

# Κρυπτογραφία

Μονόδρομες συναρτήσεις - Συναρτήσεις σύνοψης

Άρης Παγουρτζής - Πέτρος Ποτίκας

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

# Περιεχόμενα

- 1 Μονόδρομες Συναρτήσεις
- 2 Συναρτήσεις σύνοψης (hash functions)
- 3 Δένδρα Merkle

# Συναρτήσεις μονόδρομες ή μονής-κατεύθυνσης (one-way functions)

- ▶ Συνάρτηση που είναι εύκολο να υπολογιστεί, αλλά “δύσκολο” να αντιστραφεί

# Συναρτήσεις μονόδρομες ή μονής-κατεύθυνσης (one-way functions)

- ▶ Συνάρτηση που είναι εύκολο να υπολογιστεί, αλλά “δύσκολο” να αντιστραφεί
- ▶ Απαραίτητη προϋπόθεση για κρυπτογραφία ιδιωτικού κλειδιού

# Συναρτήσεις μονόδρομες ή μονής-κατεύθυνσης (one-way functions)

- ▶ Συνάρτηση που είναι εύκολο να υπολογιστεί, αλλά “δύσκολο” να αντιστραφεί
- ▶ Απαραίτητη προϋπόθεση για κρυπτογραφία ιδιωτικού κλειδιού
- ▶ Γεννήτριες ψευδοτυχειότητας βασίζονται στην υπόθεση ύπαρξης μονόδρομων συναρτήσεων

## Συναρτήσεις μονόδρομες ή μονής-κατεύθυνσης (one-way functions)

- ▶ Συνάρτηση που είναι εύκολο να υπολογιστεί, αλλά “δύσκολο” να αντιστραφεί
- ▶ Απαραίτητη προϋπόθεση για κρυπτογραφία ιδιωτικού κλειδιού
- ▶ Γεννήτριες ψευδοτυχειότητας βασίζονται στην υπόθεση ύπαρξης μονόδρομων συναρτήσεων
- ▶ Μόνο με αμελητέα πιθανότητα μπορώ να αντιστρέψω μια μονόδρομη συνάρτηση

## Συναρτήσεις μονόδρομες ή μονής-κατεύθυνσης (one-way functions)

- ▶ Συνάρτηση που είναι εύκολο να υπολογιστεί, αλλά “δύσκολο” να αντιστραφεί
- ▶ Απαραίτητη προϋπόθεση για κρυπτογραφία ιδιωτικού κλειδιού
- ▶ Γεννήτριες ψευδοτυχειότητας βασίζονται στην υπόθεση ύπαρξης μονόδρομων συναρτήσεων
- ▶ Μόνο με αμελητέα πιθανότητα μπορώ να αντιστρέψω μια μονόδρομη συνάρτηση
- ▶ Με εξαντλητική αναζήτηση (εκθετικό χρόνο) μπορώ να αντιστρέψω μια μονόδρομη συνάρτηση

# Μονόδρομες Συναρτήσεις

Έστω συνάρτηση  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$ .



# Μονόδρομες Συναρτήσεις

Έστω συνάρτηση  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$ .

Ορίζουμε για κάθε αλγόριθμο  $\mathcal{A}$  και κάθε παράμετρο ασφαλείας  $n$  το

**Πείραμα αντιστρεψιμότητας**  $Invert_{\mathcal{A},f}(n)$

1. Διάλεξε  $x \leftarrow \{0, 1\}^n$ . Υπολόγισε  $y = f(x)$
2. Ο  $\mathcal{A}$  με είσοδο το  $1^n$  και το  $y$  επιστρέφει το  $x'$
3. Η έξοδος είναι 1, αν  $f(x') = y$ , αλλιώς 0

# Μονόδρομες Συναρτήσεις

Έστω συνάρτηση  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$ .

Ορίζουμε για κάθε αλγόριθμο  $\mathcal{A}$  και κάθε παράμετρο ασφαλείας  $n$  το

**Πείραμα αντιστρεψιμότητας**  $Invert_{\mathcal{A},f}(n)$

1. Διάλεξε  $x \leftarrow \{0, 1\}^n$ . Υπολόγισε  $y = f(x)$
2. Ο  $\mathcal{A}$  με είσοδο το  $1^n$  και το  $y$  επιστρέφει το  $x'$
3. Η έξοδος είναι 1, αν  $f(x') = y$ , αλλιώς 0

Παρατήρηση: Δε χρειάζεται να βρούμε το ίδιο το  $x$ , αλλά οποιαδήποτε  $x'$ , τ.ώ.  $f(x') = y = f(x)$ .

# Μονόδρομες Συναρτήσεις

## Ορισμός

Μία συνάρτηση  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$  είναι *μονόδρομη συνάρτηση* αν είναι:

1. (Εύκολα υπολογίσιμη) Υπάρχει πολυωνυμικού χρόνου αλγόριθμος  $M_f$  που την υπολογίζει, δηλ.  $M_f(x) = f(x), \forall x$
2. (Δύσκολα αντιστρέψιμη) Για κάθε PPT αλγόριθμο  $\mathcal{A}$  υπάρχει αμελητέα συνάρτηση  $\epsilon$  έτσι ώστε:

$$\Pr[\text{Invert}_{\mathcal{A},f}(n) = 1] \leq \epsilon(n)$$

# Μονόδρομες Συναρτήσεις

## Ορισμός

Μία συνάρτηση  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$  είναι *μονόδρομη συνάρτηση* αν είναι:

1. (Εύκολα υπολογίσιμη) Υπάρχει πολυωνυμικού χρόνου αλγόριθμος  $M_f$  που την υπολογίζει, δηλ.  $M_f(x) = f(x), \forall x$
2. (Δύσκολα αντιστρέψιμη) Για κάθε PPT αλγόριθμο  $\mathcal{A}$  υπάρχει αμελητέα συνάρτηση  $\epsilon$  έτσι ώστε:

$$\Pr[\text{Invert}_{\mathcal{A},f}(n) = 1] \leq \epsilon(n)$$

Πιο αναλυτικά,

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}(1^n, f(x)) \in f^{-1}(f(x))] \leq \epsilon(n)$$

# Μονόδρομες Συναρτήσεις

Παρατηρήσεις:

1. Μια συνάρτηση που δεν είναι μονόδρομη δεν είναι απαραίτητο να αντιστρέφεται πάντα εύκολα (ή “συχνά”).

# Μονόδρομες Συναρτήσεις

Παρατηρήσεις:

1. Μια συνάρτηση που δεν είναι μονόδρομη δεν είναι απαραίτητο να αντιστρέφεται πάντα εύκολα (ή “συχνά”).  
Π.χ. αν υπάρχει αντίπαλος που αντιστρέφει μια συνάρτηση με πιθανότητα  $n^{-10}$  για όλους άρτιους  $n$  (αλλά αποτυγχάνει για τους μονούς), τότε δεν είναι μονόδρομη.

# Μονόδρομες Συναρτήσεις

Παρατηρήσεις:

1. Μια συνάρτηση που δεν είναι μονόδρομη δεν είναι απαραίτητο να αντιστρέφεται πάντα εύκολα (ή “συχνά”).  
Π.χ. αν υπάρχει αντίπαλος που αντιστρέφει μια συνάρτηση με πιθανότητα  $n^{-10}$  για όλους άρτιους  $n$  (αλλά αποτυγχάνει για τους μονούς), τότε δεν είναι μονόδρομη.
2. Αν έχουμε εκθετικό χρόνο, τότε αν μας δίνεται ένα  $y$  και η παράμετρος ασφαλείας  $1^n$ , τότε μπορούμε να δοκιμάσουμε όλα τα  $x \in \{0, 1\}^n$ , μέχρι να βρούμε ένα  $x$ , τέτοιο ώστε  $f(x) = y$ .

# Μονόδρομες Μεταθέσεις

Μια συνάρτηση λέμε ότι *διατηρεί το μήκος* αν  $|f(x)| = |x|, \forall x$ .



# Μονόδρομες Μεταθέσεις

Μια συνάρτηση λέμε ότι *διατηρεί το μήκος* αν  $|f(x)| = |x|, \forall x$ .

