

Υπολογιστική Θεωρία Αριθμών και Κρυπτογραφία

Κρυπτογραφία Δημοσίου Κλειδιού

Άρης Παγουρτζής – Στάθης Ζάχος

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Ορισμός RSA

- ▶ Παραγωγή κλειδιών

Ορισμός RSA

► Παραγωγή κλειδιών

1. Εύρεση πρώτων p, q μεγάλου μήκους (> 1000 ψηφία) – χρήση ελέγχου πρώτων αριθμών (π.χ. Miller-Rabin).

Ορισμός RSA

► Παραγωγή κλειδιών

1. Εύρεση πρώτων p, q μεγάλου μήκους (> 1000 ψηφία) – χρήση *ελέγχου πρώτων αριθμών* (π.χ. Miller-Rabin).
2. Υπολογισμός $n = p \cdot q$ και $\varphi(n) = (p - 1) \cdot (q - 1)$.

Ορισμός RSA

► Παραγωγή κλειδιών

1. Εύρεση πρώτων p, q μεγάλου μήκους (> 1000 ψηφία) – χρήση *ελέγχου πρώτων αριθμών* (π.χ. Miller-Rabin).
2. Υπολογισμός $n = p \cdot q$ και $\varphi(n) = (p - 1) \cdot (q - 1)$.
3. Επιλογή $e \in U(\mathbb{Z}_n) : \text{gcd}(e, \varphi(n)) = 1$.

Ορισμός RSA

► Παραγωγή κλειδιών

1. Εύρεση πρώτων p, q μεγάλου μήκους (> 1000 ψηφία) – χρήση *ελέγχου πρώτων αριθμών* (π.χ. Miller-Rabin).
2. Υπολογισμός $n = p \cdot q$ και $\varphi(n) = (p - 1) \cdot (q - 1)$.
3. Επιλογή $e \in U(\mathbb{Z}_n) : \gcd(e, \varphi(n)) = 1$.
4. Υπολογισμός $d : e \cdot d \equiv 1 \pmod{\varphi(n)}$ – χρήση *Επεκτεταμένου Ευκλείδειου αλγόριθμου*.

Ορισμός RSA

► Παραγωγή κλειδιών

1. Εύρεση πρώτων p, q μεγάλου μήκους (> 1000 ψηφία) – χρήση *ελέγχου πρώτων αριθμών* (π.χ. Miller-Rabin).
2. Υπολογισμός $n = p \cdot q$ και $\varphi(n) = (p - 1) \cdot (q - 1)$.
3. Επιλογή $e \in U(\mathbb{Z}_n) : \gcd(e, \varphi(n)) = 1$.
4. Υπολογισμός $d : e \cdot d \equiv 1 \pmod{\varphi(n)}$ – χρήση *Επεκτεταμένου Ευκλείδειου αλγόριθμου*.

Δημόσιο κλειδί: e, n .

Ορισμός RSA

► Παραγωγή κλειδιών

1. Εύρεση πρώτων p, q μεγάλου μήκους (> 1000 ψηφία) – χρήση *ελέγχου πρώτων αριθμών* (π.χ. Miller-Rabin).
2. Υπολογισμός $n = p \cdot q$ και $\varphi(n) = (p - 1) \cdot (q - 1)$.
3. Επιλογή $e \in U(\mathbb{Z}_n) : \gcd(e, \varphi(n)) = 1$.
4. Υπολογισμός $d : e \cdot d \equiv 1 \pmod{\varphi(n)}$ – χρήση *Επεκτεταμένου Ευκλείδειου αλγόριθμου*.

Δημόσιο κλειδί: e, n .

Ιδιωτικό κλειδί: d .

Ορισμός RSA

► Παραγωγή κλειδιών

1. Εύρεση πρώτων p, q μεγάλου μήκους (> 1000 ψηφία) – χρήση *ελέγχου πρώτων αριθμών* (π.χ. Miller-Rabin).
2. Υπολογισμός $n = p \cdot q$ και $\varphi(n) = (p - 1) \cdot (q - 1)$.
3. Επιλογή $e \in U(\mathbb{Z}_n) : \gcd(e, \varphi(n)) = 1$.
4. Υπολογισμός $d : e \cdot d \equiv 1 \pmod{\varphi(n)}$ – χρήση *Επεκτεταμένου Ευκλείδειου αλγόριθμου*.

Δημόσιο κλειδί: e, n .

Ιδιωτικό κλειδί: d .

► Κρυπτογράφηση

$$enc(m) = m^e \bmod n \quad (m \in \mathbb{Z}_n) .$$

Ορισμός RSA

► Παραγωγή κλειδιών

1. Εύρεση πρώτων p, q μεγάλου μήκους (> 1000 ψηφία) – χρήση *ελέγχου πρώτων αριθμών* (π.χ. Miller-Rabin).
2. Υπολογισμός $n = p \cdot q$ και $\varphi(n) = (p - 1) \cdot (q - 1)$.
3. Επιλογή $e \in U(\mathbb{Z}_n) : \gcd(e, \varphi(n)) = 1$.
4. Υπολογισμός $d : e \cdot d \equiv 1 \pmod{\varphi(n)}$ – χρήση *Επεκτεταμένου Ευκλείδειου αλγόριθμου*.

Δημόσιο κλειδί: e, n .

Ιδιωτικό κλειδί: d .

► Κρυπτογράφηση

$$\text{enc}(m) = m^e \bmod n \quad (m \in \mathbb{Z}_n) .$$

► Αποκρυπτογράφηση

$$\text{dec}(c) = c^d \bmod n .$$

► Ορθότητα

$$\text{dec}(m^e \bmod n) \equiv (m^e)^d \equiv m^{k \cdot \varphi(n) + 1} \equiv m \pmod{n}$$

► Ορθότητα

$$\text{dec}(m^e \bmod n) \equiv (m^e)^d \equiv m^{k \cdot \varphi(n) + 1} \equiv m \pmod{n}$$

Αποδεικνύεται εύκολα για $m \in U(\mathbb{Z}_n)$, αλλά ισχύει και για κάθε $m \in \mathbb{Z}_n \setminus U(\mathbb{Z}_n)$ (άσκηση).

- ▶ Ορθότητα

$$\text{dec}(m^e \bmod n) \equiv (m^e)^d \equiv m^{k \cdot \varphi(n) + 1} \equiv m \pmod{n}$$

Αποδεικνύεται εύκολα για $m \in U(\mathbb{Z}_n)$, αλλά ισχύει και για κάθε $m \in \mathbb{Z}_n \setminus U(\mathbb{Z}_n)$ (άσκηση).

- ▶ Και οι τρεις διαδικασίες (παραγωγή κλειδιών, κρυπτογράφηση, αποκρυπτογράφηση) υλοποιούνται αποδοτικά.

RSA και παραγοντοποίηση

Αν κάποιος μπορεί να βρει τα p και q μπορεί εύκολα να υπολογίσει το $\varphi(n)$ και επομένως και το d :

$$\text{RSA-decrypt}(c, e, n) \leq^P \text{FindSecrExp}(e, n) \leq^P$$

$$\varphi(n) - \text{Computation} \leq^P \text{Factoring}(n)$$

RSA και παραγοντοποίηση

Αν κάποιος μπορεί να βρει τα p και q μπορεί εύκολα να υπολογίσει το $\varphi(n)$ και επομένως και το d :

$$\text{RSA-decrypt}(c, e, n) \leq^P \text{FindSecrExp}(e, n) \leq^P$$

$$\varphi(n) - \text{Computation} \leq^P \text{Factoring}(n)$$

Η εύρεση του $\varphi(n)$ οδηγεί σε παραγοντοποίηση (άρα είναι παρόμοιας δυσκολίας) λύνοντας ως προς p, q :

$$n = p \cdot q$$

$$\varphi(n) = (p - 1) \cdot (q - 1)$$

RSA και παραγοντοποίηση

Αν κάποιος μπορεί να βρει τα p και q μπορεί εύκολα να υπολογίσει το $\varphi(n)$ και επομένως και το d :

$$\text{RSA-decrypt}(c, e, n) \leq^P \text{FindSecrExp}(e, n) \leq^P$$

$$\varphi(n) - \text{Computation} \leq^P \text{Factoring}(n)$$

Η εύρεση του $\varphi(n)$ οδηγεί σε παραγοντοποίηση (άρα είναι παρόμοιας δυσκολίας) λύνοντας ως προς p, q :

$$n = p \cdot q$$

$$\varphi(n) = (p - 1) \cdot (q - 1)$$

Επομένως:

$$\text{RSA-decrypt}(c, e, n) \leq^P \text{FindSecrExp}(e, n) \leq^P$$

$$\varphi(n) - \text{Computation} \equiv^P \text{Factoring}(n)$$

Εύρεση μυστικού εκθέτη $d \Rightarrow$ παραγοντοποίηση του n

Πρόταση

Η εύρεση του ιδιωτικού κλειδιού d (εκθέτη αποκρυπτογράφησης) του RSA, οδηγεί στην παραγοντοποίηση του n με πολύ μεγάλη πιθανότητα.

