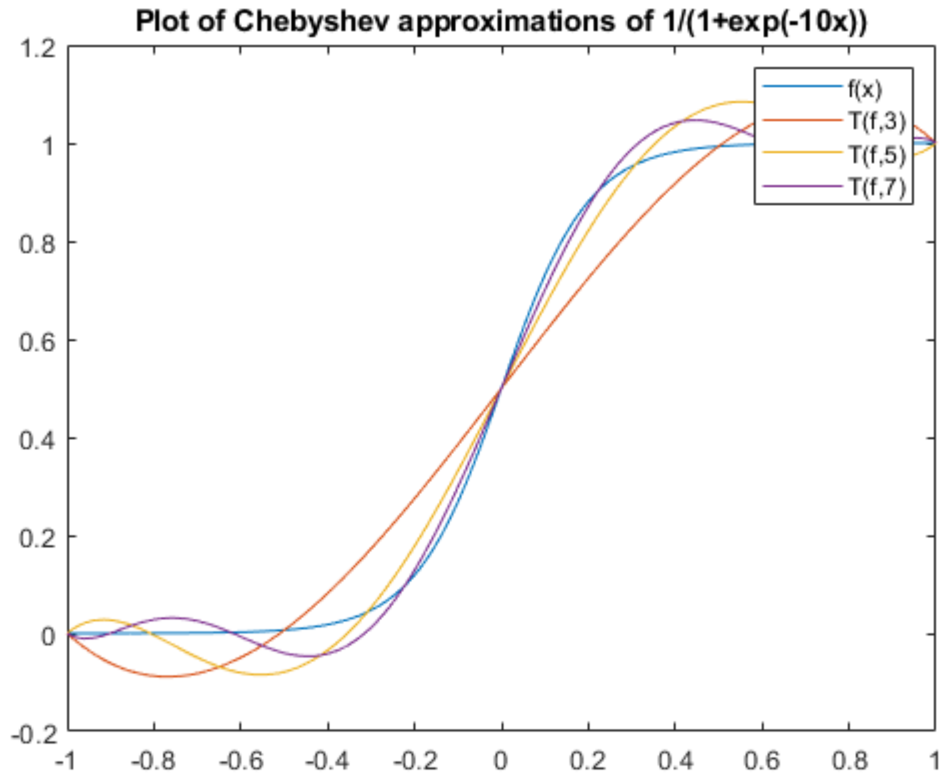
HE for nonlinear functions

- Function on finite intervals can be approximated by <u>polynomials</u>.

- There's many measures for approximation, e.g. , the infinity norm

  $$p(f, n) \coloneqq argmin\{\ |f(x) - p(x)|_\infty : \deg(p) \le n\}$$

- The <u>Chebyshev interpolation</u> is often a good candidate



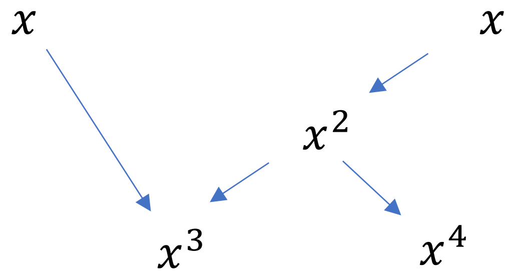
How to evaluate a polynomial

- Say we have $p(x) = \sum a_i x^i$. How to evaluate it efficiently in HE?
- Horner's method

$$p(x) = a_0 + x\left(a_1 + x\left(a_2 + \cdots + x(a_{n-1} + \, x a_n)\right)\cdots\right)$$

  - Required Levels: O(n)

- Tree computation
  - Compute all powers of x in a tree
  - Then, perform a dot product between
    - (1, x, …, x^n-1) and (a_0, a_1, \ldots, a_{n-1})
  - Required Levels: O(log n)



- Smaller levels -> smaller parameters -> better efficiency
- Level is not multiplicative: Lvl(ct1*ct2) = max(Lvl(ct1), Lvl(ct2)) + 1.

HE operation cost cheat-sheet

Operation	Asymptotic cost	Actual Cost in ms *
Ctxt rotation (generic/single)	N log N	809 / 161
Ct-Ct mult (Relin+Rescale included)	N log N	202
Ct-Pt mult	N	3.7
Ct-Ct add	N	1.2

* N = 32768

Reducing ct-ct mults in polynomial evaluation

- The tree method still requires O(n) ct-ct mults.
- Ct-ct mults cost much more than Ct-pt mults.
  - Thus, we wish to optimize.
- Babystep-Giantstep => $3\sqrt{n}$ ct-ct mults
- Paterson-Stockmeyer => $\sqrt{2n} + O(\log n)$ ct-ct mults

Babystep-Giantstep

- Say $n = n_1 \cdot n_2$. We can write
  $$p(x) = \sum_i a_i x^i = \sum_{j,k} a_{jn_1+k} x^{jn_1+k} = \sum_j x^{jn_1} \sum_k a_{jn_1+k} x^k$$
- Then, we can evaluate the x^k, and x^jn_1, and then compute the products.
- Requires a total of $2n_2 + n_1$ ct-ct mults. Now set $n_1, n_2 \approx \sqrt{n}$.
- How many levels does it consume?
- $Lvl\left(x^{jn_1}\right) \leq \log(n)$, $Lvl\left(x^k\right) \leq \log(n_2)$
- So $Lvl\left(p(x)\right) \leq \log(n) + 1$=> just takes 1 more level than tree compute

