# Ψηφιακές Υπογραφές

Παναγιώτης Γροντάς - Άρης Παγουρτζής

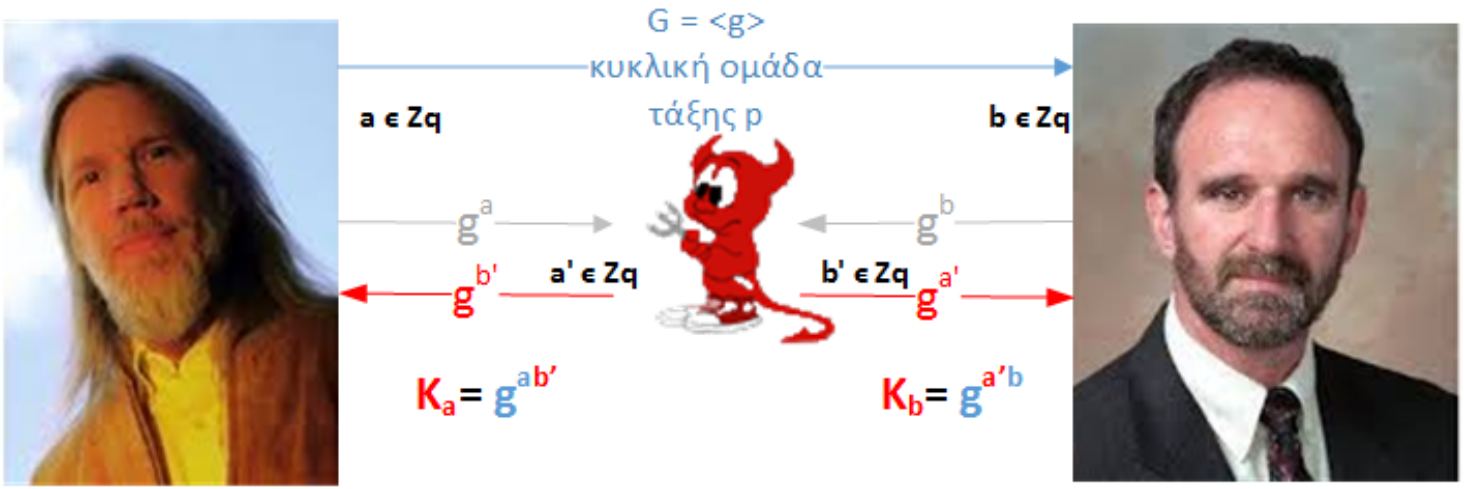
ΕΜΠ - Κρυπτογραφία - (2016-2017)

09/12/2016

# Περιεχόμενα

- Ορισμός - Μοντελοποίηση Ασφάλειας
- Ψηφιακές Υπογραφές RSA
- Επιθέσεις - Παραλλαγές
- Το μοντέλο του τυχαίου μαντείου
- Ψηφιακές Υπογραφές ElGamal-DSA-ECDSA
- Υποδομή Δημοσίου Κλειδιού
- Διαμοιρασμός Απορρήτων (Secret sharing)

# 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)
- 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 $key_{ver}$)

# Πλεονεκτήματα

- Εύκολη διανομή κλειδιού
- Δημόσια Επαληθευσιμότητα
  - Δεν επαληθεύει μόνο ο παραλήπτης
- Μη αποκήρυξη (non repudiation)
  - Εσωτερικός αντίπαλος
  - Ο υπογράφων δεν μπορεί να αρνηθεί τις υπογραφές του
- Επιπλέον λειτουργίες
  - Αυθεντικοποίηση χρηστών (λόγω *κατοχής* του ιδιωτικού κλειδού)
  - Ανωνυμία (τυφλές υπογραφές)
  - Αντιπροσωπεία από ομάδα (ομαδικές υπογραφές)
  - ...

