Κρυπτογραφικά Πρωτόκολλα

Παναγιώτης Γροντάς

ΕΜΠ - Κρυπτογραφία - (2017-2018)

05/12/2017

Περιεχόμενα

- Ασφαλής Υπολογισμός Πολλών Συμμετεχόντων
- Πρωτόκολλα
  - Πολλοί συμμετέχοντες
  - Πολλά μηνύματα
- Διαμοιρασμός Απορρήτων (secret sharing)
- Μη συνειδητή μεταφορά (oblivious transfer)

Το πρόβλημα

- m παίκτες θέλουν να υπολογίσουν από κοινού την τιμή της συνάρτησης $f(x_1, x_2, \cdots x_m)$
- Κάθε παίκτης P_i συνεισφέρει την είσοδο x_i
- Γενίκευση: κάθε παίκτης διαθέτει τη δική του συνάρτηση f_i, αλλά χρειάζεται είσοδο από όλους
- Μπορεί να γίνει;
  - Χωρίς να αποκαλυφθεί καμία πληροφορία εκτός από το αποτέλεσμα
  - Υποθέσεις ασφάλειας
  - Πολυπλοκότητα: Υπολογισμών / Επικοινωνίας
- Δεν είναι αποδεκτή η χρήση TTP

Διαμοιρασμός απορρήτων - Εισαγωγή

Βασικό συστατικό Secure Multi Party Computation

Το πρόβλημα

Κλειδιά: κρίσιμα κρυπτογραφικά δεδομένα (όχι τα μόνα)

Για παράδειγμα: ιδιωτικό κλειδί

- Δύναμη αποκρυπτογράφησης
- Δύναμη υπογραφής

Λύση

Δεν θέλουμε να είναι στην φυσική κατοχή μίας οντότητας (μόνο)

Additive secret sharing

Έστω $(\mathbb{G}, +)$ μια ομάδα και $s \in \mathbb{G}$ το μυστικό το οποίο θέλουμε να μοιράσουμε σε n παίκτες

- Διαλέγουμε τυχαία $s_1, \cdots s_{n-1} \in \mathbb{G}$
- Θέτουμε $s_n = s - \sum_{i=1}^{n-1} s_i$
- Μοιράζουμε τα $\{s_i\}_{i=1}^{n}$ στους παίκτες
- Ανακατασκευή $s = \sum_{i=1}^{n} s_i$

Παραλλαγή: Αν $s \in \{0,1\}^l$ τότε υλοποίηση με XOR
Πρόβλημα: Ένας παίκτης μπορεί να ακυρώσει την ανακατασκευή

Threshold Secret Sharing

(t, n) threshold secret sharing

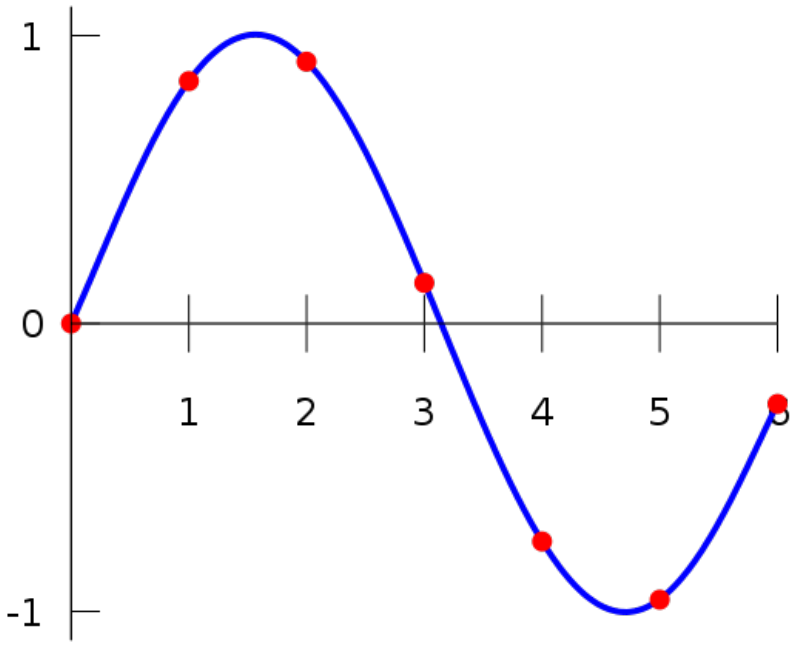
- Ένα μυστικό s πρέπει να μοιραστεί σε n παίκτες $P_1, P_2, \cdots P_n$ ώστε:
  - Οποιοδήποτε υποσύνολο από τουλάχιστον $t+1$ παίκτες να μπορεί να το ανακτήσει
  - Κανένα υποσύνολο με t παίκτες να μην μπορεί
- Υπόθεση Εμπιστεύομαστε τον διανομέα D και τους παίκτες

