ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Δημόσια Επαληθεύσιμοι Υπολογισμοί
Publicly Verifiable & Delegatable Computing

και εισαγωγή στην Pairing-based κρυπτογραφία

Υπολογιστική Κρυπτογραφία

Ανδρέας Στάμος

Αριθμός μητρώου: 03120***

Διεύθυνση ηλεκτρονικού ταχυδρομείου: stamos.aa@gmail.com

9ο Εξάμηνο — 2024-2025

Περιεχόμενα

1 Εισαγωγή στην Pairing-based κρυπτογραφία

1.1 Εισαγωγή

Μια από τις κλασικές κρυπτογραφικές εικασίες στις οποίες στηρίζονται πολλά σύγχρονα κρυπτογραφικά συστήματα είναι η υπολογιστική δυσκολίας προβλημάτων τύπου Υπολογισμού Διακριτού Λογάριθμου σε μια ομάδα (DLP), με το πιο σημαντικό εκπρόσωπο, το Πρόβλημα Υπολογισμού Diffie-Hellman (CDH).

Με βάση αυτή την υπόθεση, κατασκευάζεται το πρωτόκολλο ανταλλαγής κλειδιού Diffie-Hellman με το οποίο δύο χρήστες μπορούν να αποκτήσουν ένα κοινό μυστικό μέσα από μη ασφαλές κανάλι επικοινωνίας.

Οι πρώτοι προβληματισμοί για επέκταση ξεκινάνε από τον Joux, που σκοπεύει να επεκτείνει το πρωτόκολλο ώστε 3 χρήστες να ανταλλάξουν ένα κοινό μυστικό.

Μια πρώτη απόπειρα είναι η επέκταση του πρωτοκόλλου Diffie-Hellman:

Ειδικότερα, ας υποθέσουμε μια ομάδα G, τάξης n, με προσθετικό συμβολισμό (ο λόγος είναι πως συνήθως χρησιμοποιούνται ομάδες ελλειπτικών καμπύλων) και ένα σημείο βάσης P. Προσέχουμε πως στον προσθετικό συμβολισμό, το πρόβλημα DLP γίνεται δοθέντος (P, aP) να υπολογίστει το a.

Ας θεωρήσουμε τους 3 παίκτες σε έναν λογικό δακτύλιο.

Οι 3 παίκτες διαλέγουν ένα ομοιόμορφα τυχαίο στοιχείο της $\mathbb{Z}_n$, αντίστοιχα καθένας, a, b, c, υπολογίζουν το aP, bP, cP, το στέλνουν στον επόμενο παίχτη, έπειτα πολλαπλασιάζουν αυτό που λαμβάνουν με τον αριθμό τους, ξαναστέλνουν στον επόμενο και στο τέλος απλά πολλαπλασιάζουν αυτό που λάβουν ξανά με τον αριθμό τους.

Έτσι στο τέλος και οι 3 παίχτες έχουν το κοινό μυστικό $abcP$. Ένας αντίπαλος για να βρει το μυστικό πρέπει να υπολογίσει το $abcP$ από τα $P, aP, bP, cP, abP, acP, bcP$, που είναι τουλάχιστον όσο δύσκολο όσο το CDH.

Θεώρημα 1.1. *Το πρόβλημα υπολογισμού του $abcP$ από τα $P, aP, bP, cP, abP, acP, bcP$ είναι τουλάχιστον όσο δύσκολο το CDH.*

Απόδειξη. Έστω ένα στιγμιότυπο του CDH, όπου έχουμε να υπολογίσουμε xyQ από Q, xQ, yQ.

Επιλέγουμε ένα τυχαίο z, υπολογίζουμε τα zQ, zxQ, zyQ. Έχουμε έτσι $Q, xQ, yQ, zQ, xyQ, zxQ, zyQ$. Τα δίνουμε στο μαντείο και μας επιστρέφει το $xyzQ$. Υπολογίζουμε το $z^{-1} \pmod p$ και τελικά υπολογίζουμε $z^{-1}xyzQ = xyQ$. ☐

Το πρωτόκολλο αυτό, αν και ασφαλές, απαιτεί δύο γύρους εκτέλεσης.