Απόδειξη.

Αν γνωρίζουμε το d μπορούμε να σχεδιάσουμε τον παρακάτω πιθανοτικό αλγόριθμο:

- ▶ Υπολογίζουμε $u = ed - 1$. Ισχύει $\varphi(n) \mid ed - 1 \Rightarrow \forall a \in U(\mathbb{Z}_n) : a^u \equiv 1 \pmod{n}$.
- ▶ Χρησιμοποιούμε την ίδια ιδέα που είδαμε στην απόδειξη του Miller-Rabin:
Γράφουμε $u = t \cdot 2^s$ και παίρνουμε sequences $\langle b^t, b^{2t}, \dots, b^{2^{i-1}t}, \dots, b^{2^s t} \rangle \pmod{n}$, για διάφορες τυχαία επιλεγμένες τιμές του b .

Όπως και στο Miller-Rabin, αποδεικνύεται ότι τουλάχιστον τα μισά $b \in U(\mathbb{Z}_n)$ δίνουν factoring sequences. □

Η σχέση των προβλημάτων ως τώρα

$$\text{RSA-decrypt}(c, e, n) \leq^P \text{FindSecrExp}(e, n) \leq^P$$

$$\varphi(n) - \text{Computation} \equiv^P \text{Factoring}(n) \leq^{rP} \text{FindSecrExp}(e, n)$$

Επίθεση Κοινού Γινομένου

Σενάριο: μια Έμπιστη Αρχή (*Trusted Third Party*) διανέμει το ίδιο γινόμενο n και διαφορετικά ζεύγη (e_1, d_1) και (e_2, d_2) σε δύο χρήστες. Οι πρώτοι αριθμοί p, q είναι γνωστοί μόνο στην Έμπιστη Αρχή. Τι πρόβλημα υπάρχει;

Επίθεση Κοινού Γινομένου

Σενάριο: μια Έμπιστη Αρχή (*Trusted Third Party*) διανέμει το ίδιο γινόμενο n και διαφορετικά ζεύγη (e_1, d_1) και (e_2, d_2) σε δύο χρήστες. Οι πρώτοι αριθμοί p, q είναι γνωστοί μόνο στην Έμπιστη Αρχή. Τι πρόβλημα υπάρχει;

(i) Ο χρήστης 1 μπορεί να υπολογίσει $(e_1 \cdot d_1 - 1)$ και να παραγοντοποιήσει το n με τον πιθανοτικό αλγόριθμο.

Επίθεση Κοινού Γινομένου

Σενάριο: μια Έμπιστη Αρχή (*Trusted Third Party*) διανέμει το ίδιο γινόμενο n και διαφορετικά ζεύγη (e_1, d_1) και (e_2, d_2) σε δύο χρήστες. Οι πρώτοι αριθμοί p, q είναι γνωστοί μόνο στην Έμπιστη Αρχή. Τι πρόβλημα υπάρχει;

(i) Ο χρήστης 1 μπορεί να υπολογίσει $(e_1 \cdot d_1 - 1)$ και να παραγοντοποιήσει το n με τον πιθανοτικό αλγόριθμο.

(ii) Μπορεί επίσης να υπολογίσει έναν εκθέτη αποκρυπτογράφησης χωρίς παραγοντοποίηση του n ως εξής:

Επίθεση Κοινού Γινομένου

Σενάριο: μια *Έμπιστη Αρχή (Trusted Third Party)* διανέμει το ίδιο γινόμενο n και διαφορετικά ζεύγη (e_1, d_1) και (e_2, d_2) σε δύο χρήστες. Οι πρώτοι αριθμοί p, q είναι γνωστοί μόνο στην Έμπιστη Αρχή. Τι πρόβλημα υπάρχει;

(i) Ο χρήστης 1 μπορεί να υπολογίσει $(e_1 \cdot d_1 - 1)$ και να παραγοντοποιήσει το n με τον πιθανοτικό αλγόριθμο.

(ii) Μπορεί επίσης να υπολογίσει έναν εκθέτη αποκρυπτογράφησης χωρίς παραγοντοποίηση του n ως εξής:

► Γνωρίζει ότι $g_0 = e_1 \cdot d_1 - 1 = k \cdot \varphi(n)$, για κάποιο $k \in \mathbb{N}$.

Επίθεση Κοινού Γινομένου

Σενάριο: μια Έμπιστη Αρχή (Trusted Third Party) διανέμει το ίδιο γινόμενο n και διαφορετικά ζεύγη (e_1, d_1) και (e_2, d_2) σε δύο χρήστες. Οι πρώτοι αριθμοί p, q είναι γνωστοί μόνο στην Έμπιστη Αρχή. Τι πρόβλημα υπάρχει;

(i) Ο χρήστης 1 μπορεί να υπολογίσει $(e_1 \cdot d_1 - 1)$ και να παραγοντοποιήσει το n με τον πιθανοτικό αλγόριθμο.

(ii) Μπορεί επίσης να υπολογίσει έναν εκθέτη αποκρυπτογράφησης χωρίς παραγοντοποίηση του n ως εξής:

- ▶ Γνωρίζει ότι $g_0 = e_1 \cdot d_1 - 1 = k \cdot \varphi(n)$, για κάποιο $k \in \mathbb{N}$.
- ▶ Από κατασκευή ισχύει $\gcd(e_2, \varphi(n)) = 1$.

Επίθεση Κοινού Γινομένου

Σενάριο: μια Έμπιστη Αρχή (*Trusted Third Party*) διανέμει το ίδιο γινόμενο n και διαφορετικά ζεύγη (e_1, d_1) και (e_2, d_2) σε δύο χρήστες. Οι πρώτοι αριθμοί p, q είναι γνωστοί μόνο στην Έμπιστη Αρχή. Τι πρόβλημα υπάρχει;

(i) Ο χρήστης 1 μπορεί να υπολογίσει $(e_1 \cdot d_1 - 1)$ και να παραγοντοποιήσει το n με τον πιθανοτικό αλγόριθμο.

(ii) Μπορεί επίσης να υπολογίσει έναν εκθέτη αποκρυπτογράφησης χωρίς παραγοντοποίηση του n ως εξής:

- ▶ Γνωρίζει ότι $g_0 = e_1 \cdot d_1 - 1 = k \cdot \varphi(n)$, για κάποιο $k \in \mathbb{N}$.
- ▶ Από κατασκευή ισχύει $\gcd(e_2, \varphi(n)) = 1$.
- ▶ Επομένως, διαιρώντας διαδοχικά το g_0 με τους κοινούς παράγοντές του με το e_2 βρίσκουμε $a = g_i = k' \cdot \varphi(n)$, $\gcd(e_2, a) = 1$.