Shamir Secret Sharing I

Πολυωνυμική παρεμβολή

- Έστω ένα πολυώνυμο βαθμού t: $f(x) = a_0 + a_1 x + \cdots + a_t x^t$
- Μπορεί να ανακατασκευαστεί από $t+1$ σημεία $(x_i, f(x_i))$ με διαφορετικές τετμημένες (με μοναδικό τρόπο)
- Υπάρχουν άπειρα πολυώνυμα βαθμού t που περνούν από t τέτοια σημεία
- Ανάκτηση πολυωνύμου: συντελεστές Lagrange
- $\lambda_i(x) = \prod_{k=0, k\neq i}^{t} \frac{x - x_k}{x_i - x_k}$
- Προκύπτει το $L(x) = \sum_{i=0}^{t} y_i \lambda_i(x) = y_0\lambda_0(x) + y_1\lambda_1(x) + \cdots y_t\lambda_t(x)$
- Αποδεικνύεται ότι είναι μοναδικό δηλ: $L = f$

Shamir Secret Sharing II



Shamir Secret Sharing III

Εφαρμογή στο διαμοιρασμό απορρήτων

Υποθέτουμε ότι διαθέτουμε έναν έμπιστο διανομέα:

- Επιλέγει και δημοσιοποιεί ένα πρώτο p
- Επιλέγει t συντελεστές ενός πολυωνύμου βαθμού t $\{a_t, \cdots, a_1\} \in_R \mathbb{Z}_p$
- Θέτει ως σταθερό όρο το μυστικό s
- Προκύπτει το πολυώνυμο $f(x) = a_t \cdot x^t + a_{t-1} \cdot x^{t-1} + \cdots + a_1 \cdot x + s \pmod{p}$
- $f(0) = s$
- Μοιράζει στον παίκτη i την τιμή $(i, f(i))$

Shamir Secret Sharing IV

Ανακατασκευή

- Παρατήρηση: Δεν μας ενδιαφέρει να υπολογίσουμε το πολυώνυμο f αλλά το $f(0)$
- Κάθε παίκτης i υπολογίζει τους συντελεστές Lagrange
- $\lambda_i(0) = \prod_{k=1, k\neq i}^{t+1} \frac{-k}{i-k} \bmod p$
- $t+1$ παίκτες μπορούν να υπολογίσουν το $f(0)$ ως: $\sum_{i=1}^{t+1} f(i)\lambda_i(0) \bmod p$
- Ανακτούν το μυστικό υπολογίζοντας το $p(0)$

Παρατηρήσεις I

- Πληροφοριοθεωρητική ασφάλεια αν ο αντίπαλος διαθέτει λιγότερα μερίδια
- Μπορούν να προστεθούν εύκολα καινούρια μερίδια, χωρίς να αλλάξουν τα παλιά: Υπολογισμός νέων σημείων
- Εύκολη αντικατάσταση μεριδίων: Υπολογισμός νέων σημείων (πρέπει να γίνει ασφαλής καταστροφή των παλιών)
- Σημαντικοί παίκτες: περισσότερα από ένα μερίδια
- Αλλαγή Μεριδίων: Τροποποίηση πολυωνύμου χωρίς να αλλάξει το μυστικό
- Ομομορφικές ιδιότητες (άθροισμα πολυωνύμων είναι πολυώνυμο)
  $s_1 + s_2 = f(0) + g(0) = (f + g)(0)$

Παρατηρήσεις II

- Μειονεκτήματα: Εμπιστοσύνη
  - Κακόβουλος διανομέας: Λανθασμένα μερίδια σε τμήμα των παικτών
  - Κακόβουλος παίκτης: Παροχή λανθασμένων μεριδίων κατά τη διάρκεια της ανακατασκευής
- Λύση: Συνδυασμός με σχήμα δέσμευσης (Verifiable Secret Sharing)
  - Ο διανομέας μαζί με τα μερίδια παρέχει και δεσμεύσεις για τους συντελεστές
  - Οι παίκτες επαληθεύουν ότι οι δεσμεύσεις δίνουν το σημείο τους