## Ορισμός

Μια μονόδρομη συνάρτηση που διατηρεί το μήκος και είναι 1-1, είναι μια *μονόδρομη μετάθεση*.

Η τιμή  $y$  καθορίζει μοναδικά το  $x$  από το οποίο προήλθε. Παρόλα αυτά είναι δύσκολο να βρούμε το  $x$  σε πολυωνυμικό χρόνο.

# Υποψήφιες μονόδρομες συναρτήσεις

Υπάρχουν μονόδρομες συναρτήσεις με την προϋπόθεση πως κάποια προβλήματα είναι δύσκολα, π.χ. παραγοντοποίηση ακεραίων

# Υποψήφιες μονόδρομες συναρτήσεις

Υπάρχουν μονόδρομες συναρτήσεις με την προϋπόθεση πως κάποια προβλήματα είναι δύσκολα, π.χ. παραγοντοποίηση ακεραίων

## Παράδειγμα 1

$$f_{mult}(x, y) = xy,$$

# Υποψήφιες μονόδρομες συναρτήσεις

Υπάρχουν μονόδρομες συναρτήσεις με την προϋπόθεση πως κάποια προβλήματα είναι δύσκολα, π.χ. παραγοντοποίηση ακεραίων

## Παράδειγμα 1

$f_{mult}(x, y) = xy$ , όμως με μεγάλη πιθανότητα, το αποτέλεσμα άρτιος, οπότε το  $(2, xy/2)$  είναι αντίστροφος.

# Υποψήφιες μονόδρομες συναρτήσεις

Υπάρχουν μονόδρομες συναρτήσεις με την προϋπόθεση πως κάποια προβλήματα είναι δύσκολα, π.χ. παραγοντοποίηση ακεραίων

## Παράδειγμα 1

$f_{mult}(x, y) = xy$ , όμως με μεγάλη πιθανότητα, το αποτέλεσμα άρτιος, οπότε το  $(2, xy/2)$  είναι αντίστροφος. Με περιορισμό, είναι μονόδρομη:

# Υποψήφιες μονόδρομες συναρτήσεις

Υπάρχουν μονόδρομες συναρτήσεις με την προϋπόθεση πως κάποια προβλήματα είναι δύσκολα, π.χ. παραγοντοποίηση ακεραίων

## Παράδειγμα 1

$f_{mult}(x, y) = xy$ , όμως με μεγάλη πιθανότητα, το αποτέλεσμα άρτιος, οπότε το  $(2, xy/2)$  είναι αντίστροφος. Με περιορισμό, είναι μονόδρομη:

1.  $f_{mult}(x, y) = (xy, |x|, |y|)$ , (εναλλακτικά,  $x, y$  έχουν ίδιο μήκος)

# Υποψήφιες μονόδρομες συναρτήσεις

Υπάρχουν μονόδρομες συναρτήσεις με την προϋπόθεση πως κάποια προβλήματα είναι δύσκολα, π.χ. παραγοντοποίηση ακεραίων

## Παράδειγμα 1

$f_{mult}(x, y) = xy$ , όμως με μεγάλη πιθανότητα, το αποτέλεσμα άρτιος, οπότε το  $(2, xy/2)$  είναι αντίστροφος. Με περιορισμό, είναι μονόδρομη:

1.  $f_{mult}(x, y) = (xy, |x|, |y|)$ , (εναλλακτικά,  $x, y$  έχουν ίδιο μήκος)
2.  $x, y$  πρώτοι αριθμοί ίσου μήκους

# Υποψήφιες μονόδρομες συναρτήσεις

Υπάρχουν μονόδρομες συναρτήσεις με την προϋπόθεση πως κάποια προβλήματα είναι δύσκολα, π.χ. παραγοντοποίηση ακεραίων

## Παράδειγμα 1

$f_{mult}(x, y) = xy$ , όμως με μεγάλη πιθανότητα, το αποτέλεσμα άρτιος, οπότε το  $(2, xy/2)$  είναι αντίστροφος. Με περιορισμό, είναι μονόδρομη:

1.  $f_{mult}(x, y) = (xy, |x|, |y|)$ , (εναλλακτικά,  $x, y$  έχουν ίδιο μήκος)
2.  $x, y$  πρώτοι αριθμοί ίσου μήκους

## Παράδειγμα 2

Η συνάρτηση  $f(x_1, \dots, x_n, J) = (x_1, \dots, x_n, \sum_{j \in J} x_j)$ , όπου κάθε  $x_i$  είναι ένας ακέραιος και  $J \subseteq \{1, 2, \dots, n\}$ . Εύρεση αντιστρόφου είναι το γνωστό  $\mathcal{NP}$ -πλήρες πρόβλημα Subset Sum. Είναι μονόδρομη;



# Υποψήφιες μονόδρομες μεταθέσεις

## Παράδειγμα 3

Έστω ένας πρώτος αριθμός  $p$  μήκους  $n$ -bits και ένας γεννήτορας  $g \in \mathbb{Z}_p^*$ .  
Έστω ένα στοιχείο  $x \in \mathbb{Z}_p^*$ .

# Υποψήφιες μονόδρομες μεταθέσεις

## Παράδειγμα 3

Έστω ένας πρώτος αριθμός  $p$  μήκους  $n$ -bits και ένας γεννήτορας  $g \in \mathbb{Z}_p^*$ .  
Έστω ένα στοιχείο  $x \in \mathbb{Z}_p^*$ . Ορίζουμε

$$f_{p,g}(x) = g^x \bmod p$$

# Υποψήφιες μονόδρομες μεταθέσεις

## Παράδειγμα 3

Έστω ένας πρώτος αριθμός  $p$  μήκους  $n$ -bits και ένας γεννήτορας  $g \in \mathbb{Z}_p^*$ .  
Έστω ένα στοιχείο  $x \in \mathbb{Z}_p^*$ . Ορίζουμε

$$f_{p,g}(x) = g^x \bmod p$$

- ▶ Η συνάρτηση αυτή διατηρεί το μήκος και είναι 1-1, άρα μετάθεση.
- ▶ Η δυσκολία αντιστροφής της βασίζεται στη δυσκολία του προβλήματος διακριτού λογάριθμου.

# Υποψήφιες μονόδρομες μεταθέσεις

## Παράδειγμα 3

Έστω ένας πρώτος αριθμός  $p$  μήκους  $n$ -bits και ένας γεννήτορας  $g \in \mathbb{Z}_p^*$ .  
Έστω ένα στοιχείο  $x \in \mathbb{Z}_p^*$ . Ορίζουμε

$$f_{p,g}(x) = g^x \bmod p$$

- ▶ Η συνάρτηση αυτή διατηρεί το μήκος και είναι 1-1, άρα μετάθεση.
- ▶ Η δυσκολία αντιστροφής της βασίζεται στη δυσκολία του προβλήματος διακριτού λογάριθμου.

Πρακτικά συστήματα, όπως το AES, δίνουν μονόδρομες συναρτήσεις, π.χ. με την υπόθεση ότι είναι ψευδοτυχαία μετάθεση.

# Μονόδρομες συναρτήσεις καταπακτής (Trapdoor one-way functions)

Μονόδρομες συναρτήσεις που είναι δύσκολο να αντιστραφούν, εκτός και αν ξέρουμε κάποιο μυστικό, την *καταπακτή* (trapdoor).

## Παράδειγμα

Έστω  $N = pq$ , όπου  $p, q$  μεγάλοι πρώτοι αριθμοί.

Η συνάρτηση  $f_N(x) = x^2 \bmod N$  είναι μια μονόδρομη μετάθεση με καταπακτή.

Βασίζεται στην δυσκολία εύρεσης τετραγωνικών ριζών  $\bmod N$ , για σύνθετο  $N$ , εκτός και αν ξέρουμε την παραγοντοποίηση του.

