קריפטוגרפיה מודרנית

פרויקט, חלק III ואחרון

תאריך הגשה: 15.8.2024

חלק זה של הפרויקט מתבסס על החלק הקודם.

הגשת הפרויקט נעשית על ידי תיאום מפגש איתי סמוך לתאריך היעד.

בחלק זה נוסיף נממש את מרכיב החישוב הפרטי לסימולציה שיישמתם, כאשר כהרגלנו, ב-log ההדפסות מתועדות הפעולות שנעשות. עד כה, שניים מהמשתתפים שיתפו את המידע הפרטי שלהם עם המשתתף השלישי, שעשה בעצמו את כל החישובים. אחרי מימוש שלב זה, כל שלושת המשתתפים משתתפים יחד בחישוב, תוך שמירה על פרטיות המידע שהיה ברשותם. המרכיבים העיקריים בחלק זה הם:

1. שיתוף סוד (secret sharing): במקום ששניים מהמשתתפים ישלחו את ביטי הקלט שברשותם למשתמש השלישי (באופן מאובטח), כל אחד משלושת המשתתפים יפצל כל אחד מהביטים שברשותו לשלושה חלקים על האלגוריתם הנלמד בכיתה, ישמור חלק אחד לעצמו ואת שני החלקים האחרים יעביר לשותפיו, אחד לכל אחד, כשהם מוצפנים. בסיום החישוב, כל אחד משלושת המשתתפים יקבל משותפיו (שוב, באופן מוצפן), את המידע הנדרש כדי להרכיב את הפלט שלו.
2. חישוב השערים הלוגיים באופן מאובטח (MPC): כל שער לוגי שעשיתם בו שימוש יוחלף בחישוב מאובטח, בו שלושת המשתתפים מחשבים את השער באופן מאובטח, כאשר הם אינם יודעים מה הכניסות לשערים (כל אחד מהם מחזיק share מתוך המידע הזה) וגם לא יודעים מה ערכה של היציאה של השער, אך שוב, כל אחד מהם מחזיק share של המידע. במינימום עליכם לממש חישוב מאובטח של שער NOT ושל שער AND, כאשר האחרון דורש מימוש גם של אלגוריתם Oblivious Transfer. אם השתמשתם גם בשער XOR, ממשו גרסה מאובטחת שלו. כל שער אחר אמור להיות ממומש כהרכבה של שערים פשוטים יותר ולכן להיות מאובטח באופן אוטומטי.
3. ההדפסות ל-log צריכות להיות מפורטות, מצד אחד, אך במבנה קריא, מצד שני. עשו שימוש בכל אמצעי ויזואלי (כולל אמצעים פשוטים כמו הזחה/אינדנטציה, או שתילת שורת רווח בין חלקים שונים בלוגיקה) כדי לגרום ל-log לענות על שתי דרישות אלו.

הנחיות מפורטות יותר:

(אם נדרש:) תיקון מימוש החישוב בשערים לוגיים

חלק הקוד שמבצע את החישוב שבמרכזה של העבודה אמור להיות מבוסס על הפעלה של שערים לוגיים בלבד. ניתן להשתמש בשער NOT, ובשערים AND ו-XOR עם שתי כניסות (אם אינכם זקוקים לשער XOR, מותר שלא יופיע בקוד). שערים אחרים יש לממש בעזרת שערי NOT, AND, XOR. בנוסף, אם מסיבה כלשהי אתם זקוקים לקבועים (הביטים 0 ו-1) ניתן להשתמש גם בהם. כמו כן, פעולה של שרשור ביטים (הצמדה של ביטים זה לזה לצרכי הדפסה או "שליחה" למשתמש אחר) ופיצול קבוצת ביטים לביטים בודדים (למשל, מתוך הודעה שהתקבלה ממשתמש אחר) מותרת, מכיוון שאינה דורשת התייחסות מפורשת לערכי הביטים. חשבו על החלק הזה של הקוד כמעגל מודפס שבו בצד אחד נכנסים ביטי הקלט של המשתמשים, בצד שני יוצאים ביטי הפלט שלהם ובאמצע יכולים להיות רק שערים לוגים, חיווט של הקלטים לכניסות של שערים וחיווט של היציאות של השערים לכניסות של שערים אחרים, או לפלט. כל דבר אחר הוא אסור וצריך להיות מתוקן.
להלן שתי שגיאות שהיו נפוצות בשלב הקודם של העבודה, ודרך אפשרית לתקנן:

1. שימוש בפקודת if. זכרו ש-if אינו שער לוגי, ובשלב הזה של הפרויקט אין דרך לדעת בזמן אמת האם התנאי של ה-if מתקיים או לא, ולכן בזמן אמת (בניגוד לקוד שמכיל פקודת if) לא ניתן לדעת מה יש לחשב. דרך אפשרית להיפטר מפקודת if מיותרת היא על ידי החלפתו בשער MUX. כניסת ה-0 לשער תהיה תוצאת החישוב שיש לבצע אם תנאי ה-if אינו מתקיים (כלומר, מה שבא אחרי ה-else);

כניסת ה-1 לשער תהיה תוצאת החישוב אם ה-if כן מתקיים; תנאי ה-if יכנס לכניסת הבורר של שער ה-MUX.