Feldman Verifiable Secret Sharing

Υποθέσεις

- Ομάδα $\mathbb{G}$ τάξης q με γεννήτορα g με δύσκολο DLP
- Υπολογιστική Ασφάλεια
- Για απλότητα χρήση συνάρτησης σύνοψης $\mathcal{H}$ για δέσμευση
- Απαιτείται έντιμη πλειοψηφία (το πολύ t corrupted / τουλάχιστον $t+1$ honest)

Feldman Verifiable Secret Sharing

Φάση διαμοιρασμού μυστικού s

- Επιλογή $a_0 \in_R \mathbb{Z}_q$
- Διαμοιρασμός του a_0 με shamir secret sharing
  - Επιλογή $a_1, \cdots, a_t \in_R \mathbb{Z}_q$
  - Ορισμός $p(x) = \sum_{j=0}^{t} a_j \cdot x^j$
  - Αποστολή $s_i = p(i)$ στον P_i
- Δημοσιοποίηση: $\{A_j = g^{a_j}\}_{j=0}^{t}$ και
- $c = \mathcal{H}(a_0) \oplus s$

Feldman Verifiable Secret Sharing

Φάση επαλήθευσης s_i

- Κάθε παίκτης P_i υπολογίζει $c_i = \prod_{j=0}^{t} A_j^{i^j}$
- Αν ο διανομέας είναι έμπιστος ισχύει: $c_i = g^{s_i} = g^{p(i)}$
- Αν όχι τερματισμός πρωτοκόλλου από P_i
- Αν τερματίσουν πάνω από t χρήστες, επανάληψη πρωτοκόλλου με άλλον διανομέα
- Αλλιώς επανάληψη s_i

Ανακατασκευή s

- Συγκέντρωση τουλάχιστον $t+1$ μεριδίων - υπολογισμός a_0
- Υπολογισμός $s = \mathcal{H}(a_0) \oplus c$

Εφαρμογή: Threshold ElGamal I

- Δημιουργία Κλειδιών
  - Επιλογή δύο μεγάλων πρώτων p, q ώστε $q \mid (p-1)$
  - Επιλογή της υποομάδας τάξης q του $\mathbb{Z}_p^*$ και γεννήτορα g
  - Επιλογή τυχαίου $x \in \mathbb{Z}_q$
  - Κανονικός υπολογισμός δημοσίου κλειδιού $y = g^x \bmod p$
  - Χρήση σχήματος Shamir για διαμοιρασμό του ιδιωτικού $x \pmod q$
  - Αποτέλεσμα $\texttt{KeyGen}(1^\lambda) = (y, \{i, f(i)\}_{i=1}^n)$
- Κρυπτογράφηση
  - Κανονικά $\texttt{Encrypt}(y, m) = (G, M) = (g^r, m \cdot y^r)$

Εφαρμογή: Threshold ElGamal II

- Αποκρυπτογράφηση
  Σε δύο βήματα
  1. **'Αποκρυπτογράφηση' μεριδίων**
     - Κάθε παίκτης υπολογίζει και δημοσιοποιεί το $c_i = G^{f(i)} \bmod p$
  2. **Συνδυασμός**
     - Συγκεντρώνονται $t+1$ 'αποκρυπτογραφημένα' μερίδια (i, c_i) τα οποία συνδυάζονται ως:

$$C = \prod_i c_i^{\lambda_i(0)} = \prod_i G^{f(i)\lambda_i(0)} = G^{\sum_i f(i)\lambda_i(0))} = G^{f(0)} = G^x$$

     όπου λ_i οι συντελεστές Lagrange
     - Αποκρυπτογράφηση ως:

$$\frac{M}{C}$$

Παρατηρήσεις

- Υπολογιστική ασφάλεια ως προς τα c_i
- Ίδια κρυπτογράφηση
- Αποκρυπτογράφηση χωρίς ανακατασκευή του ιδιωτικού κλειδιού (δυνατότητα επαναχρησιμοποίησης)