Γνωστή ως μονόδρομη μετάθεση Rabin (κρυπτοσύστημα)

# Συναρτήσεις σύνοψης (hash functions)

- ▶ Γνωστές και ως **συναρτήσεις κατακερματισμού**.
- ▶ Σημαντικές ιδιότητες:
  - ▶ **Συμπίεση**  $h : X \rightarrow Y, |Y| < |X|$ .  
Συνήθως  $X = \Sigma^*$ ,  $Y = \Sigma^n$ , δηλαδή η  $h(x)$  έχει συγκεκριμένο μήκος για οποιαδήποτε είσοδο  $x$ .
  - ▶ **Ευκολία Υπολογισμού** Ο υπολογισμός της τιμής  $h(x)$  για κάποιο  $x$  γίνεται “εύκολα”. Δηλαδή υπάρχει αλγόριθμος  $A$  πολυωνυμικού χρόνου, έτσι ώστε για κάθε  $x$  να ισχύει  $h(x) = A(x)$ .
- ▶ Μια συνάρτηση σύνοψης ορίζει σχέση ισοδυναμίας:

$$x \sim x' : \quad h(x) = h(x')$$

Δύο στοιχεία στην ίδια κλάση ισοδυναμίας λέμε ότι προκαλούν **σύγκρουση (collision)**.

## Συναρτήσεις σύννοψης (hash functions): επιθυμητές ιδιότητες

Έστω hash function  $h : X \rightarrow Y$ . Η  $h$  έχει:

1. Αντίσταση πρώτου ορίσματος (preimage resistance), αν για  $y \in Y$  είναι υπολογιστικά δύσκολο να βρεθεί  $x \in X$  τ.ώ.  $h(x) = y$ .
2. Αντίσταση δεύτερου ορίσματος (2nd preimage resistance), αν για  $x \in X$  είναι υπολογιστικά δύσκολο να βρεθεί  $x' \in X$  τ.ώ.  $x \neq x'$  και  $h(x) = h(x')$ .
3. Δυσκολία εύρεσης συγκρούσεων (collision resistance / freeness), αν είναι υπολογιστικά δύσκολο να βρεθούν  $x, x' \in X$  έτσι ώστε  $h(x) = h(x')$ .

Άλλα ονόματα: για το (2) weak collision freeness, για το (1) non-invertibility.

Σειρά ισχύος: (3)  $\Rightarrow$  (2)  $\Rightarrow$  (1) (υπό προϋποθέσεις).

One-way hash functions (OWHFs): (1) & (2).

Collision-resistant hash functions (CRHFs): (1) & (2) & (3).

# Συναρτήσεις σύνοψης (hash functions): παραδείγματα

1.  $f(x) = (x^2 - c) \bmod p$ :



# Συναρτήσεις σύνοψης (hash functions): παραδείγματα

1.  $f(x) = (x^2 - c) \bmod p$ : δεν είναι μονής κατεύθυνσης αφού η εύρεση τετραγωνικών ριζών στο  $\mathbb{Z}_p$  είναι δυνατή σε πολυωνυμικό χρόνο.

## Συναρτήσεις σύνοψης (hash functions): παραδείγματα

1.  $f(x) = (x^2 - c) \bmod p$ : δεν είναι μονής κατεύθυνσης αφού η εύρεση τετραγωνικών ριζών στο  $\mathbb{Z}_p$  είναι δυνατή σε πολυωνυμικό χρόνο.
2.  $g(x) = x^2 \bmod n$ ,  $n = pq$ ,  $p, q$  κρυφοί:

## Συναρτήσεις σύνοψης (hash functions): παραδείγματα

1.  $f(x) = (x^2 - c) \bmod p$ : δεν είναι μονής κατεύθυνσης αφού η εύρεση τετραγωνικών ριζών στο  $\mathbb{Z}_p$  είναι δυνατή σε πολυωνυμικό χρόνο.
2.  $g(x) = x^2 \bmod n$ ,  $n = pq$ ,  $p, q$  κρυφοί: αντίσταση πρώτου ορίσματος, αλλά όχι αντίσταση δεύτερου ορίσματος (γιατί;), επομένως δεν είναι CRHF.

## Συναρτήσεις σύνοψης (hash functions): παραδείγματα

1.  $f(x) = (x^2 - c) \bmod p$ : δεν είναι μονής κατεύθυνσης αφού η εύρεση τετραγωνικών ριζών στο  $\mathbb{Z}_p$  είναι δυνατή σε πολυωνυμικό χρόνο.
2.  $g(x) = x^2 \bmod n$ ,  $n = pq$ ,  $p, q$  κρυφοί: αντίσταση πρώτου ορίσματος, αλλά όχι αντίσταση δεύτερου ορίσματος (γιατί;), επομένως δεν είναι CRHF.
3.  $h : \mathbb{Z}_q^2 \rightarrow \mathbb{Z}_p^*$ ,  $h(x_1, x_2) = \alpha^{x_1} \beta^{x_2} \bmod p$ ,  $p, q$  πρώτοι,  $p = 2q + 1$ ,  $\alpha, \beta$  γεννήτορες του  $\mathbb{Z}_p^*$ .  
Είναι γνωστή ως συνάρτηση σύνοψης **Chaum-van Heijst-Pfitzman** και είναι CRHF αν ισχύει η Υπόθεση Διακριτού Λογαρίθμου στη  $\mathbb{Z}_p^*$ .

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

## Θεώρημα

*Έστω συνάρτηση σύνοψης  $h : X \rightarrow Y$  και η  $h(x) \in Y$  ακολουθεί ομοιόμορφη κατανομή πιθανότητας όταν η  $x \in X$  ακολουθεί ομοιόμορφη κατανομή. Η πιθανότητα να βρεθεί σύγκρουση μετά από τυχαία επιλογή  $x_1, x_2, \dots, x_k$  είναι περίπου  $\frac{1}{2}$  όταν  $k \cong 1.17\sqrt{n}$ , όπου  $n = |Y|$ .*

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

## Θεώρημα

Έστω συνάρτηση σύνοψης  $h : X \rightarrow Y$  και η  $h(x) \in Y$  ακολουθεί ομοιόμορφη κατανομή πιθανότητας όταν η  $x \in X$  ακολουθεί ομοιόμορφη κατανομή. Η πιθανότητα να βρεθεί σύγκρουση μετά από τυχαία επιλογή  $x_1, x_2, \dots, x_k$  είναι περίπου  $\frac{1}{2}$  όταν  $k \cong 1.17\sqrt{n}$ , όπου  $n = |Y|$ .

## Απόδειξη

$NoColl_i$ : δεν έχουμε σύγκρουση στα  $\{y_1, y_2, \dots, y_i\}$

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

## Θεώρημα

Έστω συνάρτηση σύννοψης  $h : X \rightarrow Y$  και η  $h(x) \in Y$  ακολουθεί ομοιόμορφη κατανομή πιθανότητας όταν η  $x \in X$  ακολουθεί ομοιόμορφη κατανομή. Η πιθανότητα να βρεθεί σύγκρουση μετά από τυχαία επιλογή  $x_1, x_2, \dots, x_k$  είναι περίπου  $\frac{1}{2}$  όταν  $k \cong 1.17\sqrt{n}$ , όπου  $n = |Y|$ .

## Απόδειξη

$NoColl_i$ : δεν έχουμε σύγκρουση στα  $\{y_1, y_2, \dots, y_i\}$   
Έχουμε  $NoColl_k$  αν  $NoColl_i$  για όλα τα  $i \leq k$ , δηλαδή

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

## Θεώρημα

Έστω συνάρτηση σύνοψης  $h : X \rightarrow Y$  και η  $h(x) \in Y$  ακολουθεί ομοιόμορφη κατανομή πιθανότητας όταν η  $x \in X$  ακολουθεί ομοιόμορφη κατανομή. Η πιθανότητα να βρεθεί σύγκρουση μετά από τυχαία επιλογή  $x_1, x_2, \dots, x_k$  είναι περίπου  $\frac{1}{2}$  όταν  $k \cong 1.17\sqrt{n}$ , όπου  $n = |Y|$ .