Babystep-Giantstep polyeval in action

- Code example: evaluate a random degree-15 polynomial

- Code can be accessed at https://github.com/haochenuw/algorithms-in-SEAL/

Method	Ctxt size	Time
Horner	? (Exercise)	2,000ms
Tree	?	350ms
Bstep-Gstep	?	240ms

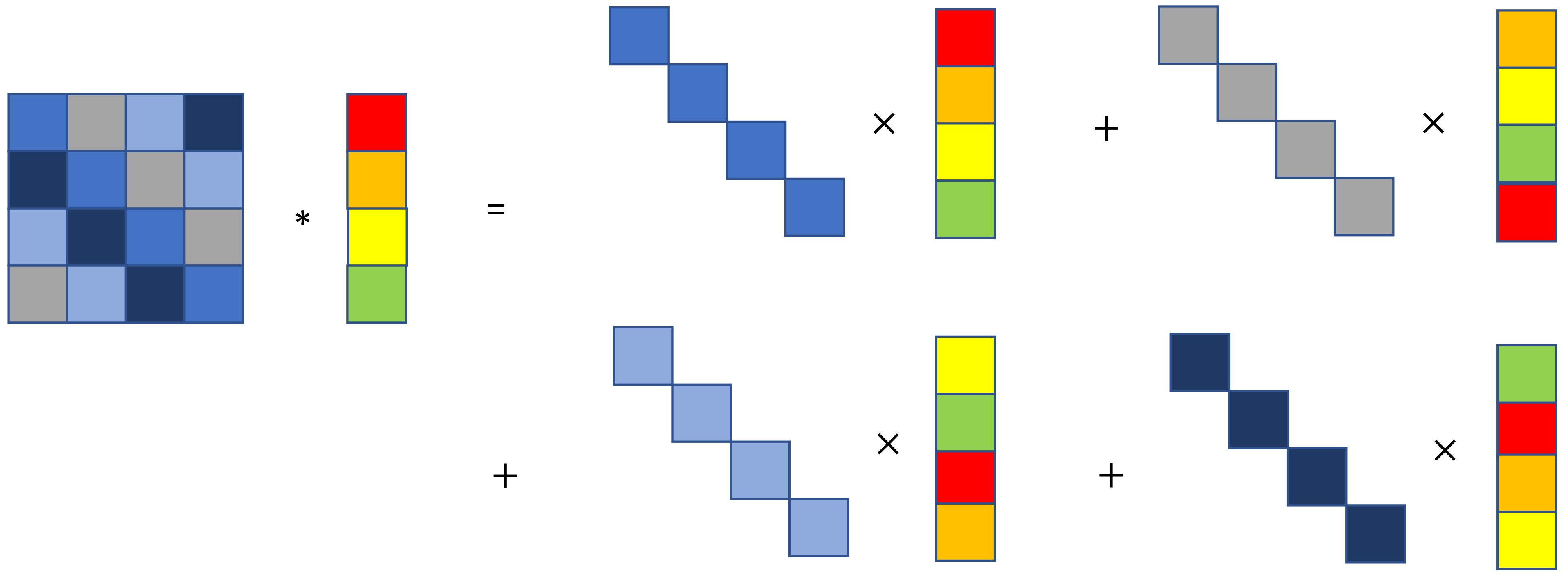
Linear operations in HE

- Matrix-vector product, or matrix-matrix product
- Native operations:
  - a + b
  - a * b = (a[1]b[1], a[2]b[2], …, a[n] b[n])
  - Rot(a, k) = (a[k+1], a[k+2] , … , a[(k+n) mod n])
- Goal: design algorithms for linear operations, using as few native operations as possible

- Are both operands encrypted? If not, might have a faster algorithm

Plain matrix times encrypted vector

- “Diagonal” method from Halevi and Shoup



It requires n ct-pt mults and n rotations – can we do better?

Code example: diagonal method

Babystep-giantstep again!

- The diagonal method for matrix-vector product can be written as

$$\sum_{i=0}^{n-1} diag(M, i) * rot(v, i)$$

When $n = n_1 n_2$, this can again be factored as

$$\sum_{k=0}^{n_2-1} \sum_{j=0}^{n_1-1} diag(M, kn_1 + j) * rot(v, kn_1 + j)$$

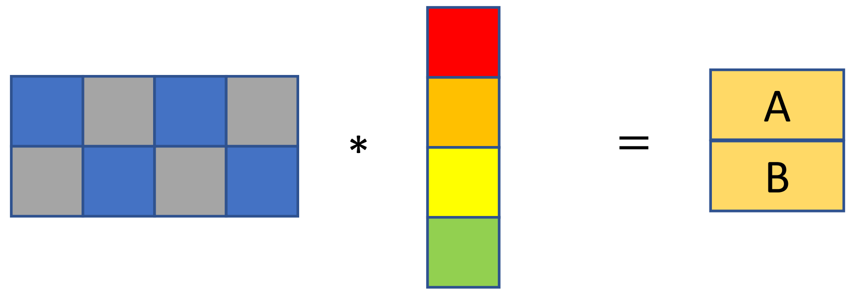
$= \sum_{k=0}^{n_2-1} Rot(\sum_{j=0}^{n_1-1} diag(M, kn_1 + j)' * Rot(v, j), kn_1)$

