Μελέτη Πρωτοκόλλων Ασφαλούς Υπολογισμού Πολλών Μερών (Study on Secure Multi Party Computation Protocols)

Θεόδωρος Συμεωνίδης
Α.Μ. 1064870

Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Πατρών

27 Οκτωβρίου 2022

Εισαγωγή

Αντικείμενο μελέτης

Το αντικείμενο αυτής της διπλωματικής εργασίας :

- Μελέτη Πρωτοκόλλων Ασφαλούς Υπολογισμού Πολλών Μερών (Secure Multi Party Computation (SMPC) Protocols)
- Εφαρμογή τους στην κατασκευή μιας Ασφαλούς Υπολογισμού Δύο Μερών BLAS Level-1 Βιβλιοθήκης, για πρώτη φορά στη βιβλιογραφία

Για να επιτευχθεί το παραπάνω έγινε εισαγωγή και αναπτύχθηκε το εξής θεωρητικό υπόβαθρο :

- Μαθηματικά της κρυπτογραφίας
- Σύγχρονα κρυπτογραφικά εργαλεία που χρησιμοποιούνται σε πρωτόκολλα SMPC
- Θεωρητικά αποδείξιμη ασφάλεια

Βασικοί ορισμοί

Ασφαλής Υπολογισμός : Όλες οι υπολογιστικές μέθοδοι που επιτρέπουν τον υπολογισμό επάνω σε δεδομένα κρατώντας τα δεδομένα μυστικά. Χρησιμοποιείται σε περιπτώσει όπου δεδομένα ή ένα μέρος των δεδομένων μπορεί να ανήκουν σε κάποιον τρίτο ο οποίος δε θέλει να μας φανερώσει τις τιμές τους.

Παραδείγματα :

- Υπολογισμός του εσωτερικού γινομένου δυό διανυσμάτων από ένα άτομο ενώ τα διανύσματα παρέχονται από κάποιο άλλο άτομο. Το πρώτο άτομο μπορεί να υπολογίσει το αποτέλεσμα χωρίς να μάθει τις πραγματικές τιμές των διανυσμάτων αυτών του δεύτερου ατόμου
- Υπολογισμός της Ευκλείδιας Νόρμας ενός διανύσματος από το οποίο κάθε άτομο μιας ομάδας άτομο διαθέτει ένα στοιχείο

Επαληθεύσιμος Υπολογισμός : Όλες οι υπολογιστικές μέθοδοι που επιτρέπουν την επαλήθευση πως η έξοδος του υπολογισμού που εξήγαγαν είναι πράγματι η έξοδος του υπολογισμού και των δεδομένων που τους δόθηκαν.

Βασικοί ορισμοί

Κατηγορίες Ασφαλούς και Επαληθεύσιμου Υπολογισμού

- Ασφαλής Αναθέσιμος Υπολογισμός : Μη διαδραστικά πρωτόκολλα υπολογισμού (π.χ. HE, SWHE, FHE)
- **Ασφαλής Υπολογισμός Πολλών Μερών** : Διαδραστικά πρωτόκολλα υπολογισμού

Ασφαλής Υπολογισμός Πολλών Μερών (Secure Multi Party Computation (SMPC))

Ορισμός

Ασφαλής Υπολογισμός Πολλών Μερών (Secure Multi Party Computation ή SMPC) : Είναι μια κλάση διαδραστικών κρυπτογραφικών σχημάτων που επιτρέπουν σε συμμετέχοντες $P_1, P_2, \ldots, P_n$ με αντίστοιχες εισόδους $x_1, x_2, \ldots, x_n$ να υπολογίσουν τη συνάρτηση την έξοδο y της συνάρτησης f, ως $y = f(x_1, x_2, \ldots, x_n)$ και ικανοποιούν τις παρακάτω ιδιότητες :

- Ορθότητα : Η έξοδος y είναι η σωστή έξοδος της συνάρτησης για τις δεδομένες εισόδους.
- Ιδιωτικότητα : Η έξοδος y είναι η μόνη πληροφορία που φανερώνει το πρωτόκολλο σε έναν αντίπαλο που ελέγχει κάποιους διεφθαρμένους συμμετέχοντες.