## Απόδειξη

$NoColl_i$ : δεν έχουμε σύγκρουση στα  $\{y_1, y_2, \dots, y_i\}$

Έχουμε  $NoColl_k$  αν  $NoColl_i$  για όλα τα  $i \leq k$ , δηλαδή

$$Pr[NoColl_k] = Pr[NoColl_1]Pr[NoColl_2|NoColl_1] \cdots Pr[NoColl_k|NoColl_{k-1}]$$



# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

## Θεώρημα

Έστω συνάρτηση σύνοψης  $h : X \rightarrow Y$  και η  $h(x) \in Y$  ακολουθεί ομοιόμορφη κατανομή πιθανότητας όταν η  $x \in X$  ακολουθεί ομοιόμορφη κατανομή. Η πιθανότητα να βρεθεί σύγκρουση μετά από τυχαία επιλογή  $x_1, x_2, \dots, x_k$  είναι περίπου  $\frac{1}{2}$  όταν  $k \cong 1.17\sqrt{n}$ , όπου  $n = |Y|$ .

## Απόδειξη

$NoColl_i$ : δεν έχουμε σύγκρουση στα  $\{y_1, y_2, \dots, y_i\}$

Έχουμε  $NoColl_k$  αν  $NoColl_i$  για όλα τα  $i \leq k$ , δηλαδή

$$Pr[NoColl_k] = Pr[NoColl_1]Pr[NoColl_2|NoColl_1] \cdots Pr[NoColl_k|NoColl_{k-1}]$$

- ▶  $Pr[NoColl_1] = 1$
- ▶ Αν συμβαίνει το  $NoColl_i$ , τότε η πιθανότητα να συγκρουστεί το  $y_{i+1}$  με τα προηγούμενα είναι  $\frac{i}{n}$

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$Pr[NoColl_k] = \frac{n(n-1)\dots(n-k+1)}{n^k} = \prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right)$$

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$Pr[NoColl_k] = \frac{n(n-1)\dots(n-k+1)}{n^k} = \prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right)$$

Ισχύει  $\forall x \in \mathbb{R}, 1 + x \leq e^x$ , οπότε:

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$Pr[NoColl_k] = \frac{n(n-1)\dots(n-k+1)}{n^k} = \prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right)$$

Ισχύει  $\forall x \in \mathbb{R}, 1 + x \leq e^x$ , οπότε:

$$\prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right) \leq \prod_{i=1}^{k-1} e^{-\frac{i}{n}} = e^{-\frac{\sum_{i=1}^{k-1} i}{n}} = e^{-\frac{k(k-1)}{2n}} \Rightarrow$$

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$Pr[NoColl_k] = \frac{n(n-1)\dots(n-k+1)}{n^k} = \prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right)$$

Ισχύει  $\forall x \in \mathbb{R}, 1 + x \leq e^x$ , οπότε:

$$\prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right) \leq \prod_{i=1}^{k-1} e^{-\frac{i}{n}} = e^{-\frac{\sum_{i=1}^{k-1} i}{n}} = e^{-\frac{k(k-1)}{2n}} \Rightarrow$$

$$Pr[Coll_k] \geq 1 - e^{-\frac{k(k-1)}{2n}}$$



## Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$\Pr[\text{Coll}_k] \geq 1 - e^{-\frac{k(k-1)}{2n}}$$

Για να είναι επομένως η πιθανότητα σύγκρουσης τουλάχιστον  $p$  αρκεί:

## Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$\Pr[\text{Coll}_k] \geq 1 - e^{-\frac{k(k-1)}{2n}}$$

Για να είναι επομένως η πιθανότητα σύγκρουσης τουλάχιστον  $p$  αρκεί:

$$1 - e^{-\frac{k(k-1)}{2n}} \geq p \Rightarrow \ln(1 - p) \geq -\frac{k(k-1)}{2n} \Rightarrow k^2 - k - 2n \ln \frac{1}{1-p} \geq 0$$

# Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$\Pr[\text{Coll}_k] \geq 1 - e^{-\frac{k(k-1)}{2n}}$$

Για να είναι επομένως η πιθανότητα σύγκρουσης τουλάχιστον  $p$  αρκεί:

$$1 - e^{-\frac{k(k-1)}{2n}} \geq p \Rightarrow \ln(1 - p) \geq -\frac{k(k-1)}{2n} \Rightarrow k^2 - k - 2n \ln \frac{1}{1-p} \geq 0$$

Λύνοντας ως προς  $k$ :  $k \geq 1 + \sqrt{2n \ln \frac{1}{1-p}}$



## Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$\Pr[\text{Coll}_k] \geq 1 - e^{-\frac{k(k-1)}{2n}}$$

Για να είναι επομένως η πιθανότητα σύγκρουσης τουλάχιστον  $p$  αρκεί:

$$1 - e^{-\frac{k(k-1)}{2n}} \geq p \Rightarrow \ln(1 - p) \geq -\frac{k(k-1)}{2n} \Rightarrow k^2 - k - 2n \ln \frac{1}{1-p} \geq 0$$

Λύνοντας ως προς  $k$ :  $k \geq 1 + \sqrt{2n \ln \frac{1}{1-p}}$

Για  $p = \frac{1}{2}$  προκύπτει  $k \geq 1.17\sqrt{n} + 1$ . Για  $n = 365$ ,  $k \geq 23$ . □

## Επίθεση τετραγωνικής ρίζας (Παράδοξο Γενεθλίων)

απόδειξη – συν.

$$\Pr[\text{Coll}_k] \geq 1 - e^{-\frac{k(k-1)}{2n}}$$

Για να είναι επομένως η πιθανότητα σύγκρουσης τουλάχιστον  $p$  αρκεί:

$$1 - e^{-\frac{k(k-1)}{2n}} \geq p \Rightarrow \ln(1 - p) \geq -\frac{k(k-1)}{2n} \Rightarrow k^2 - k - 2n \ln \frac{1}{1-p} \geq 0$$

Λύνοντας ως προς  $k$ :  $k \geq 1 + \sqrt{2n \ln \frac{1}{1-p}}$

Για  $p = \frac{1}{2}$  προκύπτει  $k \geq 1.17\sqrt{n} + 1$ . Για  $n = 365$ ,  $k \geq 23$ . □

Σημαντική εφαρμογή (μεταξύ άλλων): **μέθοδος παραγοντοποίησης  $\rho$**

## Επιθέσεις γενεθλίων

- ▶ Συμπέρασμα, αν  $h : \{0, 1\}^* \mapsto \{0, 1\}^l$ , τότε αν πάρω  $k = \mathcal{O}(2^{l/2})$  τυχαία στοιχεία από το  $\{0, 1\}^*$ , η πιθανότητα να έχω σύγκρουση είναι  $1/2$

## Επιθέσεις γενεθλίων

- ▶ Συμπέρασμα, αν  $h : \{0, 1\}^* \mapsto \{0, 1\}^l$ , τότε αν πάρω  $k = \mathcal{O}(2^{l/2})$  τυχαία στοιχεία από το  $\{0, 1\}^*$ , η πιθανότητα να έχω σύγκρουση είναι  $1/2$
- ▶ Διαφορά  $2^l, 2^{l/2}$  στην πράξη: για ασφάλεια 128 bits, πρέπει η συνάρτηση hash να δίνει έξοδο 256 bits.