# Μειονεκτήματα

Λύσαμε το πρόβλημα **διανομής** κλειδιού
Δημιουργήσαμε το πρόβλημα **αυθεντικότητας** κλειδιού

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

# Ορισμός

## Σχήμα Υπογραφής

Μια τριάδα από αλγόριθμους

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

## Ορθότητα

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

Έγκυρες υπογραφές: ικανοποιούν την απαίτηση της ορθότητας

# Επιθέσεις

## 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

# Αντίπαλοι Ι

## 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 προηγουμενες έγκυρες υπογραφές

# Αντίπαλοι II

Ασφάλεια ως προς τον δυνατότερο αντίπαλο - ευκολότερη επίθεση

## Διαισθητικά

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

# Ορισμός Ασφάλειας

## Το παιχνίδι πλαστογράφησης $Forge - Game$

- Ο $S$ εκτελεί τον αλγόριθμο $\texttt{KeyGen}(1^\lambda)$ και παράγει τα $(pk, sk)$
- Ο $\mathcal{A}$ έχει πρόσβαση σε ένα μαντειο υπογραφών $\texttt{Sign}(sk, \cdot)$ 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 $(m, \sigma)$
- $Forge - Game(\mathcal{A}) = 1 \Leftrightarrow \texttt{Verify}(key_{ver}, m, \sigma) = 1 \wedge (m, \sigma) \notin Q$

Ο $\mathcal{A}$ κερδίζει το παιχνίδι αν
$Pr[Forge - Game(\mathcal{A}) = 1] = non - negl(\lambda)$

# 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

**Υπογραφή** - Αποκρυπτογράφηση

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

**Επαλήθευση** - Κρυπτογράφηση

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

**Ορθότητα**

$\texttt{Verify}((e, n), m, m^d \bmod n) = \quad m^{de} = m \pmod{n}$

...αλλά καθόλου ασφάλεια

# 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, το $m$ μπορεί να είναι αποτέλεσμα κωδικοποίησης

# 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 $\sigma = \sigma_1\sigma_2 = m_1^d(\frac{m}{m_1})^d = m^d \bmod n$
- Η $\sigma$ είναι έγκυρη υπογραφή για το $m$ και $\notin Q$

# RSA - FDH (Full Domain Hash) I

**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 $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_n^*$

**Υπογραφή**

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

**Επαλήθευση**

- Υπολογισμός $\mathcal{H}(m)$
- $\texttt{Verify}((e, N), m, \sigma) = \quad \sigma^e =_? \mathcal{H}(m) \pmod{N}$

# RSA - FDH (Full Domain Hash) II

**Ορθότητα**

$\texttt{Verify}((e, N), m, \mathcal{H}(m)^d) = \quad \mathcal{H}(m)^{de} = \mathcal{H}(m) \pmod{N}$

Υλοποίηση: συνάρτηση σύνοψης με δυσκολία εύρεσης συγκρούσεων

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

# RSA - FDH (Full Domain Hash) III

- 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 $m : \mathcal{H}(m) = h$
  - Δυσκολία αντιστροφής

# RSA - FDH (Full Domain Hash) IV

- 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);
  - ΟΧΙ
- Το μοντέλο του τυχαίου μαντείου (M. Bellare, P. Rogaway, -1993)

# 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 |
|---|---|
| $0 \cdots 00$ | $r_1$ |
| $0 \cdots 01$ | $r_2$ |
| $\cdots$ | $\cdots$ |
| $1 \cdots 11$ | $r_{l(n)}$ |

  - Συμπίεση: Μείωση τυχαιότητας

Ακόμα και να μπορούσαν να κατασκευαστούν
αδύνατη αποθήκευση
εκθετική αποτίμηση (μη αποδεκτή και για χρήστη και για αντίπαλο)

# Συναρτήσεις σύνοψης και αποδείξεις ασφάλειας Ι

## 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)

# Συναρτήσεις σύνοψης και αποδείξεις ασφάλειας II

## 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

# Συναρτήσεις σύνοψης και αποδείξεις ασφάλειας III

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)

# Απόδειξη Ασφάλειας Hashed 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 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 Ι