Επίθεση Κοινού Γινομένου

Σενάριο: μια Έμπιστη Αρχή (Trusted Third Party) διανέμει το ίδιο γινόμενο n και διαφορετικά ζεύγη (e_1, d_1) και (e_2, d_2) σε δύο χρήστες. Οι πρώτοι αριθμοί p, q είναι γνωστοί μόνο στην Έμπιστη Αρχή. Τι πρόβλημα υπάρχει;

(i) Ο χρήστης 1 μπορεί να υπολογίσει $(e_1 \cdot d_1 - 1)$ και να παραγοντοποιήσει το n με τον πιθανοτικό αλγόριθμο.

(ii) Μπορεί επίσης να υπολογίσει έναν εκθέτη αποκρυπτογράφησης χωρίς παραγοντοποίηση του n ως εξής:

- ▶ Γνωρίζει ότι $g_0 = e_1 \cdot d_1 - 1 = k \cdot \varphi(n)$, για κάποιο $k \in \mathbb{N}$.
- ▶ Από κατασκευή ισχύει $\gcd(e_2, \varphi(n)) = 1$.
- ▶ Επομένως, διαιρώντας διαδοχικά το g_0 με τους κοινούς παράγοντές του με το e_2 βρίσκουμε $a = g_i = k' \cdot \varphi(n)$, $\gcd(e_2, a) = 1$.
- ▶ Το $d'_2 = (e_2)^{-1} \pmod{a}$ μπορεί να χρησιμοποιηθεί ως εκθέτης αποκρυπτογράφησης (γιατί;).

Μερική ανάκτηση πληροφοριών στο RSA

Σχετική έννοια: **Semantic Security**, το υπολογιστικό ανάλογο της Perfect Secrecy.

Μερική ανάκτηση πληροφοριών στο RSA

Σχετική έννοια: **Semantic Security**, το υπολογιστικό ανάλογο της Perfect Secrecy.

Ενδιαφέρει η ποσότητα πληροφορίας που μπορεί να διαρρεύσει σε εφικτό υπολογιστικό χρόνο.

Μερική ανάκτηση πληροφοριών στο RSA

Σχετική έννοια: **Semantic Security**, το υπολογιστικό ανάλογο της Perfect Secrecy.

Ενδιαφέρει η ποσότητα πληροφορίας που μπορεί να διαρρεύσει σε εφικτό υπολογιστικό χρόνο.

Διαρροή της τιμής του συμβόλου Jacobi

Έστω $c = m^e \bmod n$. Τότε:

$$\left(\frac{c}{n}\right) = \left(\frac{m^e}{p}\right) \cdot \left(\frac{m^e}{q}\right) = \left(\frac{m}{p}\right)^e \cdot \left(\frac{m}{q}\right)^e = \left(\frac{m}{p}\right) \cdot \left(\frac{m}{q}\right) = \left(\frac{m}{n}\right)$$

Μερική ανάκτηση πληροφοριών στο RSA

Σχετική έννοια: **Semantic Security**, το υπολογιστικό ανάλογο της Perfect Secrecy.

Ενδιαφέρει η ποσότητα πληροφορίας που μπορεί να διαρρεύσει σε εφικτό υπολογιστικό χρόνο.

Διαρροή της τιμής του συμβόλου Jacobi

Έστω $c = m^e \bmod n$. Τότε:

$$\left(\frac{c}{n}\right) = \left(\frac{m^e}{p}\right) \cdot \left(\frac{m^e}{q}\right) = \left(\frac{m}{p}\right)^e \cdot \left(\frac{m}{q}\right)^e = \left(\frac{m}{p}\right) \cdot \left(\frac{m}{q}\right) = \left(\frac{m}{n}\right)$$

Αυτή η διαρροή δεν θεωρείται απειλητική για την ασφάλεια του RSA.

Μερική ανάκτηση πληροφοριών στο RSA

Έστω $c = m^e \bmod n$.

Μερική ανάκτηση πληροφοριών στο RSA

Έστω $c = m^e \bmod n$.

Μπορούμε από τα (c, e, n) να μάθουμε το τελευταίο bit του m ;

Μερική ανάκτηση πληροφοριών στο RSA

Έστω $c = m^e \bmod n$.

Μπορούμε από τα (c, e, n) να μάθουμε το τελευταίο bit του m ;

Ή το 'bit' που μας λέει αν $m > \frac{n}{2}$;

Μερική ανάκτηση πληροφοριών στο RSA

Έστω $c = m^e \bmod n$.

Μπορούμε από τα (c, e, n) να μάθουμε το τελευταίο bit του m ;

Ή το 'bit' που μας λέει αν $m > \frac{n}{2}$;

Θα δούμε ότι κάθε μία από τις δύο αυτές πληροφορίες είναι ισοδύναμη με το σπάσιμο του κρυπτοσυστήματος.

Μερική ανάκτηση πληροφοριών στο RSA

Έστω $c = m^e \bmod n$.

Μπορούμε από τα (c, e, n) να μάθουμε το τελευταίο bit του m ;

Ή το 'bit' που μας λέει αν $m > \frac{n}{2}$;

Θα δούμε ότι κάθε μία από τις δύο αυτές πληροφορίες είναι ισοδύναμη με το σπάσιμο του κρυπτοσυστήματος.

$$\text{parity}_{n,e}(c) = \begin{cases} 0, & \text{αν } m \text{ είναι άρτιος} \\ 1, & \text{αν } m \text{ είναι περιττός} \end{cases}$$

$$\text{loc}_{n,e}(c) = \begin{cases} 0, & \text{αν } m \leq \frac{n}{2} \\ 1, & \text{αν } m > \frac{n}{2} \end{cases}$$

όπου m το μοναδικό $m \in \mathbb{Z}_n : m^e \bmod n = c$

Μερική ανάκτηση πληροφοριών στο RSA

Πρόταση

Αν μπορούμε να υπολογίσουμε οποιαδήποτε από τις συναρτήσεις loc ή $rarity$ (για όλες τις εισόδους) τότε μπορούμε να βρούμε το απλό κείμενο (σπάσιμο του RSA.)

Απόδειξη.

Στηρίζεται στην πολλαπλασιαστική ιδιότητα της κρυπτογράφησης RSA:

$$enc_{n,e}(m_1) \cdot enc_{n,e}(m_2) = enc_{n,e}(m_1 \cdot m_2)$$

Μερική ανάκτηση πληροφοριών στο RSA

Πρόταση

Αν μπορούμε να υπολογίσουμε οποιαδήποτε από τις συναρτήσεις loc ή $parity$ (για όλες τις εισόδους) τότε μπορούμε να βρούμε το απλό κείμενο (σπάσιμο του RSA.)

Απόδειξη.

Στηρίζεται στην πολλαπλασιαστική ιδιότητα της κρυπτογράφησης RSA:

$$enc_{n,e}(m_1) \cdot enc_{n,e}(m_2) = enc_{n,e}(m_1 \cdot m_2)$$

Παρατηρήστε ότι:

$$\blacktriangleright loc_{n,e}(c) = parity_{n,e}(enc_{n,e}(2 \cdot m)) = parity_{n,e}(c \cdot enc_{n,e}(2))$$

Μερική ανάκτηση πληροφοριών στο RSA

Πρόταση

Αν μπορούμε να υπολογίσουμε οποιαδήποτε από τις συναρτήσεις loc ή $parity$ (για όλες τις εισόδους) τότε μπορούμε να βρούμε το απλό κείμενο (σπάσιμο του RSA.)

Απόδειξη.

Στηρίζεται στην πολλαπλασιαστική ιδιότητα της κρυπτογράφησης RSA:

$$enc_{n,e}(m_1) \cdot enc_{n,e}(m_2) = enc_{n,e}(m_1 \cdot m_2)$$

Παρατηρήστε ότι:

- ▶ $loc_{n,e}(c) = parity_{n,e}(enc_{n,e}(2 \cdot m)) = parity_{n,e}(c \cdot enc_{n,e}(2))$
- ▶ $parity_{n,e}(c) = loc_{n,e}(enc_{n,e}(m \cdot 2^{-1} \pmod{n})) =$

Μερική ανάκτηση πληροφοριών στο RSA

Πρόταση

Αν μπορούμε να υπολογίσουμε οποιαδήποτε από τις συναρτήσεις loc ή $parity$ (για όλες τις εισόδους) τότε μπορούμε να βρούμε το απλό κείμενο (σπάσιμο του RSA.)