Εφαρμογές

Εφαρμογές Ασφαλούς Υπολογισμού Πολλών Μερών :

- Κατανεμημένες ψηφοφορίες και ιδιωτικές ψήφους
- Μηχανική Μάθηση που Διατηρεί την Ιδιωτικότητα (Privacy Preserving Machine Learning)
- Στατιστική που Διατηρεί την Ιδιωτικότητα (Privacy Preserving Statistics)
- Ψηφιακές Υπογραφές Κατωφλίου (Threshold Digital Signatures) με μοναδικό ιδιωτικό κλειδί το οποίο διανέμεται σε κατάλληλη μορφή στους συμμετέχοντες ώστε αν ο κατάλληλος αριθμός συμμετεχόντων συνεργαστεί να μπορεί να παράξει μια υπογραφή χωρίς να χρειάζεται να ανακατασκευαστεί το κλειδί

Κατηγορίες ως προς το μοντέλο αναπαράστασης υπολογισμού

Κατηγορίες SMPC πρωτοκόλλων ως προς το μοντέλο αναπαράσταση του υπολογισμού :

- Αναπαράσταση ως Boolean δίκτυο/κύκλωμα Συνδυαστικής Λογικής
- Αναπαράσταση ως Αριθμητικό δίκτυο/κύκλωμα
- Πιο σύνθετες αναπαραστάσεις, όπως ως Μηχανή Turing ή RAM (Random Access Machine)

Κατηγορίες ως προς τις ιδιότητες ασφάλειας

Κατηγορίες πρωτοκόλλων SMPC ως προς τις ιδιότητες ασφάλειας :

- Πρωτόκολλα ασφαλή ενάντια σε Παθητικούς Αντιπάλους
  - Ασφαλή ενάντια σε διεφθαρμένη μειοψηφία
  - Ασφαλή ενάντια σε διεφθαρμένη πλειοψηφία
- Πρωτόκολλα ασφαλή ενάντια σε Ενεργητικούς Αντιπάλους
  - Ασφαλή ενάντια σε διεφθαρμένη μειοψηφία
  - Ασφαλή ενάντια σε διεφθαρμένη πλειοψηφία

Παραδείγματα πρωτοκόλλων

	Υπολογιστικό Μοντέλο / Μοντέλο Ασφάλειας	Αριθμητικό δίκτυο		Boolean κύκλωμα	
		$modp$ ή $GF(2^n)$	$mod2^n$	Δυαδική Διαμοίραση Μυστικών	Μπερδεμένα Δίκτυα
Ενεργητικός	Διεφθαρμένη Πλειοψηφία	MASCOT / LowGear / HighGear	SPDZ2k	Tiny / Tinier	BMR
	Εμπιστή Πλειοψηφία	Shamir / Rep3 / PS / SY	Brain / Rep3 / PS / SY	Rep3 / CCD / PS	BMR
	Έμπιστη Υπερ-πλειοψηφία	Rep4	Rep4	Rep4	N/A
Συγκαλλημένος	Διεφθαρμένη Πλειοψηφία	CowGear / ChaiGear	N/A	N/A	N/A
Παθητικός	Διεφθαρμένη Πλειοψηφία	Semi / Hemi / Temi / Soho	Semi2k	SemiBin	Yao's GC / BMR
	Έμπιστη Πλειοψηφία	Shamir / ATLAS / Rep3	Rep3	Rep3 / CCD	BMR
	Dealer	Dealer	Dealer	Dealer	N/A

Πίνακας: Παράδειγμα υποστηριζόμενων πρωτοκόλλων της βιβλιοθήκης MP-SPDZ. Ο παρόν είναι εμπνευσμένος από έναν παρόμοιο, αλλά δυσανάγνωστο, πίνακα που υπάρχει στα εγχειρίδια χρήσης της.