## Υπόθεση

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

## Συνέπεια

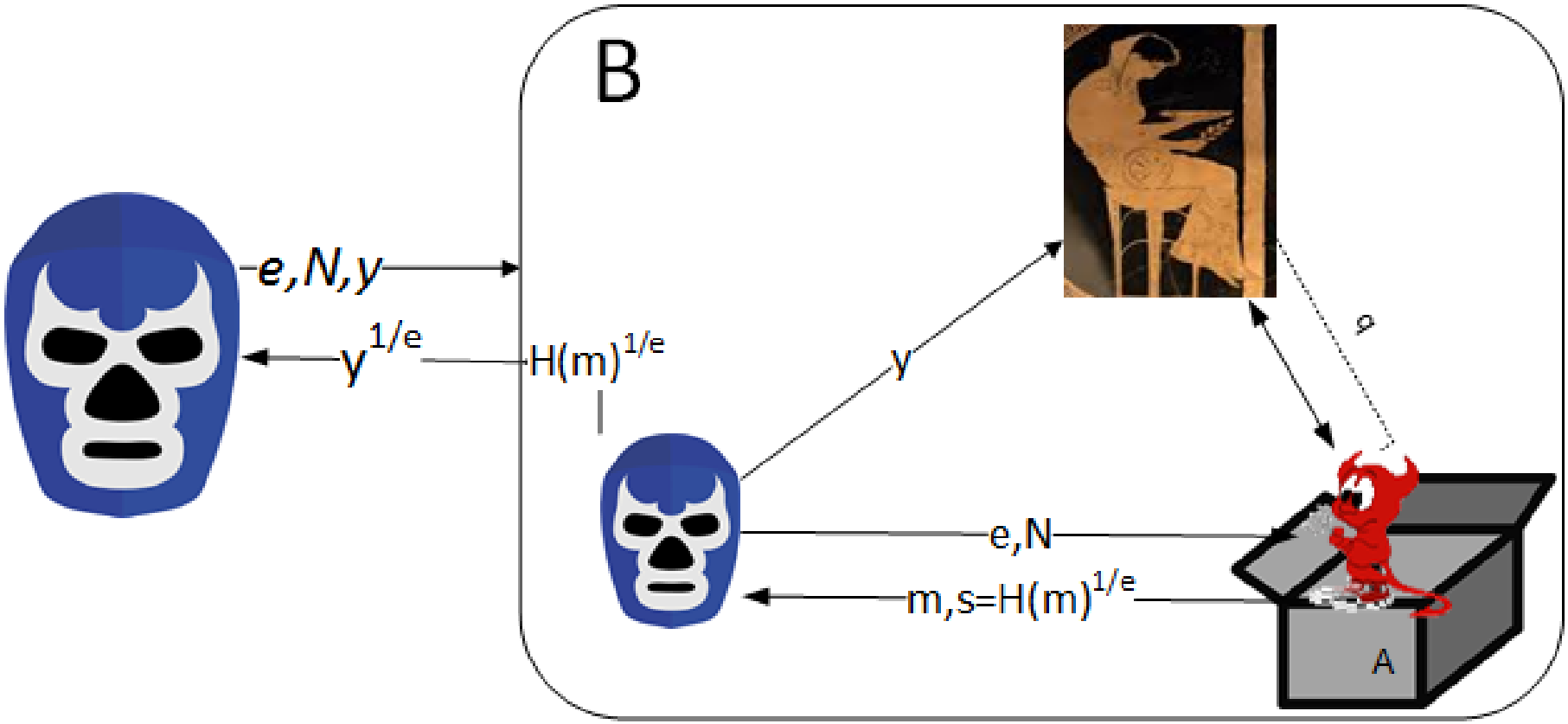
Εφόσον η πλαστογράφηση είναι έγκυρη υπογραφή πρέπει $\sigma^e = \mathcal{H}(m)$ Άρα $\sigma = \mathcal{H}(m)^{\frac{1}{e}}$

# Απόδειξη Ασφάλειας Hashed RSA
# Επίθεση χωρίς μήνυμα II

- Ο $\mathcal{B}$ προωθεί το $(e, N)$ στον $\mathcal{A}$
- Ο $\mathcal{A}$ κάνει $q = poly(\lambda)$ 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 $(m_i^*)$ με το $y$
- Ο $\mathcal{B}$ *ελπίζει* ότι στο $(m_i^*)$ 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 $\frac{\mathrm{p}}{q}$ θα ισχύει $\sigma = y^{\frac{1}{e}}$