Απόδειξη.

Στηρίζεται στην πολλαπλασιαστική ιδιότητα της κρυπτογράφησης RSA:

$$enc_{n,e}(m_1) \cdot enc_{n,e}(m_2) = enc_{n,e}(m_1 \cdot m_2)$$

Παρατηρήστε ότι:

- ▶ $loc_{n,e}(c) = parity_{n,e}(enc_{n,e}(2 \cdot m)) = parity_{n,e}(c \cdot enc_{n,e}(2))$
- ▶ $parity_{n,e}(c) = loc_{n,e}(enc_{n,e}(m \cdot 2^{-1} \pmod{n})) = loc_{n,e}(c \cdot enc_{n,e}(\frac{n+1}{2}))$

Μερική ανάκτηση πληροφοριών στο RSA

Πρόταση

Αν μπορούμε να υπολογίσουμε οποιαδήποτε από τις συναρτήσεις loc ή $parity$ (για όλες τις εισόδους) τότε μπορούμε να βρούμε το απλό κείμενο (σπάσιμο του RSA.)

Απόδειξη.

Στηρίζεται στην πολλαπλασιαστική ιδιότητα της κρυπτογράφησης RSA:

$$enc_{n,e}(m_1) \cdot enc_{n,e}(m_2) = enc_{n,e}(m_1 \cdot m_2)$$

Παρατηρήστε ότι:

- ▶ $loc_{n,e}(c) = parity_{n,e}(enc_{n,e}(2 \cdot m)) = parity_{n,e}(c \cdot enc_{n,e}(2))$
- ▶ $parity_{n,e}(c) = loc_{n,e}(enc_{n,e}(m \cdot 2^{-1} \pmod{n})) = loc_{n,e}(c \cdot enc_{n,e}(\frac{n+1}{2}))$

Επομένως οι δύο συναρτήσεις είναι ισοδύναμες υπολογιστικά (ως προς πολωνυμικό χρόνο). □

Απόδειξη (συν.)

Μένει να εφαρμόσουμε δυαδική αναζήτηση, χρησιμοποιώντας την loc , για να βρούμε το m :

$$loc_{n,e}(enc(m)) = 0 \iff x \in [0, \frac{n}{2})$$

$$loc_{n,e}(enc(2m)) = 0 \iff x \in [0, \frac{n}{2}) \cup [\frac{n}{2}, \frac{3n}{4})$$

...

κ.ο.κ. για $\log n$ βήματα.

Απόδειξη (συν.)

Μένει να εφαρμόσουμε δυαδική αναζήτηση, χρησιμοποιώντας την loc , για να βρούμε το m :

$$loc_{n,e}(enc(m)) = 0 \iff x \in [0, \frac{n}{2})$$

$$loc_{n,e}(enc(2m)) = 0 \iff x \in [0, \frac{n}{2}) \cup [\frac{n}{2}, \frac{3n}{4})$$

...

κ.ο.κ. για $\log n$ βήματα.

Επομένως, αποδοτικός υπολογισμός της loc (ή της $parity$) οδηγεί σε αποκρυπτογράφηση. Συμπέρασμα;

Έστω τα δημόσια κλειδιά των Bob, Charlie και Diane $p_B = (n_1, 3)$, $p_C = (n_2, 3)$, $p_D = (n_3, 3)$, έχουν δηλαδή τον ίδιο μικρό εκθέτη. Η Alice στέλνει σε όλους το ίδιο μήνυμα m .

Επίθεση μικρού εκθέτη

Έστω τα δημόσια κλειδιά των Bob, Charlie και Diane $p_B = (n_1, 3)$, $p_C = (n_2, 3)$, $p_D = (n_3, 3)$, έχουν δηλαδή τον ίδιο μικρό εκθέτη. Η Alice στέλνει σε όλους το ίδιο μήνυμα m .

Η Eve σχηματίζει το σύστημα

$$c_1 = m^3 \pmod{n_1}$$

$$c_2 = m^3 \pmod{n_2}$$

$$c_3 = m^3 \pmod{n_3}$$

Επίθεση μικρού εκθέτη

Έστω τα δημόσια κλειδιά των Bob, Charlie και Diane $p_B = (n_1, 3)$, $p_C = (n_2, 3)$, $p_D = (n_3, 3)$, έχουν δηλαδή τον ίδιο μικρό εκθέτη. Η Alice στέλνει σε όλους το ίδιο μήνυμα m .

Η Eve σχηματίζει το σύστημα

$$c_1 = m^3 \pmod{n_1}$$

$$c_2 = m^3 \pmod{n_2}$$

$$c_3 = m^3 \pmod{n_3}$$

Ερώτηση: τι δίνει το σύστημα αυτό με χρήση CRT;

Επίθεση μικρού εκθέτη

Έστω τα δημόσια κλειδιά των Bob, Charlie και Diane $p_B = (n_1, 3)$, $p_C = (n_2, 3)$, $p_D = (n_3, 3)$, έχουν δηλαδή τον ίδιο μικρό εκθέτη. Η Alice στέλνει σε όλους το ίδιο μήνυμα m .

Η Eve σχηματίζει το σύστημα

$$c_1 = m^3 \pmod{n_1}$$

$$c_2 = m^3 \pmod{n_2}$$

$$c_3 = m^3 \pmod{n_3}$$

Ερώτηση: τι δίνει το σύστημα αυτό με χρήση CRT;

Απάντηση: την τιμή του m^3 στο $\mathbb{Z}_{n_1 n_2 n_3}$, δηλαδή το m^3 (γιατί;).

Συνιστώμενες παράμετροι του RSA

- ▶ $|n| \geq 2048$ (μέχρι το 2030 περίπου, μετά $|n| \geq 3072$).

Συνιστώμενες παράμετροι του RSA

- ▶ $|n| \geq 2048$ (μέχρι το 2030 περίπου, μετά $|n| \geq 3072$).
- ▶ p, q περίπου ίδιου μήκους.

Συνιστώμενες παράμετροι του RSA

- ▶ $|n| \geq 2048$ (μέχρι το 2030 περίπου, μετά $|n| \geq 3072$).
- ▶ p, q περίπου ίδιου μήκους.
- ▶ $p - q > 2^{\frac{|n|}{2} - 100}$.

Συνιστώμενες παράμετροι του RSA

- ▶ $|n| \geq 2048$ (μέχρι το 2030 περίπου, μετά $|n| \geq 3072$).
- ▶ p, q περίπου ίδιου μήκους.
- ▶ $p - q > 2^{\frac{|n|}{2} - 100}$.
- ▶ $p - 1, q - 1$ έχουν και ‘μεγάλους’ πρώτους παράγοντες (αποφυγή κυκλικών επιθέσεων).

Συνιστώμενες παράμετροι του RSA

- ▶ $|n| \geq 2048$ (μέχρι το 2030 περίπου, μετά $|n| \geq 3072$).
- ▶ p, q περίπου ίδιου μήκους.
- ▶ $p - q > 2^{\frac{|n|}{2} - 100}$.
- ▶ $p - 1, q - 1$ έχουν και ‘μεγάλους’ πρώτους παράγοντες (αποφυγή **κυκλικών επιθέσεων**).
- ▶ $66537 < e \leq 2^{256}$. Επιλέγεται πριν από τα p, q .

Συνιστώμενες παράμετροι του RSA

- ▶ $|n| \geq 2048$ (μέχρι το 2030 περίπου, μετά $|n| \geq 3072$).
- ▶ p, q περίπου ίδιου μήκους.
- ▶ $p - q > 2^{\frac{|n|}{2} - 100}$.
- ▶ $p - 1, q - 1$ έχουν και ‘μεγάλους’ πρώτους παράγοντες (αποφυγή **κυκλικών επιθέσεων**).
- ▶ $66537 < e \leq 2^{256}$. Επιλέγεται πριν από τα p, q .
- ▶ $ed \equiv 1 \pmod{\lambda(n) = \text{lcm}(p - 1, q - 1)}$.