Αρχικός σκοπός είναι η κατασκευή πρωτοκόλλου που θα τρέχει σε έναν μόνο γύρο εκτέλεσης. Θα επιστρέψουμε αργότερα σε αυτό το πρόβλημα ως πρώτο παράδειγμα.

1.2 Bilinear Pairing – ορισμός και σχετικά υπολογιστικά προβλήματα

Έστω μια κυκλική ομάδα G_1 με τάξη $n \in \mathbb{P}$ και ένας γεννήτορας της P. Η G_1 θα γράφεται με προσθετικό συμβολισμό. Το ουδέτερο στοιχείο της συμβολίζεται ∞. (οι συμβολισμοί προκύπτουν από τους συμβολισμούς ελλειπτικών καμπύλων. συνήθως η G_1 επιλέγεται ως η κυκλική ομάδα που γεννά σημείο βάσης P.)

Έστω μια ομάδα G_T με ίδια τάξη n με την G_1. Η G_2 θα γράφεται με πολλαπλασιαστικό συμβολισμό. Το ουδέτερο στοιχείο της συμβολίζεται 1.

Ορισμός 1.1. *Ως Bilinear Pairing στην δυάδα (G_1, G_T) ορίζεται μια συνάρτηση $e : G_1 \times G_1 \mapsto G_T$, που ικανοποιεί τις εξής 3 ιδιότητες:*

1. *(διγραμμικότητα/bilinearity) Για κάθε $R, S, T \in G_1$ ισχύει $e(R + S, T) = e(R, T)e(S, T)$ και $e(R, S + T) = e(R, S)e(R, T)$.*
2. *(μη εκφυλισιμότητα/non-degeneracy) Για κάθε $P \in G_1$ ισχύει $e(P, P) \neq 1$.*
3. *(αποδοτικός υπολογισμός) Η συνάρτηση e μπορεί να υπολογιστεί αποδοτικά (συνήθως αυτό σημαίνει σε πολυωνυμικό χρόνο, αλλα γενικά σκοπός είναι να μπορεί να εκετελείται γρήγορα ώστε να μπορεί πρακτικά να χρησιμοποιηθεί.)*

Σημαντική Παρατήρηση: Εξ ορισμού και μόνο του Bilinear Pairing, το πρόβλημα απόφασης Diffie-Hellman (DDH) είναι υπολογιστικά εύκολο στην G_1: Έστω ότι δίνονται $P, aP, bP, cP \in G_1$ και καλούμαστε να εξετάσουμε αν $cP = abP$. Υπολογίζω $\gamma_1 = e(P, cP)$ και $\gamma_2 = e(aP, bP)$, αν $\gamma_1 = \gamma_2$ επιστρέφω πως ισχύει $cP = abP$, αλλιώς πως δεν ισχύει. Ισχύει $e(P, cP) = e(aP, bP) \iff e(P,P)^c = e(P,P)^{ab}$. Επειδή $\operatorname{ord}(G_T) = n \in \mathbb{P}$ και επειδή από την 2η ιδιότητα $e(P,P) \neq 1$, τότε $\operatorname{ord}_{G_T}(e(P,P)) = n$, οπότε τελικά

$$e(P, cP) = e(aP, bP) \iff e(P,P)^c = e(P,P)^{ab} \iff c \equiv ab \pmod n$$

Μια συνήθης εικασία για κρυπτογραφικές εφαρμογές που στηρίζονται σε Bilinear Pairing είναι η υπολογιστική δυσκολία του προβλήματος Υπολογισμού Διγραμμικού Diffie-Hellman (BDHP), που ορίζεται ως εξής:

Ορισμός 1.2. *Έστω δύο ομάδες G_1, G_T και ένα bilinear pairing e σε αυτές. Το Πρόβλημα Υπολογισμού Διγραμμικού Diffie-Hellman BDHP ορίζεται ως εξής:*

Δοθέντος P, aP, bP, cP να υπολογιστεί το $e(P,P)^{abc}$

Αν το BDHP είναι υπολογιστικά δύσκολο, τότε το CDH είναι υπολογιστικά δύσκολο και στις δύο G_1, G_T. Αν το CDH λύνεται αποδοτικά στην G_1 τότε λύνω το BDHP ως εξής: Από τα aP, bP υπολογίζω με το CDH το abP και έπειτα υπολογίζω $e(abP, cP) = e(P,P)^{abc}$. Αντίστοιχα, αν το CDH λύνεται αποδοτικά στην G_T λύνω το BDHP ως εξής: Υπολογίζω τα $g = e(P,P)$, $e(aP, bP) = e(P,P)^{ab} = g^{ab}$ και $e(P, cP) = e(P,P)^c$. Έπειτα με το CDH στο (g, g^{ab}, g^c) βρίσκω το $g^{abc} = e(P,P)^{abc}$.