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

# Απόδειξη Ασφάλειας Hashed RSA
# Επίθεση χωρίς μήνυμα III



# Απόδειξη Ασφάλειας Hashed RSA
# Επίθεση επιλεγμένου μηνύματος Ι

## 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

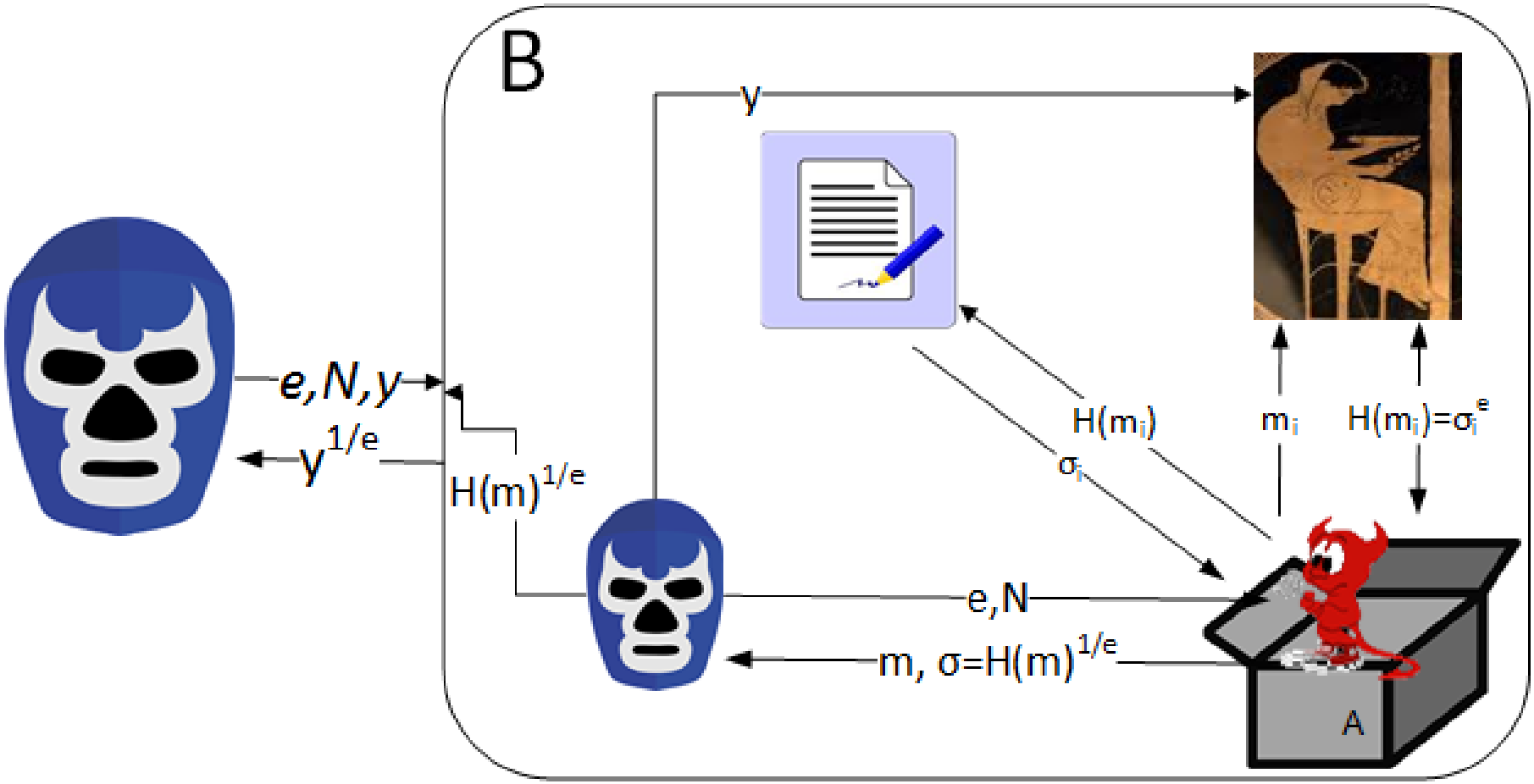
- Ο $\mathcal{B}$ προωθεί το $(e, N)$ στον $\mathcal{A}$
- Ο $\mathcal{A}$ κάνει $q$ ερωτήσεις στο μαντείο για μηνύματα $\{m_i\}_{i=1}^{q}$
- Κάθε ερώτηση απαντάται από τον $\mathcal{B}$ ως εξής:
  - Επιλέγει τυχαίο $\sigma_i \in \mathbb{Z}_n^*$
  - Υπολογίζει $y_i = \mathcal{H}(m_i) = \sigma_i^e \bmod N$
  - 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 πλαστογράφηση. Το απαντάει με $y$
- Για το συγκεκριμένο δεν θα ζητηθεί υπογραφή, αλλά το $\sigma$ θα παραχθεί από τον $\mathcal{A}$ (πλαστογράφηση)

# Απόδειξη Ασφάλειας Hashed RSA
# Επίθεση επιλεγμένου μηνύματος III

- 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 $\frac{\mathrm{p}}{q}$ αμελητέο