Multi party computation from secret sharing

- Υπολογισμός οποιουδήποτε πολυωνύμου στο $\mathbb{F}_p$ (Ben-Or, Goldwasser, Wigderson)
- Ομομορφικές ιδιότητες πολυωνύμων
- Secret sharing όλων των τιμών εισόδου
- Για τέλεια ασφάλεια:
  - Honest majority για παθητικό αντίπαλο
  - Honest $> 2/3$ για ενεργό αντίπαλο
- Προβλήματα αποδοτικότητας
- Δεν μπορεί να εφαρμοστεί για 2 party computation λόγω honest majority

Two party computation

Το πρόβλημα των εκατομμυριούχων (Yao-1982)

- Δύο εκατομμυριούχοι (Alice, Bob) θέλουν να δουν ποιος είναι πιο πλούσιος
- Χωρίς να αποκαλυφθεί η περιουσία τους
- $f(a, b) =$ if $a < b$ then 1 else 0
- Υπόθεση: $1 \leq a, b \leq n$

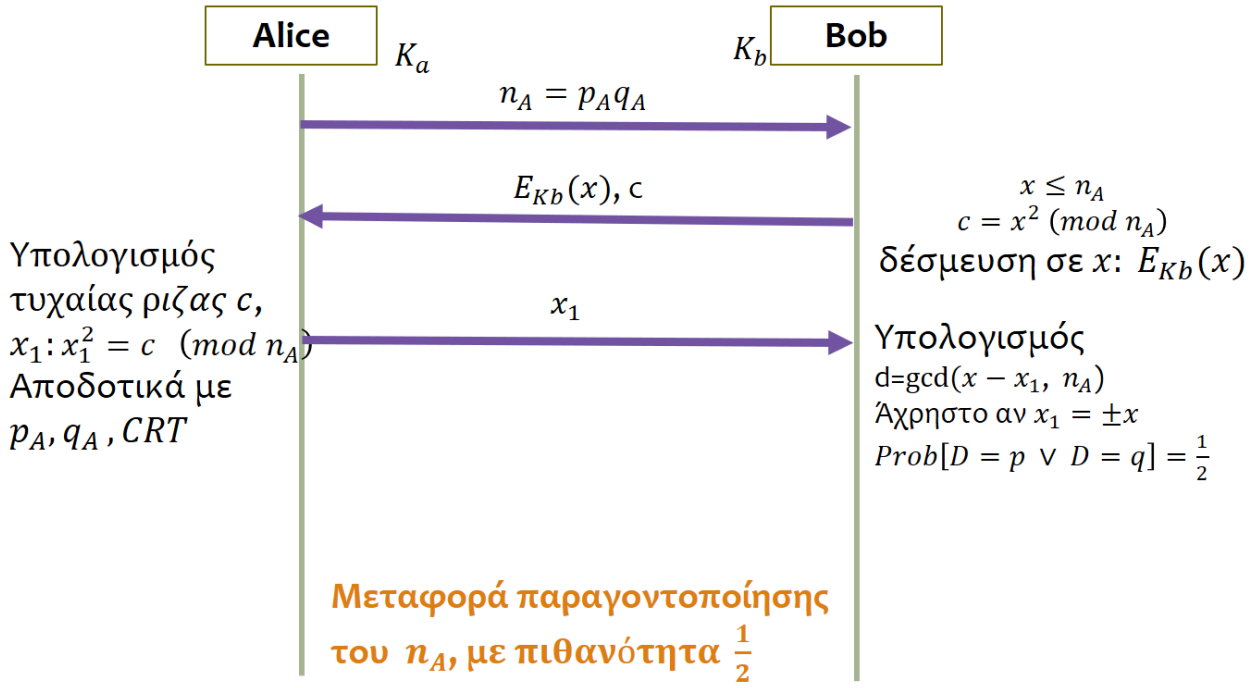
Το πρόβλημα των εκατομμυριούχων - Η λύση του Yao

- Ο Bob
  - Δημιουργεί n ταυτόσημα κουτιά (σχήμα δέσμευσης)
  - Διαλέγει έναν αριθμό x και τον τοποθετεί στο κουτί b
  - Στα υπόλοιπα τοποθετεί τυχαίους αριθμούς
- Η Alice
  - Ανοίγει όλες τις δεσμεύσεις
  - Αφήνει τα πρώτα a κουτιά ίδια
  - Προσθέτει 1 στα υπόλοιπα $n - a$
  - Τα στέλνει πίσω στον Bob
- Ο Bob
  - Ελέγχει τα κουτιά
  - Αν στο κουτί b υπάρχει το $x + 1$ είναι πλουσιότερος
  - Αλλιώς: Η Alice είναι
- Προβλήματα
  - Εκθετικό πλήθος δεσμεύσεων (ως προς τα bits της περιουσίας)
  - Ενεργοί αντίπαλοι (τερματισμός πριν την αποκάλυψη)

