

# Ψηφιακές Υπογραφές

---

Παναγιώτης Γροντάς - Άρης Παγουρτζής

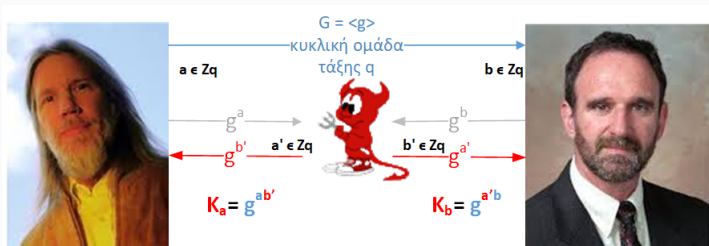
ΕΜΠ - Κρυπτογραφία

- Ορισμός - Μοντελοποίηση Ασφάλειας
- Ψηφιακές Υπογραφές RSA
- Επιθέσεις - Παραλλαγές
- Το μοντέλο του τυχαίου μαντείου
- Ψηφιακές Υπογραφές ElGamal-DSA
- Υποδομή Δημοσίου Κλειδιού

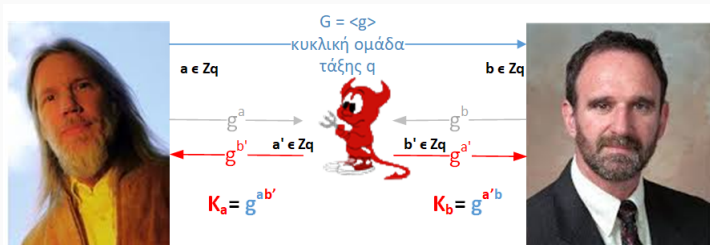
# Εισαγωγή

---

# Εισαγωγή - Το πρόβλημα



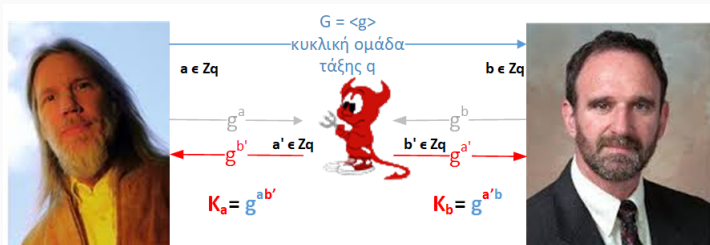
# Εισαγωγή - Το πρόβλημα



Αποφυγή MITM attacks σε DHKE

- **Ακεραιότητα:** Το μήνυμα είναι αυτό που έστειλε ο αποστολέας
- **Αυθεντικοποίηση:** Το μήνυμα το έστειλε αυτός που φαίνεται ως αποστολέας

# Εισαγωγή - Το πρόβλημα



Αποφυγή MITM attacks σε DHKE

- **Ακεραιότητα:** Το μήνυμα είναι αυτό που έστειλε ο αποστολέας
- **Αυθεντικοποίηση:** Το μήνυμα το έστειλε αυτός που φαίνεται ως αποστολέας

Μία λύση: MACs **Μειονεκτήματα** συμμετρικής κρυπτογραφίας

# Ψηφιακές υπογραφές-Ασύμμετρα MACs

- Ο αποστολέας (υπογράφων  $S$ ) εκτελεί αλγόριθμο **KeyGen** και παράγει τα  $(key_{sign}, key_{ver})$ 
  - Το κλειδί επαλήθευσης πρέπει να είναι δημόσιο
  - Το κλειδί υπογραφής πρέπει να διατηρείται μυστικό
- Δημοσιοποιεί το κλειδί επαλήθευσης (web site, κατάλογο)

# Ψηφιακές υπογραφές-Ασύμμετρα MACs

- Ο αποστολέας (υπογράφων  $S$ ) εκτελεί αλγόριθμο **KeyGen** και παράγει τα  $(key_{sign}, key_{ver})$ 
  - Το κλειδί επαλήθευσης πρέπει να είναι δημόσιο
  - Το κλειδί υπογραφής πρέπει να διατηρείται μυστικό
- Δημοσιοποιεί το κλειδί επαλήθευσης (web site, κατάλογο)
- Πριν την αποστολή 'υπογράφει το μήνυμα' (με το  $key_{sign}$ ) παράγοντας την υπογραφή  $\sigma$



# Ψηφιακές υπογραφές-Ασύμμετρα MACs

- Ο αποστολέας (υπογράφων  $S$ ) εκτελεί αλγόριθμο **KeyGen** και παράγει τα  $(key_{sign}, key_{ver})$ 
  - Το κλειδί επαλήθευσης πρέπει να είναι δημόσιο
  - Το κλειδί υπογραφής πρέπει να διατηρείται μυστικό
- Δημοσιοποιεί το κλειδί επαλήθευσης (web site, κατάλογο)
- Πριν την αποστολή 'υπογράφει το μήνυμα' (με το  $key_{sign}$ ) παράγοντας την υπογραφή  $\sigma$
- Αποστέλλει το ζεύγος  $(m, \sigma)$ 
  - Η υπογραφή εξαρτάται από το μήνυμα
  - Η υπογραφή είναι άχρηστη χωρίς το μήνυμα

# Ψηφιακές υπογραφές-Ασύμμετρα MACs

- Ο αποστολέας (υπογράφων  $S$ ) εκτελεί αλγόριθμο **KeyGen** και παράγει τα  $(key_{sign}, key_{ver})$ 
  - Το κλειδί επαλήθευσης πρέπει να είναι δημόσιο
  - Το κλειδί υπογραφής πρέπει να διατηρείται μυστικό
- Δημοσιοποιεί το κλειδί επαλήθευσης (web site, κατάλογο)
- Πριν την αποστολή 'υπογράφει το μήνυμα' (με το  $key_{sign}$ ) παράγοντας την υπογραφή  $\sigma$
- Αποστέλλει το ζεύγος  $(m, \sigma)$ 
  - Η υπογραφή εξαρτάται από το μήνυμα
  - Η υπογραφή είναι άχρηστη χωρίς το μήνυμα
- Ο παραλήπτης (επαληθεύων  $V$ ) ελέγχει αν η υπογραφή που έλαβε είναι έγκυρη (με το  $key_{ver}$ )

- Εύκολη διανομή κλειδιού

- Εύκολη διανομή κλειδιού
- Δημόσια Επαληθευσιμότητα
  - Δεν επαληθεύει μόνο ο παραλήπτης
  - Δημόσιο κλειδί: Μπορεί να επαληθεύσει οποιοσδήποτε

- Εύκολη διανομή κλειδιού
- Δημόσια Επαληθευσιμότητα
  - Δεν επαληθεύει μόνο ο παραλήπτης
  - Δημόσιο κλειδί: Μπορεί να επαληθεύσει οποιοσδήποτε
- Μη αποκήρυξη (non repudiation)
  - Αντιμετώπιση εσωτερικού αντίπαλου που προσπαθεί να αρνηθεί τις υπογραφές του
  - Μαθηματική σχέση κλειδιών υπογραφής - επαλήθευσης

- Εύκολη διανομή κλειδιού
- Δημόσια Επαληθευσιμότητα
  - Δεν επαληθεύει μόνο ο παραλήπτης
  - Δημόσιο κλειδί: Μπορεί να επαληθεύσει οποιοσδήποτε
- Μη αποκήρυξη (non repudiation)
  - Αντιμετώπιση εσωτερικού αντίπαλου που προσπαθεί να αρνηθεί τις υπογραφές του
  - Μαθηματική σχέση κλειδιών υπογραφής - επαλήθευσης
- Επιπλέον λειτουργίες
  - Αυθεντικοποίηση χρηστών (λόγω κατοχής του ιδιωτικού κλειδού)

- Εύκολη διανομή κλειδιού
- Δημόσια Επαληθευσιμότητα
  - Δεν επαληθεύει μόνο ο παραλήπτης
  - Δημόσιο κλειδί: Μπορεί να επαληθεύσει οποιοσδήποτε
- Μη αποκήρυξη (non repudiation)
  - Αντιμετώπιση εσωτερικού αντίπαλου που προσπαθεί να αρνηθεί τις υπογραφές του
  - Μαθηματική σχέση κλειδιών υπογραφής - επαλήθευσης
- Επιπλέον λειτουργίες
  - Αυθεντικοποίηση χρηστών (λόγω κατοχής του ιδιωτικού κλειδού)
  - Αωνυμία (τυφλές υπογραφές)

- Εύκολη διανομή κλειδιού
- Δημόσια Επαληθευσιμότητα
  - Δεν επαληθεύει μόνο ο παραλήπτης
  - Δημόσιο κλειδί: Μπορεί να επαληθεύσει οποιοσδήποτε
- Μη αποκήρυξη (non repudiation)
  - Αντιμετώπιση εσωτερικού αντίπαλου που προσπαθεί να αρνηθεί τις υπογραφές του
  - Μαθηματική σχέση κλειδιών υπογραφής - επαλήθευσης
- Επιπλέον λειτουργίες
  - Αυθεντικοποίηση χρηστών (λόγω κατοχής του ιδιωτικού κλειδού)
  - Ανωνυμία (τυφλές υπογραφές)
  - Αντιπροσωπεία από ομάδα (ομαδικές υπογραφές)
  - ...



Λύσαμε τα πρόβληματα **διανομής** κλειδιού, **αυθεντικότητας** και **ακεραιότητας** μηνύματος

Δημιουργήσαμε το πρόβλημα **αυθεντικότητας** κλειδιού

- Πώς είμαστε σίγουροι πως το ζεύγος κλειδιών αντιστοιχεί όντως στον  $S$ ;

Λύσαμε τα πρόβλήματα **διανομής** κλειδιού, **αυθεντικότητας** και **ακεραιότητας** μηνύματος

Δημιουργήσαμε το πρόβλημα **αυθεντικότητας** κλειδιού

- Πώς είμαστε σίγουροι πως το ζεύγος κλειδιών αντιστοιχεί όντως στον  $S$ ;
- Πώς είμαστε σίγουροι πως το  $key_{sign}$  ήταν στην κατοχή του  $S$  κατά τη δημιουργία της υπογραφής;

Λύσαμε τα πρόβληματα **διανομής** κλειδιού, **αυθεντικότητας** και **ακεραιότητας** μηνύματος

Δημιουργήσαμε το πρόβλημα **αυθεντικότητας** κλειδιού

- Πώς είμαστε σίγουροι πως το ζεύγος κλειδιών αντιστοιχεί όντως στον  $S$ ;
- Πώς είμαστε σίγουροι πως το  $key_{sign}$  ήταν στην κατοχή του  $S$  κατά τη δημιουργία της υπογραφής;

Μαθηματικές και μη λύσεις

## Σχήμα Υπογραφής

Μια τριάδα από αλγόριθμους

- $\text{KeyGen}(1^\lambda) = (key_{sign}, key_{ver})$

## Σχήμα Υπογραφής

Μια τριάδα από αλγόριθμους

- $\text{KeyGen}(1^\lambda) = (\text{key}_{\text{sign}}, \text{key}_{\text{ver}})$
- $\text{Sign}(\text{key}_{\text{sign}}, m) = \sigma, \quad m \in \{0, 1\}^*$

## Σχήμα Υπογραφής

Μια τριάδα από αλγόριθμους

- $\text{KeyGen}(1^\lambda) = (key_{sign}, key_{ver})$
- $\text{Sign}(key_{sign}, m) = \sigma, \quad m \in \{0, 1\}^*$
- $\text{Verify}(key_{ver}, m, \sigma) \in \{0, 1\}$

## Σχήμα Υπογραφής

Μια τριάδα από αλγόριθμους

- $\text{KeyGen}(1^\lambda) = (key_{sign}, key_{ver})$
- $\text{Sign}(key_{sign}, m) = \sigma, \quad m \in \{0, 1\}^*$
- $\text{Verify}(key_{ver}, m, \sigma) \in \{0, 1\}$

## Ορθότητα

$$\text{Verify}(key_{ver}, m, \text{Sign}(key_{sign}, m)) = 1$$

## Σχήμα Υπογραφής

Μια τριάδα από αλγόριθμους

- $\text{KeyGen}(1^\lambda) = (key_{sign}, key_{ver})$
- $\text{Sign}(key_{sign}, m) = \sigma, \quad m \in \{0, 1\}^*$
- $\text{Verify}(key_{ver}, m, \sigma) \in \{0, 1\}$

## Ορθότητα

$\text{Verify}(key_{ver}, m, \text{Sign}(key_{sign}, m)) = 1$

**Έγκυρες** υπογραφές: ικανοποιούν την απαίτηση της ορθότητας



## Πλαστογραφία(Forgery)

Ο  $\mathcal{A}$  με δεδομένα το δημόσιο κλειδί επαλήθευσης και ένα μήνυμα παράγει μια έγκυρη υπογραφή χωρίς την συμμετοχή του  $S$ .

## Πλαστογραφία(Forgery)

Ο  $\mathcal{A}$  με δεδομένα το δημόσιο κλειδί επαλήθευσης και ένα μήνυμα παράγει μια έγκυρη υπογραφή χωρίς την συμμετοχή του  $S$ .

## Είδη Επιθέσεων

- **Καθολική πλαστογράφηση:** Ο  $\mathcal{A}$  μπορεί να παράγει έγκυρες υπογραφές σε όποιο μήνυμα θέλει ( $\Leftrightarrow$  κατοχή ιδιωτικού κλειδιού)

## Πλαστογραφία(Forgery)

Ο  $\mathcal{A}$  με δεδομένα το δημόσιο κλειδί επαλήθευσης και ένα μήνυμα παράγει μια έγκυρη υπογραφή χωρίς την συμμετοχή του  $S$ .

### Είδη Επιθέσεων

- **Καθολική πλαστογράφηση:** Ο  $\mathcal{A}$  μπορεί να παράγει έγκυρες υπογραφές σε όποιο μήνυμα θέλει ( $\Leftrightarrow$  κατοχή ιδιωτικού κλειδιού)
- **Επιλεκτική πλαστογράφηση:** Ο  $\mathcal{A}$  μπορεί να παράγει 1 έγκυρη υπογραφή σε μήνυμα (με νόημα) της επιλογής του

## Πλαστογραφία(Forgery)

Ο  $\mathcal{A}$  με δεδομένα το δημόσιο κλειδί επαλήθευσης και ένα μήνυμα παράγει μια έγκυρη υπογραφή χωρίς την συμμετοχή του  $S$ .