# Επιθέσεις γενεθλίων

- ▶ Συμπέρασμα, αν  $h : \{0, 1\}^* \mapsto \{0, 1\}^l$ , τότε αν πάρω  $k = \mathcal{O}(2^{l/2})$  τυχαία στοιχεία από το  $\{0, 1\}^*$ , η πιθανότητα να έχω σύγκρουση είναι  $1/2$
- ▶ Διαφορά  $2^l, 2^{l/2}$  στην πράξη: για ασφάλεια 128 bits, πρέπει η συνάρτηση hash να δίνει έξοδο 256 bits.
- ▶ Αδυναμίες επίθεσης γενεθλίων (για τον επιτιθέμενο):
  1. τυχαίες τιμές μπορεί να είναι άχρηστες

# Επιθέσεις γενεθλίων

- ▶ Συμπέρασμα, αν  $h : \{0, 1\}^* \mapsto \{0, 1\}^l$ , τότε αν πάρω  $k = \mathcal{O}(2^{l/2})$  τυχαία στοιχεία από το  $\{0, 1\}^*$ , η πιθανότητα να έχω σύγκρουση είναι  $1/2$
- ▶ Διαφορά  $2^l, 2^{l/2}$  στην πράξη: για ασφάλεια 128 bits, πρέπει η συνάρτηση hash να δίνει έξοδο 256 bits.
- ▶ Αδυναμίες επίθεσης γενεθλίων (για τον επιτιθέμενο):
  1. τυχαίες τιμές μπορεί να είναι άχρηστες
  2. πολυπλοκότητα με ταξινόμηση  $\mathcal{O}(l 2^{l/2})$

# Επιθέσεις γενεθλίων

- ▶ Συμπέρασμα, αν  $h : \{0, 1\}^* \mapsto \{0, 1\}^l$ , τότε αν πάρω  $k = \mathcal{O}(2^{l/2})$  τυχαία στοιχεία από το  $\{0, 1\}^*$ , η πιθανότητα να έχω σύγκρουση είναι  $1/2$
- ▶ Διαφορά  $2^l, 2^{l/2}$  στην πράξη: για ασφάλεια 128 bits, πρέπει η συνάρτηση hash να δίνει έξοδο 256 bits.
- ▶ Αδυναμίες επίθεσης γενεθλίων (για τον επιτιθέμενο):
  1. τυχαίες τιμές μπορεί να είναι άχρηστες
  2. πολυπλοκότητα με ταξινόμηση  $\mathcal{O}(l 2^{l/2})$
  3. εξαιρετικά μεγάλος χώρος

# Επιθέσεις γενεθλίων με επιλεγμένα μηνύματα

- ▶ Επιλογή των μηνυμάτων:



## Επιθέσεις γενεθλίων με επιλεγμένα μηνύματα

- ▶ Επιλογή των μηνυμάτων: Οι τιμές που δίνουμε για να πετύχουμε σύγκρουση, μπορούν να έχουν σχέση μεταξύ τους, για παράδειγμα η Alice απολύεται και θέλει να βρει δύο μηνύματα  $x$  και  $x'$  έτσι ώστε  $H(x) = H(x')$ , όπου το πρώτο λέει τους λόγους της απόλυσής της, ενώ το δεύτερο επαινετικά λόγια.

## Επιθέσεις γενεθλίων με επιλεγμένα μηνύματα

- ▶ Επιλογή των μηνυμάτων: Οι τιμές που δίνουμε για να πετύχουμε σύγκρουση, μπορούν να έχουν σχέση μεταξύ τους, για παράδειγμα η Alice απολύεται και θέλει να βρει δύο μηνύματα  $x$  και  $x'$  έτσι ώστε  $H(x) = H(x')$ , όπου το πρώτο λέει τους λόγους της απόλυσής της, ενώ το δεύτερο επαινετικά λόγια.
- ▶ Φτιάχνουμε  $k = \Theta(2^{l/2})$  μηνύματα από τον πρώτο τύπο και άλλα τόσα από το δεύτερο και τις εικόνες τους.

## Επιθέσεις γενεθλίων με επιλεγμένα μηνύματα

- ▶ Επιλογή των μηνυμάτων: Οι τιμές που δίνουμε για να πετύχουμε σύγκρουση, μπορούν να έχουν σχέση μεταξύ τους, για παράδειγμα η Alice απολύεται και θέλει να βρει δύο μηνύματα  $x$  και  $x'$  έτσι ώστε  $H(x) = H(x')$ , όπου το πρώτο λέει τους λόγους της απόλυσής της, ενώ το δεύτερο επαινετικά λόγια.
- ▶ Φτιάχνουμε  $k = \Theta(2^{l/2})$  μηνύματα από τον πρώτο τύπο και άλλα τόσα από το δεύτερο και τις εικόνες τους.

“Είναι δύσκολο/απίθανο/σπάνιο να βρεις μια τόσο αποδοτική/εργατική/φιλότιμη υπάλληλο σαν την Alice. Η δουλειά της είναι καταπληκτική/ασύγκριτη/πρωτοποριακή.”

## Επιθέσεις γενεθλίων με επιλεγμένα μηνύματα

- ▶ Επιλογή των μηνυμάτων: Οι τιμές που δίνουμε για να πετύχουμε σύγκρουση, μπορούν να έχουν σχέση μεταξύ τους, για παράδειγμα η Alice απολύεται και θέλει να βρει δύο μηνύματα  $x$  και  $x'$  έτσι ώστε  $H(x) = H(x')$ , όπου το πρώτο λέει τους λόγους της απόλυσής της, ενώ το δεύτερο επαινετικά λόγια.
- ▶ Φτιάχνουμε  $k = \Theta(2^{l/2})$  μηνύματα από τον πρώτο τύπο και άλλα τόσα από το δεύτερο και τις εικόνες τους.

“Είναι δύσκολο/απίθανο/σπάνιο να βρεις μια τόσο αποδοτική/εργατική/φιλότιμη υπάλληλο σαν την Alice. Η δουλειά της είναι καταπληκτική/ασύγκριτη/πρωτοποριακή.”

- ▶ Από παράδοξο γενεθλίων έχουμε καλή πιθανότητα να πετύχουμε σύγκρουση μεταξύ μηνυμάτων των δύο τύπων.

## Επιθέσεις γενεθλίων με επιλεγμένα μηνύματα

- ▶ Επιλογή των μηνυμάτων: Οι τιμές που δίνουμε για να πετύχουμε σύγκρουση, μπορούν να έχουν σχέση μεταξύ τους, για παράδειγμα η Alice απολύεται και θέλει να βρει δύο μηνύματα  $x$  και  $x'$  έτσι ώστε  $H(x) = H(x')$ , όπου το πρώτο λέει τους λόγους της απόλυσής της, ενώ το δεύτερο επαινετικά λόγια.
- ▶ Φτιάχνουμε  $k = \Theta(2^{l/2})$  μηνύματα από τον πρώτο τύπο και άλλα τόσα από το δεύτερο και τις εικόνες τους.

“Είναι δύσκολο/απίθανο/σπάνιο να βρεις μια τόσο αποδοτική/εργατική/φιλότιμη υπάλληλο σαν την Alice. Η δουλειά της είναι καταπληκτική/ασύγκριτη/πρωτοποριακή.”

- ▶ Από παράδοξο γενεθλίων έχουμε καλή πιθανότητα να πετύχουμε σύγκρουση μεταξύ μηνυμάτων των δύο τύπων.
- ▶ Όμως: το πρόβλημα του μεγάλου χώρου παραμένει.

# Βελτιωμένες επιθέσεις γενεθλίων

Μια πρώτη ιδέα:

Επίθεση γενεθλίων γραμμικού πλήθους συγκρίσεων

# Βελτιωμένες επιθέσεις γενεθλίων

Μια πρώτη ιδέα:

## Επίθεση γενεθλίων γραμμικού πλήθους συγκρίσεων

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i := 1, 2, \dots$  υπολόγισε  $x_i = H(x_{i-1})$

# Βελτιωμένες επιθέσεις γενεθλίων

Μια πρώτη ιδέα:

## Επίθεση γενεθλίων γραμμικού πλήθους συγκρίσεων

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i := 1, 2, \dots$  υπολόγισε  $x_i = H(x_{i-1})$
2. Σύγκρινε  $x_i$  με  $x_{2^{\lfloor \log i \rfloor}}$ : **γραμμικό πλήθος συγκρίσεων**.



# Βελτιωμένες επιθέσεις γενεθλίων

Μια πρώτη ιδέα:

## Επίθεση γενεθλίων γραμμικού πλήθους συγκρίσεων

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i := 1, 2, \dots$  υπολόγισε  $x_i = H(x_{i-1})$
2. Σύγκρινε  $x_i$  με  $x_{2^{\lfloor \log i \rfloor}}$ : γραμμικό πλήθος συγκρίσεων.
3. Εξήγηση:  $x_i = x_j \Rightarrow \forall k \geq 1, x_{i+k} = x_{j+k}$ .