Πρωτόκολλο GMW

- Προτάθηκε από τους Goldreich, Micali, Wigderson το 1987
- Βασίζεται σε Δυαδική Διαμοίραση Μετοχών
- Είναι ασφαλές ενάντια σε παθητικούς αντιπάλους
- Μπορεί να χρησιμοποιηθεί για $n \geq 2$ συμμετέχοντες
- Μπορεί να εκτελέσει Boolean και Αριθμητικά κυκλώματα
- Οι Boolean πύλες NOT και XOR μπορούν να αποτιμηθούν, μη διαδραστικά
- Η Boolean πύλη AND απαιτεί διάδραση για την αποτίμηση της

Πρωτόκολλο GMW

Θα εξετάσουμε την περίπτωση δύο συμμετεχόντων, P_1 και P_2, και μιας Boolean πύλης με δύο bit εισόδου και ένα bit εξόδου :

- **Πύλη εισόδου** : Έστω πως ο συμμετέχον P_1 διαθέτει ένα ιδιωτικό bit εισόδου x_i το οποίο αντιστοιχεί σε ένα καλώδιο w_i του κυκλώματος. Έστω επίσης s_i^j η μετοχή για το καλώδιο w_i του παίχτη j. Για κάθε καλώδιο i επιθυμούμε $s_i^j \oplus s_i^{j-1} = w_i$. Ο P_1 δημιουργεί τις μετοχές ως εξής :
  - Στέλνει στον P_2 το $r_i \leftarrow\$ \{0,1\}$, η οποία είναι η μετοχή του P_2 για το καλώδιο w_i, $s_i^2 = r_i$
  - Δημιουργεί τη δική του μετοχή του w_i ως εξής : $s_i^1 = x_i \oplus r_i$

Πρωτόκολλο GMW

- **Πύλη NOT** : Έστω το καλώδιο εισόδου w_k και το καλώδιο εξόδου w_{k+1}. Στην περίπτωση της Πύλης NOT, ένας από τους δύο συμμετέχοντες (όχι και οι δύο), έστω ο P_1, το μόνο που έχει να κάνει είναι να αντιστρέψει το bit της μετοχής του, δηλαδή $s^1_{k+1} = 1 - s^1_k$
- **Πύλη XOR** : Έστω τα καλώδια εισόδου w_k, w_{k+1} και το καλώδιο εξόδου w_{k+2}. Στην περίπτωση της Πύλης XOR, οι δύο συμμετέχοντες κάνουν XOR τις μετοχές που έχουν για κάθε μία από τις δύο εισόδους της πύλης. Δηλαδή $s^1_{k+2} = s^1_k \oplus s^1_{k+1}$ και $s^2_{k+2} = s^2_k \oplus s^2_{k+1}$

Πρωτόκολλο GMW

- **Πύλη AND**: Έστω τα καλώδια εισόδου w_k, w_{k+1} και το καλώδιο εξόδου w_{k+2}. Στην περίπτωση της Πύλης AND, ο ένας από τους δύο συμμετέχοντες, έστω ο P_1 πρέπει να ετοιμάσει ένα 1-4 OT της παρακάτω συνάρτησης :

$$S = S_{s_i^1, s_j^1}\left(s_i^2, s_j^2\right) = \left(s_i^1 \oplus s_i^2\right) \wedge \left(s_j^1 \oplus s_j^2\right)$$

Αυτό το κάνει επιλέγοντας $r \leftarrow\!\!\$\ \{0,1\}$ και στη συνέχεια δημιουργώντας τον παρακάτω πίνακα από τον οποία στέλνει την κατάλληλη γραμμή μέσω 1/4 OT ανάλογα με το τι τιμή έχουν οι μετοχές του P_2 :

$$T_G = \begin{pmatrix} r \oplus S(0,0) \\ r \oplus S(0,1) \\ r \oplus S(1,0) \\ r \oplus S(1,1) \end{pmatrix}$$

BLAS (Basic Linear Algebra Subroutines)

Βασικοί ορισμοί

BLAS : Είναι μια διεπαφή/πρότυπο συναρτήσεων και ρουτινών γραμμικής που επιτρέπει σε έναν κατασκευαστή υλικού να δημιουργήσει μια δική του υλοποίηση του προτύπου είναι παραμετροποιημένη για το συγκεκριμένο υλικό. Χωρίζεται σε τρία επίπεδα.