# Απόδειξη Ασφάλειας Hashed RSA
# Επίθεση επιλεγμένου μηνύματος IV



# Το μοντέλο του τυχαίου μαντείου - κριτική

## 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
Ύπαρξη '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

# 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

$$r = g^k \bmod p$$

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

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

# Σχήμα Υπογραφής ElGamal II

**Επαλήθευση υπογραφής στο** $m$

$$\texttt{Verify}(y, m, (r, s)) = \begin{cases} 1, y^r \cdot r^s \equiv g^m \pmod{p} \\ 0, y^r \cdot r^s \neq g^m \pmod{p} \end{cases} \quad \text{και } k < p$$

## 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 $s$

# Παρατηρήσεις

- Πιθανοτικό σχήμα υπογραφής - πολλές έγκυρες υπογραφές για ένα μήνυμα $m$ (τυχαίο $k$)
- Η συνάρτηση επαλήθευσης δέχεται οποιαδήποτε από αυτές ως έγκυρη
- Χειρισμός Τυχαιότητας
  - 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 $s_1 = (m_1 - xr)k^{-1}$
- $sign(x, m_2) = (r, s_2)$ με $s_2 = (m_2 - xr)k^{-1}$
- Αφαιρούμε κατά μέλη:
- $s_1 - s_2 = (m_1 - m_2)k^{-1}$
- Υπάρχουν $d = gcd(s_1 - s_2, p - 1)$ λύσεις της μορφής
- $k = \frac{m_1 - m_2}{d} \frac{s_1 - s_2}{d}^{-1} + i\frac{p-1}{d} \bmod p - 1, \quad i \in \{0, \cdots, d - 1\}$
- Δοκιμή όλων των $k$ ως προς το γνωστό $r$
- Υπολογισμός ιδιωτικού κλειδιού

# Ασφάλεια έναντι πλαστογράφησης I

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

1. 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
2. Επιλέγω $r$ και $s$, ψάχνω $m$: DLP ξανά.

# Ασφάλεια έναντι πλαστογράφησης II

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.
- 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.

# Παραγωγή κλειδιών DSS

1. Επιλογή πρώτων $q$ μεγέθους 160-bit και $p$ μεγέθους $n$-bit, $n = 64\lambda$, $\lambda = 8, 9, 10, \ldots, 16$, 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: $x$.

# DSS:Δημιουργία υπογραφής

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: $(r, s)$.

# DSS:Επαλήθευση υπογραφής DSA

Ο $B$ υπολογίζει:

$$h = \mathcal{H}(m)$$
$$e_1 = s^{-1} h \bmod q$$
$$e_2 = r s^{-1} \bmod q$$

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

## Ορθότητα

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

# Ψηφιακές Υπογραφές - ECDSA I

**Δημιουργία κλειδιών**

- Δημόσια Διαθέσιμες Παράμετροι: $(p, a, b, \#\mathcal{E}, q, G)$
- Ιδιωτικό κλειδί: Ένας τυχαίος ακέραιος $x \in \{1, \cdots, q-1\}$
- Δημόσιο κλειδί: Το σημείο $Y = xG \in \mathcal{E}$

# Ψηφιακές Υπογραφές - ECDSA II

**Υπογραφή**

- Υπολογισμός σύνοψης του μηνύματος $h = \mathcal{H}(M)$ και προσαρμογή της στο $[0, \cdots, q-1]$
- Επιλογή τυχαίου αριθμού $k$ στο σύνολο $\{1, \cdots, q-1\}$
- Υπολογισμός του σημείου $P = kG = (x_P, y_P)$.
- Υπολογισμός του $r = x_P \bmod q$
- Αν $r = 0 \pmod q$ τότε επιλέγεται καινούριο $k$ και η διαδικασία επαναλαμβάνεται.
- Υπολογισμός του $s = k^{-1}(h + r \cdot x) \bmod q$
- Αν $s = 0$ τότε επιλέγεται καινούριο $k$ και η διαδικασία επαναλαμβάνεται.
- Η υπογραφή είναι το ζεύγος $(r, s)$

# Ψηφιακές Υπογραφές - ECDSA III

**Επαλήθευση**

- Υπολογισμός του $u_1 = s^{-1}h \bmod q$
- Υπολογισμός του $u_2 = s^{-1}r \bmod q$
- Υπολογισμός του σημείου $P' = u_1 G + u_2 Y$
- Η υπογραφή είναι έγκυρη αν $r = x_{P'} \pmod q$