## Είδη Επιθέσεων

- **Καθολική πλαστογράφηση:** Ο  $\mathcal{A}$  μπορεί να παράγει έγκυρες υπογραφές σε όποιο μήνυμα θέλει ( $\Leftrightarrow$  κατοχή ιδιωτικού κλειδιού)
- **Επιλεκτική πλαστογράφηση:** Ο  $\mathcal{A}$  μπορεί να παράγει 1 έγκυρη υπογραφή σε μήνυμα (με νόημα) της επιλογής του
- **Υπαρξιακή πλαστογράφηση:** Ο  $\mathcal{A}$  μπορεί να παράγει 1 έγκυρη υπογραφή (τυχαία bits) σε τυχαίο μήνυμα

## Είδη Αντιπάλων

- Παθητικός (passive): Απλά γνωρίζει το κλειδί επαλήθευσης και ζεύγη μηνυμάτων, έγκυρων υπογραφών
- Ενεργός (active): Μπορεί να αποκτήσει έγκυρες υπογραφές σε μηνύματα της επιλογής του
- Ενεργός με προσαρμοστικότητα (adaptive active): Μπορεί να αποκτήσει έγκυρες υπογραφές σε μηνύματα της επιλογής του που εξαρτώνται από προηγούμενες έγκυρες υπογραφές

Ασφάλεια ως προς τον δυνατότερο αντίπαλο - γενικότερη επίθεση

### **Ασφάλεια**

Ένα σχήμα υπογραφής είναι ασφαλές αν δεν επιτρέπει σε έναν ενεργό αντίπαλο με προσαρμοστικότητα να επιτύχει υπαρξιακή πλαστογράφηση

## Το παιχνίδι πλαστογράφησης *Forge* – *Game*

- Ο  $S$  εκτελεί τον αλγόριθμο  $\text{KeyGen}(1^\lambda)$  και παράγει τα  $(pk, sk)$

## Το παιχνίδι πλαστογράφησης *Forge – Game*

- Ο  $S$  εκτελεί τον αλγόριθμο  $\text{KeyGen}(1^\lambda)$  και παράγει τα  $(pk, sk)$
- Ο  $\mathcal{A}$  έχει πρόσβαση σε ένα μαντείο υπογραφών  $\text{Sign}(sk, \cdot)$  με το οποίο αποκτά ένα σύνολο έγκυρων υπογραφών  $Q = \{(m_i, \sigma_i)\}$



## Το παιχνίδι πλαστογράφησης *Forge – Game*

- Ο  $S$  εκτελεί τον αλγόριθμο  $\text{KeyGen}(1^\lambda)$  και παράγει τα  $(pk, sk)$
- Ο  $\mathcal{A}$  έχει πρόσβαση σε ένα μαντείο υπογραφών  $\text{Sign}(sk, \cdot)$  με το οποίο αποκτά ένα σύνολο έγκυρων υπογραφών  $Q = \{(m_i, \sigma_i)\}$

γιατί στην ‘πραγματική ζωή’ μπορεί να χρησιμοποιήσει παλιότερες υπογραφές

## Το παιχνίδι πλαστογράφησης *Forge* – *Game*

- Ο  $S$  εκτελεί τον αλγόριθμο  $\text{KeyGen}(1^\lambda)$  και παράγει τα  $(pk, sk)$
- Ο  $\mathcal{A}$  έχει πρόσβαση σε ένα μαντείο υπογραφών  $\text{Sign}(sk, \cdot)$  με το οποίο αποκτά ένα σύνολο έγκυρων υπογραφών  $Q = \{(m_i, \sigma_i)\}$   
γιατί στην ‘πραγματική ζωή’ μπορεί να χρησιμοποιήσει παλιότερες υπογραφές
- Ο  $\mathcal{A}$  επιλέγει ένα μήνυμα  $m$  και παράγει το ζεύγος  $(m, \sigma)$

## Το παιχνίδι πλαστογράφησης *Forge – Game*

- Ο  $S$  εκτελεί τον αλγόριθμο  $\text{KeyGen}(1^\lambda)$  και παράγει τα  $(pk, sk)$
- Ο  $\mathcal{A}$  έχει πρόσβαση σε ένα μαντείο υπογραφών  $\text{Sign}(sk, \cdot)$  με το οποίο αποκτά ένα σύνολο έγκυρων υπογραφών  $Q = \{(m_i, \sigma_i)\}$   
γιατί στην ‘πραγματική ζωή’ μπορεί να χρησιμοποιήσει παλιότερες υπογραφές
- Ο  $\mathcal{A}$  επιλέγει ένα μήνυμα  $m$  και παράγει το ζεύγος  $(m, \sigma)$
- Νίκη  $\mathcal{A}$ :  $\text{Forge} - \text{Game}(\mathcal{A}) = 1 \Leftrightarrow \text{Verify}(key_{ver}, m, \sigma) = 1 \wedge (m, \sigma) \notin Q$

## Το παιχνίδι πλαστογράφησης *Forge – Game*

- Ο  $S$  εκτελεί τον αλγόριθμο  $\text{KeyGen}(1^\lambda)$  και παράγει τα  $(pk, sk)$
- Ο  $\mathcal{A}$  έχει πρόσβαση σε ένα μαντείο υπογραφών  $\text{Sign}(sk, \cdot)$  με το οποίο αποκτά ένα σύνολο έγκυρων υπογραφών  $Q = \{(m_i, \sigma_i)\}$   
γιατί στην ‘πραγματική ζωή’ μπορεί να χρησιμοποιήσει παλιότερες υπογραφές
- Ο  $\mathcal{A}$  επιλέγει ένα μήνυμα  $m$  και παράγει το ζεύγος  $(m, \sigma)$
- Νίκη  $\mathcal{A}$ :  $\text{Forge} - \text{Game}(\mathcal{A}) = 1 \Leftrightarrow \text{Verify}(key_{ver}, m, \sigma) = 1 \wedge (m, \sigma) \notin Q$

Ο  $\mathcal{A}$  κερδίζει το παιχνίδι αν

$$\Pr[\text{Forge} - \text{Game}(\mathcal{A}) = 1] = \text{non} - \text{negl}(\lambda)$$

# Ψηφιακές Υπογραφές RSA

---

Δημιουργία Κλειδιών:  $KeyGen(1^\lambda) = (d, (e, n))$

- $n = p \cdot q$ ,  $p, q$  πρώτοι αριθμοί  $\frac{\lambda}{2}$  bits
- Επιλογή  $e$  ώστε  $\gcd(e, \phi(n)) = 1$
- $d = e^{-1} \pmod{\phi(n)}$  με EGCD

Δημιουργία Κλειδιών:  $KeyGen(1^\lambda) = (d, (e, n))$

- $n = p \cdot q$ ,  $p, q$  πρώτοι αριθμοί  $\frac{\lambda}{2}$  bits
- Επιλογή  $e$  ώστε  $\gcd(e, \phi(n)) = 1$
- $d = e^{-1} \pmod{\phi(n)}$  με EGCD

Υπογραφή - Αποκρυπτογράφηση

- $Sign(d, m) = m^d \pmod{n}$

Δημιουργία Κλειδιών:  $KeyGen(1^\lambda) = (d, (e, n))$

- $n = p \cdot q$ ,  $p, q$  πρώτοι αριθμοί  $\frac{\lambda}{2}$  bits
- Επιλογή  $e$  ώστε  $\gcd(e, \phi(n)) = 1$
- $d = e^{-1} \pmod{\phi(n)}$  με EGCD

Υπογραφή - Αποκρυπτογράφηση

- $Sign(d, m) = m^d \pmod{n}$

Επαλήθευση - Κρυπτογράφηση

- $Verify((e, n), m, \sigma) = \sigma^e \stackrel{?}{=} m \pmod{n}$



# Ψηφιακές Υπογραφές RSA

Δημιουργία Κλειδιών:  $KeyGen(1^\lambda) = (d, (e, n))$

- $n = p \cdot q$ ,  $p, q$  πρώτοι αριθμοί  $\frac{\lambda}{2}$  bits
- Επιλογή  $e$  ώστε  $\gcd(e, \phi(n)) = 1$
- $d = e^{-1} \pmod{\phi(n)}$  με EGCD

Υπογραφή - Αποκρυπτογράφηση

- $Sign(d, m) = m^d \pmod n$

Επαλήθευση - Κρυπτογράφηση

- $Verify((e, n), m, \sigma) = \sigma^e \stackrel{?}{=} m \pmod n$

Ορθότητα

$$Verify((e, n), m, m^d \pmod n) = m^{d^e} = m \pmod n$$

# Ψηφιακές Υπογραφές RSA

Δημιουργία Κλειδιών:  $KeyGen(1^\lambda) = (d, (e, n))$

- $n = p \cdot q$ ,  $p, q$  πρώτοι αριθμοί  $\frac{\lambda}{2}$  bits
- Επιλογή  $e$  ώστε  $\gcd(e, \phi(n)) = 1$
- $d = e^{-1} \pmod{\phi(n)}$  με EGCD

Υπογραφή - Αποκρυπτογράφηση

- $Sign(d, m) = m^d \pmod n$

Επαλήθευση - Κρυπτογράφηση

- $Verify((e, n), m, \sigma) = \sigma^e \stackrel{?}{=} m \pmod n$

Ορθότητα

$$Verify((e, n), m, m^d \pmod n) = m^{d^e} = m \pmod n$$

...αλλά καθόλου ασφάλεια

## Επίθεση Χωρίς Μήνυμα (No message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$

## Επίθεση Χωρίς Μήνυμα (No message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$
- $Q = \emptyset$  - δεν υποβάλλονται μηνύματα για υπογραφή

## Επίθεση Χωρίς Μήνυμα (No message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$
- $Q = \emptyset$  - δεν υποβάλλονται μηνύματα για υπογραφή
- Επιλογή τυχαίου  $\sigma \in \mathbb{Z}_n^*$

## Επίθεση Χωρίς Μήνυμα (No message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$
- $Q = \emptyset$  - δεν υποβάλλονται μηνύματα για υπογραφή
- Επιλογή τυχαίου  $\sigma \in \mathbb{Z}_n^*$
- 'Κρυπτογράφηση'  $\sigma$ :  $\sigma^e \bmod n = m$

## Επίθεση Χωρίς Μήνυμα (No message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$
- $Q = \emptyset$  - δεν υποβάλλονται μηνύματα για υπογραφή
- Επιλογή τυχαίου  $\sigma \in \mathbb{Z}_n^*$
- 'Κρυπτογράφηση'  $\sigma$ :  $\sigma^e \bmod n = m$
- Το ζεύγος  $(m, \sigma)$  είναι έγκυρο και  $\notin Q$

## Επίθεση Χωρίς Μήνυμα (No message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$
- $Q = \emptyset$  - δεν υποβάλλονται μηνύματα για υπογραφή
- Επιλογή τυχαίου  $\sigma \in \mathbb{Z}_n^*$
- 'Κρυπτογράφηση'  $\sigma$ :  $\sigma^e \bmod n = m$
- Το ζεύγος  $(m, \sigma)$  είναι έγκυρο και  $\notin Q$
- Ο  $\mathcal{A}$  κερδίζει με πιθανότητα 1

Έχει νόημα; - Ναι, με επαναλήψεις μπορούν να βρεθούν  $m$  όπου κάποια bits μπορεί να είναι έγκυρα τμήματα μηνυμάτων



## Επίθεση Επιλεγμένων Μηνυμάτων (Chosen message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$  και θέλει να πλαστογραφήσει υπογραφή για  $m \in \mathbb{Z}_n^*$

## Επίθεση Επιλεγμένων Μηνυμάτων (Chosen message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$  και θέλει να πλαστογραφήσει υπογραφή για  $m \in \mathbb{Z}_n^*$
- Ο  $\mathcal{A}$  χρησιμοποιώντας το μαντείο **αποκτά τις υπογραφές** 2 μηνυμάτων  $Q = \{(m_1, \sigma_1), (\frac{m}{m_1}, \sigma_2)\}$  με  $m_1 \in_R \mathbb{Z}_n^*$

## Επίθεση Επιλεγμένων Μηνυμάτων (Chosen message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$  και θέλει να πλαστογραφήσει υπογραφή για  $m \in \mathbb{Z}_n^*$
- Ο  $\mathcal{A}$  χρησιμοποιώντας το μαντείο **αποκτά τις υπογραφές** 2 μηνυμάτων  $Q = \{(m_1, \sigma_1), (\frac{m}{m_1}, \sigma_2)\}$  με  $m_1 \in_R \mathbb{Z}_n^*$
- Υπολογισμός  $\sigma = \sigma_1 \sigma_2 = m_1^d (\frac{m}{m_1})^d = m^d \pmod n$

## Επίθεση Επιλεγμένων Μηνυμάτων (Chosen message attack)

- Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$  και θέλει να πλαστογραφήσει υπογραφή για  $m \in \mathbb{Z}_n^*$
- Ο  $\mathcal{A}$  χρησιμοποιώντας το μαντείο **αποκτά τις υπογραφές** 2 μηνυμάτων  $Q = \{(m_1, \sigma_1), (\frac{m}{m_1}, \sigma_2)\}$  με  $m_1 \in_R \mathbb{Z}_n^*$
- Υπολογισμός  $\sigma = \sigma_1 \sigma_2 = m_1^d (\frac{m}{m_1})^d = m^d \pmod n$
- Η  $\sigma$  είναι έγκυρη υπογραφή για το  $m$  και  $\notin Q$

Δημιουργία Κλειδιών:  $KeyGen(1^\lambda) = (d, (e, n))$

- $n = p \cdot q$ ,  $p, q$  πρώτοι αριθμοί  $\frac{\lambda}{2}$  bits
- Επιλογή  $e$  ώστε  $\gcd(e, \phi(n)) = 1$
- $d = e^{-1} \pmod{\phi(n)}$  με EGCD
- Χρήση δημόσια διαθέσιμης τυχαίας συνάρτησης  $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{Z}_n^*$

Υπογραφή

- Υπολογισμός  $\mathcal{H}(m)$
- $Sign(d, m) = \mathcal{H}(m)^d \pmod{n}$

Επαλήθευση

- Υπολογισμός  $\mathcal{H}(m)$
- $\text{Verify}((e, n), m, \sigma) = \sigma^e \stackrel{?}{=} \mathcal{H}(m) \pmod{n}$

## Ορθότητα

$$\text{Verify}((e, n), m, \mathcal{H}(m)^d) = \mathcal{H}(m)^{d^e} = \mathcal{H}(m) \pmod{n}$$

Υλοποίηση: συνάρτηση σύνοψης με δυσκολία εύρεσης συγκρούσεων

Πλεονέκτημα: Μπορεί να χρησιμοποιηθεί για υπογραφή τυχαίων συμβολοσειρών και όχι μόνο στοιχείων του  $\mathbb{Z}_n^*$