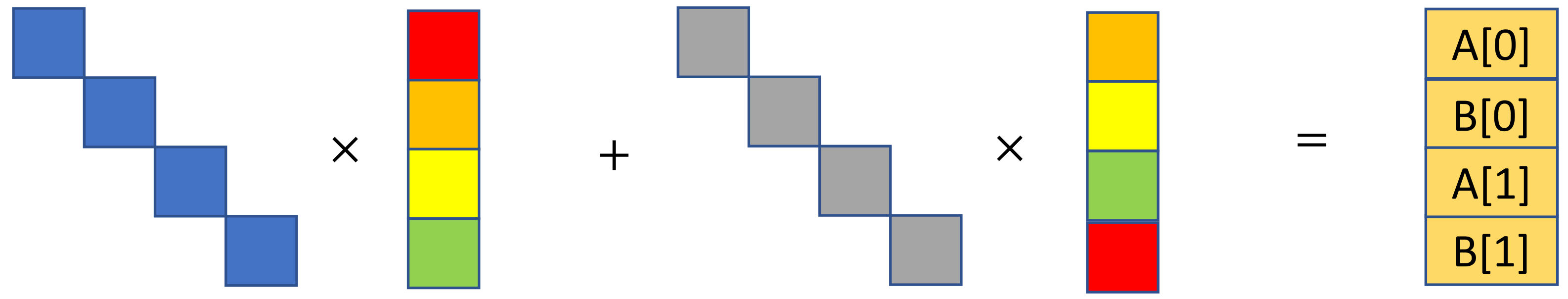
This reduces the number of rotations to $n_1 + n_2$

Matrix-vector: non-square case

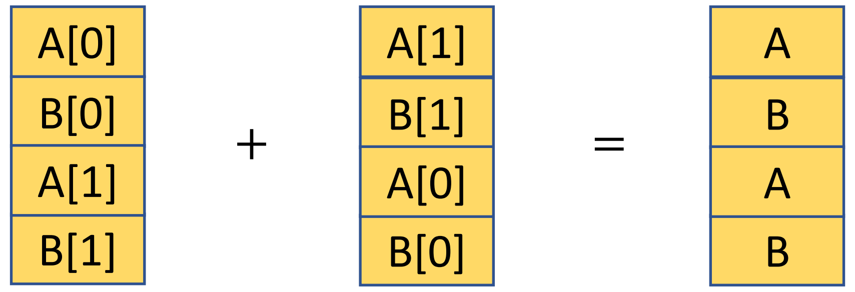
- We are multiplying a m-by-n matrix to a length-n vector.
- “Hybrid” method (Gazelle, Usenix ’18) works well when n = mk
- Work with “extended diagonals”
- Requires m + log(k) rotations

Hybrid method





(m rotations)



(log(n/m) rotations)

Remark on circuit privacy

- Informally, a ciphertext should not leak information about “how it is generated”
- Formally, there should be a “Sanitize” procedure, such that
    - Sanitize(ct1) $\approx$ Sanitize(ct2) if ct1 and ct2 encrypt same message.
- If not ensured, can leak information to the party with secret key.

- One solution: use noise flooding $\rightarrow$ adding an encryption of zero with large noise.

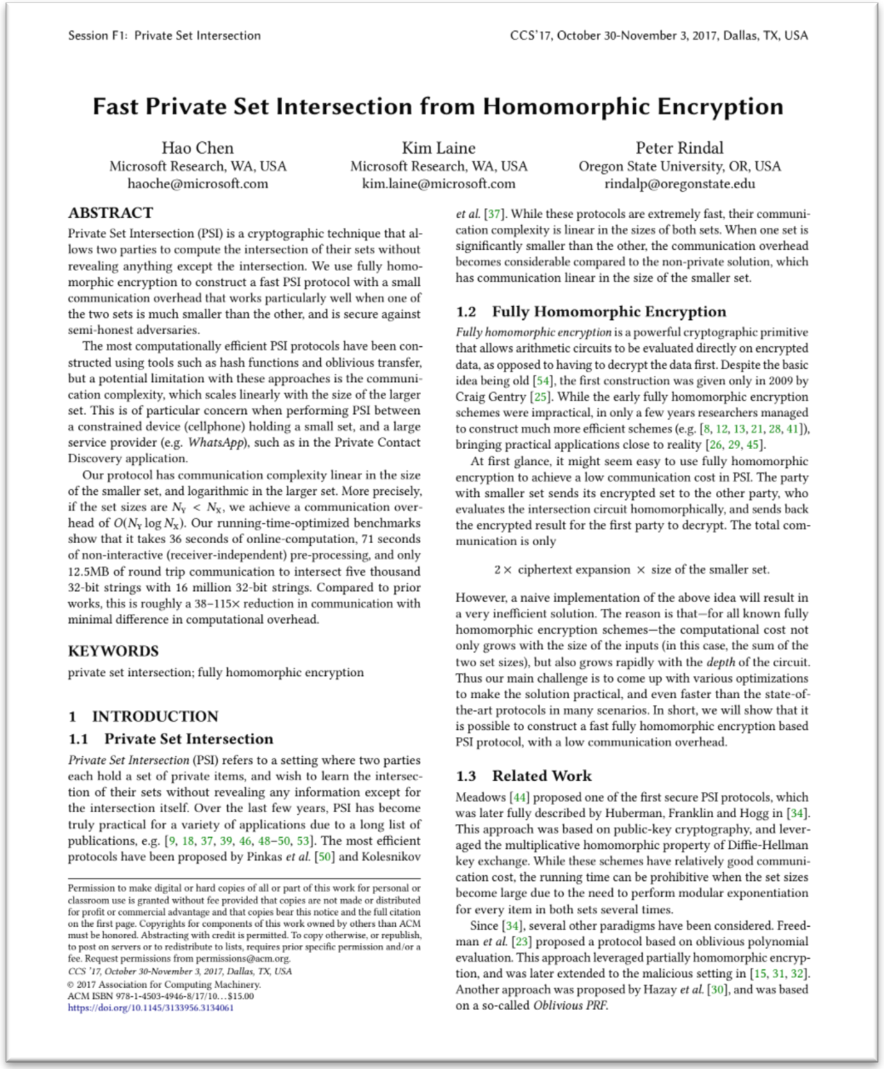
Future research problems

- Beyond semi-honest threat model?
- Model selection, Model Validation
- Deeper integration with MPC
- Faster FHE?
- Beyond circuit model?