Συνιστώμενες παράμετροι του RSA

- ▶ $|n| \geq 2048$ (μέχρι το 2030 περίπου, μετά $|n| \geq 3072$).
- ▶ p, q περίπου ίδιου μήκους.
- ▶ $p - q > 2^{\frac{|n|}{2} - 100}$.
- ▶ $p - 1, q - 1$ έχουν και ‘μεγάλους’ πρώτους παράγοντες (αποφυγή **κυκλικών επιθέσεων**).
- ▶ $66537 < e \leq 2^{256}$. Επιλέγεται πριν από τα p, q .
- ▶ $ed \equiv 1 \pmod{\lambda(n) = \text{lcm}(p - 1, q - 1)}$.
- ▶ $2^{\frac{|n|}{2}} < d < \text{lcm}(p - 1, q - 1)$.

Συνιστώμενες παράμετροι του RSA

- ▶ $|n| \geq 2048$ (μέχρι το 2030 περίπου, μετά $|n| \geq 3072$).
- ▶ p, q περίπου ίδιου μήκους.
- ▶ $p - q > 2^{\frac{|n|}{2} - 100}$.
- ▶ $p - 1, q - 1$ έχουν και ‘μεγάλους’ πρώτους παράγοντες (αποφυγή **κυκλικών επιθέσεων**).
- ▶ $66537 < e \leq 2^{256}$. Επιλέγεται πριν από τα p, q .
- ▶ $ed \equiv 1 \pmod{\lambda(n) = \text{lcm}(p - 1, q - 1)}$.
- ▶ $2^{\frac{|n|}{2}} < d < \text{lcm}(p - 1, q - 1)$.
- ▶ Περισσότερα: NIST.

Το κρυπτοσύστημα Rabin

Ορισμός

Δημόσιο κλειδί: $n = pq, b < n$. Ιδιωτικό κλειδί: p, q .

Το κρυπτοσύστημα Rabin

Ορισμός

Δημόσιο κλειδί: $n = pq$, $b < n$. Ιδιωτικό κλειδί: p, q .

$$enc(x) = (x \cdot (x + b)) \bmod n$$

Το κρυπτοσύστημα Rabin

Ορισμός

Δημόσιο κλειδί: $n = pq$, $b < n$. Ιδιωτικό κλειδί: p, q .

$$enc(x) = (x \cdot (x + b)) \bmod n$$

$$dec(y) = x' - \frac{b}{2} \bmod n, \quad x'^2 \equiv y + \frac{b^2}{4} \pmod{n}$$

Το κρυπτόςστημα Rabin

Ορισμός

Δημόσιο κλειδί: $n = pq$, $b < n$. Ιδιωτικό κλειδί: p, q .

$$enc(x) = (x \cdot (x + b)) \bmod n$$

$$dec(y) = x' - \frac{b}{2} \bmod n, \quad x'^2 \equiv y + \frac{b^2}{4} \pmod{n}$$

Η αποκρυπτογράφηση συνίσταται ουσιαστικά στην εύρεση τετραγωνικών ριζών $(\bmod n)$ του $y' = y + \frac{b^2}{4}$:

Το κρυπτοσύστημα Rabin

Ορισμός

Δημόσιο κλειδί: $n = pq$, $b < n$. Ιδιωτικό κλειδί: p, q .

$$enc(x) = (x \cdot (x + b)) \bmod n$$

$$dec(y) = x' - \frac{b}{2} \bmod n, \quad x'^2 \equiv y + \frac{b^2}{4} \pmod{n}$$

Η αποκρυπτογράφηση συνίσταται ουσιαστικά στην εύρεση τετραγωνικών ριζών $(\bmod n)$ του $y' = y + \frac{b^2}{4}$:

$\pm y'^{(p+1)/4} \pmod{p}$, $\pm y'^{(q+1)/4} \pmod{q}$, αν γνωρίζουμε p, q και $p \equiv q \equiv 3 \pmod{4}$.

Το κρυπτοσύστημα Rabin

Ορισμός

Δημόσιο κλειδί: $n = pq$, $b < n$. Ιδιωτικό κλειδί: p, q .

$$enc(x) = (x \cdot (x + b)) \bmod n$$

$$dec(y) = x' - \frac{b}{2} \bmod n, \quad x'^2 \equiv y + \frac{b^2}{4} \pmod{n}$$

Η αποκρυπτογράφηση συνίσταται ουσιαστικά στην εύρεση τετραγωνικών ριζών $(\bmod n)$ του $y' = y + \frac{b^2}{4}$:

$\pm y'^{(p+1)/4} \pmod{p}$, $\pm y'^{(q+1)/4} \pmod{q}$, αν γνωρίζουμε p, q και $p \equiv q \equiv 3 \pmod{4}$.

Σημαντικό: η αποκρυπτογράφηση χωρίς γνώση των p, q είναι ισοδύναμη με παραγοντοποίηση του n .

Διακριτός Λογάριθμος

Ορισμός

Έστω G μία πεπερασμένη κυκλική ομάδα τάξης n , α ένας γεννήτορας της G και $\beta \in G$.

Ο **διακριτός λογάριθμος (discrete logarithm)** του β στη βάση α , που συμβολίζεται $\log_{\alpha}\beta$, είναι ο μοναδικός ακέραιος $x \in \mathbb{Z}_n$ τέτοιος ώστε $\beta = \alpha^x$.

Διακριτός Λογάριθμος

Ορισμός

Έστω G μία πεπερασμένη κυκλική ομάδα τάξης n , α ένας γεννήτορας της G και $\beta \in G$.

Ο **διακριτός λογάριθμος (discrete logarithm)** του β στη βάση α , που συμβολίζεται $\log_{\alpha}\beta$, είναι ο μοναδικός ακέραιος $x \in \mathbb{Z}_n$ τέτοιος ώστε $\beta = \alpha^x$.

Παράδειγμα. Για $p = 97$, η \mathbb{Z}_{97}^* είναι κυκλική ομάδα τάξης $n = 96$.

Ένας γεννήτορας της \mathbb{Z}_{97}^* είναι ο $\alpha = 5$.

Αφού $5^{32} \equiv 35 \pmod{97}$, έχουμε ότι $\log_5 35 = 32$ στο \mathbb{Z}_{97}^* .

Το πρόβλημα του Διακριτού Λογαρίθμου στο \mathbb{Z}_p^*

Discrete Logarithm Problem (DLP)

Δίνονται: ένας πρώτος αριθμός p , ένας γεννήτορας α του \mathbb{Z}_p^* και ένα στοιχείο $\beta \in \mathbb{Z}_p^*$.

Ζητείται: Να βρεθεί ακέραιος x , $0 \leq x \leq p - 2$, τέτοιος ώστε

$$\alpha^x \equiv \beta \pmod{p} \quad (1)$$

Το πρόβλημα του Διακριτού Λογαρίθμου στο \mathbb{Z}_p^*

Discrete Logarithm Problem (DLP)

Δίνονται: ένας πρώτος αριθμός p , ένας γεννήτορας α του \mathbb{Z}_p^* και ένα στοιχείο $\beta \in \mathbb{Z}_p^*$.

Ζητείται: Να βρεθεί ακέραιος x , $0 \leq x \leq p - 2$, τέτοιος ώστε

$$\alpha^x \equiv \beta \pmod{p} \quad (1)$$

Το πρόβλημα DLP (στο \mathbb{Z}_p^*) θεωρείται **υπολογιστικά δύσκολο** (υπό κάποιες προϋποθέσεις). Δεν γνωρίζουμε πολυωνυμικό αλγόριθμο που να το επιλύει.

Δυσκολία του DLP: ανεξάρτητη του γεννήτορα

Πρόταση

Η δυσκολία του DLP είναι ανεξάρτητη από την επιλογή του γεννήτορα α του \mathbb{Z}_p^ .*

Απόδειξη.

Έστω α και γ δύο γεννήτορες του \mathbb{Z}_p^* , και $\beta \in \mathbb{Z}_p^*$.