- Επίθεση χωρίς μήνυμα
  - Επιλογή τυχαίου  $\sigma \in \mathbb{Z}_n^*$
  - Η 'κρυπτογράφηση' δίνει τη σύνοψη  $h = \sigma^e \bmod n$  - όχι το μήνυμα
  - Για το μήνυμα πρέπει να βρεθεί  $m : \mathcal{H}(m) = h$
  - Δυσκολία αντιστροφής

- Επίθεση επιλεγμένων μηνυμάτων
  - Ο  $\mathcal{A}$  έχει στη διάθεση του δημόσιο κλειδί  $(e, n)$  και θέλει να πλαστογραφήσει υπογραφή για  $m \in \mathbb{Z}_n^*$
  - Ο  $\mathcal{A}$  χρησιμοποιώντας το μαντείο αποκτά τις υπογραφές 2 μηνυμάτων  $Q = \{(m_1, \sigma_1), (\frac{m}{m_1}, \sigma_2)\}$  με  $m_1 \in_R \mathbb{Z}_n^*$
  - Υπολογισμός  $\sigma = \sigma_1 \sigma_2 = \mathcal{H}(m_1) \mathcal{H}(\frac{m}{m_1})$
  - Δυσκολία αντιστροφής
- Απόδειξη Ασφάλειας: Πρέπει η  $\mathcal{H}$  να δίνει 'τυχαίες' τιμές
- Αρκούν οι ιδιότητες τους (one-way-ness, collision resistance);
  - OXI
- Το μοντέλο του τυχαίου μαντείου (M. Bellare, P. Rogaway, -1993)



# Το μοντέλο του τυχαίου μαντείου

---

## Συναρτήσεις σύνοψης ως τυχαίες συναρτήσεις - informal

- Θεωρητικά θα θέλαμε να συμπεριφέρονται ως τυχαίες συναρτήσεις

## Συναρτήσεις σύνοψης ως τυχαίες συναρτήσεις - informal

- Θεωρητικά θα θέλαμε να συμπεριφέρονται ως τυχαίες συναρτήσεις
- Πρακτικά όμως: αδύνατον να κατασκευαστούν

## Συναρτήσεις σύνοψης ως τυχαίες συναρτήσεις - informal

- Θεωρητικά θα θέλαμε να συμπεριφέρονται ως τυχαίες συναρτήσεις
- Πρακτικά όμως: αδύνατον να κατασκευαστούν
  - Συνάρτηση  $\mathcal{H} : \{0, 1\}^n \rightarrow \{0, 1\}^{l(n)}$

# Συναρτήσεις σύνοψης ως τυχαίες συναρτήσεις - informal

- Θεωρητικά θα θέλαμε να συμπεριφέρονται ως τυχαίες συναρτήσεις
- Πρακτικά όμως: **αδύνατον να κατασκευαστούν**
  - Συνάρτηση  $\mathcal{H} : \{0, 1\}^n \rightarrow \{0, 1\}^{l(n)}$
  - Κατασκευή ως πίνακας τιμών: Απαιτούνται  $2^n$  γραμμές

Εισοδος	Έξοδος
0...00	$r_1$
0...01	$r_2$
...	...
1...11	$r_{l(n)}$

# Συναρτήσεις σύνοψης ως τυχαίες συναρτήσεις - informal

- Θεωρητικά θα θέλαμε να συμπεριφέρονται ως τυχαίες συναρτήσεις
- Πρακτικά όμως: **αδύνατον να κατασκευαστούν**
  - Συνάρτηση  $\mathcal{H} : \{0, 1\}^n \rightarrow \{0, 1\}^{l(n)}$
  - Κατασκευή ως πίνακας τιμών: Απαιτούνται  $2^n$  γραμμές

Έισοδος	Έξοδος
0...00	$r_1$
0...01	$r_2$
...	...
1...11	$r_{l(n)}$

- Συμπίεση: Μείωση τυχειότητας

# Συναρτήσεις σύνοψης ως τυχαίες συναρτήσεις - informal

- Θεωρητικά θα θέλαμε να συμπεριφέρονται ως τυχαίες συναρτήσεις
- Πρακτικά όμως: **αδύνατον να κατασκευαστούν**
  - Συνάρτηση  $\mathcal{H} : \{0, 1\}^n \rightarrow \{0, 1\}^{l(n)}$
  - Κατασκευή ως πίνακας τιμών: Απαιτούνται  $2^n$  γραμμές

Εισοδος	Έξοδος
0...00	$r_1$
0...01	$r_2$
...	...
1...11	$r_{l(n)}$

- Συμπέραση: Μείωση τυχειότητας

Ακόμα και να μπορούσαν να κατασκευαστούν

αδύνατη αποθήκευση

εκθετική αποτίμηση (μη αποδεκτή και για χρήστη και για

αντίπαλο)

## Τυχαίο Μαντείο - Αφαιρετική αναπαράσταση συνάρτησης σύνοψης

- Μαύρο κουτί - απαντάει σε ερωτήσεις
- (Τέλεια) Ασφάλεια στο κανάλι επικοινωνίας (μοντελοποίηση τοπικής αποτίμησης)
- Είναι συνάρτηση (ίδια είσοδος - ίδια έξοδος σε κάθε κλήση)
- Είναι συνάρτηση σύνοψης (υπάρχουν συγκρούσεις - αλλά είναι δύσκολο να βρεθούν)



## Lazy Evaluation

- Εσωτερικός πίνακας - αρχικά άδειος
- Για κάθε ερώτηση: έλεγχος αν έχει ήδη απαντηθεί
- Αν ναι, τότε ανάκτηση της απάντησης
- Αν όχι, απάντηση με τυχαία τιμή και αποθήκευση για μελλοντική αναφορά

Αποδείξεις στο μοντέλο τυχαίου μαντείου (Bellare - Rogaway)

- Ο  $\mathcal{A}$  νομίζει ότι αλληλεπιδρά με το τυχαίο μαντείο
- Στην πραγματικότητα το προσομοιώνει η αναγωγή (programmability)
- Μπορούμε να μάθουμε τις ερωτήσεις του  $\mathcal{A}$
- Στο πραγματικό πρωτόκολλο το τυχαίο μαντείο αντικαθίσταται από μία πραγματική συνάρτηση (πχ. SHA256)

## Theorem

*Αν το πρόβλημα RSA είναι δύσκολο, τότε οι υπογραφές Hashed RSA παρέχουν ασφάλεια έναντι πλαστογράφησης στο μοντέλο του τυχαίου μαντείου.*

## Theorem

*Αν το πρόβλημα RSA είναι δύσκολο, τότε οι υπογραφές Hashed RSA παρέχουν ασφάλεια έναντι πλαστογράφησης στο μοντέλο του τυχαίου μαντείου.*

Γενική κατασκευή:

- Ο  $\mathcal{A}$  μπορεί να κατασκευάσει πλαστογράφηση υπογραφής

## Theorem

*Αν το πρόβλημα RSA είναι δύσκολο, τότε οι υπογραφές Hashed RSA παρέχουν ασφάλεια έναντι πλαστογράφησης στο μοντέλο του τυχαίου μαντείου.*

Γενική κατασκευή:

- Ο  $\mathcal{A}$  μπορεί να κατασκευάσει πλαστογράφηση υπογραφής
- Κατασκευή  $\mathcal{B}$  που με χρήση του  $\mathcal{A}$  και ενός τυχαίου μαντείου μπορεί να αντιστρέψει το RSA

## Theorem

Αν το πρόβλημα RSA είναι δύσκολο, τότε οι υπογραφές Hashed RSA παρέχουν ασφάλεια έναντι πλαστογράφησης στο μοντέλο του τυχαίου μαντείου.

Γενική κατασκευή:

- Ο  $\mathcal{A}$  μπορεί να κατασκευάσει πλαστογράφιση υπογραφής
- Κατασκευή  $\mathcal{B}$  που με χρήση του  $\mathcal{A}$  και ενός τυχαίου μαντείου μπορεί να αντιστρέψει το RSA
- Είσοδος  $\mathcal{B}$ 
  - Δημόσιο κλειδί  $(e, N)$

## Theorem

Αν το πρόβλημα RSA είναι δύσκολο, τότε οι υπογραφές Hashed RSA παρέχουν ασφάλεια έναντι πλαστογράφησης στο μοντέλο του τυχαίου μαντείου.

Γενική κατασκευή:

- Ο  $\mathcal{A}$  μπορεί να κατασκευάσει πλαστογράφηση υπογραφής
- Κατασκευή  $\mathcal{B}$  που με χρήση του  $\mathcal{A}$  και ενός τυχαίου μαντείου μπορεί να αντιστρέψει το RSA
- **Είσοδος  $\mathcal{B}$** 
  - Δημόσιο κλειδί  $(e, N)$
  - Στοιχείο  $y \in \mathbb{Z}_n^*$

## Theorem

Αν το πρόβλημα RSA είναι δύσκολο, τότε οι υπογραφές Hashed RSA παρέχουν ασφάλεια έναντι πλαστογράφησης στο μοντέλο του τυχαίου μαντείου.

Γενική κατασκευή:

- Ο  $\mathcal{A}$  μπορεί να κατασκευάσει πλαστογράφηση υπογραφής
- Κατασκευή  $\mathcal{B}$  που με χρήση του  $\mathcal{A}$  και ενός τυχαίου μαντείου μπορεί να αντιστρέψει το RSA
- **Είσοδος  $\mathcal{B}$** 
  - Δημόσιο κλειδί  $(e, N)$
  - Στοιχείο  $y \in \mathbb{Z}_n^*$
- **Έξοδος  $\mathcal{B}$** 
  - $x = y^{\frac{1}{e}}$



# Απόδειξη Ασφάλειας Hashed RSA

## Επίθεση χωρίς μήνυμα $i$

### Υπόθεση

Για την πλαστογράφηση  $(m, \sigma)$  έχει προηγουμένως ερωτηθεί στο μαντείο το  $\mathcal{H}(m)$

### Συνέπεια

Εφόσον η πλαστογράφηση είναι έγκυρη υπογραφή πρέπει

$$\sigma^e = \mathcal{H}(m) \text{ Άρα } \sigma = \mathcal{H}(m)^{\frac{1}{e}}$$

# Απόδειξη Ασφάλειας Hashed RSA

## Επίθεση χωρίς μήνυμα ii

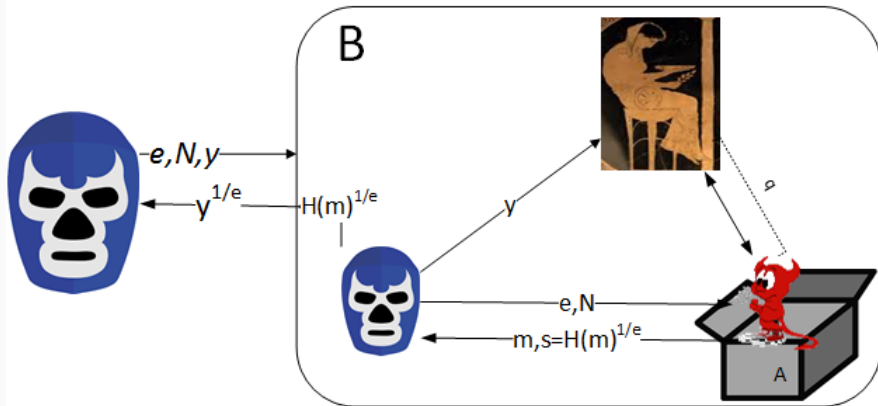
- Ο  $\mathcal{B}$  προωθεί το  $(e, N)$  στον  $\mathcal{A}$
- Ο  $\mathcal{A}$  κάνει  $q = \text{poly}(\lambda)$  ερωτήσεις στο μαντείο για μηνύματα  $\{m_i\}_{i=1}^q$  και λαμβάνει τις απαντήσεις  $\{\mathcal{H}(m_i)\}_{i=1}^q \in_r \mathbb{Z}_n^*$
- Ο  $\mathcal{B}$  επιλέγει τυχαία μία ερώτηση και αντικαθιστά την απάντηση  $\mathcal{H}(m_i^*)$  με το  $y$
- Ο  $\mathcal{B}$  ελπίζει ότι στο  $\mathcal{H}(m_i^*)$  θα γίνει η πλαστογράφηση
- Αν έχει δίκιο, τότε ο  $\mathcal{A}$  εξάγει την πλαστογραφία  $(m, \sigma)$  με πιθανότητα  $p$
- Δηλαδή:  $\sigma^e = y \Rightarrow \sigma = y^{\frac{1}{e}}$
- Ο  $\mathcal{B}$  προωθεί το  $\sigma$  στην έξοδο
- Με πιθανότητα επιτυχίας  $\frac{p}{q}$  θα ισχύει  $\sigma = y^{\frac{1}{e}}$

# Απόδειξη Ασφάλειας Hashed RSA

## Επίθεση χωρίς μήνυμα iii

Αν  $p$  αμελητέο τότε  $\frac{p}{q}$  αμελητέο

# Απόδειξη Ασφάλειας Hashed RSA Επίθεση χωρίς μήνυμα iv



# Απόδειξη Ασφάλειας Hashed RSA

## Επίθεση επιλεγμένου μηνύματος $i$

### Σενάριο

- $\mathcal{A}$  πρέπει να υπολογίσει έγκυρες υπογραφές
- Ζητάει συνόψεις και υπογραφές από τον  $\mathcal{B}$
- Συνόψεις: το τυχαίο μαντείο
- Υπογραφές: Πρέπει να τις απαντήσει ο  $\mathcal{B}$
- ...χωρίς το ιδιωτικό κλειδί

### Λύση

Αντικατάσταση  $\mathcal{H}(m)$  με  $\sigma^e$  για γνωστό  $\sigma$

Τετριμμένη επαλήθευση  $\sigma^e = \sigma^e (= \mathcal{H}(m))$

# Απόδειξη Ασφάλειας Hashed RSA

## Επίθεση επιλεγμένου μηνύματος ii

- Ο  $\mathcal{B}$  προωθεί το  $(e, N)$  στον  $\mathcal{A}$
- Ο  $\mathcal{A}$  κάνει  $q$  ερωτήσεις στο μαντείο για μηνύματα  $\{m_i\}_{i=1}^q$
- Κάθε ερώτηση απαντάται από τον  $\mathcal{B}$  ως εξής:
  - Επιλέγει τυχαίο  $\sigma_i \in \mathbb{Z}_n^*$
  - Υπολογίζει  $y_i = \mathcal{H}(m_i) = \sigma_i^e \bmod N$
  - Επιστρέφει  $y_i$
  - Αποθηκεύει τις τριάδες  $\mathcal{T} = (m_i, y_i, \sigma_i)$
- Ο  $\mathcal{A}$  ζητάει υπογραφές
  - Για κάθε σύνοψη  $y_i$  γίνεται αναζήτηση στον  $\mathcal{T}$  για την τριάδα και επιστρέφεται το  $\sigma_i$
  - Οι υπογραφές είναι έγκυρες αφού  $\sigma_i^e = y_i$
- Ο  $\mathcal{B}$  μαντεύει ποιο ερώτημα στο RO θα οδηγήσει στην πλαστογράφιση. Το απαντάει με  $y$