2. שימוש באינדקס כדי לגשת למערך. ניתן לחשוב על שימוש באינדקס למערך כסדרה של פעולות if: אם קבוצת ביטים מסוימת שווה ל-0, התוצאה היא הערך שנמצא באינדקס 0 של המערך; אם קבוצת הביטים היא המספר הבינארי 1, התוצאה היא הערך שנמצא באינדקס 1 של המערך, וכן הלאה. על ידי שימוש במספר שערי MUX (או בשערים גדולים יותר, כמו 4-1 MUX), ניתן להחליף פניה למערך על פי אינדקס (פעולה אסורה) בסדרת שערים לוגיים.

שיתוף סוד (secret sharing)

כשלב מקדמי, כל אחד משלושת המשתתפים מפצל את ביטי הקלט שלו לשלושה חלקים, לפי האלגוריתם ל-secret sharing עם n=k=3. כלומר, כל ביט בקלט מוחלף בשלושה ביטים, שניים מהם אקראיים והשלישי מחושב כך ש-XOR של שלושת הביטים ייתן חזרה את הביט המקורי.

כל משתתף, עבור כל ביט קלט, שומר לעצמו אחד משלושת החלקים אליו פורק הביט הזה ומעביר לשני המשתתפים האחרים את שני החלקים הנוספים של הביט, אחד לכל אחד. העברת הביטים תעשה באופן מאובטח, על ידי הצפנת ההודעה עם Randomized RSA שכבר מימשתם בחלק הקודם של העבודה. מספר דגשים על כך:

- בכל פעם שעליכם להשתמש באקראיות, מותר להשתמש בפונקציית random מובנית של שפת התכנות שלכם. הדבר נכון גם במקומות אחרים בפרויקט בו יש להשתמש בערכים אקראיים.
- בשימוש ב-Randomized RSA, השולח מצפין את ההודעה בעזרת המפתח הציבורי של הנמען.

הנמען מפענח בעזרת המפתח הפרטי שלו. לכל אחד משלושת המשתתפים צריך להיות זוג מפתחות משלו, כאשר את המפתחות קבעתם מראש והם hard coded ואינם מתחלפים. אין צורך להפיץ את המפתחות הציבוריים וניתן להניח שכולם מכירים אותם.

- כפי שלמדנו, בהודעה שמועברת בעזרת Randomized RSA יש שני חלקים. הראשון הוא חזקה (מודולו N כלשהו) של מספר אקראי והשני הוא הצפנה של המידע עצמו. המידע עצמו יכול להיות מספר ביטים, החל מביט בודד ועד מחרוזת של ביטים בגודל של מחצית מהחלק הראשון של ההודעה. מותר ואך מומלץ להעביר בבת אחת מספר ביטים בהודעה אחת. מטעמי קריאות, אין חובה לשרשר פיזית את שני חלקי ההודעה, ועדיף להדפיס אותה ללוג כשני חלקים מופרדים.
- אם, למרות שאין בכך צורך, אתם משתמשים בקבוע כלשהו בחלק של החישוב המאובטח (הביט הקבוע 0 או הביט הקבוע 1), גם הוא צריך להיות secret shared בין המשתתפים. מכיוון שערכו אינו באמת פרטי, אפשר לחלק אותו לשלושה חלקים באופן נאיבי ו-hard coded ללא שימוש באקראיות. שימו לב שלא מדובר בביט קלט אלא רק בקבוע (לדוגמה, אם במקום שער NOT החלטתם לבצע XOR בין ביט כלשהו לבין הקבוע 1, תצטרכו "לחלק" גם את הקבוע 1 בין המשתתפים).

הרכבת סוד

בסיום החישוב, כל משתתף צריך להרכיב מחדש את הפלט שלו. מכיוון שהוא מחזיק רק ב-share מתוך כל ביט שהוא זקוק לו, שני המשתתפים האחרים ישלחו אליו, באופן מאובטח, את ה-shares שלהם של הביטים המתאימים. כל משתתף, בהינתן שלושת החלקים של הביטים שהוא זקוק להם, יוכל להרכיב את הפלט שלו ולהדפיס אותו ל-log. מספר דגשים:

- אם לכל משתתף יש פלט אחר, הוא יקבל הודעות שיאפשרו לו להרכיב רק את הביטים שהוא צריך.
- גם כאן, שיתוף ה-shares יעשה על ידי Randomized RSA. עשו בו שימוש אפילו אם למערכת יש רק פלט אחד משותף לכל שלושת המשתתפים (כלומר, הפלט אינו ערך פרטי כי כולם יודעים אותו ואין צורך לשמור על סודיות). במקרה זה, כולם ישלחו לכולם, באופן מוצפן, את כל ה-shares שלהם של ביטי פלט.