Το τελευταίο σε συνδυασμό με την προηγούμενη παρατήρηση είναι ήδη ενδεικτική του γεγονότος πως χρειάζεται μια ειδική κατασκευή ώστε να προκύψει Bilinear Pairing, αφού θα χρειαστούμε ομάδες που τελικά θα πρέπει να έχουν εύκολο DDH αλλά δύσκολο CDH. Στις συνηθείς κρυπτογραφικές ομάδες και το DDH θεωρείται υπολογιστικά δύσκολο. Θα αναφερθούμε, αναλυτικότερα, αν και όχι στο πλήρες βάθος, παρακάτω στην κατασκευή ομάδων με Bilinear Pairing (στηριζόμενοι σε ελλειπτικές καμπύλες).

1.3 Εφαρμογές

1.3.1 Μια πρώτη εφαρμογή των Bilinear Pairings: Το πρωτόκολλο Joux ανταλλαγής κλειδιού 3 παικτών σε 1 γύρο

Θα λύσουμε το πρόβλημα που αναφέραμε παραπάνω στην εισαγωγή.

Τρεις παίχτες θέλουν να αποκτήσουν και οι τρεις ένα κοινό μυστικό, με έναν μόνο γύρο επικοινωνιών, που γίνονται σε μη ασφαλή κανάλια επικοινωνίας.

Εδω θεωρούμε δύο ομάδες G_1, G_T για τις οποίες ορίζεται ένα Bilinear Pairing e που δίνει υπολογιστικά δύσκολο BDHP. Θεωρούμε και ένα σημείο βάσης P της G_1.

Οι τρεις παίχτες επιλέγουν καθένας έναν ομοιόμορφα τυχαίο αριθμό, αντίστοιχα $a, b, c \xleftarrow{R} \mathbb{Z}_n^*$, υπολογίζουν, αντίστοιχα τα aP, bP, cP και στέλνουν την τιμή τους στους άλλους δύο παίχτες. Οι τρεις παίχτες υπολογίζουν τότε το εξής κοινό μυστικό:

$$K = e(aP, bP)^c = e(bP, cP)^b = e(aP, cP)^a = e(P,P)^{abc}$$

(μπορούν και οι 3 να το υπολογίσουν αφού ξέρουν τον αριθμό τους a, b, c.)

Για να βρει ένας αντίπαλος το K πρέπει με είσοδο τα P, aP, bP, cP να υπολογίσει το $e(P,P)^{abc}$ δηλαδή να λύσει το BDHP, που όμως έχει υποτεθεί υπολογιστικά δύσκολο.

Το παραπάνω είναι το πρωτόκολλο του Joux, και ήταν η αρχή της Pairing-based κρυπτογραφίας.

1.3.2 Υπογραφές Boneh-Lynn-Shacham (BLS)

Οι Boneh, Lynn και Shacham πρότειναν ένα σχήμα υπογραφών χρησιμοποιώντας ένα bilinear pairing και μια συνάρτηση κατακερματισμού $H : \{0,1\}^* \mapsto G_1 - \{\infty\}$. Η ασφάλεια εδώ απαιτεί υπολογιστική δυσκολία του CDH στην G_1.

Το ιδιωτικό κλειδί επιλέγεται ως ένας ομοιόμορφα τυχαίος αριθμός $a \xleftarrow{R} \mathbb{Z}_n^*$ και το δημόσιο κλείδι υπολογίζεται από αυτό ως $A = aP$.

Η συνάρτηση υπογραφής ορίζεται ως $S = aM$, όπου $M = H(m) \in G_1$.