# Βελτιωμένες επιθέσεις γενεθλίων

Μια πρώτη ιδέα:

## Επίθεση γενεθλίων γραμμικού πλήθους συγκρίσεων

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i := 1, 2, \dots$  υπολόγισε  $x_i = H(x_{i-1})$
2. Σύγκρινε  $x_i$  με  $x_{2^{\lfloor \log i \rfloor}}$ : **γραμμικό πλήθος συγκρίσεων**.
3. Εξήγηση:  $x_i = x_j \Rightarrow \forall k \geq 1, x_{i+k} = x_{j+k}$ .
4. Η σύγκρουση  $x_i = x_j$  θα εντοπιστεί το αργότερο στην θέση  $x_{4j}$ . (Γιατί; Άσκηση!)

# Βελτιωμένες επιθέσεις γενεθλίων

Μια πρώτη ιδέα:

## Επίθεση γενεθλίων γραμμικού πλήθους συγκρίσεων

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i := 1, 2, \dots$  υπολόγισε  $x_i = H(x_{i-1})$
2. Σύγκρινε  $x_i$  με  $x_{2^{\lfloor \log i \rfloor}}$ : γραμμικό πλήθος συγκρίσεων.
3. Εξήγηση:  $x_i = x_j \Rightarrow \forall k \geq 1, x_{i+k} = x_{j+k}$ .
4. Η σύγκρουση  $x_i = x_j$  θα εντοπιστεί το αργότερο στην θέση  $x_{4j}$ . (Γιατί; Άσκηση!)

Μπορεί να υλοποιηθεί σε σταθερό χώρο: κάθε φορά χρειαζόμαστε μόνο τα  $x_i, x_{2^{\lfloor \log i \rfloor}}$ .

# Βελτιωμένες επιθέσεις γενεθλίων

Μια πρώτη ιδέα:

## Επίθεση γενεθλίων γραμμικού πλήθους συγκρίσεων

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i := 1, 2, \dots$  υπολόγισε  $x_i = H(x_{i-1})$
2. Σύγκρινε  $x_i$  με  $x_{2^{\lfloor \log i \rfloor}}$ : **γραμμικό πλήθος συγκρίσεων**.
3. Εξήγηση:  $x_i = x_j \Rightarrow \forall k \geq 1, x_{i+k} = x_{j+k}$ .
4. Η σύγκρουση  $x_i = x_j$  θα εντοπιστεί το αργότερο στην θέση  $x_{4j}$ . (Γιατί; Άσκηση!)

Μπορεί να υλοποιηθεί σε σταθερό χώρο: κάθε φορά χρειαζόμαστε μόνο τα  $x_i, x_{2^{\lfloor \log i \rfloor}}$ .

Μια ακόμη ταχύτερη επίθεση σταθερού χρόνου περιγράφεται παρακάτω.

# Βελτιωμένες επιθέσεις γενεθλίων

Μια πρώτη ιδέα:

## Επίθεση γενεθλίων γραμμικού πλήθους συγκρίσεων

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i := 1, 2, \dots$  υπολόγισε  $x_i = H(x_{i-1})$
2. Σύγκρινε  $x_i$  με  $x_{2^{\lfloor \log i \rfloor}}$ : **γραμμικό πλήθος συγκρίσεων**.
3. Εξήγηση:  $x_i = x_j \Rightarrow \forall k \geq 1, x_{i+k} = x_{j+k}$ .
4. Η σύγκρουση  $x_i = x_j$  θα εντοπιστεί το αργότερο στην θέση  $x_{4j}$ . (Γιατί; Άσκηση!)

Μπορεί να υλοποιηθεί σε σταθερό χώρο: κάθε φορά χρειαζόμαστε μόνο τα  $x_i, x_{2^{\lfloor \log i \rfloor}}$ .

Μια ακόμη ταχύτερη επίθεση σταθερού χρόνου περιγράφεται παρακάτω.

Οι ίδιες ιδέες εφαρμόζονται και στην **μέθοδο παραγοντοποίησης  $\rho$** .

# Βελτιωμένες επιθέσεις γενεθλίων

## Επίθεση γενεθλίων σταθερού χώρου

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i > 0$  υπολόγισε  $x_i = H(x_{i-1})$  και  $x_{2i} = H(H(x_{2(i-1)}))$

# Βελτιωμένες επιθέσεις γενεθλίων

## Επίθεση γενεθλίων σταθερού χώρου

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i > 0$  υπολόγισε  $x_i = H(x_{i-1})$  και  $x_{2i} = H(H(x_{2(i-1)}))$
2. Σε κάθε επανάληψη έλεγξε  $x_i \stackrel{?}{=} x_{2i}$ . Εάν ίσα, τότε ψάξε από το  $x_0$  έως το  $x_{i-1}$  για σύγκρουση.

# Βελτιωμένες επιθέσεις γενεθλίων

## Επίθεση γενεθλίων σταθερού χώρου

1. Πάρε τυχαία αρχική τιμή  $x_0$  και για  $i > 0$  υπολόγισε  $x_i = H(x_{i-1})$  και  $x_{2i} = H(H(x_{2(i-1)}))$
2. Σε κάθε επανάληψη έλεγξε  $x_i \stackrel{?}{=} x_{2i}$ . Εάν ίσα, τότε ψάξε από το  $x_0$  έως το  $x_{i-1}$  για σύγκρουση.
3. Βρες το μικρότερο  $j$  ώστε  $x_j = x_{j+i}$  και τύπωσε τα  $x_{j-1}, x_{j+i-1}$



# Βελτιωμένες επιθέσεις γενεθλίων

## Αλγόριθμος Επίθεσης Γενεθλίων Σταθερού Χώρου

**Είσοδος:** Συνάρτηση σύνοψης  $H : \{0, 1\}^* \mapsto \{0, 1\}^l$

**Έξοδος:**  $x \neq x'$ , με  $H(x) = H(x')$

$x_0 \leftarrow \{0, 1\}^{l+1}, x' = x = x_0$

**for**  $i = 1, 2, \dots$  **do** :

$x = H(x)$

$x' = H(H(x'))$

// τώρα  $x = H^{(i)}(x_0)$  και  $x' = H^{(2i)}(x_0)$

**if**  $x = x'$  **break**

$x' = x, x = x_0$

**for**  $j = 1 \rightarrow i$  **do** :

**if**  $H(x) == H(x')$  **return**  $x, x'$

**else**  $x = H(x), x' = H(x')$

// τώρα  $x = H^{(j)}(x_0)$  και  $x' = H^{(j+i)}(x_0)$

► Σταθερός χώρος: δύο στοιχεία  $x_i, x_{2i}$ .

# Βελτιωμένες επιθέσεις γενεθλίων

## Αλγόριθμος Επίθεσης Γενεθλίων Σταθερού Χώρου

**Είσοδος:** Συνάρτηση σύνοψης  $H : \{0, 1\}^* \mapsto \{0, 1\}^l$

**Έξοδος:**  $x \neq x'$ , με  $H(x) = H(x')$

$x_0 \leftarrow \{0, 1\}^{l+1}, x' = x = x_0$

**for**  $i = 1, 2, \dots$  **do** :

$x = H(x)$

$x' = H(H(x'))$

// τώρα  $x = H^{(i)}(x_0)$  και  $x' = H^{(2i)}(x_0)$

**if**  $x = x'$  **break**

$x' = x, x = x_0$

**for**  $j = 1 \rightarrow i$  **do** :

**if**  $H(x) == H(x')$  **return**  $x, x'$

**else**  $x = H(x), x' = H(x')$

// τώρα  $x = H^{(j)}(x_0)$  και  $x' = H^{(j+i)}(x_0)$

- ▶ Σταθερός χώρος: δύο στοιχεία  $x_i, x_{2i}$ .
- ▶ Επιτυχία με πιθανότητα  $\geq 1/2$  σε  $\Theta(2^{l/2})$  βήματα.

# Βελτιωμένες επιθέσεις γενεθλίων

## Λήμμα

Έστω  $x_1, \dots, x_q$  η ακολουθία τιμών με  $x_m = H(x_{m-1})$ . Αν  $x_I = x_J$ , με  $1 \leq I < J \leq q$ , τότε υπάρχει ένα  $i < J$  τέτοιο ώστε  $x_i = x_{2i}$ .