Έστω $x = \log_\alpha \beta$, $y = \log_\gamma \beta$ και $z = \log_\alpha \gamma$.

Τότε $\alpha^x \equiv \beta \equiv \gamma^y \equiv (\alpha^z)^y \pmod{p}$, δηλαδή $x \equiv zy \pmod{p-1}$.

Αλλά τότε $y \equiv xz^{-1} \pmod{p-1}$, δηλαδή:

$$\log_\gamma \beta \equiv (\log_\alpha \beta)(\log_\alpha \gamma)^{-1} \pmod{p-1}$$



Δυσκολία του DLP: ανεξάρτητη του γεννήτορα

Πρόταση

Η δυσκολία του DLP είναι ανεξάρτητη από την επιλογή του γεννήτορα α του \mathbb{Z}_p^* .

Απόδειξη.

Έστω α και γ δύο γεννήτορες του \mathbb{Z}_p^* , και $\beta \in \mathbb{Z}_p^*$.

Έστω $x = \log_\alpha \beta$, $y = \log_\gamma \beta$ και $z = \log_\alpha \gamma$.

Τότε $\alpha^x \equiv \beta \equiv \gamma^y \equiv (\alpha^z)^y \pmod{p}$, δηλαδή $x \equiv zy \pmod{p-1}$.

Αλλά τότε $y \equiv xz^{-1} \pmod{p-1}$, δηλαδή:

$$\log_\gamma \beta \equiv (\log_\alpha \beta)(\log_\alpha \gamma)^{-1} \pmod{p-1}$$

□

Επομένως αν μπορούμε να υπολογίσουμε τον διακριτό λογάριθμο σε μία βάση α τότε μπορούμε να τον υπολογίσουμε σε οποιαδήποτε βάση γ , όπου α, γ γεννήτορες του \mathbb{Z}_p^* .

Προφανής αλγόριθμος: $\tilde{O}(p)$.

Αλγόριθμοι για το DLP

Προφανής αλγόριθμος: $\tilde{O}(p)$.

Αλγόριθμος με προεπεξεργασία: υπολογίζουμε όλα τα ζεύγη (x, a^x) και ταξινομούμε ως προς δεύτερη συντεταγμένη. Χρόνος και χώρος προεπεξεργασίας $\tilde{O}(p)$, χρόνος απάντησης ερωτήματος $\tilde{O}(1)$.

Αλγόριθμοι για το DLP

Προφανής αλγόριθμος: $\tilde{O}(p)$.

Αλγόριθμος με προεπεξεργασία: υπολογίζουμε όλα τα ζεύγη (x, a^x) και ταξινομούμε ως προς δεύτερη συντεταγμένη. Χρόνος και χώρος προεπεξεργασίας $\tilde{O}(p)$, χρόνος απάντησης ερωτήματος $\tilde{O}(1)$.

Βελτιωμένη ιδέα: **αλγόριθμος Shanks**.

Αλγόριθμος Shanks

Είσοδος: πρώτος p , α γεννήτορας του \mathbb{Z}_p^* , $\beta \in \mathbb{Z}_p^*$.

Εξοδος: $x \in \mathbb{Z}_p^* : \alpha^x \equiv \beta \pmod{p}$.

1. $m := \lceil \sqrt{p-1} \rceil$
2. Υπολόγισε $\alpha^{mj} \pmod{p}$, $0 \leq j \leq m-1$
3. Ταξινόμησε τα m διατεταγμένα ζεύγη $(j, \alpha^{mj} \pmod{p})$ βάσει της δεύτερης συντεταγμένης (δηλαδή του $\alpha^{mj} \pmod{p}$), σε μια λίστα L_1
4. Υπολόγισε $\beta\alpha^{-i} \pmod{p}$, $0 \leq i \leq m-1$
5. Ταξινόμησε τα m διατεταγμένα ζεύγη $(i, \beta\alpha^{-i} \pmod{p})$ βάσει της δεύτερης συντεταγμένης (δηλαδή του $\beta\alpha^{-i} \pmod{p}$), σε μια λίστα L_2
6. Αναζήτησε ζεύγος $(j, y) \in L_1$ τέτοιο ώστε $(i, y) \in L_2$
7. Επίστρεψε $mj + i \pmod{p-1}$

Ορθότητα αλγορίθμου Shanks:

$$\alpha^{mj} \equiv y \equiv \beta\alpha^{-i} \pmod{p} \Rightarrow \alpha^{mj+i} \equiv \beta p \pmod{p}$$

Ορθότητα αλγορίθμου Shanks:

$$\alpha^{mj} \equiv y \equiv \beta \alpha^{-i} \pmod{p} \Rightarrow \alpha^{mj+i} \equiv \beta \pmod{p}$$

Πολυπλοκότητα: $\tilde{O}(\sqrt{p})$ σε χρόνο και $\tilde{O}(\sqrt{p})$ σε χώρο.

Ανταλλαγή Κλειδιού Diffie-Hellman

Πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman

Ανταλλαγή Κλειδιού Diffie-Hellman

Πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman

1. Επιλογή κοινού πρώτου p , και γεννήτορα α του \mathbb{Z}_p^* .

Ανταλλαγή Κλειδιού Diffie-Hellman

Πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman

1. Επιλογή κοινού πρώτου p , και γεννήτορα α του \mathbb{Z}_p^* .
2. Η **Αλίκη** επιλέγει έναν τυχαίο ακέραιο x που το γνωρίζει μόνο αυτή και στέλνει στον **Βασίλη** το μήνυμα: $\alpha^x \bmod p$

Ανταλλαγή Κλειδιού Diffie-Hellman

Πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman

1. Επιλογή κοινού πρώτου p , και γεννήτορα α του \mathbb{Z}_p^* .
2. Η **Αλίκη** επιλέγει έναν τυχαίο ακέραιο x που το γνωρίζει μόνο αυτή και στέλνει στον **Βασίλη** το μήνυμα: $\alpha^x \bmod p$
3. Ο **Βασίλης** επιλέγει έναν τυχαίο ακέραιο y που γνωρίζει μόνο αυτός και στέλνει στην **Αλίκη** το μήνυμα: $\alpha^y \bmod p$

Ανταλλαγή Κλειδιού Diffie-Hellman

Πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman

1. Επιλογή κοινού πρώτου p , και γεννήτορα α του \mathbb{Z}_p^* .
2. Η **Αλίκη** επιλέγει έναν τυχαίο ακέραιο x που το γνωρίζει μόνο αυτή και στέλνει στον **Βασίλη** το μήνυμα: $\alpha^x \bmod p$
3. Ο **Βασίλης** επιλέγει έναν τυχαίο ακέραιο y που γνωρίζει μόνο αυτός και στέλνει στην **Αλίκη** το μήνυμα: $\alpha^y \bmod p$
4. **Βασίλης**: $k = (\alpha^x)^y \bmod p$

Ανταλλαγή Κλειδιού Diffie-Hellman

Πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman

1. Επιλογή κοινού πρώτου p , και γεννήτορα α του \mathbb{Z}_p^* .
2. Η **Αλίκη** επιλέγει έναν τυχαίο ακέραιο x που το γνωρίζει μόνο αυτή και στέλνει στον **Βασίλη** το μήνυμα: $\alpha^x \bmod p$
3. Ο **Βασίλης** επιλέγει έναν τυχαίο ακέραιο y που γνωρίζει μόνο αυτός και στέλνει στην **Αλίκη** το μήνυμα: $\alpha^y \bmod p$
4. **Βασίλης**: $k = (\alpha^x)^y \bmod p$
Αλίκη: $k = (\alpha^y)^x \bmod p$

Ανταλλαγή Κλειδιού Diffie-Hellman

Πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman

1. Επιλογή κοινού πρώτου p , και γεννήτορα α του \mathbb{Z}_p^* .
2. Η **Αλίκη** επιλέγει έναν τυχαίο ακέραιο x που το γνωρίζει μόνο αυτή και στέλνει στον **Βασίλη** το μήνυμα: $\alpha^x \bmod p$
3. Ο **Βασίλης** επιλέγει έναν τυχαίο ακέραιο y που γνωρίζει μόνο αυτός και στέλνει στην **Αλίκη** το μήνυμα: $\alpha^y \bmod p$
4. **Βασίλης**: $k = (\alpha^x)^y \bmod p$
Αλίκη: $k = (\alpha^y)^x \bmod p$

Η ασφάλεια του πρωτοκόλλου αυτού φαίνεται να βασίζεται στην δυσκολία του DLP. Αυτό δεν είναι απόλυτα ακριβές.