Η συνάρτηση ελέγχου ψηφιακής υπογραφής S σε μήνυμα m ορίζεται ως εξής: Βρίσκουμε το $M = H(m)$. Υπάρχει ένα k ώστε $M = kP$ (δεν χρειάζεται να το βρούμε.) Για έγκυρη υπογραφή είναι $S = aM = akP$. Συνέπως για να ελέγξουμε την ψηφιακή υπογραφή ελέγχουμε αν το (P, A, M, S) είναι μια τετράδα Diffie-Hellman (αυτό γίνεται αποδοτικά, όπως δείξαμε παραπάνω, ελέγχοντας αν $e(P, A) = e(M, S)$). Αν η υπογραφή είναι έγκυρη, τότε ο έλεγχος πετυχαίνει. Αντίστροφα, ένας αντίπαλος για να βρει μια υπογραφή χρειάζεται να λύσει το πρόβλημα

Υπολογισμού Diffie-Hellman στην G_1 (για το στιγμιότυπο $(P, A, M) = (P, aP, kP)$), που όμως έχει υποτεθεί υπολογιστικά δύσκολο.

Το ενδιάφερον αυτού του σχήματος υπογραφών, είναι η συντομία της υπογραφής. Στα συνήθη σχήματα ψηφιακών υπογραφών, που σχετίζονται με υπολογιστική δυσκολία προβλημάτων διακριτού λογαρίθμου, όπως ενδεικτικά το ECDSA, η υπογραφή αποτελείται από δύο αριθμούς, καθένας πλήθους bits όσο η παράμετρος ασφάλειας. Αντίθετα οι υπογραφές Boneh-Lynn-Shacham αποτελούνται από έναν μόνο αριθμό, οπότε χρειάζονται τον μισό χώρο. Το γεγονός αυτό μπορεί να είναι ασήμαντο για μεγάλα μηνύματα, όμως αν αποστέλλονται πολλά μικρά μηνύματα, οι μεγάλες ψηφιακές υπογραφές μπορούν να καταλήξουν να καταναλώνουν σημαντικό μέρος του διαθέσιμου εύρους ζώνης.

1.3.3 Identity-Based Κρυπτογραφία

1.3.3.1 Εισαγωγή στην Identity-Based κρυπτογραφία

Στα κλασικά συστήματα κρυπτογραφίας δημόσιου κλειδιού, για να γίνει κρυπτογράφηση ενός μηνύματος χρειάζεται ο αποστολέας να διαθέτει το δήμοσιο κλειδί του παραλήπτη, και να έχει εξασφαλίσει με κάποιον τρόπο την αυθεντικότητά του.

Αυτό το πρόβλημα, συνήθως (όπως συμβαίνει σε όλες τις γνωστές εφαρμογές) λύνεται με μια Certificate Authority (CA) που αναλαμβάνει να ελέγξει με κάποιο εξωτερικό τρόπο την αντιστοιχία μια φυσικής οντότητας (π.χ. μιας εταιρίας, ενός ανθρώπου, κ.λπ.) με ένα δημόσιο κλειδί και να εκδόσει ένα πιστοποιητικό για αυτό, που το υπογράφει με το δικό της ιδιωτικό κλειδί.

Έτσι ο αποστολέας αντί να αποκτήσει ένα δημόσιο κλειδί του παραλήπτη, αποκτά με κάποιο τρόπο (μπορεί να είναι μη αξιόπιστος ο τρόπος), ένα πιστοποιητικό της CA που έχει το δημόσιο κλειδί του παραληπτή και έχει πάνω μια ψηφιακή υπογραφή της CA που πιστοποιεί ότι το δημόσιο κλειδί αντιστοιχεί στον παραλήπτη.

Βέβαια, υποθέτουμε ότι ο αποστολέας μπορεί να βρει αξιόπιστα το δημόσιο κλειδί της CA. Αυτό μπορεί να φαίνεται ότι είναι αυτοαναφορικό λύνοντας το πρόβλημα, δημιουργώντας το ίδιο πρόβλημα, όμως στην πράξη, ο αποστολέας χρειάζεται να έχει μόνο λίγα δημόσια κλειδιά από κάποιες λίγες CA υψηλής αξιοπιστίας και με αυτά μπoροεί να στειλεί μήνυμα σε σημαντικά περισσότερους παραλήπτες.