## Απόδειξη.

Η ακολουθία  $x_I, x_{I+1}, \dots$  επαναλαμβάνεται με περίοδο  $\Delta = J - I$ . Δηλ. για κάθε  $i \geq I$  και  $k = 0, 1, \dots$ , έχουμε  $x_i = x_{i+k\Delta}$ . Έστω  $i$  το μικρότερο πολλαπλάσιο του  $\Delta$  που είναι μεγαλύτερο ή ίσο του  $I$ . Έχουμε  $i < J$  (γιατί;). Επειδή  $i \geq I$ , το  $2i$  είναι πολλαπλάσιο του  $\Delta$ , έχουμε  $x_i = x_{2i}$ .  $\square$

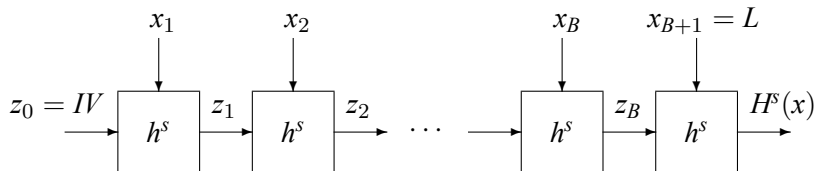
# Επέκταση συναρτήσεων σύνοψης

## Merkle-Damgård Hash Function Extension

Έστω  $h$  μια συνάρτηση σύνοψης που απεικονίζει είσοδο μήκους  $2n$  σε έξοδο μήκους  $n$ . Κατασκευάζουμε μια συνάρτηση σύνοψης  $H$  μεταβλητού μήκους ως εξής:

- ▶  $H$ : με είσοδο ένα string  $x \in \{0, 1\}^*$  μήκους  $L \leq 2^n$ :
  1. Θέσε  $B = \lceil \frac{L}{n} \rceil$  (πλήθος block του  $x$ ). (Πρόσθεσε μηδενικά στο  $x$  ώστε το μήκος να είναι πολλαπλάσιο του  $n$ ) ( $x = x_1, \dots, x_B$ ). Θέσε  $x_{B+1} = L$  (το  $L$  δυαδική αναπαράσταση)
  2. Θέσε  $z_0 = 0^n$  (Initialization vector)
  3. Για  $i = 1, \dots, B + 1$ , υπολόγισε το  $z_i = h(z_{i-1} || x_i)$
  4. Έξοδος:  $z_{B+1}$

# Κατασκευή Merkle-Damgård



Σχήμα: Merkle-Damgård

# Επέκταση συναρτήσεων σύνοψης

## Θεώρημα

Αν η συνάρτηση σύνοψης  $h$  είναι *collision resistant*, τότε και η  $H$  που κατασκευάζεται με τη μέθοδο Merkle-Damgård είναι επίσης *collision resistant*.

## Απόδειξη.

Έστω  $x' = x'_1 \dots x'_{B'} \neq x = x_1 \dots x_B : x'_{B'+1} = L', x_{B+1} = L$ , με  $H(x) = H(x')$ .

Τότε έχουμε δύο περιπτώσεις:

1.  $L \neq L'$ , οπότε στο τελευταίο βήμα είναι  $z_{B+1} = h(z_B || L)$  και  $z'_{B'+1} = h(z'_{B'} || L')$ , άρα σύγκρουση στην  $h$ , αφού τα strings  $z_B || L$  και  $z'_{B'} || L'$  είναι διαφορετικά.
2.  $L = L'$ , οπότε  $B = B'$ . Έστω  $z_0, \dots, z_{B+1}$  οι τιμές που παράγονται από την  $H(x)$ , και  $I_i = z_{i-1} || x_i$ ,  $I_{B+2} = z_{B+1}$ . Έστω  $N$  ο μεγαλύτερος δείκτης, ώστε  $I_N \neq I'_N$  (υπάρχει). Αφού ο  $N$  μέγιστος, έχουμε  $I_{N+1} = I'_{N+1}$  (ειδικά  $z_N = z'_N$ ). Αλλά τότε τα  $I_N, I'_N$  είναι σύγκρουση στην  $h$ .



## Συναρτήσεις σύνοψης: μερικές ακόμη παρατηρήσεις

- ▶ Οι πιο διάσημες συναρτήσεις, MD5 και SHA-1 στηρίζονται σε πράξεις που θυμίζουν συμμετρική κρυπτογραφία (rotation, XOR, πρόσθεση mod $2^{32}$ , δυαδικές πράξεις) και στην κατασκευή Merkle-Damgård.

## Συναρτήσεις σύνοψης: μερικές ακόμη παρατηρήσεις

- ▶ Οι πιο διάσημες συναρτήσεις, MD5 και SHA-1 στηρίζονται σε πράξεις που θυμίζουν συμμετρική κρυπτογραφία (rotation, XOR, πρόσθεση  $\text{mod } 2^{32}$ , δυαδικές πράξεις) και στην κατασκευή Merkle-Damgård.
- ▶ Υπέστησαν εντατικές επιθέσεις (επίθεση γενεθλίων κ.ά.). Η MD5 δεν θεωρείται πλέον ασφαλής, η SHA-1 αντικαταστάθηκε από την (οικογένεια) SHA-2, ενώ έχει αναπτυχθεί και η SHA-3 (Keccak).



# Δένδρα Merkle

- ▶ Ένας χρήστης θέλει να ανεβάσει αρχείο  $x$  σε έναν server.

# Δένδρα Merkle

- ▶ Ένας χρήστης θέλει να ανεβάσει αρχείο  $x$  σε έναν server.
- ▶ Όταν το κατεβάσει, θέλει να ελέγξει αν είναι το ίδιο.

# Δένδρα Merkle

- ▶ Ένας χρήστης θέλει να ανεβάσει αρχείο  $x$  σε έναν server.
- ▶ Όταν το κατεβάσει, θέλει να ελέγξει αν είναι το ίδιο.
- ▶ Λύση: αποθηκεύει τοπικά το  $h = H(x)$ , και όταν καταβάζει το ζητούμενο αρχείο  $x'$  ελέγχει  $H(x') \stackrel{?}{=} h$ .

# Δένδρα Merkle

- ▶ Ένας χρήστης θέλει να ανεβάσει αρχείο  $x$  σε έναν server.
- ▶ Όταν το κατεβάσει, θέλει να ελέγξει αν είναι το ίδιο.
- ▶ Λύση: αποθηκεύει τοπικά το  $h = H(x)$ , και όταν καταβάζει το ζητούμενο αρχείο  $x'$  ελέγχει  $H(x') \stackrel{?}{=} h$ .
- ▶ Αν έχει πολλά αρχεία;

# Δένδρα Merkle

- ▶ Ένας χρήστης θέλει να ανεβάσει αρχείο  $x$  σε έναν server.
- ▶ Όταν το κατεβάσει, θέλει να ελέγξει αν είναι το ίδιο.
- ▶ Λύση: αποθηκεύει τοπικά το  $h = H(x)$ , και όταν καταβάζει το ζητούμενο αρχείο  $x'$  ελέγχει  $H(x') \stackrel{?}{=} h$ .
- ▶ Αν έχει πολλά αρχεία; Υπάρχουν διάφορες λύσεις.

# Δένδρα Merkle

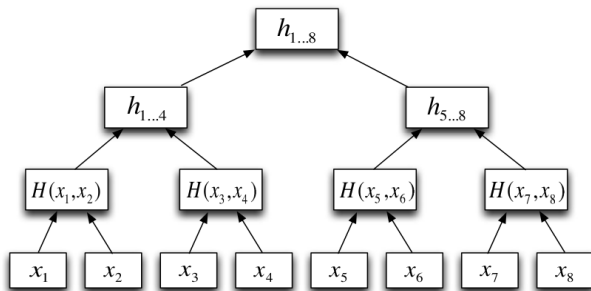
- ▶ Δένδρο Merkle με είσοδο  $x_1, x_2, \dots, x_t$ :

## Δένδρα Merkle

- ▶ Δένδρο Merkle με είσοδο  $x_1, x_2, \dots, x_t$ : ένα δυαδικό δένδρο με φύλλα τα  $x_1, \dots, x_t$  και εσωτερικούς κόμβους τις τιμές σύνοψης των παιδιών του.

