Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Σημειώσεις Διαλέξεων

Στοιχεία Θεωρίας Αριθμών

&

Εφαρμογές στην Κρυπτογραφία

Διδάσκοντες:
Στάθης Ζάχος
Άρης Παγουρτζής

Επιμέλεια σημειώσεων:
Κωνσταντίνος Μάστακας
Στράτος Παλαιολόγος

Παρουσίαση:
Γιώργος Ζηρδέλης
Δημήτρης Σακαβάλας

30 Νοεμβρίου 2012 (v2)

1 Αναγωγή ElGamal σε DDH (και το αντίστροφο)

Α. Κρυπτοσύστημα ElGamal

Έστω p πρώτος, όπου ο $p-1$ έχει τουλάχιστον ένα μεγάλο πρώτο παράγοντα, και g γεννήτορας του $\mathbb{Z}_p^*$. Έστω ότι ο Bob θέλει να στείλει ένα μήνυμα στην Alice.

1. Η Alice διαλέγει ένα τυχαίο $a \in \{1, 2, \ldots, p-1\}$ (ιδιωτικό κλειδί) και υπολογίζει $A = g^a \pmod p$ (δημόσιο κλειδί).
2. Ο Bob διαλέγει $m \in \{1, 2, \ldots, p-1\}$ και τυχαίο $k \in \{2, 3, \ldots, p-1\}$.
3. (κρυπτογράφηση Bob) $r = g^k \pmod p$ και $c = mA^k \pmod p$ και στέλνει (r, c) (2-to-1 message expansion)
4. (απόκρυπτογράφηση Alice) $m = c \cdot r^{-a}$

Επειδή το r είναι τυχαίο η κρυπτογράφηση είναι πιθανοτική.

Β. Ανταλλαγή κλειδιού Diffie - Hellman

1. *Decisional DH (DDH)*
   Δίνονται πρώτος p και $g, g^x, g^y, g^z \in \mathbb{Z}_p^*$
   Ζητείται η απάντηση στο αν ισχύει η ισότητα: $g^z = g^{xy} \pmod p$ (YES ή NO)
   Πρόκειται για πρόβλημα απόφασης!
2. *Computational DH (CDH)*
   Δίνονται πρώτος p και $g, g^x, g^y \in \mathbb{Z}_p^*$
   Ζητείται να βρεθεί το $g^z \in \mathbb{Z}_p^*$ τ.ω. $g^z = g^{xy} \pmod p$
   Πρόκειται για πρόβλημα υπολογισμού!

Ισχύει: $DDH \leq CDH \leq DLP$
Ανοιχτό πρόβλημα παραμένει αν ισχύει επίσης $DLP \leq CDH$.

Πρόταση 1: Μια μηχανή που αποφασίζει την εγκυρότητα κρυπτοκειμένων ElGamal μπορεί να χρησιμοποιηθεί για να αποφασίσουμε το Decisional DH. Ισχύει και το αντίστροφο. Θα δεχθούμε ότι οι μηχανές χρειάζονται πολυωνυμικό χρόνο στα bits του p (είναι η παράμετρος ασφάλειάς μας).

Απόδειξη. "$\Rightarrow$" Ευθύ:

Έχουμε μηχανή M_1 που με είσοδο $p, g, A, m, (r, c)$ (στιγμιότυπο ElGamal) δίνει YES αν $m = Dec_a(r, c)$ και NO διαφορετικά. Θέλουμε να δώσουμε είσοδο p, g, g^x, g^y, g^z στη M_1 ώστε να μας απαντήσει YES ή NO στο ερώτημα $g^{x \cdot y} = g^z$

$(\text{mod } p)$. Επομένως θα διαλέξω κατάλληλα τις παραμέτρους εισόδου στη M_1 ώστε το DDH να αναχθεί στον έλεγχο εγκυρότητας κρυπτοκειμένων ElGamal.

Η είσοδός μας λοιπόν γίνεται: $p, g, A = g^x, m = 1, (r = g^y, c = g^z)$ οπότε η M_1 θα απαντήσει στην έξοδο **αν**:

$m = c \cdot r^{-x} \Rightarrow 1 = g^z \cdot (g^y)^{-x} \Rightarrow g^{x \cdot y} = g^z \ (\text{mod } p) \Rightarrow$ άρα αποφασίζει το DDH.

"$\Leftarrow$" Αντίστροφο:

Έχουμε μηχανή M_2 που με είσοδο $p, g, g^x, g^y, g^z \in \mathbb{Z}_p^*$ (στιγμιότυπο DDH) δίνει YES αν $g^{x \cdot y} = g^z \ (\text{mod } p)$ και NO διαφορετικά. Θέλουμε να δώσουμε είσοδο $p, g, A, m, (r, c)$ στη M_2 ώστε να μας απαντήσει YES αν $m = Dec_a(r, c)$ ή NO διαφορετικά. Αντίστοιχα με το "ευθύ"($\Rightarrow$) θα διαλέξω μια κατάλληλη είσοδο. Επειδή το πρόβλημά μου είναι η απόφαση της εγκυρότητας κρυπτοκειμένων ElGamal και όχι η εύρεση του plaintext m, τότε μπορώ να βάλω στην είσοδο και το m. Έτσι βάζω είσοδο: $p, g, g^x = g^a, g^y = g^k, g^z = c \cdot m^{-1}$ οπότε η M_2 θα απαντήσει στην εξόδο **αν**:

$g^{x \cdot y} = g^z \Rightarrow g^{a \cdot k} = c \cdot m^{-1} \Rightarrow m = c \cdot g^{-a \cdot k} \Rightarrow m = c \cdot (g^k)^{-a} \Rightarrow m = c \cdot r^{-a} \Rightarrow$ αποφασίζει την εγκυρότητα κρυπτοκειμένων ElGamal. □

Η σχέση ενός κρυπτοσυστήματος με ένα δύσκολο πρόβλημα, μέσω αναγωγής, μας δίνει πληροφορία για την ασφάλεια του κρυπτοσυστήματος.