Ο Shamir, απέναντι στην λύση των Certificate Authority, πρότεινε μια διαφορετική λύση, την Identity-Based κρυπτογραφία. Εδώ όλοι οι παραλήπτες έχουν ένα μοναδικό identifier που είναι δημόσιο και παραμένει σταθερό όπως μια διεύθυνση ηλεκτρονικού ταχυδρομείου ή ακόμα και ένα UUID (Universally Unique Identifier – τυχαίος αριθμός αρκετών bits ώστε η σύγκρουση σε ίδιων αριθμών να είναι απίθανη).

Και πάλι υποθέτουμε μια έμπιστη άρχη (την συμβολίζουμε TTP – Trusted Third Party). Η TTP αναλαμβάνει για κάθε παραλήπτη, με βάση τον identifier του και το δικό της Ιδιωτικό Κλειδί, να εκδίδει ένα Ιδιωτικό Κλειδί για αυτόν, και να του τον δίνει (φυσικά με έμπιστη μετάδοση).

Ο αποστολέας για να στείλει ένα μήνυμα το κρυπτογραφεί με χρήση του Δημοσίου Κλειδιού της TTP και του Identifier του παραλήπτη.

Συνοψίζοντας, ο αποστολέας κρυπτογραφεί με βάση τον identifier και ο παραλήπτης βρίσκει το ιδιωτικό κλειδί από τον identifier του, ρωτώντας την TTP.

1.3.3.2 Υλοποίηση Boneh-Franklin

Οι Boneh και Franklin, με χρήση της Pairing-based κρυπτογραφίας, όρισαν το πρώτο πρακτικό σύστημα identity-based κρυπτογραφίας, το οποίο στηρίζεται στην υπολογιστική δυσκολία του BDHP.

Έστω δύο ομάδες G_1 (με σημείο βάσης P), G_T (τάξης $n \in \mathbb{P}$) για τις οποίες ορίζεται ένα bilinear pairing e, για το οποίο ισχύει η υπολογιστική δυσκολία του BDHP.

Έστω επίσης δύο συναρτήσεις κατακερματισμού $H_1 : \{0,1\}^* \mapsto G_1 - \{\infty\}$ και $H_2 : G_T \mapsto \{0,1\}^l$, όπου l το μήκος των μηνυμάτων m.

Το ιδιωτικό κλειδί της TTP επιλέγεται ως ένας ομοιόμορφα τυχαίος αριθμός $t \xleftarrow{R} \mathbb{Z}_n^*$ και το δημόσιο κλειδί της υπολογίζεται ως $T = tP$.

Έστω ότι ένας παραλήπτης έχει identifier ID_A (μια δυαδική συμβολοσειρά). Παρακάτω θεωρούμε ότι όλοι (όπου απαιτείται) υπολογίζουν το $Q_A = H_1(ID_A) \in G_1$.

Η TTP υπολογίζει το ιδιωτικό κλειδί για τον ID_A ως εξής: $d_A = tQ_A$.

Για να κρυπτογραφήσει ο αποστολέας ένα μήνυμα m προς τον ID_A εκτελεί τα εξής: Επιλέγει έναν ομοιόμορφα τυχαίο αριθμό $r \xleftarrow{R} \mathbb{Z}_n^*$ και έπειτα υπολογίζει και στέλνει το κρυπτοκείμενο:

$$(R, c) = \Big(rP, m \oplus H_2\Big(e(Q_A, T)^r\Big)\Big)$$

Για να αποκρυπτογραφήσει ο παραλήπτης υπολογίζει:

$$m = c \oplus H_2\Big(e(d_A, R)\Big)$$

Η ορθότητα του κρυπτοσυστήματος προκύπτει ως εξής:

$$\begin{aligned} & c \oplus H_2\Big(e(d_A, R)\Big) \\ &= m \oplus H_2\Big(e(Q_A, tT)^r\Big) \oplus H_2\Big(e(tQ_A, rR)\Big) \\ &= m \oplus H_2\Big(e(Q_A, T)^{rt}\Big) \oplus H_2\Big(e(Q_A, T)^{rt}\Big) \\ &= m \end{aligned}$$

Για να ανακτήσει ένας αντίπαλος το m από το (R, c) (υποθέτοντας pre-image resistance της H_2), πρέπει να υπολογίσει το $e(Q_A, T)^r$ από τα (P, Q_A, T, R). Υπάρχει k ώστε $Q_A = kP$. Συνέπως πρέπει να υπολογιστεί το $e(Q_A, T)^r = e(P, P)^{ktr}$ από τα $(P, Q_A, T, R) = (P, kP, tP, rP)$, δηλαδή πρέπει να λυθεί το πρόβλημα BDHP, που όμως έχει υποτεθεί υπολογιστικά δύσκολο.