# Δένδρα Merkle

- ▶ Δένδρο Merkle με είσοδο  $x_1, x_2, \dots, x_t$ : ένα δυαδικό δένδρο με φύλλα τα  $x_1, \dots, x_t$  και εσωτερικούς κόμβους τις τιμές σύνοψης των παιδιών του.

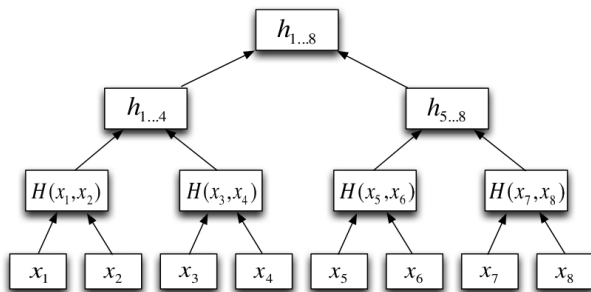


Σχήμα: Δένδρο Merkle



## Δένδρα Merkle

- ▶ Δένδρο Merkle με είσοδο  $x_1, x_2, \dots, x_t$ : ένα δυαδικό δένδρο με φύλλα τα  $x_1, \dots, x_t$  και εσωτερικούς κόμβους τις τιμές σύνοψης των παιδιών του.



Σχήμα: Δένδρο Merkle

- ▶ Για δοσμένη συνάρτηση σύνοψης  $H$ , συμβολίζουμε με  $\mathcal{MT}_t$  τη συνάρτηση που με είσοδο τα  $x_1, \dots, x_t$ , υπολογίζει το δένδρο Merkle και τη ρίζα του δένδρου.

# Δένδρα Merkle

# Δένδρα Merkle

## Θεώρημα

*Έστω  $(Gen_H, H)$  συνάρτηση σύνοψης ελεύθερη συγκρούσεων. Τότε και η  $(Gen_H, MT_t)$  είναι ελεύθερη συγκρούσεων για κάθε σταθερό  $t$ .*

# Δένδρα Merkle

- ▶ Ο χρήστης υπολογίζει το  $h = \mathcal{MT}_t(x_1, \dots, x_t)$ , ανεβάζει τα  $x_1, \dots, x_t$  στον server και αποθηκεύει το  $h$  (και το  $t$ )

# Δένδρα Merkle

- ▶ Ο χρήστης υπολογίζει το  $h = \mathcal{MT}_t(x_1, \dots, x_t)$ , ανεβάζει τα  $x_1, \dots, x_t$  στον server και αποθηκεύει το  $h$  (και το  $t$ )
- ▶ Όταν ο χρήστης θέλει το  $i$ -οστό αρχείο, ο server του στέλνει το  $x_i$  μαζί με μια “απόδειξη”  $\pi_i$  ότι είναι το σωστό αρχείο

# Δένδρα Merkle

- ▶ Ο χρήστης υπολογίζει το  $h = \mathcal{MT}_t(x_1, \dots, x_t)$ , ανεβάζει τα  $x_1, \dots, x_t$  στον server και αποθηκεύει το  $h$  (και το  $t$ )
- ▶ Όταν ο χρήστης θέλει το  $i$ -οστό αρχείο, ο server του στέλνει το  $x_i$  μαζί με μια “απόδειξη”  $\pi_i$  ότι είναι το σωστό αρχείο
- ▶ Η απόδειξη αποτελείται από τις τιμές που είναι γειτονικές στο μονοπάτι από το  $x_i$  προς τη ρίζα.

# Δένδρα Merkle

- ▶ Ο χρήστης υπολογίζει το  $h = \mathcal{MT}_t(x_1, \dots, x_t)$ , ανεβάζει τα  $x_1, \dots, x_t$  στον server και αποθηκεύει το  $h$  (και το  $t$ )
- ▶ Όταν ο χρήστης θέλει το  $i$ -οστό αρχείο, ο server του στέλνει το  $x_i$  μαζί με μια “απόδειξη”  $\pi_i$  ότι είναι το σωστό αρχείο
- ▶ Η απόδειξη αποτελείται από τις τιμές που είναι γειτονικές στο μονοπάτι από το  $x_i$  προς τη ρίζα.

**Παράδειγμα** Έστω ότι ζητάει το  $x_3$ . Τότε ο server του στέλνει το  $x_3$  μαζί και τα  $x_4, h_{1..2}, h_{5..8}$

# Δένδρα Merkle

- ▶ Αν η  $H$  είναι ελεύθερη συγκρούσεων, τότε είναι αδύνατο ο server να στείλει ψεύτικο αρχείο (και απόδειξη) που να επαληθεύεται.



# Δένδρα Merkle

- ▶ Αν η  $H$  είναι ελεύθερη συγκρούσεων, τότε είναι αδύνατο ο server να στείλει ψεύτικο αρχείο (και απόδειξη) που να επαληθεύεται.
- ▶ Ο χρήστης χρειάζεται σταθερό χώρο και  $\mathcal{O}(\log t)$  επικοινωνία με τον server για να πάρει το αρχείο.

# Δένδρα Merkle

- ▶ Αν η  $H$  είναι ελεύθερη συγκρούσεων, τότε είναι αδύνατο ο server να στείλει ψεύτικο αρχείο (και απόδειξη) που να επαληθεύεται.
- ▶ Ο χρήστης χρειάζεται σταθερό χώρο και  $\mathcal{O}(\log t)$  επικοινωνία με τον server για να πάρει το αρχείο.
- ▶ Σημείωση: Συνήθως έχουμε τις hash τιμές των  $x_i$  στα φύλλα του δένδρου.

## Χρήσεις συναρτήσεων σύνοψης

- ▶ Ψηφιακές υπογραφές. Σε συνδυασμό με αλγόριθμο υπογραφής, για επιτάχυνση της διαδικασίας. Παραδείγματα: MD5, που χρησιμοποιείται με RSA στο PGP, SHA-1 (τώρα SHA-2), που χρησιμοποιείται στο DSS (Digital Signature Standard), κ.ά.
- ▶ Έλεγχος γνησιότητας μηνύματος – αυθεντικοποίηση (με συμμετρικό κλειδί): keyed hash functions, π.χ. HMAC.
- ▶ Ακεραιότητα δεδομένων (με ή χωρίς κλειδί).

# Χρήσεις συναρτήσεων σύνοψης

- ▶ Ψηφιακές υπογραφές. Σε συνδυασμό με αλγόριθμο υπογραφής, για επιτάχυνση της διαδικασίας. Παραδείγματα: MD5, που χρησιμοποιείται με RSA στο PGP, SHA-1 (τώρα SHA-2), που χρησιμοποιείται στο DSS (Digital Signature Standard), κ.ά.
- ▶ Έλεγχος γνησιότητας μηνύματος – αυθεντικοποίηση (με συμμετρικό κλειδί): keyed hash functions, π.χ. HMAC.
- ▶ Ακεραιότητα δεδομένων (με ή χωρίς κλειδί).
- ▶ **Bitcoin**: blockchain, proof of work, **Merkle trees**.

# Χρήσεις συναρτήσεων σύνοψης

- ▶ Ψηφιακές υπογραφές. Σε συνδυασμό με αλγόριθμο υπογραφής, για επιτάχυνση της διαδικασίας. Παραδείγματα: MD5, που χρησιμοποιείται με RSA στο PGP, SHA-1 (τώρα SHA-2), που χρησιμοποιείται στο DSS (Digital Signature Standard), κ.ά.
- ▶ Έλεγχος γνησιότητας μηνύματος – αυθεντικοποίηση (με συμμετρικό κλειδί): keyed hash functions, π.χ. HMAC.
- ▶ Ακεραιότητα δεδομένων (με ή χωρίς κλειδί).
- ▶ **Bitcoin**: blockchain, proof of work, **Merkle trees**.
- ▶ Γεννήτριες ψευδοτυχαίων αριθμών (με random seed + counter).
- ▶ Stream ciphers, αλλά και block ciphers (SHACAL).