References

- HE for ML
  - Chen, Hao, Ran Gilad-Bachrach, Kyoohyung Han, Zhicong Huang, Amir Jalali, Kim Laine, and Kristin Lauter. "Logistic regression over encrypted data from fully homomorphic encryption."
  - Kim, Andrey, Yongsoo Song, Miran Kim, Keewoo Lee, and Jung Hee Cheon. "Logistic regression model training based on the approximate homomorphic encryption."
  - Jiang, Xiaoqian, Miran Kim, Kristin Lauter, and Yongsoo Song. "Secure outsourced matrix computation and application to neural networks."
  - Chen, Hao, Ilaria Chillotti, Yihe Dong, Oxana Poburinnaya, Ilya Razenshteyn, and M. Sadegh Riazi. "Sanns: Scaling up secure approximate k-nearest neighbors search."
- "non-ML" applications of HE
  - Chen, Hao, Zhicong Huang, Kim Laine, and Peter Rindal. "Labeled PSI from Fully Homomorphic Encryption with Malicious Security."
  - Angel, Sebastian, Hao Chen, Kim Laine, and Srinath Setty. "PIR with compressed queries and amortized query processing."
- Bootstrapping
  - Chen, Hao, and Kyoohyung Han. "Homomorphic lower digits removal and improved FHE bootstrapping."
  - Cheon, Jung Hee, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. "Bootstrapping for approximate homomorphic encryption."
  - Chen, Hao, Ilaria Chillotti, and Yongsoo Song. "Improved bootstrapping for approximate homomorphic encryption."

Private Set Intersection



Session F1: Private Set Intersection

CCS'17, October 30-November 3, 2017, Dallas, TX, USA

Fast Private Set Intersection from Homomorphic Encryption

Hao Chen
Microsoft Research, WA, USA
haoche@microsoft.com

Kim Laine
Microsoft Research, WA, USA
kim.laine@microsoft.com

Peter Rindal
Oregon State University, OR, USA
rindalp@oregonstate.edu

ABSTRACT

Private Set Intersection (PSI) is a cryptographic technique that allows two parties to compute the intersection of their sets without revealing anything except the intersection. We use fully homomorphic encryption to construct a fast PSI protocol with a small communication overhead that works particularly well when one of the two sets is much smaller than the other, and is secure against semi-honest adversaries.

The most computationally efficient PSI protocols have been constructed using tools such as hash functions and oblivious transfer, but a potential limitation with these approaches is the communication complexity, which scales linearly with the size of the larger set. This is of particular concern when performing PSI between a constrained device (cellphone) holding a small set, and a large service provider (e.g. *WhatsApp*), such as in the Private Contact Discovery application.

Our protocol has communication complexity linear in the size of the smaller set, and logarithmic in the larger set. More precisely, if the set sizes are $N_Y < N_X$, we achieve a communication overhead of $O(N_Y \log N_X)$. Our running-time-optimized benchmarks show that it takes 36 seconds of online-computation, 71 seconds of non-interactive (receiver-independent) pre-processing, and only 12.5MB of round trip communication to intersect five thousand 32-bit strings with 16 million 32-bit strings. Compared to prior works, this is roughly a 38–115× reduction in communication with minimal difference in computational overhead.

KEYWORDS

private set intersection; fully homomorphic encryption

1 INTRODUCTION

1.1 Private Set Intersection

Private Set Intersection (PSI) refers to a setting where two parties each hold a set of private items, and wish to learn the intersection of their sets without revealing any information except for the intersection itself. Over the last few years, PSI has become truly practical for a variety of applications due to a long list of publications, e.g. [9, 18, 37, 39, 46, 48–50, 53]. The most efficient protocols have been proposed by Pinkas *et al.* [50] and Kolesnikov *et al.* [37]. While these protocols are extremely fast, their communication complexity is linear in the sizes of both sets. When one set is significantly smaller than the other, the communication overhead becomes considerable compared to the non-private solution, which has communication linear in the size of the smaller set.

1.2 Fully Homomorphic Encryption

Fully homomorphic encryption is a powerful cryptographic primitive that allows arithmetic circuits to be evaluated directly on encrypted data, as opposed to having to decrypt the data first. Despite the basic idea being old [54], the first construction was given only in 2009 by Craig Gentry [25]. While the early fully homomorphic encryption schemes were impractical, in only a few years researchers managed to construct much more efficient schemes (e.g. [8, 12, 13, 21, 28, 41]), bringing practical applications close to reality [26, 29, 45].