- Level-1 : Πράξεις μεταξύ διανυσμάτων
- Level-2 : Πράξεις πινάκων με διανύσματα
- Level-3 : Πράξεις μεταξύ πινάκων

Η διεπαφή της BLAS υπάρχει για δύο γλώσσες προγραμματισμού :

- FORTAN και αποκαλείται BLAS
- **C/C++ και αποκαλείται CBLAS**

Η BLAS διεπαφή χρησιμοποιείται σχεδόν από κάθε πρόγραμμα που κάνει πράξεις γραμμικής άλγεβρας, ενδεικτικά αναφέρουμε τα εξής :

- Βιβλιοθήκες/Προγράμματα επιστημονικού υπολογισμού όπως η NumPy, η Sci-Py και το MATLAB
- Βιβλιοθήκες μηχανικής μάθησης όπως η Tensorflow και η PyTorch

Παραδείγματα προτύπου CBLAS

```
1 float   cblas_sdot(const CBLAS_INT N,
2                    const float   *X, const CBLAS_INT incX,
3                    const float   *Y, const CBLAS_INT incY);
```

```
1 void  cblas_saxpy(const CBLAS_INT N,
2                    const float alpha,
3                    const float *X, const CBLAS_INT incX,
4                    float *Y, const CBLAS_INT incY);
```

Πρόβλημα

Περιγραφή προβλήματος

- Οι BLAS βιβλιοθήκες χρησιμοποιούνται σχεδόν παντού, από τον Επιστημονικό Υπολογισμό μέχρι την Μηχανική Μάθηση κσι σε τομείς που η ιδιωτικότητα των επεξεργαζόμενων δεδομένων είναι ιδιαίτερα επιθυμητή
- Ωστόσο, δεν υπάρχει στη βιβλιογραφία, τουλάχιστον στη δική μας γνώση, κάποια βιβλιοθήκη που να επιτρέπει τη χρήση BLAS συναρτήσεων για Ασφαλή Υπολογισμό Πολλών Μερών και να μπορεί να λειτουργήσει ως σχεδόν drop-in αντικατάστατο μιας κανονικής βιβλιοθήκης

Επίλυση προβλήματος

Επίλυση προβλήματος

Εφαρμογή του SMPC πρωτοκόλλου GMW στην κατασκευή της βιβλιοθήκης MPC-BLAS, μιας Ασφαλούς Υπολογισμού Δύο Μερών BLAS Level-1 βιβλιοθήκης, η οποία κατά τη γνώση μας, παρουσιάζεται πρώτη φορά στη βιβλιογραφία.

Η βιβλιοθήκη MPC-BLAS

Περιγραφή :

- Ένα πρόγραμμα που είναι χρησιμοποιεί τη διεπαφή CBLAS απαιτεί ελάχιστες αλλαγές στον πηγαίο του κώδικα για να χρησιμοποιήσει την διεπαφή MPC-BLAS
- Απαιτεί δύο διεργασίες, μία για κάθε συμμετέχοντα οι οποίες μπορούν να τρέχουν στο ίδιο μηχάνημα ή σε διαφορετικό
- Υποστηρίζει μόνο το λειτουργικό σύστημα Linux, διότι είναι το μοναδικό που υποστηρίζεται από την βιβλιοθήκη ABY. Έχει αναπτυχθεί και ελεγχθεί σε Ubuntu 22.04 LTS
- Υλοποιεί όλες τις Level-1 BLAS ρουτίνες εκτός από τις xROTx, διότι η βιβλιοθήκη ABY έχει πολύ περιορισμένη υποστήριξη για πράξεις πινάκων

Η βιβλιοθήκη MPC-BLAS

Τεχνικές Προδιαγραφές :

- Γραμμένη σε C++20 και χτισμένη με CMake
- Βασισμένη στην βιβλιοθήκη ABY για τις κρυπτογραφικές πράξεις και την εκτέλεση του πρωτοκόλλου GMW
- Απαιτεί στατική σύνδεση (static linking), λόγω περιορισμών της βιβλιοθήκης ABY

Προσαρμογή προγράμματος από CBLAS σε MPC-BLAS 1

```
1 float result_1 =
2     cblas_sdot(4,
3                test_vector_1,1,
4                test_vector_2,1);
```

Προσαρμογή προγράμματος από CBLAS σε MPC-BLAS 1

```
1  mpcblas_initialize(SERVER,
2                     "127.0.0.1",
3                     7766,
4                     bitlen);
5
6  float result_1 =
7      mpcblas_sdot(4,
8                   test_vector_1,1,
9                   MPCBLAS_IGNORE,1)
10               ->get_value();
11
12 mpcblas_uninitialize();
```

```
1  mpcblas_initialize(SERVER,
2                     "127.0.0.1",
3                     7766,
4                     bitlen);
5
6  float result_1 =
7      mpcblas_sdot(4,
8                   MPCBLAS_IGNORE,1,
9                   test_vector_2,1)
10               ->get_value();
11
12 mpcblas_uninitialize();
```