Το παραπάνω σχήμα, βέβαια, είναι ευάλωτο σε Chosen Ciphertext Attack (CCA), καθώς ο αντίπαλος αν αλλάξει μόλις ένα bit του c, αυτό αντιστοιχεί στο m με αλλαγμένο ένα μόνο bit (λόγω της πράξης XOR), που δυνητικά μπορεί να είναι έγκυρο μήνυμα. Έτσι ο αντίπαλος αλλάζει ένα bit του c, το μαντείο αποκρπυογράφησης του δίνει ένα m' στο οποίο έχει αλλάξει μόνο αυτό το bit, ο αντίπαλος το ξανααλλάζει, και έτσι ανακτά το m.

Το πρόβλημα διορθώνεται με πάρομοιο τρόπο όπως στο RSA OAEP, που είναι επίσης η μέθοδος για να αποκτήσει το RSA ασφάλεια CCA. Χρησιμοποιούνται δύο ακόμα συναρτήσεις κατακερματισμού $H_3 : \{0,1\}^* \mapsto \mathbb{Z}_n^*$ και $H_4 : \{0,1\}^l \mapsto \{0,1\}^l$.

Για την συνάρτηση κρυπτογράφησης, επιλέγεται πρώτα μια ομοιόμορφα τυχαία συμβολοσειρά $\sigma \xleftarrow{R} \{0,1\}^l$. Αντί τυχαίου r, υπολογίζεται το $r = H_3(\sigma \mid\mid m)$ (που επίσης θεωρείται ομοιόμορφα τυχαίο στο μοντέλο του τυχαίου μαντείου μαντείου). Κρυπτογραφούμε, ακριβώς όπως πριν, την σ, υπολογίζοντας τα $R = rP$ και $c_1 = \sigma \oplus H_2\Big(e(Q_A, T)^r\Big)$. Όμως, επιπρόσθετα τώρα, υπολογίζουμε και το $c_2 = m \oplus H_4(\sigma)$. Στέλνουμε ως κρυπτοκείμενο το (R, c_1, c_2).

Για την συνάρτηση αποκρυπτογράφησης αποκρυπτογραφούμε, όπως πριν το (R, c_1) βρίσκοντας το $\sigma = c \oplus H_2\Big(e(d_A, R)\Big)$. Υπολογίζουμε, έπειτα $m = c_3 \oplus H_4(\sigma)$. Προτού επιστρέψουμε, ελέγχουμε αν $R = rP = H_3(\sigma \mid\mid m)$. Αν ισχύει επιστρέφουμε το m, αλλιώς δηλώνουμε αποτυχία (παραχαραγμένο μήνυμα).

Η ασφάλεια, εντός και του μοντέλου CCA (Chosen Ciphertext Attack), προκύπτει επειδή αν προκύψει ένα κρυπτοκείμενο με στο οποίο έχει μεταβληθεί οποιοδήποτε από τα R, σ, m και προκύψουν νέες τιμές (R, σ, m), για να ισχύει και μετά $R' = H_3(\sigma' \mid\mid m')$, επειδή η H_3 θεωρείται τυχαίο μαντείο, το m' είναι ανεξάρτητο του m, οπότε ο αντίπαλος δεν μπορεί να κερδίσει καμία πληροφορία για το m. Αντίστροφα, αν δεν μεταβληθεί τίποτα από τα R, σ, m, τότε έχουμε το ίδιο κρυπτοκείμενο με αρχικά, που όμως το μαντείο απροκρυπτογράφησης δεν μπορεί να αποκρυπτογραφήσει.