At first glance, it might seem easy to use fully homomorphic encryption to achieve a low communication cost in PSI. The party with smaller set sends its encrypted set to the other party, who evaluates the intersection circuit homomorphically, and sends back the encrypted result for the first party to decrypt. The total communication is only

$$2\times \text{ ciphertext expansion } \times \text{ size of the smaller set.}$$

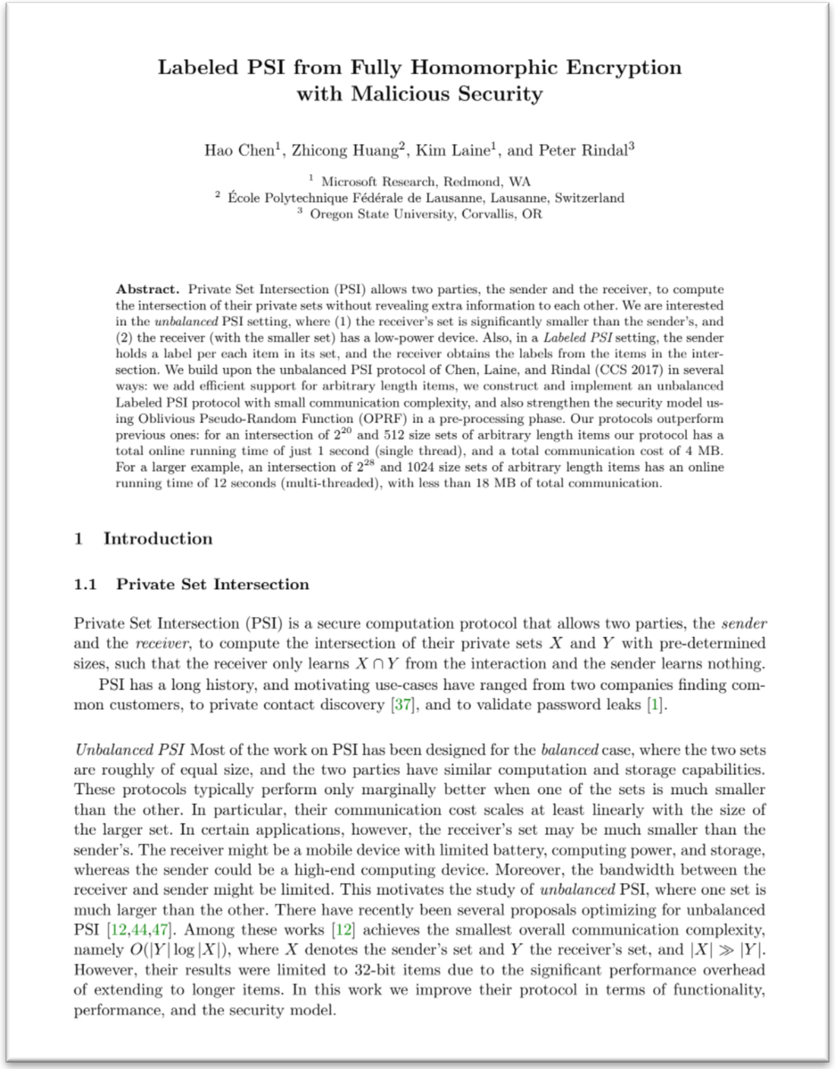
However, a naive implementation of the above idea will result in a very inefficient solution. The reason is that—for all known fully homomorphic encryption schemes—the computational cost not only grows with the size of the inputs (in this case, the sum of the two set sizes), but also grows rapidly with the *depth* of the circuit. Thus our main challenge is to come up with various optimizations to make the solution practical, and even faster than the state-of-the-art protocols in many scenarios. In short, we will show that it is possible to construct a fast fully homomorphic encryption based PSI protocol, with a low communication overhead.

1.3 Related Work

Meadows [44] proposed one of the first secure PSI protocols, which was later fully described by Huberman, Franklin and Hogg in [34]. This approach was based on public-key cryptography, and leveraged the multiplicative homomorphic property of Diffie-Hellman key exchange. While these schemes have relatively good communication cost, the running time can be prohibitive when the set sizes become large due to the need to perform modular exponentiation for every item in both sets several times.

Since [34], several other paradigms have been considered. Freedman *et al.* [23] proposed a protocol based on oblivious polynomial evaluation. This approach leveraged partially homomorphic encryption, and was later extended to the malicious setting in [15, 31, 32]. Another approach was proposed by Hazay *et al.* [30], and was based on a so-called *Oblivious PRF*.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CCS '17, October 30-November 3, 2017, Dallas, TX, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4946-8/17/10…$15.00
https://doi.org/10.1145/3133956.3134061



Labeled PSI from Fully Homomorphic Encryption with Malicious Security

Hao Chen[1], Zhicong Huang[2], Kim Laine[1], and Peter Rindal[3]

[1] Microsoft Research, Redmond, WA
[2] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[3] Oregon State University, Corvallis, OR

Abstract. Private Set Intersection (PSI) allows two parties, the sender and the receiver, to compute the intersection of their private sets without revealing extra information to each other. We are interested in the *unbalanced* PSI setting, where (1) the receiver's set is significantly smaller than the sender's, and (2) the receiver (with the smaller set) has a low-power device. Also, in a *Labeled PSI* setting, the sender holds a label per each item in its set, and the receiver obtains the labels from the items in the intersection. We build upon the unbalanced PSI protocol of Chen, Laine, and Rindal (CCS 2017) in several ways: we add efficient support for arbitrary length items, we construct and implement an unbalanced Labeled PSI protocol with small communication complexity, and also strengthen the security model using Oblivious Pseudo-Random Function (OPRF) in a pre-processing phase. Our protocols outperform previous ones: for an intersection of 2^{20} and 512 size sets of arbitrary length items our protocol has a total online running time of just 1 second (single thread), and a total communication cost of 4 MB. For a larger example, an intersection of 2^{28} and 1024 size sets of arbitrary length items has an online running time of 12 seconds (multi-threaded), with less than 18 MB of total communication.