**Ορθότητα**: Υπολογισμός ίδιου σημείου με 2 τρόπους

- Υπογραφή $P = kG$
- Επαλήθευση $P' = u_1 G + u_2 Y$

$$P' = u_1 G + u_2 Y = s^{-1}(h + rx)G = k(h + rx)^{-1}(h + rx)G = kG = P$$

# Πρακτική χρήση ψηφιακών υπογραφών

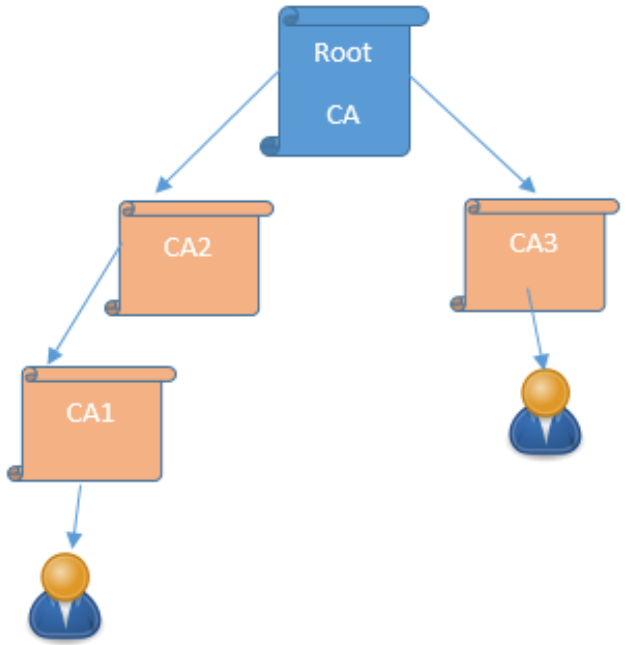
- Διαφορά Συμμετρικών - Ασύμμετρων Κρυπτοσυστημάτων
  - Συμμετρικά: Δύσκολη διανομή, Εύκολη Αυθεντικότητα (λόγω φυσικών υποθέσεων)
  - Ασύμμετρα: Εύκολη διανομή, Δύσκολη Αυθεντικότητα
- Αντιστοιχία (?) Ταυτότητας Χρήστη - Δημοσίου, Ιδιωτικού Κλειδιού (binding)
- Ενεργός αντίπαλος - Πλαστοπροσωπία - αλλαγή 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)
  - 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 έμπτιστο τρόπο
  - Μόνο το κλειδί της CA
  - Για τα υπόλοιπα 'εγγύαται' 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

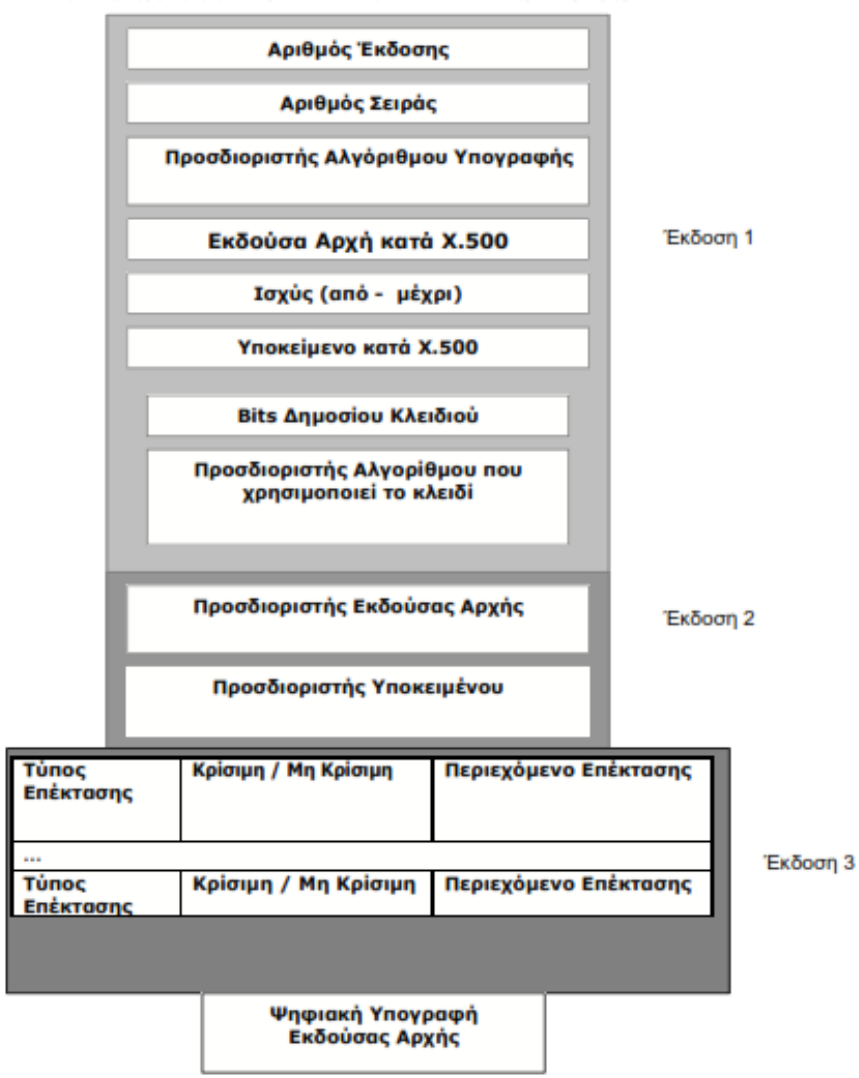
- Ενδιάμεσες Αρχές: Υπογραφή από ανώτερη αρχή
- Ριζικές (Root) Αρχές: Υπογράφουν μόνες τους
- Συνήθως 3-4 επίπεδα