1.4 Παρατηρήσεις και Σχόλια για την κατασκευή Bilinear Pairing

Μέχρι τώρα, έχουμε παρουσιάσει διάφορες κρυπτογραφικές εφαρμογές των bilinear pairings, όμως δεν έχουμε σχολιάσει καθόλου το πως θα γίνουν realize τα bilinear pairings.

Μια συνήθης κατασκευή είναι το *Tate pairing.*

Η κατασκευή γίνεται πάνω σε ελλειπτική καμπύλη E πάνω σε πεπερασμένο πεδίο $\mathbb{F}_q$, όπου ορίζεται η συνήθης πράξη ομάδας που ορίζεται στην Κρυπτογραφία Ελλειπτικής Καμπύλης.

Υποθέτουμε ένα σημείο βάσης P και θέτουμε ως G_1 την (κυκλική) υποομάδα $\langle P \rangle$ που γεννά το P.

Έστω πως η $G_1 = \langle P \rangle$ έχει τάξη n.

Ορίζεται το embdedding degree, που είναι ο μικρότερος μη μηδενικός φυσικός αριθμός k ώστε $n \mid (q^k - 1)$.

Το Tate Pairing δίνει ομάδα G_T που είναι η (μοναδική) υποομάδα τάξης n του $\mathbb{F}_{q^k}^*$ (προσέχουμε πως είναι η πολλαπλασιαστική ομάδα του πεδίου και όχι η ομάδα της ελλειπτικής καμπύλης).

Το Tate Pairing υπολογίζεται, για δύο στοιχεία της G_1, με το *αλγόριθμο Miller*, που απαιτεί $O(\log n)$ επαναλήψεις καθεμία από τις οποίες χρειάζεται $O(1)$ αριθμητικές πράξεις στο $\mathbb{F}_{q^k}$.

Θυμόμαστε, όπως σχολιάστηκε παραπάνω το bilinear pairing μπορεί να χρησιμοποιηθεί για να υπολογιστεί ένα πρόβλημα Διακριτού Λογάριθμου στην G_1, στην G_T αντί της G_1.

Εδώ η G_T είναι μια υπόομαδα του $\mathbb{F}_{q^k}^*$ οπότε μπορούν να χρησιμοποιηθούν υποεκθετικοί αλγόριθμοι τύπου Index Calculus για αυτό.

Πριν την έλευση της Pairing-based κρυπτογραφίας, στα κλασικά κρυπτοσυστήματα Ελλειπτικής Καμπύλης, το *Tate pairing* και ο *αλγόριθμος Miller* μπορούσαν να χρησιμοποιηθούν για να οριστεί ένα bilinear pairing απλά και μόνο για να αναχθεί το Πρόβλημα Διακριτού Λογάριθμου στην ομάδα της Ελλειπτικής Καμπύλης, σε ένα Πρόβλημα Διακριτού Λογάριθμου σε μια υποομάδα του $\mathbb{F}_{q^k}^*$ όπου θα μπορούσαν να χρησιμοποιηθούν οι αποδοτικοί αλγόριθμοι Index Calculus.

Το πλεονέκτημα που εξαρχής έφεραν οι Ελλειπτικές Καμπύλες στην κρυπτογραφία ήταν πως δεν ήταν ευάλωτες σε υποεκθετικές επιθέσεις τύπου Index Calculus. Έτσι, αν το embedding degree k προέκυπτε χαμηλό, θα χανόταν το πλεονέκτημα που εξαρχής ερχόταν.

Έτσι, στα κλασικά κρυπτοσύστημα, στόχος ήταν το υψηλό embedding degree, και μάλιστα αυτό προβλεπόταν και στα πρότυπα, όπως στο ANSI X9.62. Εξάλλου, για μια τυχαία ελλειπτική καμπύλη, το embedding degree προκύπτει $k \approx n$, οπότε η ομάδα G_T γίνεται τάξης $q^k \approx q^n$.

Αν λ είναι η παράμετρος ασφάλειας, στόχος είναι επίσης και $n = O(2^\lambda)$ ώστε να έχουμε υπολογιστική δυσκολία στο Πρόβλημα Διακριτού Λογάριθμου. Έτσι, συνήθως, η ομάδα G_T αποκτούσε τάξη $q^k \approx q^n = O\left((2^\lambda)^{(2^\lambda)}\right)$, οπότε ακόμα και υποεκθετικός αλγόριθμος δεν θα ήταν πιο αποδοτικός από το να λύσουμε με εκθετικό αλγόριθμο το DLP στην αρχική ομάδα.