1 Introduction

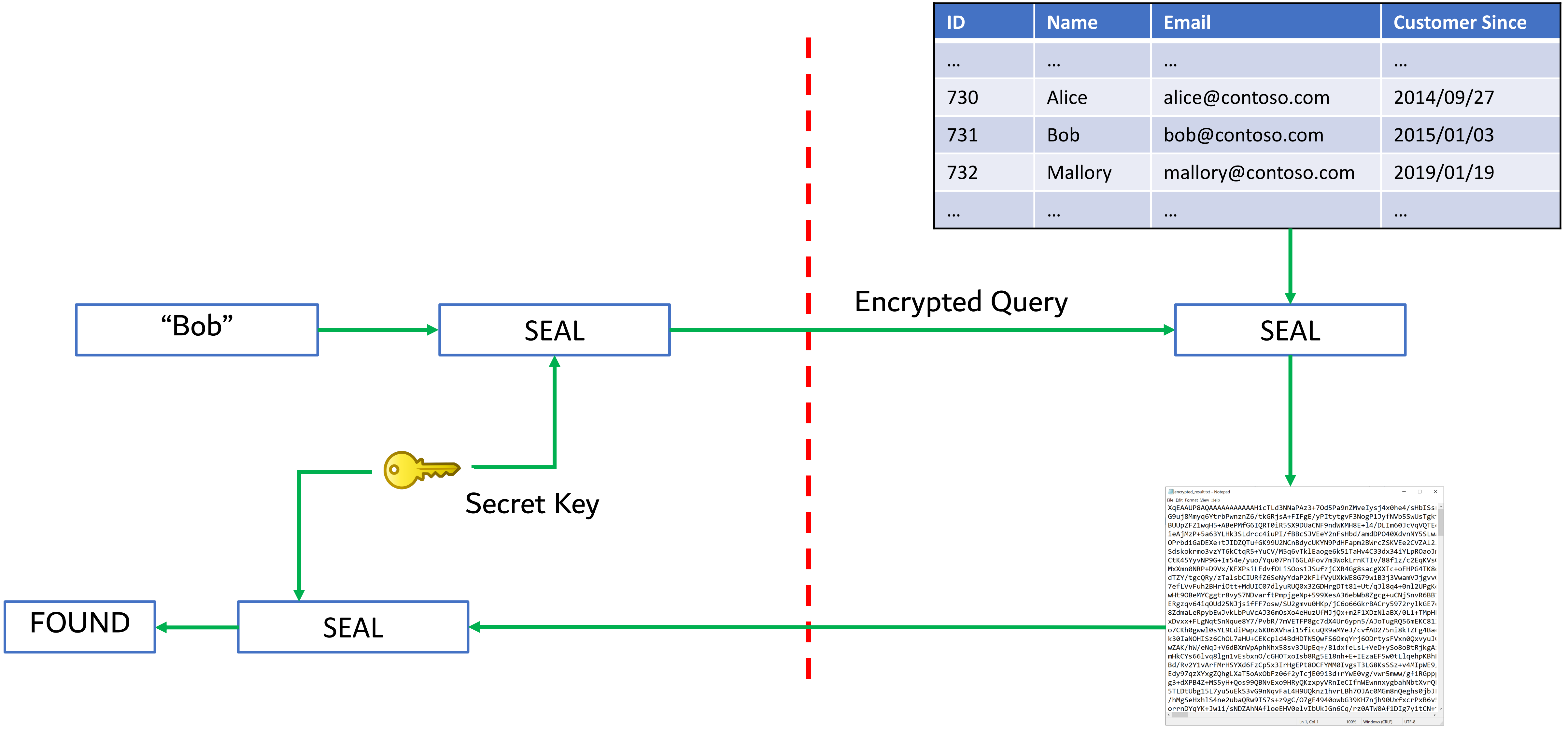
1.1 Private Set Intersection

Private Set Intersection (PSI) is a secure computation protocol that allows two parties, the *sender* and the *receiver*, to compute the intersection of their private sets X and Y with pre-determined sizes, such that the receiver only learns $X \cap Y$ from the interaction and the sender learns nothing.

PSI has a long history, and motivating use-cases have ranged from two companies finding common customers, to private contact discovery [37], and to validate password leaks [1].

Unbalanced PSI Most of the work on PSI has been designed for the *balanced* case, where the two sets are roughly of equal size, and the two parties have similar computation and storage capabilities. These protocols typically perform only marginally better when one of the sets is much smaller than the other. In particular, their communication cost scales at least linearly with the size of the larger set. In certain applications, however, the receiver's set may be much smaller than the sender's. The receiver might be a mobile device with limited battery, computing power, and storage, whereas the sender could be a high-end computing device. Moreover, the bandwidth between the receiver and sender might be limited. This motivates the study of *unbalanced* PSI, where one set is much larger than the other. There have recently been several proposals optimizing for unbalanced PSI [12,44,47]. Among these works [12] achieves the smallest overall communication complexity, namely $O(|Y| \log |X|)$, where X denotes the sender's set and Y the receiver's set, and $|X| \gg |Y|$. However, their results were limited to 32-bit items due to the significant performance overhead of extending to longer items. In this work we improve their protocol in terms of functionality, performance, and the security model.