Προσαρμογή προγράμματος από CBLAS σε MPC-BLAS 2

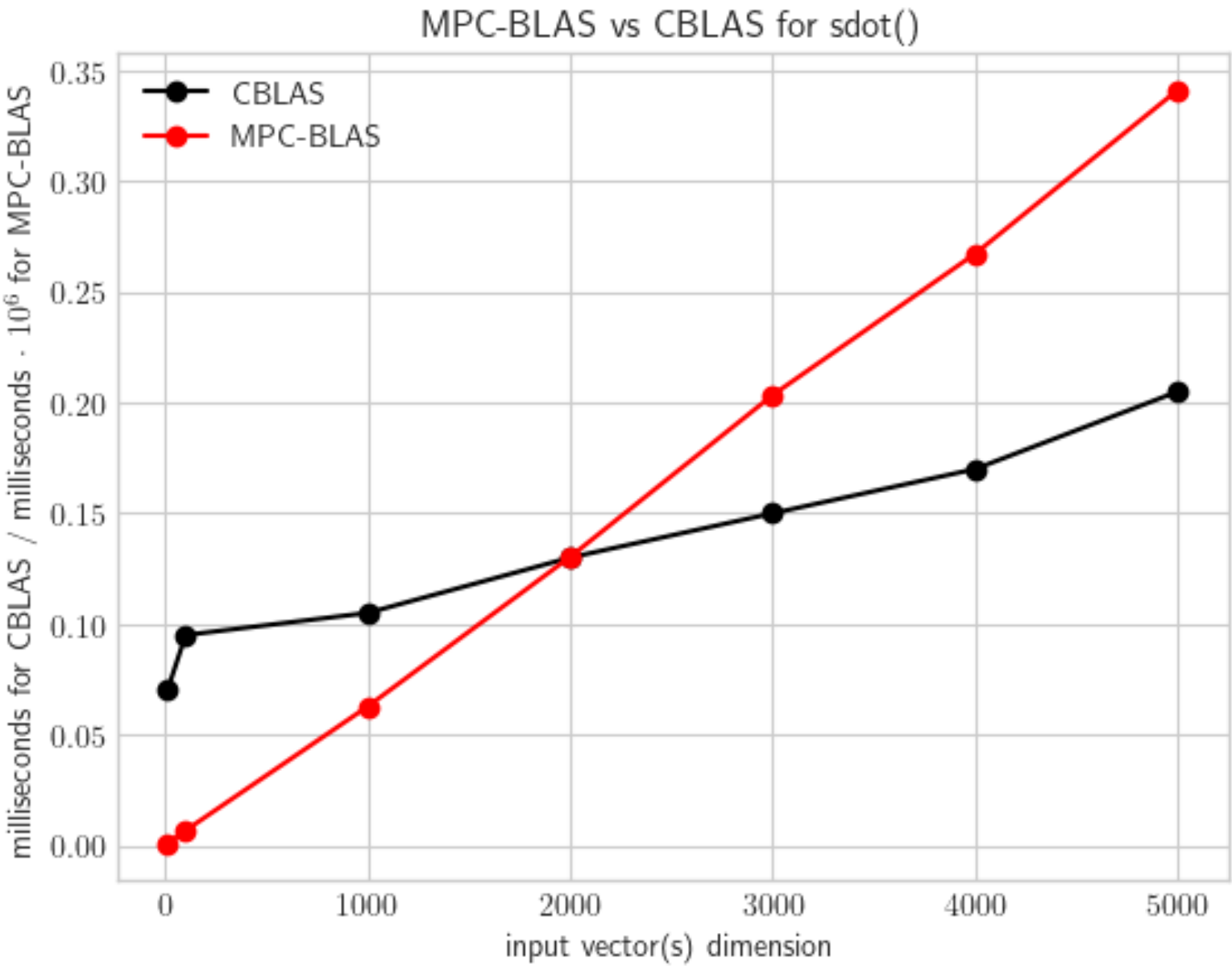
```
1 cblas_saxpy(4,
2     cblas_sdot(4,
3                test_vector_1,1,
4                test_vector_2,1),
5            test_vector_1,1,
6            test_vector_2,1);
7 float result =
8     cblas_snrm2(4,
9                test_vector_2,1);
```