Ανταλλαγή Κλειδιού Diffie-Hellman

Πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman

1. Επιλογή κοινού πρώτου p , και γεννήτορα α του \mathbb{Z}_p^* .
2. Η **Αλίκη** επιλέγει έναν τυχαίο ακέραιο x που το γνωρίζει μόνο αυτή και στέλνει στον **Βασίλη** το μήνυμα: $\alpha^x \bmod p$
3. Ο **Βασίλης** επιλέγει έναν τυχαίο ακέραιο y που γνωρίζει μόνο αυτός και στέλνει στην **Αλίκη** το μήνυμα: $\alpha^y \bmod p$
4. **Βασίλης**: $k = (\alpha^x)^y \bmod p$
Αλίκη: $k = (\alpha^y)^x \bmod p$

Η ασφάλεια του πρωτοκόλλου αυτού φαίνεται να βασίζεται στην δυσκολία του DLP. Αυτό δεν είναι απόλυτα ακριβές.

Στην πραγματικότητα, η ασφάλεια του πρωτοκόλλου Diffie-Hellman ταυτίζεται με την υπολογιστική δυσκολία του **Προβλήματος Diffie-Hellman (DHP)**.

Το Πρόβλημα Diffie-Hellman

Diffie-Hellman Problem (DHP)

- ▶ Δίνονται: ένας πρώτος αριθμός p , ένας γεννήτορας α του \mathbb{Z}_p^* και τα στοιχεία $\alpha^a \bmod p$, $\alpha^b \bmod p \in \mathbb{Z}_p^*$.

Το Πρόβλημα Diffie-Hellman

Diffie-Hellman Problem (DHP)

- ▶ Δίνονται: ένας πρώτος αριθμός p , ένας γεννήτορας α του \mathbb{Z}_p^* και τα στοιχεία $\alpha^a \bmod p, \alpha^b \bmod p \in \mathbb{Z}_p^*$.
- ▶ Ζητείται: Να βρεθεί το $\alpha^{ab} \bmod p$.

Πρόταση

Το DHP ανάγεται σε πολυωνυμικό χρόνο στο DLP: $DHP \leq^p DLP$

Το Πρόβλημα Diffie-Hellman

Diffie-Hellman Problem (DHP)

- ▶ Δίνονται: ένας πρώτος αριθμός p , ένας γεννήτορας α του \mathbb{Z}_p^* και τα στοιχεία $\alpha^a \bmod p, \alpha^b \bmod p \in \mathbb{Z}_p^*$.
- ▶ Ζητείται: Να βρεθεί το $\alpha^{ab} \bmod p$.

Πρόταση

Το DHP ανάγεται σε πολυωνυμικό χρόνο στο DLP: $DHP \leq^p DLP$

Πράγματι, αν $x = \alpha^a \bmod p$ και $y = \alpha^b \bmod p$, τότε $a = \log_\alpha x$ και $b = \log_\alpha y$. Επομένως, λύνοντας το DLP, μπορούμε να υπολογίσουμε τα a, b άρα και το $\alpha^{ab} \bmod p$.

Το Πρόβλημα Diffie-Hellman

Diffie-Hellman Problem (DHP)

- ▶ Δίνονται: ένας πρώτος αριθμός p , ένας γεννήτορας α του \mathbb{Z}_p^* και τα στοιχεία $\alpha^a \bmod p, \alpha^b \bmod p \in \mathbb{Z}_p^*$.
- ▶ Ζητείται: Να βρεθεί το $\alpha^{ab} \bmod p$.

Πρόταση

Το DHP ανάγεται σε πολυωνυμικό χρόνο στο DLP: $DHP \leq^P DLP$

Πράγματι, αν $x = \alpha^a \bmod p$ και $y = \alpha^b \bmod p$, τότε $a = \log_{\alpha} x$ και $b = \log_{\alpha} y$. Επομένως, λύνοντας το DLP, μπορούμε να υπολογίσουμε τα a, b άρα και το $\alpha^{ab} \bmod p$.

Δεν γνωρίζουμε αν ισχύει και το αντίστροφο ($DLP \leq^P DHP$).

Το Πρόβλημα Απόφασης Diffie-Hellman

Decision Diffie-Hellman Problem (DDHP)

- ▶ Δίνονται: ένας πρώτος αριθμός p , ένας γεννήτορας α του \mathbb{Z}_p^* και δύο τριάδες $\langle \alpha^a, \alpha^b, \alpha^c \rangle, \langle \alpha^a, \alpha^b, \alpha^{ab} \rangle \pmod{p}$.

Το Πρόβλημα Απόφασης Diffie-Hellman

Decision Diffie-Hellman Problem (DDHP)

- ▶ *Δίνονται*: ένας πρώτος αριθμός p , ένας γεννήτορας α του \mathbb{Z}_p^* και δύο τριάδες $\langle \alpha^a, \alpha^b, \alpha^c \rangle, \langle \alpha^a, \alpha^b, \alpha^{ab} \rangle \pmod{p}$.
- ▶ *Ζητείται*: Να βρεθεί (με πιθανότητα ‘αρκετά’ μεγαλύτερη από $1/2$) ποιά είναι η “σωστή” τριάδα, δηλαδή η $\langle \alpha^a, \alpha^b, \alpha^{ab} \rangle$.

Για αποφυγή σύγχυσης, το κλασικό πρόβλημα DHP αναφέρεται συχνά και ως **Computational Diffie-Hellman Problem (CDHP)**.

Η σχετική δυσκολία των DDHP, CDHP, DLP

Προφανώς ισχύει:

$$\text{DDHP} \leq^p \text{CDHP} \leq^p \text{DLP}$$

Η σχετική δυσκολία των DDHP, CDHP, DLP

Προφανώς ισχύει:

$$\text{DDHP} \leq^p \text{CDHP} \leq^p \text{DLP}$$

Cryptographic assumptions

Για κάθε πρόβλημα ορίζεται και η αντίστοιχη υπόθεση υπολογιστικής δυσκολίας του: **DDH**, **CDH**, **DL**.

Η σχετική δυσκολία των DDHP, CDHP, DLP

Προφανώς ισχύει:

$$\text{DDHP} \leq^p \text{CDHP} \leq^p \text{DLP}$$

Cryptographic assumptions

Για κάθε πρόβλημα ορίζεται και η αντίστοιχη υπόθεση υπολογιστικής δυσκολίας του: **DDH**, **CDH**, **DL**.

Η σειρά ισχύος των υποθέσεων:

$$\text{DDH} \Rightarrow \text{CDH} \Rightarrow \text{DL}$$

Η σχετική δυσκολία των DDHP, CDHP, DLP

Προφανώς ισχύει:

$$\text{DDHP} \leq^p \text{CDHP} \leq^p \text{DLP}$$

Cryptographic assumptions

Για κάθε πρόβλημα ορίζεται και η αντίστοιχη υπόθεση υπολογιστικής δυσκολίας του: **DDH**, **CDH**, **DL**.

Η σειρά ισχύος των υποθέσεων:

$$\text{DDH} \Rightarrow \text{CDH} \Rightarrow \text{DL}$$

Σημαντική παρατήρηση: υπάρχουν κυκλικές ομάδες όπου το DDHP είναι εύκολο (υπό προϋποθέσεις), ενώ το CDHP θεωρείται δύσκολο. Παράδειγμα: η ομάδα \mathbb{Z}_p^* (λόγω της δυνατότητας υπολογισμού του τελευταίου bit του διακριτού λογαρίθμου).