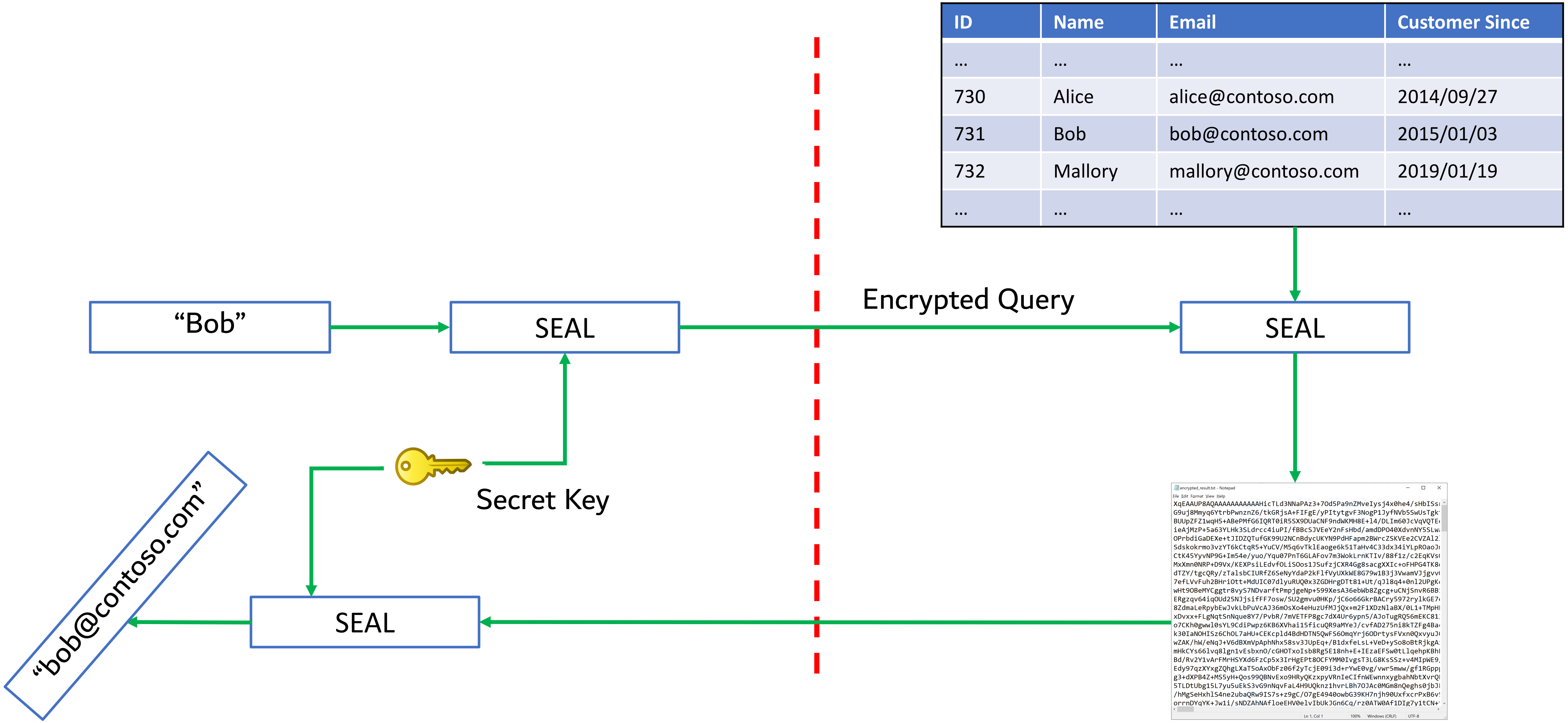
Private Set Intersection



ID	Name	Email	Customer Since
…	…	…	…
730	Alice	alice@contoso.com	2014/09/27
731	Bob	bob@contoso.com	2015/01/03
732	Mallory	mallory@contoso.com	2019/01/19
…	…	…	…

XqEAAUP8AQAAAAAAAAAAAHicTLd3NNaPAz3+7Od5Pa9nZMveIysj4x0he4/sHbISs
G9uj8Mmyq6YtrbPwnznZ6/tkGRjsA+FIFgE/yPItytgvF3NogP1JyfNVb5SwUsTgk
BUUpZFZ1wqH5+ABePMfG6IQRT0iR5SX9DUaCNF9ndWKMH8E+l4/DLIm60JcVqVQTE
ieAjMzP+5a63YLHk3SLdrcc4iuPI/fBBcSJVEeY2nFsHbd/amdDPO40XdvnNY5SLw
OPrbdiGaDEXe+tJIDZQTufGK99U2NCnBdycUKYN9PdHFapm2BWrcZSKVEe2CVZAl2
SdskokrmO3vzYT6kCtqR5+YuCV/M5q6vTklEaoge6k51TaHv4C33dx34iYLpROaoJ
CtK45YyvNP9G+Im54e/yuo/Yqu07PnT6GLAFov7m3WokLrnKTIv/88f1z/c2EqKVs
MxXmn0NRP+D9Vx/KEXPsiLEdvfOLiSOos1JSufzjCXR4Gg8sacgXXIc+oFHPG4TK8
dTZY/tgcQRy/zTalsbCIURfZ6SeNyYdaP2kFlfVyUXkWE8G79w1B3j3VwamVJjgvv
7efLVvFuh2BHriOtt+MdUIC07dlyuRUQ0x3ZGDHrgDTt81+Ut/qJl8q4+0nl2UPgK
wHt9OBeMYCggtr8vyS7NDvarftPmpjgeNp+599XesA36ebWb8Zgcg+uCNjSnvR6BB
ERgzqv64iqOUd25NJjsifFF7osw/SU2gmvu0HKp/jC6o66GkrBACry5972rylkGE7
8ZdmaLeRpybEwJvkLbPuVcAJ36mOsXo4eHuzUfMJjQx+m2F1XDzNlaBX/0L1+TMpH
xDvxx+FLgNqtSnNque8Y7/PvbR/7mVETFP8gc7dX4Ur6ypn5/AJoTugRQ56mEKC81
o7CKh0gwwl0sYL9CdiPwpz6KB6XVhai15ficuQR9aMYeJ/cvfAD275ni8kTZFg4Ba
k30IaNOHISz6ChOL7aHU+CEKcpld4BdHDTN5QwFS6OmqYrj6ODrtysFVxn0QxvyuJ
wZAK/hW/eNqJ+V6dBXmVpAphNhx58sv3JUpEq+/B1dxfeLsL+VeD+ySo8oBtRjkgA
mHkCYs66lvq8lgn1vEsbxnO/cGHOTxoIsb8Rg5E18nh+E+IEzaEFSw0tLlqehpKBh
Bd/Rv2Y1vArFMrHSYXd6FzCp5x3IrHgEPt8OCFYMM0IvgsT3LG8KsSSz+v4MIpWE9
Edy97qzXYxgZQhgLXaT5oAxObFz06f2yTcjE09i3d+rYwE0vg/vwr5mww/gf1RGpp
g3+dXPB4Z+MS5yH+Qos99QBNvExo9HRyQKzxpyVRnIeCIfnWEwnnxygbahNbtXvrQ
5TLDtUbg15L7yu5uEkS3vG9nNqvFaL4H9UQknz1hvrLBh7OJAc0MGm8nQeghs0jbJ
/hMgSeHxhlS4ne2ubaQRw9IS7s+z9gC/O7gE494OowbG39KH7njh90UxfxcrPxB6v
orrnDYqYK+Jw1i/sNDZAhNAfloeEHV0elvIbUkJGn6Cq/rz0ATW0Af1DIg7y1tCN+