# Απόδειξη Ασφάλειας Hashed RSA

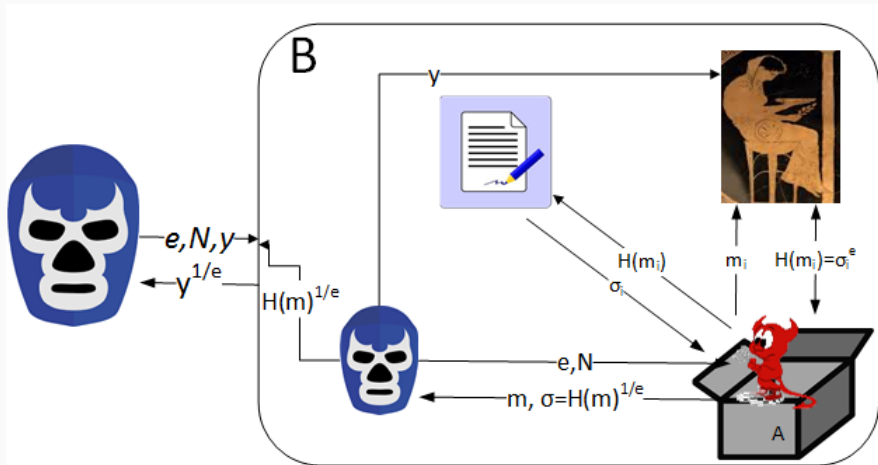
## Επίθεση επιλεγμένου μηνύματος iii

- Για το συγκεκριμένο δεν θα ζητηθεί υπογραφή, αλλά το  $\sigma$  θα παραχθεί από τον  $\mathcal{A}$  (πλαστογράφηση)
- Για να είναι έγκυρη η πλαστογράφημενη υπογραφή πρέπει  $\sigma^e = y$ , δηλαδή  $\sigma = y^{\frac{1}{e}}$
- Πιθανότητα επιτυχίας  $\mathcal{A}$   $p$  και πιθανότητα επιτυχίας  $\mathcal{B}$   $\frac{p}{q}$

Αν  $p$  αμελητέο τότε  $\frac{p}{q}$  αμελητέο

# Απόδειξη Ασφάλειας Hashed RSA

## Επίθεση επιλεγμένου μηνύματος $m_i$





## Μειονεκτήματα

‘Αχρηστη’ απόδειξη - Καμία πραγματική συνάρτηση  $H$  δεν είναι random oracle

## Μειονεκτήματα

‘Άχρηστη’ απόδειξη - Καμία πραγματική συνάρτηση  $H$  δεν είναι random oracle

Εσωτερική χρήση - Δεν φαίνονται οι τιμές στις οποίες αποτιμάται

## Μειονεκτήματα

‘Άχρηστη’ απόδειξη - Καμία πραγματική συνάρτηση  $\mathcal{H}$  δεν είναι random oracle

Εσωτερική χρήση - Δεν φαίνονται οι τιμές στις οποίες αποτιμάται

Programmability - Η περιγραφή της συνάρτησης είναι σταθερή στην πραγματικότητα

## Μειονεκτήματα

‘Άχρηστη’ απόδειξη - Καμία πραγματική συνάρτηση  $H$  δεν είναι random oracle

Εσωτερική χρήση - Δεν φαίνονται οι τιμές στις οποίες αποτιμάται

Programmability - Η περιγραφή της συνάρτησης είναι σταθερή στην πραγματικότητα

‘Υπαρξη’ ‘θεωρητικών’ σχημάτων τα οποία αποδεικνύονται ασφαλή, αλλά οποιαδήποτε κατασκευή τους είναι μη ασφαλής

# Το μοντέλο του τυχαίου μαντείου - κριτική

## Μειονεκτήματα

‘Άχρηστη’ απόδειξη - Καμία πραγματική συνάρτηση  $H$  δεν είναι random oracle

Εσωτερική χρήση - Δεν φαίνονται οι τιμές στις οποίες αποτιμάται

Programmability - Η περιγραφή της συνάρτησης είναι σταθερή στην πραγματικότητα

‘Υπαρξη ‘θεωρητικών’ σχημάτων τα οποία αποδεικνύονται ασφαλή, αλλά οποιαδήποτε κατασκευή τους είναι μη ασφαλής

## Πλεονεκτήματα

Απόδειξη με χρήση τυχαίου μαντείου είναι καλύτερη από απουσία απόδειξης

## Μειονεκτήματα

‘Αχρηστη’ απόδειξη - Καμία πραγματική συνάρτηση  $\mathcal{H}$  δεν είναι random oracle

Εσωτερική χρήση - Δεν φαίνονται οι τιμές στις οποίες αποτιμάται

Programmability - Η περιγραφή της συνάρτησης είναι σταθερή στην πραγματικότητα

‘Υπαρξη ‘θεωρητικών’ σχημάτων τα οποία αποδεικνύονται ασφαλή, αλλά οποιαδήποτε κατασκευή τους είναι μη ασφαλής

## Πλεονεκτήματα

Απόδειξη με χρήση τυχαίου μαντείου είναι καλύτερη από απουσία απόδειξης

Η μόνη αδυναμία: η συνάρτηση σύνοψης

# Το μοντέλο του τυχαίου μαντείου - κριτική

## Μειονεκτήματα

‘Αχρηστη’ απόδειξη - Καμία πραγματική συνάρτηση  $H$  δεν είναι random oracle

Εσωτερική χρήση - Δεν φαίνονται οι τιμές στις οποίες αποτιμάται

Programmability - Η περιγραφή της συνάρτησης είναι σταθερή στην πραγματικότητα

‘Υπαρξη ‘θεωρητικών’ σχημάτων τα οποία αποδεικνύονται ασφαλή, αλλά οποιαδήποτε κατασκευή τους είναι μη ασφαλής

## Πλεονεκτήματα

Απόδειξη με χρήση τυχαίου μαντείου είναι καλύτερη από απουσία απόδειξης

Η μόνη αδυναμία: η συνάρτηση σύνοψης

Δεν υπάρχουν πραγματικές επιθέσεις που να έχουν εκμεταλλευτεί την απόδειξη μέσω τυχαίου μαντείου

# Ψηφιακές Υπογραφές ElGamal

---



# Σχήμα Υπογραφής ElGamal i

## Δημιουργία Κλειδιών:

- Επιλογή πρώτου  $p$ . Δουλεύουμε στο  $\mathbb{Z}_p^*$  ΠΡΟΣΟΧΗ!
- Επιλογή γεννήτορα  $g$
- Επιλογή  $x \in \{2 \cdots p - 2\}$  και υπολογισμός του  $y = g^x \pmod{p}$
- Δημόσιο κλειδί  $(p, g, y)$ , ιδιωτικό κλειδί  $x$ .

## Υπογραφή Μηνύματος $m$

- Επιλογή τυχαίου  $k \in \mathbb{Z}_{p-1}^*$  .  $\gcd(k, p - 1) = 1$
- Υπολογισμός

$$r = g^k \pmod{p}$$

$$s = (m - xr)k^{-1} \pmod{p - 1}$$

- Υπογραφή είναι:  $(r, s)$
- Δύο ακέραιοι μεγέθους  $O(|p|)$

Επαλήθευση υπογραφής στο  $m$

$$\text{Verify}(y, m, (r, s)) = \begin{cases} 1, & y^r \cdot r^s \equiv g^m \pmod{p} \\ 0, & y^r \cdot r^s \not\equiv g^m \pmod{p} \end{cases}$$

Ορθότητα

$$y^r r^s \equiv g^{xr} g^{ks} = g^{xr+ks} \equiv g^m \pmod{p}$$

το οποίο ισχύει λόγω της κατασκευής του  $s$

- Πιθανοτικό σχήμα υπογραφής - πολλές έγκυρες υπογραφές για ένα μήνυμα  $m$  (τυχαίο  $k$ )

- Πιθανοτικό σχήμα υπογραφής - πολλές έγκυρες υπογραφές για ένα μήνυμα  $m$  (τυχαίο  $k$ )
- Η συνάρτηση επαλήθευσης δέχεται οποιαδήποτε από αυτές ως έγκυρη

- Πιθανοτικό σχήμα υπογραφής - πολλές έγκυρες υπογραφές για ένα μήνυμα  $m$  (τυχαίο  $k$ )
- Η συνάρτηση επαλήθευσης δέχεται οποιαδήποτε από αυτές ως έγκυρη
- Χειρισμός Τυχειότητας
  - Το τυχαία επιλεγμένο  $k$  πρέπει να κρατείται κρυφό
  - Η επανάληψη της χρήσης του ίδιου  $k$  καθιστά για τον  $\mathcal{A}$  εφικτό τον υπολογισμό του

## Επίθεση επανάληψης κλειδιού

Χρήση ίδιου εφήμερου κλειδιού στην υπογραφή δύο μηνυμάτων  $m_1, m_2$

- $sign(x, m_1) = (r, s_1)$  με  $s_1 = (m_1 - xr)k^{-1}$

## Επίθεση επανάληψης κλειδιού

Χρήση ίδιου εφήμερου κλειδιού στην υπογραφή δύο μηνυμάτων  $m_1, m_2$

- $sign(x, m_1) = (r, s_1)$  με  $s_1 = (m_1 - xr)k^{-1}$
- $sign(x, m_2) = (r, s_2)$  με  $s_2 = (m_2 - xr)k^{-1}$



## Επίθεση επανάληψης κλειδιού

Χρήση ίδιου εφήμερου κλειδιού στην υπογραφή δύο μηνυμάτων  $m_1, m_2$

- $sign(x, m_1) = (r, s_1)$  με  $s_1 = (m_1 - xr)k^{-1}$

- $sign(x, m_2) = (r, s_2)$  με  $s_2 = (m_2 - xr)k^{-1}$

- Υπολογισμός

$$s_1 - s_2 = (m_1 - m_2)k^{-1} \Rightarrow (s_1 - s_2)k = (m_1 - m_2)$$

## Επίθεση επανάληψης κλειδιού

Χρήση ίδιου εφήμερου κλειδιού στην υπογραφή δύο μηνυμάτων  $m_1, m_2$

- $sign(x, m_1) = (r, s_1)$  με  $s_1 = (m_1 - xr)k^{-1}$
- $sign(x, m_2) = (r, s_2)$  με  $s_2 = (m_2 - xr)k^{-1}$
- Υπολογισμός  
 $s_1 - s_2 = (m_1 - m_2)k^{-1} \Rightarrow (s_1 - s_2)k = (m_1 - m_2)$
- Δεν ισχύει γενικά  $gcd(s_1 - s_2, p - 1) = 1$

## Επίθεση επανάληψης κλειδιού

Χρήση ίδιου εφήμερου κλειδιού στην υπογραφή δύο μηνυμάτων  $m_1, m_2$

- $sign(x, m_1) = (r, s_1)$  με  $s_1 = (m_1 - xr)k^{-1}$
- $sign(x, m_2) = (r, s_2)$  με  $s_2 = (m_2 - xr)k^{-1}$
- Υπολογισμός  
$$s_1 - s_2 = (m_1 - m_2)k^{-1} \Rightarrow (s_1 - s_2)k = (m_1 - m_2)$$
- Δεν ισχύει γενικά  $gcd(s_1 - s_2, p - 1) = 1$
- Όμως υπάρχουν  $gcd(s_1 - s_2, p - 1)$  λύσεις για το  $k$  (αν διαιρεί το  $m_1 - m_2$ )
- Δοκιμή όλων των πιθανών  $g^k$  και σύγκριση με το γνωστό  $r$

## Επίθεση επανάληψης κλειδιού

Χρήση ίδιου εφήμερου κλειδιού στην υπογραφή δύο μηνυμάτων  $m_1, m_2$

- $sign(x, m_1) = (r, s_1)$  με  $s_1 = (m_1 - xr)k^{-1}$
- $sign(x, m_2) = (r, s_2)$  με  $s_2 = (m_2 - xr)k^{-1}$
- Υπολογισμός  
$$s_1 - s_2 = (m_1 - m_2)k^{-1} \Rightarrow (s_1 - s_2)k = (m_1 - m_2)$$
- Δεν ισχύει γενικά  $gcd(s_1 - s_2, p - 1) = 1$
- Όμως υπάρχουν  $gcd(s_1 - s_2, p - 1)$  λύσεις για το  $k$  (αν διαιρεί το  $m_1 - m_2$ )
- Δοκιμή όλων των πιθανών  $g^k$  και σύγκριση με το γνωστό  $r$
- Υπολογισμός ιδιωτικού κλειδιού από  $rx = m_1 - ks_1$

## Επίθεση επανάληψης κλειδιού

Χρήση ίδιου εφήμερου κλειδιού στην υπογραφή δύο μηνυμάτων  $m_1, m_2$

- $sign(x, m_1) = (r, s_1)$  με  $s_1 = (m_1 - xr)k^{-1}$
- $sign(x, m_2) = (r, s_2)$  με  $s_2 = (m_2 - xr)k^{-1}$
- Υπολογισμός  
$$s_1 - s_2 = (m_1 - m_2)k^{-1} \Rightarrow (s_1 - s_2)k = (m_1 - m_2)$$
- Δεν ισχύει γενικά  $gcd(s_1 - s_2, p - 1) = 1$
- Όμως υπάρχουν  $gcd(s_1 - s_2, p - 1)$  λύσεις για το  $k$  (αν διαιρεί το  $m_1 - m_2$ )
- Δοκιμή όλων των πιθανών  $g^k$  και σύγκριση με το γνωστό  $r$
- Υπολογισμός ιδιωτικού κλειδιού από  $rx = m_1 - ks_1$
- Δοκιμή όλων των  $gcd(r, p - 1)$  ως προς  $y = g^x$

Στόχος:  $g^{xr} \cdot r^s \equiv g^m \pmod{p}$

1. No message attack: Επιλέγω  $r$  και  $s$ , ψάχνω  $m$ : Επίλυση DLP.
2. Chosen message attack: Επιλέγω  $m$  και προσπαθώ να βρώ  $r, s$  για έγκυρη υπογραφή
  - Επιλέγω  $r$ , ψάχνω  $s$ . Πρέπει:  $r^s \equiv g^m \cdot g^{-xr} \pmod{p}$  (επίλυση DLP).
  - Επιλέγω  $s$ , ψάχνω  $r$ . Πρέπει:  $g^{xr} \equiv g^m \cdot r^{-s} \pmod{p}$   
Ανοιχτό πρόβλημα - δε γνωρίζουμε σχέση με DLP

3. Κατασκευή  $r, s, m$  ταυτόχρονα.  
Επιλέγω  $i, j$  με  $0 \leq i, j \leq p - 2$ ,  
και  $\gcd(j, p - 1) = 1$  και θέτω:

$$r = g^i \cdot (g^x)^j \bmod p$$

$$s = -r \cdot j^{-1} \bmod p - 1$$

$$m = -r \cdot i \cdot j^{-1} \bmod p - 1$$

Τα  $(r,s)$  επαληθεύουν την υπογραφή

Εφικτό σενάριο, δίνει υπογραφή για τυχαίο  $m$

Αντιμετώπιση με redundancy function / hash function

# Πρότυπο Ψηφιακής Υπογραφής (Digital Signature Standard – DSS)

## Βασικά Στοιχεία

- NIST, 1991.