Το κρυπτοσύστημα ElGamal (Taher ElGamal, Crypto'84)

Παραγωγή κλειδιών

Η Alice διαλέγει ένα πρώτο p , όπου ο $p - 1$ έχει τουλάχιστον ένα μεγάλο παράγοντα, ένα γεννήτορα g της \mathbb{Z}_p^* , και τυχαίο $a \in \mathbb{Z}_p^*$

Δημόσιο κλειδί της Alice: $p, g, g^a \pmod p$.

Ιδιωτικό κλειδί της Alice: a .

Το κρυπτοσύστημα ElGamal (Taher ElGamal, Crypto'84)

Παραγωγή κλειδιών

Η Alice διαλέγει ένα πρώτο p , όπου ο $p - 1$ έχει τουλάχιστον ένα μεγάλο παράγοντα, ένα γεννήτορα g της \mathbb{Z}_p^* , και τυχαίο $a \in \mathbb{Z}_p^*$

Δημόσιο κλειδί της Alice: $p, g, g^a \bmod p$.

Ιδιωτικό κλειδί της Alice: a .

Κρυπτογράφηση

1. Ο Bob επιλέγει τυχαίο $k \in \{2, 3, \dots, p - 2\}$.

Το κρυπτοσύστημα ElGamal (Taher ElGamal, Crypto'84)

Παραγωγή κλειδιών

Η Alice διαλέγει ένα πρώτο p , όπου ο $p - 1$ έχει τουλάχιστον ένα μεγάλο παράγοντα, ένα γεννήτορα g της \mathbb{Z}_p^* , και τυχαίο $a \in \mathbb{Z}_p^*$

Δημόσιο κλειδί της Alice: $p, g, g^a \bmod p$.

Ιδιωτικό κλειδί της Alice: a .

Κρυπτογράφηση

1. Ο Bob επιλέγει τυχαίο $k \in \{2, 3, \dots, p - 2\}$.
2. Ο Bob υπολογίζει $\gamma = g^k \bmod p$ και $\delta = m(g^a)^k \bmod p$ και στέλνει το ζευγάρι (γ, δ) στην Alice (1-to-2 message expansion).

Το κρυπτοσύστημα ElGamal (Taher ElGamal, Crypto'84)

Παραγωγή κλειδιών

Η Alice διαλέγει ένα πρώτο p , όπου ο $p - 1$ έχει τουλάχιστον ένα μεγάλο παράγοντα, ένα γεννήτορα g της \mathbb{Z}_p^* , και τυχαίο $a \in \mathbb{Z}_p^*$

Δημόσιο κλειδί της Alice: $p, g, g^a \pmod p$.

Ιδιωτικό κλειδί της Alice: a .

Κρυπτογράφηση

1. Ο Bob επιλέγει τυχαίο $k \in \{2, 3, \dots, p - 2\}$.
2. Ο Bob υπολογίζει $\gamma = g^k \pmod p$ και $\delta = m(g^a)^k \pmod p$ και στέλνει το ζευγάρι (γ, δ) στην Alice (1-to-2 message expansion).

Αποκρυπτογράφηση

1. Η Alice πρώτα υπολογίζει: $\gamma^a \equiv g^{ak} \pmod p$ και μετά αντιστρέφει σε $(g^{ak})^{-1}$.

Το κρυπτοσύστημα ElGamal (Taher ElGamal, Crypto'84)

Παραγωγή κλειδιών

Η Alice διαλέγει ένα πρώτο p , όπου ο $p - 1$ έχει τουλάχιστον ένα μεγάλο παράγοντα, ένα γεννήτορα g της \mathbb{Z}_p^* , και τυχαίο $a \in \mathbb{Z}_p^*$

Δημόσιο κλειδί της Alice: $p, g, g^a \pmod p$.

Ιδιωτικό κλειδί της Alice: a .

Κρυπτογράφηση

1. Ο Bob επιλέγει τυχαίο $k \in \{2, 3, \dots, p - 2\}$.
2. Ο Bob υπολογίζει $\gamma = g^k \pmod p$ και $\delta = m(g^a)^k \pmod p$ και στέλνει το ζευγάρι (γ, δ) στην Alice (1-to-2 message expansion).

Αποκρυπτογράφηση

1. Η Alice πρώτα υπολογίζει: $\gamma^a \equiv g^{ak} \pmod p$ και μετά αντιστρέφει σε $(g^{ak})^{-1}$.
2. Τέλος υπολογίζει:
$$(g^{ak})^{-1} \cdot \delta \pmod p \equiv (g^{ak})^{-1} (m(g^a)^k) \equiv (g^{ak})^{-1} \cdot m \cdot g^{ak} \equiv m$$

Επειδή το k είναι τυχαίο η κρυπτογράφηση είναι πιθανοτική.

Παρατηρήσεις στο ElGamal

Επειδή το k είναι τυχαίο η κρυπτογράφηση είναι πιθανοτική.

Ερώτηση: σε ποιο πρόβλημα στηρίζεται η ασφάλεια του ElGamal;

Παρατηρήσεις στο ElGamal

Επειδή το k είναι τυχαίο η κρυπτογράφηση είναι πιθανοτική.

Ερώτηση: σε ποιο πρόβλημα στηρίζεται η ασφάλεια του ElGamal;

Απάντηση: στο **DHP** (γιατί;).

Παρατηρήσεις στο ElGamal

Επειδή το k είναι τυχαίο η κρυπτογράφηση είναι πιθανοτική.

Ερώτηση: σε ποιο πρόβλημα στηρίζεται η ασφάλεια του ElGamal;

Απάντηση: στο **DHP** (γιατί;).

Και μάλιστα **με ισοδυναμία!**:

$$\text{CDHP} \equiv^p \text{ElGamal-decrypt}$$

Επανάληψη του $k \Rightarrow$ επίθεση ΚΡΑ

Έστω m_1, m_2 δύο απλά κείμενα και $(\gamma, \delta_1), (\gamma, \delta_2)$ τα αντίστοιχα κρυπτοκείμενα (με χρήση του ίδιου k).

Η Ενε γνωρίζοντας τα $\gamma, \delta_1, \delta_2$, και m_1 (ΚΡΑ), υπολογίζει το m_2 ως εξής:

Επανάληψη του $k \Rightarrow$ επίθεση ΚΡΑ

Έστω m_1, m_2 δύο απλά κείμενα και $(\gamma, \delta_1), (\gamma, \delta_2)$ τα αντίστοιχα κρυπτοκείμενα (με χρήση του ίδιου k).

Η Ενε γνωρίζοντας τα $\gamma, \delta_1, \delta_2$, και m_1 (ΚΡΑ), υπολογίζει το m_2 ως εξής:

$$\left. \begin{array}{l} \delta_1 \equiv m_1 g^{ak} \pmod{p} \\ \delta_2 \equiv m_2 g^{ak} \pmod{p} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \delta_1^{-1} m_1 \equiv g^{-ak} \pmod{p} \\ \delta_2 g^{-ak} \equiv m_2 \pmod{p} \end{array} \right.$$

Επανάληψη του $k \Rightarrow$ επίθεση ΚΡΑ

Έστω m_1, m_2 δύο απλά κείμενα και $(\gamma, \delta_1), (\gamma, \delta_2)$ τα αντίστοιχα κρυπτοκείμενα (με χρήση του ίδιου k).

Η Ενε γνωρίζοντας τα $\gamma, \delta_1, \delta_2$, και m_1 (ΚΡΑ), υπολογίζει το m_2 ως εξής:

$$\left. \begin{array}{l} \delta_1 \equiv m_1 g^{ak} \pmod{p} \\ \delta_2 \equiv m_2 g^{ak} \pmod{p} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \delta_1^{-1} m_1 \equiv g^{-ak} \pmod{p} \\ \delta_2 g^{-ak} \equiv m_2 \pmod{p} \end{array} \right.$$

Οπότε $m_2 \equiv \delta_2 \delta_1^{-1} m_1 \pmod{p}$.