Private Data Retrieval



ID	Name	Email	Customer Since
…	…	…	…
730	Alice	alice@contoso.com	2014/09/27
731	Bob	bob@contoso.com	2015/01/03
732	Mallory	mallory@contoso.com	2019/01/19
…	…	…	…

encrypted_result.txt - Notepad

```
XqEAAUP8AQAAAAAAAAAAAHicTLd3NNaPAz3+7Od5Pa9nZMveIysj4x0he4/sHbISs
G9uj8Mmyq6YtrbPwnznZ6/tkGRjsA+FIFgE/yPItytgvF3NogP1JyfNVb5SwUsTgk
BUUpZFZ1wqH5+ABePMfG6IQRT0iR5SX9DUaCNF9ndWKMH8E+l4/DLIm60JcVqVQTE
ieAjMzP+5a63YLHk3SLdrcc4iuPI/fBBcSJVEeY2nFsHbd/amdDPO40XdvnNY5SLw
OPrbdiGaDEXe+tJIDZQTufGK99U2NCnBdycUKYN9PdHFapm2BWrcZSKVEe2CVZAl2
Sdskokrmo3vzYT6kCtqR5+YuCV/M5q6vTklEaoge6k51TaHv4C33dx34iYLpROaoJ
CtK45YyvNP9G+Im54e/yuo/Yqu07PnT6GLAFov7m3WokLrnKTIv/88f1z/c2EqKVs
MxXmn0NRP+D9Vx/KEXPsiLEdvfOLiSOos1JSufzjCXR4Gg8sacgXXIc+oFHPG4TK8
dTZY/tgcQRy/zTalsbCIURfZ6SeNyYdaP2kFlfVyUXkWE8G79w1B3j3VwamVJjgvv
7efLVvFuh2BHriOtt+MdUIC07dlyuRUQ0x3ZGDHrgDTt81+Ut/qJl8q4+0nl2UPgK
wHt9OBeMYCggtr8vyS7NDvarftPmpjgeNp+599XesA36ebWb8Zgcg+uCNjSnvR6BB
ERgzqv64iqOUd25NJjsifFF7osw/SU2gmvu0HKp/jC6o66GkrBACry5972rylkGE7
8ZdmaLeRpybEwJvkLbPuVcAJ36mOsXo4eHuzUfMJjQx+m2F1XDzNlaBX/0L1+TMpH
xDvxx+FLgNqtSnNque8Y7/PvbR/7mVETFP8gc7dX4Ur6ypn5/AJoTugRQ56mEKC81
o7CKh0gwwl0sYL9CdiPwpz6KB6XVhai15ficuQR9aMYeJ/cvfAD275ni8kTZFg4Ba
k30IaNOHISz6ChOL7aHU+CEKcpld4BdHDTN5QwFS6OmqYrj6ODrtysFVxn0QxvyuJ
wZAK/hW/eNqJ+V6dBXmVpAphNhx58sv3JUpEq+/B1dxfeLsL+VeD+ySo8oBtRjkgA
mHkCYs66lvq8lgn1vEsbxnO/cGHOTxoIsb8Rg5E18nh+E+IEzaEFSw0tLlqehpKBh
Bd/Rv2Y1vArFMrHSYXd6FzCp5x3IrHgEPt8OCFYMM0IvgsT3LG8KsSSz+v4MIpWE9
Edy97qzXYxgZQhgLXaT5oAxObFz06f2yTcjE09i3d+rYwE0vg/vwr5mww/gf1RGpp
g3+dXPB4Z+MS5yH+Qos99QBNvExo9HRyQKzxpyVRnIeCIfnWEwnnxygbahNbtXvrQ
5TLDtUbg15L7yu5uEkS3vG9nNqvFaL4H9UQknz1hvrLBh7OJAc0MGm8nQeghs0jbJ
/hMgSeHxhlS4ne2ubaQRw9IS7s+z9gC/O7gE4940owbG39KH7njh90UxfxcrPxB6v
orrnDYqYK+Jw1i/sNDZAhNAfloeEHV0elvIbUkJGn6Cq/rz0ATW0Af1DIg7y1tCN+
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8