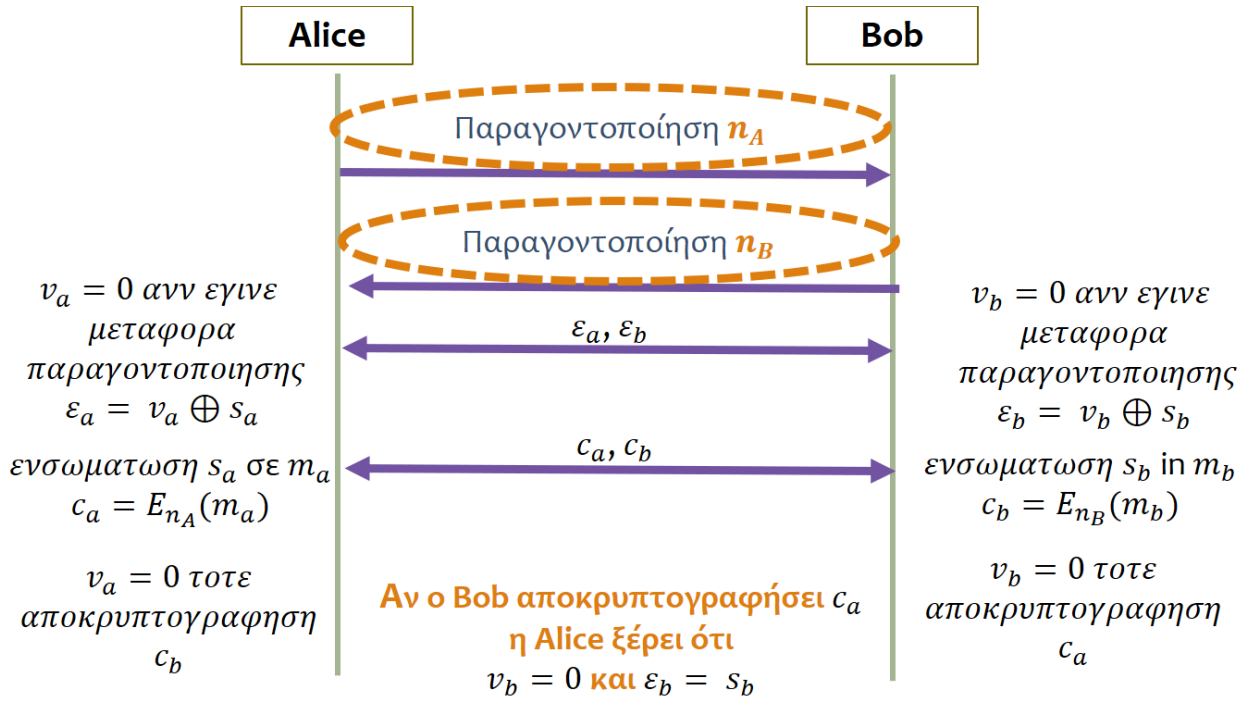
Γενίκευση: ανταλλαγή μυστικών

- Οι Alice, Bob θέλουν να ανταλλάξουν τα μυστικά s_a, s_b χωρίς TTP
- Ταυτόχρονη ανταλλαγή (ο ένας μαθαίνει αν ο άλλος έλαβε το μυστικό)
- Αποφυγή τερματισμού
- Πρόβλημα
  - $s_a = f(a_1, \cdots, a_n)$
  - $s_b = f(b_1, \cdots, b_n)$
  - $\exists k$: ώστε να μπορεί να υπολογιστεί το s_a, αλλά όχι το s_b

Η λύση του Rabin με τετραγωνικά υπόλοιπα I



Η λύση του Rabin με τετραγωνικά υπόλοιπα II



Γενίκευση: Μη συνειδητή μεταφορά (oblivious transfer)

Ορισμός $OT(S, R, M)$ (Even, Goldreich, Lempel)