חישוב מאובטח של שערים לוגיים (MPC)

כל שער לוגי צריך להיות מחושב באופן מאובטח, כך שבסיום החישוב, הפלט יהיה מחולק לשלושה shares בין המשתתפים, אך החישוב נעשה מבלי להרכיב אותם חזרה. להזכירכם, חישוב מאובטח של שער NOT דורש רק מאחד מהמשתתפים (בחרו משתתף קבוע, למשל משתתף 1) לבצע פעולת NOT על השער שלו. חישוב מאובטח של שער XOR (אם יש בכך צורך) דורש מכל אחד משלושת המשתתפים לבצע, באופן מקומי, XOR על שני ה-shares שלו. רק חישוב מאובטח של שער AND הוא מורכב יותר:

- ממשו את אלגוריתם Oblivious Transfer 4-1, על פי שקפים 19-20 במצגת האחרונה. לתשומת לבכם, אלגוריתם ה-RSA המשמש לחישוב הפרמוטציה הוא RSA רגיל (פעולת חזקה ואחריה מודולו). השתמשו באותם מפתחות RSA של המשתמשים (כלומר, ניתן לדלג על השלב הראשון בו אחד מהצדדים בוחר פרמוטציה ו-trapdoor). לחלק הבא של האלגוריתם, בו יש להשתמש ב-hardcore predicate, השתמשו בביט ה-least (הביט הימני ביותר), בלבד. השתמשו באלגוריתם הזה בכל פעם להעברת ביט בודד של מידע (מוחבא בין רביעיית ביטים).
- ממשו שער AND מאובטח לשני משתמשים בעזרת האלגוריתם משקף 21, העושה שימוש ב-OT שמימשתם כרגע.
- על פי ההסבר והדוגמה בשקפים 22-25, ממשו שער AND מאובטח לשלושה משתתפים. הדבר דורש הרצה של האלגוריתם לשני משתתפים, שלוש פעמים (פעם אחת לכל זוג) ושילוב של התוצאות.

סנכרון

לתשומת לבכם, מכיוון שמדובר בסימולציה בה קוד מרכזי מדמה שלושה משתתפים, ולא במערכת מבוזרת אמיתי, לא אמורה להיות בעיית סנכרון. הקוד שלכם יקבע איזה משתתף פועל בכל רגע, כששני האחרים ממתינים לו.

יצירת log הודעות מפורט וקריא

ההדפסה ל-log צריכה להיות מצד אחד מפורטת, על לרמת הפעולה הבודדת, ומצד שני לאפשר "לראות את היער מבין העצים". השתמשו בכל אמצעי ויזואלי שמצאתם לנכון כדי למלא אחד שתי המטרות. לדוגמה, אם במסגרת חישוב ביט הפלט הראשון של משתמש 1, מחושב שער לוגי מורכב כלשהו (למשל MUX 2-1), ובמהלך החישוב שלו מתבצע שער AND בין שני ביטים, הדורש הרצה של אלגוריתם Oblivious Transfer (יותר מפעם אחת), כל המידע הזה צריך להופיע. עשו שימוש בריווח / הזחות / צבעים / הדפסה בחלונות שונים כדי להנגיש אותו כמה שיותר. מספר טיפים:

- זכרו שההודעות הנשלחות והמתקבלות אינן אמיתיות, אלא רק הדפסות של הודעות. לכן, בהודעה המורכבת מכמה חלקים (למשל, Randomized RSA כולל הודעה בת שני חלקים) אין צורך "להדביק" יחד את שני חלקי ההודעה למחרוזת ביטים לא מובנת, ואפשר להדפיס את שני החלקים עם סימן מפריד ביניהם
- זכרו שהמערכת שלכם היא רק סימולציה ואינה באמת מבוזרת אלא מנוהלת על ידי תהליך מרכזי, שיש לו גישה לכל ה-shares של כל שלושת המשתתפים, כל הזמן. לכן, ניתן וכדאי לכלול בתצוגת ה-log את כל פיסות המידע (ה-shares) של המשתתפים ואת הערכים שהן מרכיבות. במילים אחרות, המשתתפים אינם אמורים לחבר יחד את ה-shares שלהם לפני תום הסימולציה, אך בהדפסות ל-log מותר לעשות זאת כדי להבהיר שהערכים שהם מחזיקים אכן ירכיבו ביחד את התוצאות המצופות, בכל שלב של החישוב.

בהצלחה!