# Υποδομή Δημοσίου Κλειδιού

- 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: 104e764f615ebc89
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): a06f981af315c85987c48945a82f6335

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 fc 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 (1.3.6.1.5.5.7.48
             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
```

# Απόκτηση πιστοποιητικών

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

# Αρχές Πιστοποίησης - Άλλες υπηρεσίες

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

# Ανάκληση Πιστοποιητικών

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

# Εναλλακτικές Προσεγγίσεις Ι

- 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

# Εναλλακτικές Προσεγγίσεις II

- 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 TTP
  - Παράγει τα ιδιωτικά κλειδιά από την ταυτότητα

# 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 $r$
  - $t = r^e \bmod n$
  - $s = k\ r^{\mathcal{H}(m|t)} \bmod n$
- Επαλήθευση υπογραφής με την ταυτότητα:
- Έλεγχος αν: $\mathcal{H}(id)t^{\mathcal{H}(m|t)} = s^e$
- Ορθότητα: $\mathcal{H}(id)t^{\mathcal{H}(m|t)} = k^e r^{e\mathcal{H}(m|t)} = s^e$

# Διαμοιρασμός απορρήτων - Εισαγωγή

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)

Διαμοιρασμός απορρήτων (Secret Sharing)

Επιπλέον

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

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

Πρόβλημα: Ένας παίκτης μπορεί να ακυρώσει την ανακατασκευή

# Threshold Secret Sharing

$(t, n)$ threshold secret sharing

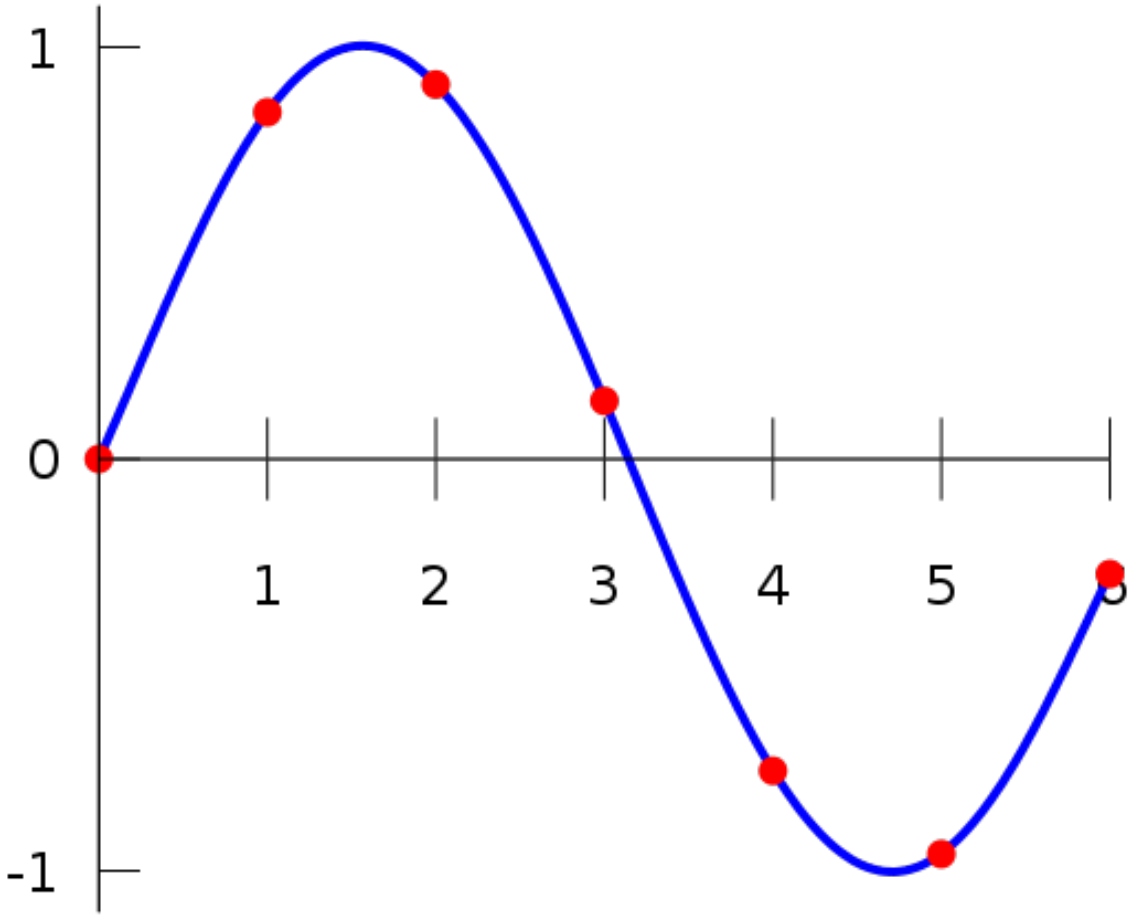
- Ένα μυστικό $s$ πρέπει να μοιραστεί σε $n$ παίκτες $P_1, P_2, \cdots P_n$ ώστε:
  - Οποιοδήποτε υποσύνολο από τουλάχιστον $t + 1$ παίκτες να μπορεί να το ανακτήσει
  - Κανένα υποσύνολο με $t$ παίκτες να μην μπορεί
- Υπόθεση Εμπιστεύομαστε τον διανομέα $D$ και τους παίκτες