Μη συνειδητή μεταφορά $OT(S, R, M)$ είναι ένα πρωτόκολλο με το οποίο ο αποστολέας S μεταφέρει ένα μηνυμα M στον παραλήπτη R έτσι ώστε ο R λαμβάνει το μήνυμα με πιθανότητα $1/2$ και:

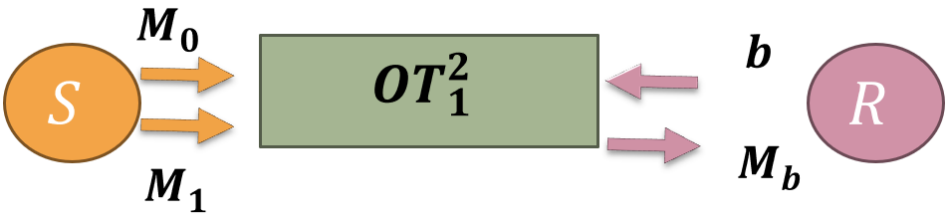
- Αν ο R δεν λάβει το μήνυμα, δεν μαθαίνει ούτε κάποια χρήσιμη πληροφορία
- Οποιαδήποτε προσπαθεια μη εκτέλεσης του πρωτοκόλλου γίνεται αντιληπτή

Αφαιρετική αναπαράσταση καναλιού με θόρυβο

Παραλλαγή: 1-από-2 Μη-Συνειδητή Μεταφορά

$OT_1^2(S, R, M_1, M_2)$

Ο R επιλέγει μεταξύ δύο μηνυμάτων για μεταφορά με πιθανότητα $1/2$ και ο S το μεταφέρει χωρίς ασφαλώς να γνωρίζει ποιο μετέφερε. Μπορούμε να προσομοιώσουμε την τυχαία επιλογή χρησιμοποιώντας ένα bit.



Άλλες παραλλαγές

$OT_1^n(S, R, M_1, \cdots, M_n)$

Ο R επιλέγει μεταξύ n μηνυμάτων να λάβει το i. Φυσικά ο S δεν το μαθαίνει, ενώ ο R δεν μαθαίνει τα $M_j, j \neq i$

k-από-n Μη-Συνειδητή Μεταφορά

- Ο R λαμβάνει ταυτόχρονα k μηνύματα
- Ο R λαμβάνει σειριακά k μηνύματα που μπορούν να τροποποιηθούν με βάση τα προηγούμενα (adaptive)

Πρακτική κατασκευή OT_1^2

- Χρήση κρυπτοσυστήματος δημοσίου κλειδιού με $\mathcal{M} = \mathcal{C}$
- Τυχαία επιλογή $x_0, x_1 \in \{0,1\}^*$
- Για να ληφθεί το M_0 ο R:
  - Στέλνει στον S το $(Enc(x_0), x_1)$
  - Ο S αποκρυπτογραφεί, παράγοντας το $(x_0, Dec(x_1))$.
  - Τελικά ο S αποστέλλει το $(M_0 \oplus x_0, M_1 \oplus Dec(x_1))$
  - Τελικά ο R ανακτά το M_0 με XOR του πρώτου συστατικού: $M_0 \oplus x_0 \oplus x_0$

Yao's Garbled Circuits

- Χρήση OT για κατασκευή κυκλώματος C που υπολογίζει ασφαλώς ως προς παθητικό αντίπαλο μια συνάρτηση f
- Οι παίκτες παρέχουν στο C τις εισόδους
- Μαθαίνουν το αποτέλεσμα χωρίς να αποκαλυφθεί οποιαδήποτε ενδιάμεση τιμή ή είσοδος

Βασική ιδέα

Κατασκευή αλλοιωμένων πινάκων τιμών για τις λογικές πύλες του κυκλώματος με χρήση OT