# Πρότυπο Ψηφιακής Υπογραφής (Digital Signature Standard – DSS)

## Βασικά Στοιχεία

- NIST, 1991.
- Παραλλαγή του ElGamal, μικρότερο μέγεθος υπογραφής.

# Πρότυπο Ψηφιακής Υπογραφής (Digital Signature Standard – DSS)

## Βασικά Στοιχεία

- NIST, 1991.
- Παραλλαγή του ElGamal, μικρότερο μέγεθος υπογραφής.
- Ιδέα: λειτουργία σε μια υποομάδα της  $\mathbb{Z}_p^*$ , τάξης  $2^{160}$ .

# Πρότυπο Ψηφιακής Υπογραφής (Digital Signature Standard – DSS)

## Βασικά Στοιχεία

- NIST, 1991.
- Παραλλαγή του ElGamal, μικρότερο μέγεθος υπογραφής.
- Ιδέα: λειτουργία σε μια υποομάδα της  $\mathbb{Z}_p^*$ , τάξης  $2^{160}$ .
- Τα  $r, s$  είναι εκθέτες δυνάμεων του γεννήτορα της υποομάδας.

1. Επιλογή πρώτων  $q$  μεγέθους 160-bit και  $p$  μεγέθους  $n$ -bit,  $n = 64\lambda$ ,  $\lambda = 8, 9, 10, \dots, 16$ , με  $q \mid (p - 1)$ .

# Παραγωγή κλειδιών DSS

1. Επιλογή πρώτων  $q$  μεγέθους 160-bit και  $p$  μεγέθους  $n$ -bit,  $n = 64\lambda$ ,  $\lambda = 8, 9, 10, \dots, 16$ , με  $q \mid (p - 1)$ .
2. Εύρεση  $g$  γεννήτορα της υποομάδας τάξης  $q$  του  $\mathbb{Z}_p^*$

# Παραγωγή κλειδιών DSS

1. Επιλογή πρώτων  $q$  μεγέθους 160-bit και  $p$  μεγέθους  $n$ -bit,  $n = 64\lambda$ ,  $\lambda = 8, 9, 10, \dots, 16$ , με  $q \mid (p - 1)$ .
2. Εύρεση  $g$  γεννήτορα της υποομάδας τάξης  $q$  του  $\mathbb{Z}_p^*$
3. Επιλογή ιδιωτικού κλειδιού  $x \in \mathbb{Z}_q$ .

# Παραγωγή κλειδιών DSS

1. Επιλογή πρώτων  $q$  μεγέθους 160-bit και  $p$  μεγέθους  $n$ -bit,  $n = 64\lambda$ ,  $\lambda = 8, 9, 10, \dots, 16$ , με  $q \mid (p - 1)$ .
2. Εύρεση  $g$  γεννήτορα της υποομάδας τάξης  $q$  του  $\mathbb{Z}_p^*$
3. Επιλογή ιδιωτικού κλειδιού  $x \in \mathbb{Z}_q$ .
4. Υπολογισμός  $g^x \bmod p$ .

# Παραγωγή κλειδιών DSS

1. Επιλογή πρώτων  $q$  μεγέθους 160-bit και  $p$  μεγέθους  $n$ -bit,  $n = 64\lambda$ ,  $\lambda = 8, 9, 10, \dots, 16$ , με  $q \mid (p - 1)$ .
2. Εύρεση  $g$  γεννήτορα της υποομάδας τάξης  $q$  του  $\mathbb{Z}_p^*$
3. Επιλογή ιδιωτικού κλειδιού  $x \in \mathbb{Z}_q$ .
4. Υπολογισμός  $g^x \bmod p$ .

Δημόσιο κλειδί:  $(p, q, g, y), y = g^x \bmod p$ .

Ιδιωτικό κλειδί:  $x$ .



1. Ο υπογράφων επιλέγει έναν τυχαίο ακέραιο  $k$ ,  $1 \leq k \leq (q - 1)$ .

1. Ο υπογράφων επιλέγει έναν τυχαίο ακέραιο  $k$ ,  $1 \leq k \leq (q - 1)$ .
2. Υπολογίζει τα

1. Ο υπογράφων επιλέγει έναν τυχαίο ακέραιο  $k$ ,  $1 \leq k \leq (q - 1)$ .
2. Υπολογίζει τα

$$r = (g^k \bmod p) \bmod q$$
$$s = (\mathcal{H}(m) + x \cdot r)k^{-1} \bmod q$$

3. Αν συμβεί  $r, s \equiv 0 \pmod{q}$  η διαδικασία επαναλαμβάνεται

1. Ο υπογράφων επιλέγει έναν τυχαίο ακέραιο  $k$ ,  $1 \leq k \leq (q - 1)$ .
2. Υπολογίζει τα

$$r = (g^k \bmod p) \bmod q$$
$$s = (\mathcal{H}(m) + x \cdot r)k^{-1} \bmod q$$

3. Αν συμβεί  $r, s \equiv 0 \pmod{q}$  η διαδικασία επαναλαμβάνεται
4. Υπογραφή:  $(r, s)$ .

Ο  $B$  υπολογίζει:

$$h = \mathcal{H}(m)$$

$$e_1 = s^{-1}h \bmod q$$

$$e_2 = rs^{-1} \bmod q$$

Ο  $B$  υπολογίζει:

$$h = \mathcal{H}(m)$$

$$e_1 = s^{-1}h \bmod q$$

$$e_2 = rs^{-1} \bmod q$$

$$\text{Verify}(y, m, (r, s)) = 1 \Leftrightarrow (g^{e_1}(y)^{e_2} \bmod p) \bmod q = r$$

Ο  $B$  υπολογίζει:

$$h = \mathcal{H}(m)$$

$$e_1 = s^{-1}h \bmod q$$

$$e_2 = rs^{-1} \bmod q$$

$$\text{Verify}(y, m, (r, s)) = 1 \Leftrightarrow (g^{e_1}(y)^{e_2} \bmod p) \bmod q = r$$

Ορθότητα

$$\begin{aligned} g^{e_1}(y)^{e_2} &= g^{hs^{-1}} \cdot g^{xrs^{-1}} \\ g^{hs^{-1}+xrs^{-1}} &= g^{(h+xr)s^{-1}} = \\ g^{kss^{-1}} &= g^k \pmod{p \bmod q} \end{aligned}$$

Ο  $B$  υπολογίζει:

$$h = \mathcal{H}(m)$$

$$e_1 = s^{-1}h \bmod q$$

$$e_2 = rs^{-1} \bmod q$$

$$\text{Verify}(y, m, (r, s)) = 1 \Leftrightarrow (g^{e_1}(y)^{e_2} \bmod p) \bmod q = r$$

Ορθότητα

$$\begin{aligned} g^{e_1}(y)^{e_2} &= g^{hs^{-1}} \cdot g^{xrs^{-1}} \\ g^{hs^{-1}+xrs^{-1}} &= g^{(h+xr)s^{-1}} = \\ g^{kss^{-1}} &= g^k \pmod{p \bmod q} \end{aligned}$$



## Υποδομή Δημοσίου Κλειδιού

---

- Διαφορά Συμμετρικών - Ασύμμετρων Κρυπτοσυστημάτων
  - Συμμετρικά: Δύσκολη διανομή, Εύκολη Αυθεντικότητα (λόγω φυσικών υποθέσεων)
  - Ασύμμετρα: Εύκολη διανομή, Δύσκολη Αυθεντικότητα
- Αντιστοιχία (?) Ταυτότητας Χρήστη - Δημοσίου, Ιδιωτικού Κλειδιού (binding)

# Πρακτική χρήση ψηφιακών υπογραφών

- Διαφορά Συμμετρικών - Ασύμμετρων Κρυπτοσυστημάτων
  - Συμμετρικά: Δύσκολη διανομή, Εύκολη Αυθεντικότητα (λόγω φυσικών υποθέσεων)
  - Ασύμμετρα: Εύκολη διανομή, Δύσκολη Αυθεντικότητα
- Αντιστοιχία (?) Ταυτότητας Χρήστη - Δημοσίου, Ιδιωτικού Κλειδιού (binding)
- Ενεργός αντίπαλος - Πλαστοπροσωπία - αλλαγή κλειδιών

- Διαφορά Συμμετρικών - Ασύμμετρων Κρυπτοσυστημάτων
  - Συμμετρικά: Δύσκολη διανομή, Εύκολη Αυθεντικότητα (λόγω φυσικών υποθέσεων)
  - Ασύμμετρα: Εύκολη διανομή, Δύσκολη Αυθεντικότητα
- Αντιστοιχία (?) Ταυτότητας Χρήστη - Δημοσίου, Ιδιωτικού Κλειδιού (binding)
- Ενεργός αντίπαλος - Πλαστοπροσωπία - αλλαγή κλειδιών
- Απαραίτητη η διασφάλιση για χρήση σε ευρεία κλίμακα

# Πρακτική χρήση ψηφιακών υπογραφών

- Διαφορά Συμμετρικών - Ασύμμετρων Κρυπτοσυστημάτων
  - Συμμετρικά: Δύσκολη διανομή, Εύκολη Αυθεντικότητα (λόγω φυσικών υποθέσεων)
  - Ασύμμετρα: Εύκολη διανομή, Δύσκολη Αυθεντικότητα
- Αντιστοιχία (?) Ταυτότητας Χρήστη - Δημοσίου, Ιδιωτικού Κλειδιού (binding)
- Ενεργός αντίπαλος - Πλαστοπροσωπία - αλλαγή κλειδιών
- Απαραίτητη η διασφάλιση για χρήση σε ευρεία κλίμακα
- **Δεν υπάρχει λύση** που να δουλεύει θεωρητικά **και** πρακτικά

# Πρακτική χρήση ψηφιακών υπογραφών

- Διαφορά Συμμετρικών - Ασύμμετρων Κρυπτοσυστημάτων
  - Συμμετρικά: Δύσκολη διανομή, Εύκολη Αυθεντικότητα (λόγω φυσικών υποθέσεων)
  - Ασύμμετρα: Εύκολη διανομή, Δύσκολη Αυθεντικότητα
- Αντιστοιχία (?) Ταυτότητας Χρήστη - Δημοσίου, Ιδιωτικού Κλειδιού (binding)
- Ενεργός αντίπαλος - Πλαστοπροσωπία - αλλαγή κλειδιών
- Απαραίτητη η διασφάλιση για χρήση σε ευρεία κλίμακα
- **Δεν υπάρχει λύση** που να δουλεύει θεωρητικά **και** πρακτικά
- Στην πράξη: μετάθεση του προβλήματος με μείωση της έκτασης (αρκεί 1 αυθεντικό κλειδί)

## Αρχές Πιστοποίησης (Certification Authorities - CAs)

- Έμπιστες Τρίτες Οντότητες - (Πάροχοι Υπηρεσιών Πιστοποίησης)

# Αρχές Πιστοποίησης (Certification Authorities - CAs)

- Έμπιστες Τρίτες Οντότητες - (Πάροχοι Υπηρεσιών Πιστοποίησης)
  - Πιστοποίηση Αντιστοιχίας Ταυτότητας Κλειδιών



# Αρχές Πιστοποίησης (Certification Authorities - CAs)

- Έμπιστες Τρίτες Οντότητες - (Πάροχοι Υπηρεσιών Πιστοποίησης)
  - Πιστοποίηση Αντιστοιχίας Ταυτότητας Κλειδιών
  - Εγγυάται ότι το δημόσιο κλειδί όντως αντιστοιχεί στον χρήστη

- Έμπιστες Τρίτες Οντότητες - (Πάροχοι Υπηρεσιών Πιστοποίησης)
  - Πιστοποίηση Αντιστοιχίας Ταυτότητας Κλειδιών
  - Εγγυάται ότι το δημόσιο κλειδί όντως αντιστοιχεί στον χρήστη
  - Πώς;

- Έμπιστες Τρίτες Οντότητες - (Πάροχοι Υπηρεσιών Πιστοποίησης)
  - Πιστοποίηση Αντιστοιχίας Ταυτότητας Κλειδιών
  - Εγγυάται ότι το δημόσιο κλειδί όντως αντιστοιχεί στον χρήστη
  - Πώς;
  - Υπογράφοντας 'ψηφιακά' το ζεύγος ( $ID, PK_{ID}$ )

# Αρχές Πιστοποίησης (Certification Authorities - CAs)

- Έμπιστες Τρίτες Οντότητες - (Πάροχοι Υπηρεσιών Πιστοποίησης)
  - Πιστοποίηση Αντιστοιχίας Ταυτότητας Κλειδιών
  - Εγγυάται ότι το δημόσιο κλειδί όντως αντιστοιχεί στον χρήστη
  - Πώς;
  - Υπογράφοντας 'ψηφιακά' το ζεύγος ( $ID, PK_{ID}$ )
- **Πλεονέκτημα:** Μείωση κλειδιών που πρέπει να αποκτήσουμε με έμπιστο τρόπο
  - Μόνο το κλειδί της CA
  - Για τα υπόλοιπα 'εγγυάται' το πιστοποιητικό
- **Μειονέκτημα** Ποιος εγγυάται την σχέση κλειδιών-ταυτότητας για την CA;

# Αρχές Πιστοποίησης (Certification Authorities - CAs)

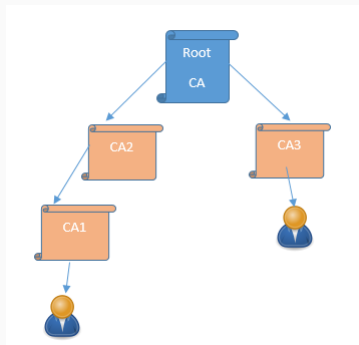
- Έμπιστες Τρίτες Οντότητες - (Πάροχοι Υπηρεσιών Πιστοποίησης)
  - Πιστοποίηση Αντιστοιχίας Ταυτότητας Κλειδιών
  - Εγγυάται ότι το δημόσιο κλειδί όντως αντιστοιχεί στον χρήστη
  - Πώς;
  - Υπογράφοντας 'ψηφιακά' το ζεύγος ( $ID, PK_{ID}$ )
- **Πλεονέκτημα:** Μείωση κλειδιών που πρέπει να αποκτήσουμε με έμπιστο τρόπο
  - Μόνο το κλειδί της CA
  - Για τα υπόλοιπα 'εγγυάται' το πιστοποιητικό
- **Μειονέκτημα** Ποιος εγγυάται την σχέση κλειδιών-ταυτότητας για την CA;
  - Η ίδια! (υπογράφει η ίδια μία δήλωση για τον εαυτό της)

# Αρχές Πιστοποίησης (Certification Authorities - CAs)

- Έμπιστες Τρίτες Οντότητες - (Πάροχοι Υπηρεσιών Πιστοποίησης)
  - Πιστοποίηση Αντιστοιχίας Ταυτότητας Κλειδιών
  - Εγγυάται ότι το δημόσιο κλειδί *όντως* αντιστοιχεί στον χρήστη
  - Πώς;
  - Υπογράφοντας 'ψηφιακά' το ζεύγος (*ID, PK<sub>ID</sub>*)
- **Πλεονέκτημα:** Μείωση κλειδιών που πρέπει να αποκτήσουμε με έμπιστο τρόπο
  - Μόνο το κλειδί της CA
  - Για τα υπόλοιπα 'εγγυάται' το πιστοποιητικό
- **Μειονέκτημα** Ποιος εγγυάται την σχέση κλειδιών-ταυτότητας για την CA;
  - Η ίδια! (υπογράφει η ίδια μία δήλωση για τον εαυτό της)
  - ή μια άλλη *ανώτερη* αρχή πιστοποίησης!