# Shamir Secret Sharing I

## Πολυωνυμική παρεμβολή

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

# Shamir Secret Sharing II



# Shamir Secret Sharing III

## Εφαρμογή στο διαμοιρασμό απορρήτων

Υποθέτουμε ότι διαθέτουμε έναν έμπιστο διανομέα:

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

# Shamir Secret Sharing IV

## Ανακατασκευή

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

# Παρατηρήσεις Ι

- 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 (άθροισμά πολυωνύμων είναι πολυώνυμο)
  $s_1 + s_2 = f(0) + g(0) = (f + g)(0)$

# Παρατηρήσεις II

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

# Εφαρμογή: Threshold ElGamal I

- Δημιουργία Κλειδιών
  - Επιλογή δύο μεγάλων πρώτων $p, q$ ώστε $q \mid (p-1)$
  - Επιλογή της υποομάδας τάξης $q$ του $\mathbb{Z}_{\mid}^*$ 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 $x$ $(\bmod\ q)$
  - Αποτέλεσμα
    $\texttt{KeyGen}(1^\lambda) = (y, \{i, f(i)\}_{i=1}^n)$
- Κρυπτογράφηση
  - Κανονικά
    $\texttt{Encrypt}(y, m) = (G, M) = (g^r, m \cdot y^r)$

# Εφαρμογή: Threshold ElGamal II

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

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

     όπου $\lambda_i$ οι συντελεστές Lagrange
     - Αποκρυπτογράφηση ως:

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

# Παρατηρήσεις

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

# Βιβλιογραφία I


1. St. Zachos and Aris Pagourtzis. Στοιχεία Θεωρίας Αριθμών και Εφαρμογές στην Κρυπτογραφία. Πανεπιστημιακές Σημειώσεις
2. Jonathan Katz and Yehuda Lindell. Introduction to Modern Cryptography (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. Adi Shamir, How to share a secret. Communications of the ACM 22.11 (1979): 612-613.
10. Helger Lipmaa, 79.159 Cryptography and Data Security, 24.03.2004 Lecture 9: Secret Sharing, Threshold Cryptography, MPC
11. J. Kuhn The Mathematics of Secret Sharing