Προσαρμογή προγράμματος από CBLAS σε MPC-BLAS 2

```
mpcblas_initialize(CLIENT,
                   "127.0.0.1",
                    7766,
                    bitlen);
auto result y =
mpcblas_saxpy(4,
    mpcblas_sdot(4,
                 test_vector_1,1,
                 MPCBLAS_IGNORE,1),
               test_vector_1,1,
               MPCBLAS_IGNORE,1);
float result_2 =
    mpcblas_snrm2(4, y, 1)
            ->get_value();
mpcblas_uninitialize();
```

```
mpcblas_initialize(CLIENT,
                   "127.0.0.1",
                    7766,
                    bitlen);
auto result y =
mpcblas_saxpy(4,
    mpcblas_sdot(4,
                 MPCBLAS_IGNORE,1,
                 test_vector_2,1),
               MPCBLAS_IGNORE,1,
               test_vector_2,1);
float result_2 =
    mpcblas_snrm2(4, y, 1)
            ->get_value();
mpcblas_uninitialize();
```

Πείραμα


MPC-BLAS vs CBLAS for sdot()
CBLAS
MPC-BLAS
milliseconds for CBLAS / milliseconds · 10^6 for MPC-BLAS
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1000
2000
3000
4000
5000
input vector(s) dimension

Ερμηνεία Πειράματος

- Η επιβάρυνση που εισάγεται από τη χρήση της MPC-BLAS σε σύγκριση με την CBLAS είναι γραμμική και είναι της τάξης των 10.000-100.000 χρονικών μονάδων
- Η διαφορά των χρονικών ασυμπτωτικών πολυπλοκοτήτων των δύο βιβλιοθηκών είναι σταθερή τιμή
- Υπάρχει τεράστιο περιθώριο βελτίωσης, ίσως και κατά αρκετές τάξεις μεγέθους

Μελλοντική Μελέτη

- Υλοποίηση των BLAS-2 και BLAS-3 υπορουτινών και ολοκλήρωση μιας εύχρηστης βιβλιοθήκης
- Υλοποίηση αποδοτικότερου αλγορίθμου Κινητής Υποδιαστολής (χρήση αριθμητικών δικτύων αντί για Boolean)
- Επέκταση της ABY για SMPC ή χρήση βιβλιοθήκης για υποστήριξη SMPC
- Επέκταση της ABY ή χρήση βιβλιοθήκης με υποστήριξη πρωτοκόλλων ασφαλών ενάντια σε Ενεργητικούς Αντιπάλους

Βιβλιογραφία I

- Demmler, Daniel, Thomas Schneider και Michael Zohner (2015). ``ABY-A framework for efficient mixed-protocol secure two-party computation.''. Στο: *NDSS*.
- Evans, David, Vladimir Kolesnikov και Mike Rosulek (Δεκ. 2018). ``A Pragmatic Introduction to Secure Multi-Party Computation''. Στο: *Found. Trends Priv. Secur.* 2.2-3, σσ. 70–246. ISSN: 2474-1558. DOI: 10.1561/3300000019. URL: https://doi.org/10.1561/3300000019.
- Goldreich, Oded, Silvio Micali και Avi Wigderson (2019). ``How to play any mental game, or a completeness theorem for protocols with honest majority''. Στο: *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, σσ. 307–328.
- Lindell, Yehuda (2020). *Secure Multiparty Computation (MPC)*. Cryptology ePrint Archive, Paper 2020/300. https://eprint.iacr.org/2020/300. DOI: 10.1145/3387108. URL: https://eprint.iacr.org/2020/300.

Βιβλιογραφία II

 Rathee, Deevashwer κ.ά. (2022). *SecFloat: Accurate Floating-Point meets Secure 2-Party Computation*. Cryptology ePrint Archive, Paper 2022/322. https://eprint.iacr.org/2022/322. URL: https://eprint.iacr.org/2022/322.

Τέλος

Ευχαριστώ για τη προσοχή σας