# Ιεραρχική Οργάνωση Αρχών Πιστοποίησης

- Ενδιάμεσες Αρχές: Υπογραφή από ανώτερη αρχή
- Ριζικές (Root) Αρχές: Υπογράφουν μόνες τους
- Συνήθως 3-4 επίπεδα



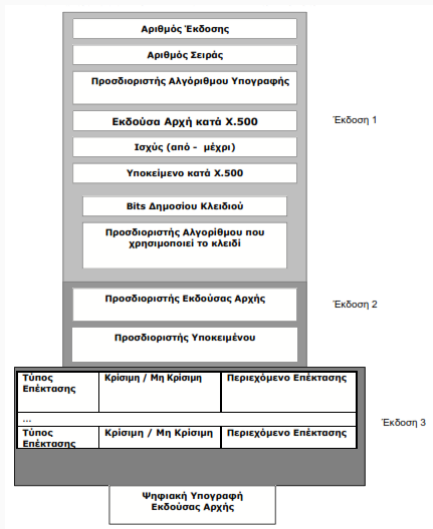
- Οργάνωση των αρχών πιστοποίησης και των σχετικών υπηρεσιών



- Οργάνωση των αρχών πιστοποίησης και των σχετικών υπηρεσιών
- Loren Kohnfelder, MIT BSc thesis, 1978

- Οργάνωση των αρχών πιστοποίησης και των σχετικών υπηρεσιών
- Loren Kohnfelder, MIT BSc thesis, 1978
- Ευρεία προτυποποίηση (ITU X.500, RFC 6818)
  - Πρόσβαση σε υπηρεσίες καταλόγου
  - X.509: Συσχέτιση οντότητας με δημόσιο κλειδί
  - Ψηφιακό Πιστοποιητικό:
    - Δήλωση σχέσης κλειδιού - ονόματος
    - Επιπλέον πληροφορίες για την επαλήθευση

# Πιστοποιητικό X.509 - Δομή



# Πιστοποιητικό X.509 - Παράδειγμα

```
X509 Certificate:
Version: 3
Serial Number: 104e764f15ebc89
Signature Algorithm:
  Algorithm ObjectID: 1.2.840.113549.1.1.11 sha256RSA
  Algorithm Parameters:
    05 00
Issuer:
  CN=Google Internet Authority G2
  O=Google Inc
  C=US
  Name Hash (sha1): f2e06af9858a1d8d709b4919237aa9b51a287e64
  Name Hash (md5): 00656cd744ec6221c3df38867186e4bb

NotBefore: 10/11/2016 18:00 μμ
NotAfter: 2/2/2017 17:31 μμ

Subject:
  CN=*.google.gr
  O=Google Inc
  L=Mountain View
  S=California
  C=US
  Name Hash (sha1): cdf2f8396ae4d2eade922220752f946e252573c7
  Name Hash (md5): a06f981af315c8f987c48945a82f6335

Public Key Algorithm:
  Algorithm ObjectID: 1.2.840.113549.1.1.1 RSA (RSA_SIGN)
  Algorithm Parameters:
    05 00
Public Key Length: 2048 bits
Public Key: UnusedBits = 0
0000 30 82 01 0a 02 82 01 01 00 b3 82 58 8f cd e0 0c
0010 18 75 1a 4f b2 85 99 88 ac 71 c7 0f aa db cd f3
0020 3c e9 a1 1e ba cc 7b 73 d4 8f b9 1d 28 04 a1 54
0030 4d 36 29 c1 e3 77 68 5b 0e 98 1e cd 89 f4 02 2f
0040 1a 0d d9 12 33 ec aa 26 d2 f2 4f cb 1b 7b 62 e5
0050 b4 03 74 33 57 19 22 ba bd de 9f 89 eb 4e 21 22
0060 c5 c4 1c fd 6e a5 a0 ae ad 1e fd 93 ec e4 0b a2
0070 62 fd e9 44 ef 01 97 c1 bb c0 23 88 ca e9 9b 16
0080 54 c8 54 7b 65 bd 32 e7 54 ba 73 ed fc 2e b5 39
0090 57 fd 4b c8 fd 97 33 b2 e0 03 55 2a db 5c 2d 1d
00a0 9e 70 e7 86 21 11 4f 8c e8 53 52 ed 9a 95 be 81
00b0 84 ed 2c dc 8d 18 0b 67 ef b5 af 4e 3f 47 a7 4e
00c0 6a 4c c3 ca 20 14 f 4e 20 cf 5c 01 a3 fa da f0
```

```
Certificate Extensions: 8
2.5.29.37: Flags = 0, Length = 16
  Enhanced Key Usage
    Server Authentication (1.3.6.1.5.5.7.3.1)
    Client Authentication (1.3.6.1.5.5.7.3.2)
2.5.29.17: Flags = 0, Length = 1a
  Subject Alternative Name
    DNS Name=*.google.gr
    DNS Name=google.gr
1.3.6.1.5.5.7.1.1: Flags = 0, Length = 5c
  Authority Information Access
    [1]Authority Info Access
      Access Method=Certification Authority Issuer (1.3.6.1.5.5.7.48.2)
      Alternative Name:
        URL=http://pki.google.com/GIAG2.crt
    [2]Authority Info Access
      Access Method=On-line Certificate Status Protocol (1.3.6.1.5.5.7.4)
      Alternative Name:
        URL=http://clients1.google.com/ocsp
2.5.29.14: Flags = 0, Length = 16
  Subject Key Identifier
    84 37 bc c5 bd 97 a3 33 92 8c 49 06 43 15 ce b7 6b 84 f2 0c
2.5.29.19: Flags = 1(Critical), Length = 2
  Basic Constraints
    Subject Type=End Entity
    Path Length Constraint=None
2.5.29.35: Flags = 0, Length = 18
  Authority Key Identifier
    KeyID=4a dd 06 16 1b bc f6 68 b5 76 f5 81 b6 bb 62 1a ba 5a 81 2f
2.5.29.32: Flags = 0, Length = 1a
  Certificate Policies
    [1]Certificate Policy:
      Policy Identifier=1.3.6.1.4.1.11129.2.5.1
    [2]Certificate Policy:
      Policy Identifier=2.23.140.1.2.2
2.5.29.31: Flags = 0, Length = 29
  CRL Distribution Points
    [1]CRL Distribution Point
```

- Προεγκατάσταση στο λειτουργικό σύστημα

# Απόκτηση πιστοποιητικών

- Προεγκατάσταση στο λειτουργικό σύστημα
- Προεγκατάσταση στον περιηγητή

# Απόκτηση πιστοποιητικών

- Προεγκατάσταση στο λειτουργικό σύστημα
- Προεγκατάσταση στον περιηγητή
- Απόκτηση από αρχείο/ιστοσελίδα

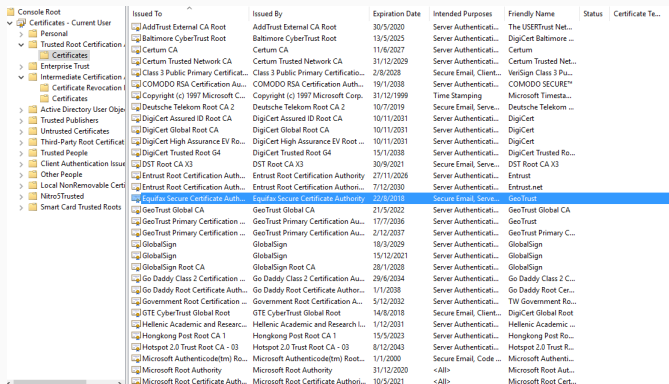
# Απόκτηση πιστοποιητικών

- Προεγκατάσταση στο λειτουργικό σύστημα
- Προεγκατάσταση στον περιηγητή
- Απόκτηση από αρχείο/ιστοσελίδα
- Απόκτηση από νομική οντότητα (εταιρεία, κράτος)



# Απόκτηση πιστοποιητικών

- Προεγκατάσταση στο λειτουργικό σύστημα
- Προεγκατάσταση στον περιηγητή
- Απόκτηση από αρχείο/ιστοσελίδα
- Απόκτηση από νομική οντότητα (εταιρεία, κράτος)



Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name	Status	Certificate Te...
AddTrust External CA Root	AddTrust External CA Root	30/5/2020	Server Authenticati...	The USERTrust Net...		
Baltimore CyberTrust Root	Baltimore CyberTrust Root	13/5/2025	Server Authenticati...	DigiCert Baltimore ...		
Certum CA	Certum CA	11/6/2027	Server Authenticati...	Certum		
Certum Trusted Network CA	Certum Trusted Network CA	31/12/2029	Server Authenticati...	Certum Trusted Net...		
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	2/8/2028	Secure Email, Client...	VeriSign Class 3 Pu...		
COMODO RSA Certification Au...	COMODO RSA Certification Auth...	19/1/2038	Server Authenticati...	COMODO SECURE™		
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.	31/12/1999	Time Stamping	Microsoft Timesta...		
Deutsche Telekom Root CA 2	Deutsche Telekom Root CA 2	10/7/2019	Secure Email, Serve...	Deutsche Telekom ...		
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	10/11/2031	Server Authenticati...	DigiCert		
DigiCert Global Root CA	DigiCert Global Root CA	10/11/2031	Server Authenticati...	DigiCert		
DigiCert High Assurance EV Ro...	DigiCert High Assurance EV Root ...	10/11/2031	Server Authenticati...	DigiCert		
DigiCert Trusted Root G4	DigiCert Trusted Root G4	15/1/2038	Server Authenticati...	DigiCert Trusted Re...		
DST Root CA X3	DST Root CA X3	30/9/2021	Secure Email, Serve...	DST Root CA X3		
Entrust Root Certification Auth...	Entrust Root Certification Authority	27/11/2026	Server Authenticati...	Entrust		
Entrust Root Certification Auth...	Entrust Root Certification Authori...	7/12/2030	Server Authenticati...	Entrust.net		
Equifax Secure Certificate Auth...	Equifax Secure Certificate Authority	22/8/2018	Secure Email, Sinc...	GeoTrust		
GeoTrust Global CA	GeoTrust Global CA	21/5/2022	Server Authenticati...	GeoTrust Global CA		
GeoTrust Primary Certification ...	GeoTrust Primary Certification Au...	17/7/2036	Server Authenticati...	GeoTrust		
GeoTrust Primary Certification ...	GeoTrust Primary Certification Au...	2/12/2037	Server Authenticati...	GeoTrust Primary C...		
GlobalSign	GlobalSign	18/3/2029	Server Authenticati...	GlobalSign		
GlobalSign	GlobalSign	15/12/2021	Server Authenticati...	GlobalSign		
GlobalSign Root CA	GlobalSign Root CA	28/1/2028	Server Authenticati...	GlobalSign		
Go Daddy Class 2 Certification Au...	Go Daddy Class 2 Certification Au...	29/8/2034	Server Authenticati...	Go Daddy Class 2 C...		
Go Daddy Root Certificate Auth...	Go Daddy Root Certificate Author...	1/1/2038	Server Authenticati...	Go Daddy Root Cer...		
Government Root Certification ...	Government Root Certification A...	5/12/2032	Server Authenticati...	TW Government Ro...		
GTE CyberTrust Global Root	GTE CyberTrust Global Root	14/8/2018	Secure Email, Client...	DigiCert Global Root		
Hellenic Academic and Researc...	Hellenic Academic and Researh...	1/12/2031	Server Authenticati...	Hellenic Academic ...		
Hongkong Post Root CA 1	Hongkong Post Root CA 1	15/5/2023	Server Authenticati...	Hongkong Post Ro...		
Hotspot 2.0 Trust Root CA - 03	Hotspot 2.0 Trust Root CA - 03	8/12/2043	Server Authenticati...	Hotspot 2.0 Trust R...		
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root...	1/1/2000	Secure Email, Code ...	Microsoft Authenti...		
Microsoft Root Authority	Microsoft Root Authority	31/12/2020	<All>	Microsoft Root Aut...		
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	10/5/2021	<All>	Microsoft Root Cert...		