Σημείωση: Η Πρόταση 1 ισχύει και στην περίπτωση που οι M_1 και M_2 είναι probabilistic polynomial time turing machines. Παρόμοια μπορεί να γίνει αναγωγή και με τις υπολογιστικές εκδοχές των προβλημάτων ElGamal και DH.

2 Αλγόριθμοι επίλυσης DLP

1. Εξαντλητική μέθοδος αναζήτησης

   Υπολογίζω όλες τις δυνάμεις του γεννήτορα: $g, g^2, g^3, ...$ μέχρι να βρω εκείνο το i για το οποίο ισχύει $g^i = y$. Για τον υπολογισμό των δυνάμεων χρησιμοποιώ την αμέσως προηγούμενη δύναμη που υπολόγισα, δηλαδή $g^i = g \cdot g^{i-1}$.

   Προφανώς είναι ο πιο αργός αλγόριθμος και ο χρόνος που χρειάζεται είναι $O(p)$

2. Αλγόριθμοι ανταλλαγής χρόνου-μνήμης

   Αυτός ο αλγόριθμος χρησιμοποιεί χώρο στη μνήμη για να "εξοικονομήσει" χρόνο. Δηλαδή, υπολογίζω όλα τα πιθανά g^i και φτιάχνω μια λίστα (ή ένα πίνακα) με τα ζευγάρια $(i, g^i), i \in \langle 1, 2, \ldots, p-1 \rangle$ και όταν δίνεται κάποιο y, κάνω μια δυαδική αναζήτηση στη λίστα και βρίσκω το ζητούμενο i. Επομένως το πρόβλημα ανάγεται ουσιαστικά σε αναζήτηση σε μια λίστα που έχω όλα τα πιθανά αποτελέσματα.

Αυτός ο αλγόριθμος έχει πολυπλοκότητα χρόνο $O(\log(p)) = \tilde{O}(1)$ αλλά πολυπλοκότητα χώρου $O(p)$, που είναι σημαντικό κόστος.

Χρησιμοποιούμε εδώ έναν καινούργιο συμβολισμό: $\tilde{O}(t(n)) = O(t(n) \cdot poly(\log(t(n))))$

Αλγόριθμος του Shanks (Baby Step - Giant Step)

Μία βελτίωση του προηγουμένου είναι αυτός ο αλγόριθμος. Η ιδέα είναι να γράψω το $x = m \cdot j + i$, όπου $j = x \, div \, m$ και $i = x \, mod \, m$. Αν θεωρήσουμε $m = \lceil \sqrt{p-1} \rceil$ τότε $0 \leq i, j < m$. Τα βήματα του αλγορίθμου είναι:

- Παίρνω $m = \lceil \sqrt{p-1} \rceil$, δηλαδή το πάνω ακέραιο μέρος του $p-1$
- Υπολογίζω το $g^{m \cdot j} \pmod p, j \in \langle 0, 1, \ldots, m-1 \rangle$ Προκύπτει λίστα $L_1 = (j, g^{m \cdot j})$ την οποία ταξινομώ ως προς τα $g^{m \cdot j}$.
- Υπολογίζω τα $\beta \cdot g^{-i} \pmod p, i \in \langle 0, 1, \ldots, m-1 \rangle$. Προκύπτει λίστα $L_2 = (i, \cdot g^{-i})$.
- Αναζητώ $(j, y) \in L_1$ τ.ω. $(i, j) \in L_2$.
- $\log_g \beta = x = m \cdot j + i$ οπότε λύθηκε το DLP.

Απόδειξη ορθότητας του αλγορίθμου:

$(j, y) \in L_1$ και $(i, j) \in L_2 \Rightarrow \begin{cases} g^{m \cdot j} = y \pmod p \\ \beta \cdot g^{-i} = y \pmod p \end{cases} \Rightarrow$
$g^{m \cdot j} = \beta \cdot g^{-i} = y \pmod p \Rightarrow g^{m \cdot j + i} = \beta \pmod p \Rightarrow x = m \cdot j + i \pmod{p-1}$

Ο αλγόριθμος αυτός έχει πολυπλοκότητα χώρου $\tilde{O}(\sqrt{p})$ και πολυπλοκότητα χρόνου $\tilde{O}(\sqrt{p})$.

Βιβλιογραφία

Zac12 : Σημειώσεις Ζάχου, ΕΜΠ, 2012.

Sti06 : D. Stinson: Cryptography: Theory and Practice, 3rd edition, CRC Press, 2005.