Παράδειγμα: Πύλη OR

- Υπολογισμός $x = s$ OR r
- Ο S παρέχει το s
- Ο R παρέχει το r

s	r	s OR r
0	0	0
0	1	1
1	0	1
1	1	1

Figure: Αρχικός πίνακας υπολογισμού OR

Παράδειγμα: Garbled OR

- Επιλογή δύο τυχαίων μεταθέσεων $v_s, v_r : \{0,1\} \to \{0,1\}$
- Εφαρμογή στον πίνακα
- Επιλογή 4 ζευγών συναρτήσεων κρυπτογράφησης και αποκρυπτογράφησης $(E_0^S, D_0^S), (E_1^S, D_1^S), (E_0^R, D_0^R), (E_1^R, D_1^R)$
- Εφαρμογή στο αποτέλεσμα της μετάθεσης
- Αποστολή στον R μαζί με τη v_r

s	r	s OR r
$v_s(0)$	$v_r(0)$	$E^S_{v_s(0)}(E^R_{v_r(0)}(0))$
$v_s(0)$	$v_r(1)$	$E^S_{v_s(0)}(E^R_{v_r(1)}(1))$
$v_s(1)$	$v_r(0)$	$E^S_{v_s(1)}(E^R_{v_r(0)}(1))$
$v_s(1)$	$v_r(1)$	$E^S_{v_s(1)}(E^R_{v_r(1)}(1))$

Figure: Αλλοιωμένος πίνακας υπολογισμού OR

Υπολογισμός με Garbled OR

- Ο S υπολογίζει το $v_s(s)$
- Στέλνει στον R το ζεύγος $(v_s(s), D^S_{v_s}(s))$
- Ο R υπολογίζει το $v_r(r)$
- Για να αποκρυπτογραφήσει χρειάζεται την συνάρτηση $D^R_{(v_r(r))}$
- Πρέπει να την πάρει από τον S χωρίς να αποκαλυφθεί το $v_r(r)$
- Χρήση $OT^2_1(S, R, D^R_0, D^R_1)$
- Τελικά ο R μπορεί να υπολογίσει το αποτέλεσμα $D^R_{v_r(r)}(D^S_{v_s(s)}(E^S_{v_s(s)}(E^R_{v_r(r)}(x))))$ και να το επιστρέψει στον S.

Γενίκευση

- Αλλοίωση όλων των πυλών
- Για κάθε πύλη
  - Μετάθεση γραμμών πίνακα αλήθειας → τυχαία μετάθεση αποτελέσματος
  - Θεώρουμε αποτέλεσμα και εισόδους ως τυχαία κλειδιά
  - Χρειάζονται 6 κλειδιά (4 είσοδοι - 2 αποτέλεσμα)
  - Υπολογισμός πύλης: γνώση κλειδιού αποτελέσματος
  - Τροφοδοσία επόμενης
- Οι τελικές έξοδοι αποκρυπτογραφούνται

Βιβλιογραφία I

1. St. Zachos and Aris Pagourtzis. Στοιχεία Θεωρίας Αριθμών και Εφαρμογές στην Κρυπτογραφία. Πανεπιστημιακές Σημειώσεις
2. Jonathan Katz and Yehuda Lindell. Introduction to Modern Cryptography 2nd edition, Chapman and Hall/CRC, 2015
3. Adi Shamir, How to share a secret. Communications of the ACM 22.11 (1979): 612-613.
4. Helger Lipmaa, 79.159 Cryptography and Data Security, 24.03.2004 Lecture 9: Secret Sharing, Threshold Cryptography, MPC
5. J. Kuhn The Mathematics of Secret Sharing
6. M. Ben-Or, S. Goldwasser and A. Wigderson, "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation," Proceedings of the 20th Annual ACM Symposium on Theory of Computing, Chicago, 1988, pp. 1-10.
7. Yao, A. C. "Protocols for secure computations" (FOCS 1982): 160–164
8. Rabin M. O. "How to exchange secrets by oblivious transfer." ,TR-81, Harvard University, 1981
9. S. Even, O. Goldreich, and A. Lempel. 1985. A randomized protocol for signing contracts. Commun. ACM 28, 6 (June 1985), 637-647
10. Claude Crépeau. 1987. Equivalence Between Two Flavours of Oblivious Transfers. In A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology (CRYPTO '87, UK, 350-354.
11. Yehuda Lindell and Benny Pinkas. 2009. A Proof of Security of Yao's Protocol for Two-Party Computation. J. Cryptol. 22, 2 (April 2009), 161-188 Ostrofski R., CS 282A/MATH 209A: Foundations of Cryptography, Lecture 10, Oblivious Transfer
12. Gabriel Bender, Cryptography and Secure Two-Party Computation, August 21, 2006
13. Ronald Cramer, Ivan Damgård, Jesper Buus Nielsen Multiparty Computation, an Introduction