- Διάδοση Πιστοποιητικών σε αποθετήρια

- Διάδοση Πιστοποιητικών σε αποθετήρια
- Εγγραφή-Επαλήθευση Ταυτότητας Χρηστών

- Διάδοση Πιστοποιητικών σε αποθετήρια
- Εγγραφή-Επαλήθευση Ταυτότητας Χρηστών
- Δημιουργία κρυπτογραφικών κλειδιών (αυστηρές προδιαγραφές ασφάλειας)

- Διάδοση Πιστοποιητικών σε αποθετήρια
- Εγγραφή-Επαλήθευση Ταυτότητας Χρηστών
- Δημιουργία κρυπτογραφικών κλειδιών (αυστηρές προδιαγραφές ασφάλειας)
- Ανάκληση Πιστοποιητικών - Ενημέρωση

- Διάδοση Πιστοποιητικών σε αποθετήρια
- Εγγραφή-Επαλήθευση Ταυτότητας Χρηστών
- Δημιουργία κρυπτογραφικών κλειδιών (αυστηρές προδιαγραφές ασφάλειας)
- Ανάκληση Πιστοποιητικών - Ενημέρωση
- Χρονοσήμανση - Αρχαιοθέτηση

Άκυρα πιστοποιητικά

- Απώλεια κλειδιού υπογραφής, Αλλαγή Στοιχείων Υποκειμένου,

## Άκυρα πιστοποιητικά

- Απώλεια κλειδιού υπογραφής, Αλλαγή Στοιχείων Υποκειμένου,
- Ενημέρωση Χρηστών με 2 τρόπους



## Άκυρα πιστοποιητικά

- Απώλεια κλειδιού υπογραφής, Αλλαγή Στοιχείων Υποκειμένου,
- Ενημέρωση Χρηστών με 2 τρόπους
- Certificate Revocation Lists (CRL):
  - 'Μαύρη' λίστα από SN για πιστοποιητικά που δεν ισχύουν
  - Υπογεγραμμένη από την CA
  - Ανάκτηση σε τακτά χρονικά διαστήματα
  - Πεδίο CDP

## Άκυρα πιστοποιητικά

- Απώλεια κλειδιού υπογραφής, Αλλαγή Στοιχείων Υποκειμένου,
- Ενημέρωση Χρηστών με 2 τρόπους
- Certificate Revocation Lists (CRL):
  - ‘Μαύρη’ λίστα από SN για πιστοποιητικά που δεν ισχύουν
  - Υπογεγραμμένη από την CA
  - Ανάκτηση σε τακτά χρονικά διαστήματα
  - Πεδίο CDP
- OCSP (Online Certificate Status Protocol)
  - Ερώτηση στην CA για ισχύ πιστοποιητικού
  - Η CA συμμετέχει σε κάθε συναλλαγή

Ομότιμη έκδοση και επαλήθευση ταυτότητας (web of trust)

- Κάθε χρήστης είναι CA

Ομότιμη έκδοση και επαλήθευση ταυτότητας (web of trust)

- Κάθε χρήστης είναι CA
- Υπογράφει αντιστοιχίες που γνωρίζει

Ομότιμη έκδοση και επαλήθευση ταυτότητας (web of trust)

- Κάθε χρήστης είναι CA
- Υπογράφει αντιστοιχίες που γνωρίζει
- Λήψη πιστοποιητικών μόνο από γνωστούς χρήστες

Ομότιμη έκδοση και επαλήθευση ταυτότητας (web of trust)

- Κάθε χρήστης είναι CA
- Υπογράφει αντιστοιχίες που γνωρίζει
- Λήψη πιστοποιητικών μόνο από γνωστούς χρήστες
- Ο κάθε χρήστης 'εγγυάται' για τους γνωστούς του

Ομότιμη έκδοση και επαλήθευση ταυτότητας (web of trust)

- Κάθε χρήστης είναι CA
- Υπογράφει αντιστοιχίες που γνωρίζει
- Λήψη πιστοποιητικών μόνο από γνωστούς χρήστες
- Ο κάθε χρήστης 'εγγυάται' για τους γνωστούς του
- PGP

- Signatures: Shamir 1984



# Identity based cryptography

- Signatures: Shamir 1984
- Encryption: Boneh-Franklin (2001)

- Signatures: Shamir 1984
- Encryption: Boneh-Franklin (2001)
- Οποιοδήποτε όνομα κάποιου χρήστη πχ. email είναι η ταυτότητα

# Identity based cryptography

- Signatures: Shamir 1984
- Encryption: Boneh-Franklin (2001)
- Οποιοδήποτε όνομα κάποιου χρήστη πχ. email είναι η ταυτότητα
- Δεν χρειάζεται διανομή κλειδιού

# Identity based cryptography

- Signatures: Shamir 1984
- Encryption: Boneh-Franklin (2001)
- Οποιοδήποτε όνομα κάποιου χρήστη πχ. email είναι η ταυτότητα
- Δεν χρειάζεται διανομή κλειδιού
- Χρειάζεται κεντρική ΤΤΡ

- Signatures: Shamir 1984
- Encryption: Boneh-Franklin (2001)
- Οποιοδήποτε όνομα κάποιου χρήστη πχ. email είναι η ταυτότητα
- Δεν χρειάζεται διανομή κλειδιού
- Χρειάζεται κεντρική ΤΤΡ
- Παράγει τα ιδιωτικά κλειδιά από την ταυτότητα

- Signatures: Shamir 1984
- Encryption: Boneh-Franklin (2001)
- Οποιοδήποτε όνομα κάποιου χρήστη πχ. email είναι η ταυτότητα
- Δεν χρειάζεται διανομή κλειδιού
- Χρειάζεται κεντρική ΤΤΡ
- Παράγει τα ιδιωτικά κλειδιά από την ταυτότητα

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$



# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο
- Υπογραφή από χρήστη  $id$

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο
- Υπογραφή από χρήστη  $id$ 
  - Επιλογή τυχαίου  $r$

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο
- Υπογραφή από χρήστη  $id$ 
  - Επιλογή τυχαίου  $r$
  - $t = r^e \bmod n$



# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο
- Υπογραφή από χρήστη  $id$ 
  - Επιλογή τυχαίου  $r$
  - $t = r^e \bmod n$
  - $s = k r^{\mathcal{H}(m|t)} \bmod n$

# Identity based signatures

- TTP έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο
- Υπογραφή από χρήστη  $id$ 
  - Επιλογή τυχαίου  $r$
  - $t = r^e \bmod n$
  - $s = k r^{\mathcal{H}(m|t)} \bmod n$
  - Η υπογραφή είναι  $(t, s)$

# Identity based signatures

- ΤΤΡ έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο
- Υπογραφή από χρήστη  $id$ 
  - Επιλογή τυχαίου  $r$
  - $t = r^e \bmod n$
  - $s = k r^{\mathcal{H}(m|t)} \bmod n$
  - Η υπογραφή είναι  $(t, s)$
- Επαλήθευση υπογραφής με την ταυτότητα:

# Identity based signatures

- ΤΤΡ έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο
- Υπογραφή από χρήστη  $id$ 
  - Επιλογή τυχαίου  $r$
  - $t = r^e \bmod n$
  - $s = k r^{\mathcal{H}(m|t)} \bmod n$
  - Η υπογραφή είναι  $(t, s)$
- Επαλήθευση υπογραφής με την ταυτότητα:
- Έλεγχος αν:  $\mathcal{H}(id)t^{\mathcal{H}(m|t)} = s^e$

# Identity based signatures

- ΤΡΡ έχει κλειδί RSA  $((e, n), d)$
- Δημιουργία ιδιωτικού κλειδιού από ταυτότητα χρήστη  $id$ 
  - Υπογραφή σύνοψης της ταυτότητας
  - $k = \mathcal{H}(id)^d \bmod n$
  - Ασφαλής Διανομή στον κάτοχο
- Υπογραφή από χρήστη  $id$ 
  - Επιλογή τυχαίου  $r$
  - $t = r^e \bmod n$
  - $s = k r^{\mathcal{H}(m|t)} \bmod n$
  - Η υπογραφή είναι  $(t, s)$
- Επαλήθευση υπογραφής με την ταυτότητα:
- Έλεγχος αν:  $\mathcal{H}(id)t^{\mathcal{H}(m|t)} = s^e$
- Ορθότητα:  $\mathcal{H}(id)t^{\mathcal{H}(m|t)} = k^e r^{e\mathcal{H}(m|t)} = s^e$

# Secret Sharing - Threshold Cryptosystems

---

## Το πρόβλημα

Κλειδιά: κρίσιμα κρυπτογραφικά δεδομένα (όχι τα μόνα)

## Το πρόβλημα

Κλειδιά: κρίσιμα κρυπτογραφικά δεδομένα (όχι τα μόνα)

Για παράδειγμα: ιδιωτικό κλειδί

- Δύναμη αποκρυπτογράφησης
- Δύναμη υπογραφής



## 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)

Βασικό συστατικό Secure Multi Party Computation

## Additive secret sharing

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

# 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  $s_1, \dots, s_{n-1} \in \mathbb{G}$

# 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  $s_1, \dots, s_{n-1} \in \mathbb{G}$
- Θέτουμε  $s_n = s - \sum_{i=1}^{n-1} s_i$

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

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

# 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  $s_1, \dots, s_{n-1} \in \mathbb{G}$
- 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

# 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  $s_1, \dots, s_{n-1} \in \mathbb{G}$
- 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

$$s_n = s \oplus \left( \bigoplus_{i=1}^{n-1} s_i \right)$$



# 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  $s_1, \dots, s_{n-1} \in \mathbb{G}$
- 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

$$s_n = s \oplus \left( \bigoplus_{i=1}^{n-1} s_i \right)$$

**Ασφάλεια:** Κανένα υποσύνολο από  $n - 1$  παίκτες δεν μπορεί να ανακατασκευάσει το  $s$

# 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  $s_1, \dots, s_{n-1} \in \mathbb{G}$
- 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

$$s_n = s \oplus \left( \bigoplus_{i=1}^{n-1} s_i \right)$$

**Ασφάλεια:** Κανένα υποσύνολο από  $n - 1$  παίκτες δεν μπορεί να ανακατασκευάσει το  $s$

**Πρόβλημα:** Ένας παίκτης μπορεί να ακυρώσει την ανακατασκευή

Παραλλαγή για ευελιξία:  $(t, n)$  threshold secret sharing

- Ένα μυστικό  $s$  πρέπει να μοιραστεί σε  $n$  παίκτες  $P_1, P_2, \dots, P_n$  ώστε:

Παραλλαγή για ευελιξία:  $(t, n)$  threshold secret sharing

- Ένα μυστικό  $s$  πρέπει να μοιραστεί σε  $n$  παίκτες  $P_1, P_2, \dots, P_n$  ώστε:
  - Οποιοδήποτε υποσύνολο από τουλάχιστον  $t$  παίκτες να μπορεί να το ανακτήσει

Παραλλαγή για ευελιξία:  $(t, n)$  threshold secret sharing

- Ένα μυστικό  $s$  πρέπει να μοιραστεί σε  $n$  παίκτες  $P_1, P_2, \dots, P_n$  ώστε:
  - Οποιοδήποτε υποσύνολο από τουλάχιστον  $t$  παίκτες να μπορεί να το ανακτήσει
  - Κανένα υποσύνολο με  $t - 1$  παίκτες να μην μπορεί

Παραλλαγή για ευελιξία:  $(t, n)$  threshold secret sharing

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

Λύση: **Shamir secret sharing** - Βασίζεται σε πολυώνυμο σε πεπερασμένο σώμα  $\mathbb{F}$  με  $s \in \mathbb{F}, |\mathbb{F}| > n$

# Πολυωνυμική παρεμβολή

- Έστω ένα πολυώνυμο βαθμού  $t - 1$ :

$$p(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$$

# Πολυωνυμική παρεμβολή

- Έστω ένα πολυώνυμο βαθμού  $t - 1$ :

$$p(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$$

- Μπορεί να ανακατασκευαστεί από  $t$  σημεία  $(x_i, p(x_i))$  με διαφορετικές τετμημένες (με μοναδικό τρόπο)



# Πολυωνυμική παρεμβολή

- Έστω ένα πολυώνυμο βαθμού  $t - 1$ :

$$p(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$$

- Μπορεί να ανακατασκευαστεί από  $t$  σημεία  $(x_i, p(x_i))$  με διαφορετικές τετμημένες (με μοναδικό τρόπο)
- Υπάρχουν άπειρα πολυώνυμα βαθμού  $t$  που περνούν από  $t$  σημεία

# Πολυωνυμική παρεμβολή

- Έστω ένα πολυώνυμο βαθμού  $t - 1$ :  
$$p(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$$
- Μπορεί να ανακατασκευαστεί από  $t$  σημεία  $(x_i, p(x_i))$  με διαφορετικές τετμημένες (με μοναδικό τρόπο)
- Υπάρχουν άπειρα πολυώνυμα βαθμού  $t$  που περνούν από  $t$  σημεία
- Υπάρχει μοναδικό πολυώνυμο βαθμού  $t - 1$  που περνά από  $t$  σημεία

# Πολυωνυμική παρεμβολή

- Έστω ένα πολυώνυμο βαθμού  $t - 1$ :  
$$p(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$$
- Μπορεί να ανακατασκευαστεί από  $t$  σημεία  $(x_i, p(x_i))$  με διαφορετικές τετμημένες (με μοναδικό τρόπο)
- Υπάρχουν άπειρα πολυώνυμα βαθμού  $t$  που περνούν από  $t$  σημεία
- Υπάρχει μοναδικό πολυώνυμο βαθμού  $t - 1$  που περνά από  $t$  σημεία
- Κατασκευή με συντελεστές Lagrange

# Πολυωνυμική παρεμβολή

- Έστω ένα πολυώνυμο βαθμού  $t - 1$ :  
$$p(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$$
- Μπορεί να ανακατασκευαστεί από  $t$  σημεία  $(x_i, p(x_i))$  με διαφορετικές τετμημένες (με μοναδικό τρόπο)
- Υπάρχουν άπειρα πολυώνυμα βαθμού  $t$  που περνούν από  $t$  σημεία
- Υπάρχει μοναδικό πολυώνυμο βαθμού  $t - 1$  που περνά από  $t$  σημεία
- Κατασκευή με συντελεστές Lagrange
- $$\lambda_i(x) = \prod_{k=1, k \neq i}^t \frac{x - x_k}{x_i - x_k}$$

# Πολυωνυμική παρεμβολή

- Έστω ένα πολυώνυμο βαθμού  $t - 1$ :  
$$p(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$$
- Μπορεί να ανακατασκευαστεί από  $t$  σημεία  $(x_i, p(x_i))$  με διαφορετικές τετμημένες (με μοναδικό τρόπο)
- Υπάρχουν άπειρα πολυώνυμα βαθμού  $t$  που περνούν από  $t$  σημεία
- Υπάρχει μοναδικό πολυώνυμο βαθμού  $t - 1$  που περνά από  $t$  σημεία
- Κατασκευή με συντελεστές Lagrange
- $\lambda_i(x) = \prod_{k=1, k \neq i}^t \frac{x - x_k}{x_i - x_k}$
- Προκύπτει το  
$$L(x) = \sum_{i=1}^t p(x_i) \lambda_i(x) = p(x_1) \lambda_1(x) + \dots + p(x_t) \lambda_t(x)$$

Υποθέτουμε ότι διαθέτουμε έναν έμπιστο διανομέα:

- Επιλέγει και δημοσιοποιεί ένα πρώτο  $p$

# Shamir secret sharing: Διανομή

Υποθέτουμε ότι διαθέτουμε έναν έμπιστο διανομέα:

- Επιλέγει και δημοσιοποιεί ένα πρώτο  $p$
- Επιλέγει  $t - 1$  συντελεστές ενός πολυωνύμου βαθμού  $t$   
 $\{a_{t-1}, \dots, a_1\} \in_R \mathbb{Z}_p$

Υποθέτουμε ότι διαθέτουμε έναν έμπιστο διανομέα:

- Επιλέγει και δημοσιοποιεί ένα πρώτο  $p$
- Επιλέγει  $t - 1$  συντελεστές ενός πολυωνύμου βαθμού  $t$   
 $\{a_{t-1}, \dots, a_1\} \in_R \mathbb{Z}_p$
- Θέτει ως σταθερό όρο το μυστικό  $s$



# Shamir secret sharing: Διανομή

Υποθέτουμε ότι διαθέτουμε έναν έμπιστο διανομέα:

- Επιλέγει και δημοσιοποιεί ένα πρώτο  $p$
- Επιλέγει  $t - 1$  συντελεστές ενός πολυωνύμου βαθμού  $t$   
 $\{a_{t-1}, \dots, a_1\} \in_R \mathbb{Z}_p$
- Θέτει ως σταθερό όρο το μυστικό  $s$
- Προκύπτει το πολυώνυμο  $p(x) = a_{t-1} \cdot x^{t-1} + \dots + a_1 \cdot x + s$   
(mod  $p$ )

Υποθέτουμε ότι διαθέτουμε έναν έμπιστο διανομέα:

- Επιλέγει και δημοσιοποιεί ένα πρώτο  $p$
- Επιλέγει  $t - 1$  συντελεστές ενός πολυωνύμου βαθμού  $t$   
 $\{a_{t-1}, \dots, a_1\} \in_R \mathbb{Z}_p$
- Θέτει ως σταθερό όρο το μυστικό  $s$
- Προκύπτει το πολυώνυμο  $p(x) = a_{t-1} \cdot x^{t-1} + \dots + a_1 \cdot x + s$   
(mod  $p$ )
- $p(0) = s$

# Shamir secret sharing: Διανομή

Υποθέτουμε ότι διαθέτουμε έναν έμπιστο διανομέα:

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

- Παρατήρηση: Δεν μας ενδιαφέρει να υπολογίσουμε το πολυώνυμο  $p$  αλλά το μυστικό  $p(0) = s$

# Shamir secret sharing: Ανακατασκευή

- Παρατήρηση: Δεν μας ενδιαφέρει να υπολογίσουμε το πολυώνυμο  $p$  αλλά το μυστικό  $p(0) = s$
- Κάθε παίκτης  $i$  υπολογίζει τους συντελεστές Lagrange

# Shamir secret sharing: Ανακατασκευή

- Παρατήρηση: Δεν μας ενδιαφέρει να υπολογίσουμε το πολυώνυμο  $p$  αλλά το μυστικό  $p(0) = s$
- Κάθε παίκτης  $i$  υπολογίζει τους συντελεστές Lagrange
- $\lambda_i(0) = \prod_{k=1, k \neq i}^t \frac{-k}{i-k} \bmod p$

# Shamir secret sharing: Ανακατασκευή

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

# Παρατηρήσεις I

- Πληροφοριοθεωρητική ασφάλεια αν ο αντίπαλος διαθέτει λιγότερα μερίδια



# Παρατηρήσεις I

- Πληροφοριοθεωρητική ασφάλεια αν ο αντίπαλος διαθέτει λιγότερα μερίδια
- Μπορούν να προστεθούν εύκολα καινούρια μερίδια, χωρίς να αλλάξουν τα παλιά: Υπολογισμός νέων σημείων

# 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)

# 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

# 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

# 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)

$$s_1 + s_2 = f(0) + g(0) = (f + g)(0)$$

- Μειονεκτήματα: Εμπιστοσύνη

- Μειονεκτήματα: Εμπιστοσύνη
  - Κακόβουλος διανομέας: Λανθασμένα μερίδια σε τμήμα των παικτών

- Μειονεκτήματα: Εμπιστοσύνη
  - Κακόβουλος διανομέας: Λανθασμένα μερίδια σε τμήμα των παικτών



- Μειονεκτήματα: Εμπιστοσύνη
  - Κακόβουλος διανομέας: Λανθασμένα μερίδια σε τμήμα των παικτών
  - Κακόβουλος παίκτης: Παροχή λανθασμένων μεριδίων κατά τη διάρκεια της ανακατασκευής

- Μειονεκτήματα: Εμπιστοσύνη
  - Κακόβουλος διανομέας: Λανθασμένα μερίδια σε τμήμα των παικτών
  - Κακόβουλος παίκτης: Παροχή λανθασμένων μεριδίων κατά τη διάρκεια της ανακατασκευής

- 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)

- 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

## Υποθέσεις

- Ομάδα  $\mathbb{G}$  τάξης  $q$  με γεννήτορα  $g$  με δύσκολο DLP

## Υποθέσεις

- Ομάδα  $\mathbb{G}$  τάξης  $q$  με γεννήτορα  $g$  με δύσκολο DLP
- Υπολογιστική Ασφάλεια

## Υποθέσεις

- Ομάδα  $\mathbb{G}$  τάξης  $q$  με γεννήτορα  $g$  με δύσκολο DLP
- Υπολογιστική Ασφάλεια
- Για απλότητα χρήση συνάρτησης σύνοψης  $\mathcal{H}$  για δέσμευση

## Υποθέσεις

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



## Υποθέσεις

- Ομάδα  $\mathbb{G}$  τάξης  $q$  με γεννήτορα  $g$  με δύσκολο DLP
- Υπολογιστική Ασφάλεια
- Για απλότητα χρήση συνάρτησης σύνοψης  $\mathcal{H}$  για δέσμευση
- Για ασφάλεια: Απαιτείται έντιμη πλειοψηφία (το πολύ  $t - 1$  corrupted / τουλάχιστον  $t$  honest)
  - Οποιοδήποτε σύνολο από  $t$  honest θα ανακατασκευάσει το μυστικό
  - Για έντιμο διανομέα: το μυστικό είναι το σωστό
  - Οι corrupted δεν μαθαίνουν τίποτα για το  $s$

Ο διανομέας:

- Επιλογή  $a_0 \in_R \mathbb{Z}_q$

Ο διανομέας:

- Επιλογή  $a_0 \in_R \mathbb{Z}_q$
- Διαμοιρασμός του  $a_0$  με Shamir Secret Sharing

## Ο διανομέας:

- Επιλογή  $a_0 \in_R \mathbb{Z}_q$
- Διαμοιρασμός του  $a_0$  με Shamir Secret Sharing
  - Επιλογή  $a_1, \dots, a_{t-1} \in_R \mathbb{Z}_q$
  - Ορισμός  $p(x) = a_0 + \sum_{j=1}^{t-1} a_j \cdot x^j$
  - Αποστολή  $s_i = p(i)$  στον  $P_i$

## Ο διανομέας:

- Επιλογή  $a_0 \in_R \mathbb{Z}_q$
- Διαμοιρασμός του  $a_0$  με Shamir Secret Sharing
  - Επιλογή  $a_1, \dots, a_{t-1} \in_R \mathbb{Z}_q$
  - Ορισμός  $p(x) = a_0 + \sum_{j=1}^{t-1} a_j \cdot x^j$
  - Αποστολή  $s_i = p(i)$  στον  $P_i$
- Broadcast:  $\{A_j = g^{a_j}\}_{j=0}^{t-1}$  και
- $c = \mathcal{H}(a_0) \oplus s$

- Κάθε παίκτης  $P_i$  υπολογίζει:

$$c_i = \prod_{j=0}^{t-1} (A_j)^{i^j} = \prod_{j=0}^{t-1} (g^{a_j})^{i^j} = \prod_{j=0}^{t-1} g^{a_j \cdot i^j} = g^{\sum_{j=0}^{t-1} a_j \cdot i^j} = g^{P(i)}$$

- Κάθε παίκτης  $P_i$  υπολογίζει:

$$c_i = \prod_{j=0}^{t-1} (A_j)^{i^j} = \prod_{j=0}^{t-1} (g^{a_j})^{i^j} = \prod_{j=0}^{t-1} g^{a_j \cdot i^j} = g^{\sum_{j=0}^{t-1} a_j \cdot i^j} = g^{P(i)}$$

- Αν ο διανομέας είναι έμπιστος θα ισχύει:  $c_i = g^{P(i)} = g^{S_i}$

- Κάθε παίκτης  $P_i$  υπολογίζει:

$$c_i = \prod_{j=0}^{t-1} (A_j)^{i^j} = \prod_{j=0}^{t-1} (g^{a_j})^{i^j} = \prod_{j=0}^{t-1} g^{a_j \cdot i^j} = g^{\sum_{j=0}^{t-1} a_j \cdot i^j} = g^{p(i)}$$

- Αν ο διανομέας είναι έμπιστος θα ισχύει:  $c_i = g^{p(i)} = g^{s_i}$
- Επαλήθευση σχέσης: Αν δεν ισχύει ο  $P_i$  τερματίζει ανεπιτυχώς



- Κάθε παίκτης  $P_i$  υπολογίζει:

$$c_i = \prod_{j=0}^{t-1} (A_j)^{i^j} = \prod_{j=0}^{t-1} (g^{a_j})^{i^j} = \prod_{j=0}^{t-1} g^{a_j \cdot i^j} = g^{\sum_{j=0}^{t-1} a_j \cdot i^j} = g^{P(i)}$$

- Αν ο διανομέας είναι έμπιστος θα ισχύει:  $c_i = g^{P(i)} = g^{S_i}$
- Επαλήθευση σχέσης: Αν δεν ισχύει ο  $P_i$  τερματίζει ανεπιτυχώς
- Αν τερματίσουν πάνω από  $t$  χρήστες, ο διανομέας δεν είναι έμπιστος: τερματισμός

- Κάθε παίκτης  $P_i$  υπολογίζει:

$$c_i = \prod_{j=0}^{t-1} (A_j)^{i^j} = \prod_{j=0}^{t-1} (g^{a_j})^{i^j} = \prod_{j=0}^{t-1} g^{a_j \cdot i^j} = g^{\sum_{j=0}^{t-1} a_j \cdot i^j} = g^{p(i)}$$

- Αν ο διανομέας είναι έμπιστος θα ισχύει:  $c_i = g^{p(i)} = g^{s_i}$
- Επαλήθευση σχέσης: Αν δεν ισχύει ο  $P_i$  τερματίζει ανεπιτυχώς
- Αν τερματίσουν πάνω από  $t$  χρήστες, ο διανομέας δεν είναι έμπιστος: τερματισμός
- Αν τερματίσουν λιγότεροι από  $t$  χρήστες: Φταίει ο  $P_i$  - επανάληψη  $s_i$

- Συγκέντρωση τουλάχιστον  $t$  μεριδίων - υπολογισμός  $a_0$

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

# Εφαρμογή: Threshold ElGamal I

- Δημιουργία κλειδιών από (trusted) dealer
- Οι παίκτες είναι 'αρχές' που συνεργάζονται στην αποκρυπτογράφηση
  - 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  $y = g^x \bmod p$

# Εφαρμογή: Threshold ElGamal I

- Δημιουργία κλειδιών από (trusted) dealer
- Οι παίκτες είναι 'αρχές' που συνεργάζονται στην αποκρυπτογράφηση
  - 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: Δημόσιο κλειδί και μερίδια  
 $\text{KeyGen}(1^\lambda) = (y, \{i, p(i)\}_{i=1}^n)$

# Εφαρμογή: Threshold ElGamal I

- Δημιουργία κλειδιών από (trusted) dealer
- Οι παίκτες είναι 'αρχές' που συνεργάζονται στην αποκρυπτογράφηση
  - 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: Δημόσιο κλειδί και μερίδια  
 $\text{KeyGen}(1^\lambda) = (y, \{i, p(i)\}_{i=1}^n)$
- Κρυπτογράφηση
  - Κανονικά  
 $\text{Encrypt}(y, m) = (G, M) = (g^r, m \cdot y^r)$

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



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

$$\begin{aligned} C &= \prod_i c_i^{\lambda_i(0)} = \prod_i G^{p(i)\lambda_i(0)} = \\ &G^{\sum_i p(i)\lambda_i(0)} = G^{p(0)} = \\ &G^x \end{aligned}$$

όπου  $\lambda_i$  οι συντελεστές Lagrange

- Αποκρυπτογράφηση: Σε δύο βήματα

## 1. 'Αποκρυπτογράφηση' μεριδίων

- Κάθε παίκτης υπολογίζει και δημοσιοποιεί το  $c_i = G^{p(i)} \bmod p$

## 2. Συνδυασμός

- Συγκεντρώνονται  $t$  'αποκρυπτογραφημένα' μερίδια  $(i, c_i)$  τα οποία συνδυάζονται ως:

$$\begin{aligned} C &= \prod_i c_i^{\lambda_i(0)} = \prod_i G^{p(i)\lambda_i(0)} = \\ &G^{\sum_i p(i)\lambda_i(0)} = G^{p(0)} = \\ &G^x \end{aligned}$$

όπου  $\lambda_i$  οι συντελεστές Lagrange

- Αποκρυπτογράφηση ως:  $\frac{M}{C}$

- Υπολογιστική ασφάλεια ως προς τα  $c_i$

- Υπολογιστική ασφάλεια ως προς τα  $c_i$
- Ίδια κρυπτογράφηση

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

1. Παγουρτζής, Α., Ζάχος, Ε., ΓΠ, 2015. Υπολογιστική κρυπτογραφία. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών
2. Jonathan Katz and Yehuda Lindell. Introduction to Modern Cryptography (Chapman and Hall/Crc Cryptography and Network Security Series). Chapman and Hall/CRC, 2007
3. Paar, Christof, and Jan Pelzl. Understanding cryptography: a textbook for students and practitioners. Springer Science-Business Media, 2009.
4. Kiayias, Aggelos [Cryptography primitives and protocols](#), UoA, 2015
5. [Nigel Smart. Introduction to cryptography](#)
6. M. Green [What is the Random Oracle Model and why should you care?](#)
7. M. Bellare, P. Rogaway, (1993). "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". ACM Conference on Computer and Communications Security: 62–73.
8. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. Journal of the ACM, 51(4):557–594, 2004.
9. Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In CRYPTO '91, pages 129–140, 1991
10. Victor Shoup [Why chosen ciphertext security matters](#), 1998
11. Adi Shamir, [How to share a secret](#). Communications of the ACM 22.11 (1979): 612-613.
12. Helger Lipmaa, 79.159 Cryptography and Data Security, 24.03.2004 Lecture 9: Secret Sharing, Threshold Cryptography, MPC
13. J. Kuhn [The Mathematics of Secret Sharing](#)