Ωστόσο, τώρα έχει προκύψει το ανάποδο πρόβλημα. Οι αριθμητικές πράξεις στο $\mathbb{F}_{q^k}$, που εκτελεί ο αλγόριθμος Miller, γίνονται σε αριθμούς μεγέθους bits $\log q^k = k \log q$ (απαιτούν πολυωνυμικό χρόνο ως προς το μέγεθος των αριθμών οι αριθμητικές πράξεις). Συνεπώς αν το embedding degree k είναι μεγάλο, το pairing δεν υπολογίζεται αποδοτικά.

Έτσι, στην pairing-based κρυπτογραφία, θέλουμε το πιο δυνατό χαμηλό embedding degree (ώστε ο αλγόριθμος Miller να είναι κατά το δυνατόν ταχύτερος), που όμως να είναι αρκούντως μεγάλο ώστε οι υποεκθετικοί αλγόριθμοι Index Calculus στην G_T να είναι πιο αργοί από τους εκθετικούς στην G_1.

Στην πράξη, αυτό οδήγησε στην δημιουργία νέων ελλειπτικών καμπύλων.

Έτσι έχουν δημιουργηθεί οι ελλειπτικές καμπύλες Barreto-Naehrig (με embedding degree $k = 12$) καθώς και άλλες, που αποκαλούνται ως "pairing-friendly".

Ενδεικτικά με τις καμπύλες Barreto-Naehrig επιτυγχάνεται στις υπογραφές Boneh-Lynn-Shacham (BLS), που περιγράφηκαν παραπάνω, ασφάλεια επιπέδου 128 bits (αυτό σημαίνει ότι χρειάζεται χρόνος $O(2^{128})$ για παραχάραξη), χρειάζεται η G_1 να έχει τάξη 2^{256} bits (για να μην είναι ευάλωτο το σύστημα σε επιθέσεις συγκρούσεων τύπου γενεθλίων – γενικό πρόβλημα στις ψηφιακές υπογραφές) και τότε η G_T έχει τάξη 2^{3072} που είναι αρκούντως μεγάλο για να μην είναι ευάλωτο το σύστημα σε επιθέσεις Index Calculus στην ομάδα G_T. Έτσι, προκύπτουν υπογραφές μεγέθους 256 bits, για ασφάλεια επιπέδου 128 bits.

2 Διαισθητική εισαγωγή στους Δημόσια Επαληθεύσιμους Υπολογισμούς (Publicly Verifiable Computations)

Στην σύγχρονη εποχή, έχει ανακύψει η ανάγκη να μεταφέρονται υπολογισμοί σε τρίτους, που συνήθως έχουν μεγαλύτερη υπολογιστική ισχύ από εκεί που χρειάζονται τα αποτελέσματα.

Ιδανικά, ένας χρήστης θα ήθελε να μπορεί να επαληθεύσει τα αποτέλεσματα που θα λάβει ώστε να μην χρειάζεται να εμπιστευτεί τον τρίτο που εκτέλεσε τον υπολογισμό, ότι οι υπολογισμοί έγιναν όπως υποσχέθηκε ότι έγιναν.

Οι εφαρμογές μπορούν να είναι ενδεικτικά στο Cloud Computing, ώστε να είμαστε βέβαιοι ότι οι υπολογισμοί γίνονται σωστά ή, σε πιο άμεση εφαρμογή, στο Blockchain στα Smart Contracts όπου θα θέλαμε να μεταφέρουμε υπολογιστικά κομμάτια από το Smart Contract σε κάποιον εξωτερικό υπολογιστή, διότι οι υπολογισμοί πάνω στο Blockchain έχουν υψηλό (οικονομικό) κόστος. Όμως, το Smart Contract πρέπει να είναι έπειτα σίγουρο πως τα αποτελέσματα που θα δοθούν προκύπτουν από υπολογισμούς που έγιναν ορθά και όπως ζήτηθηκε.

Εδώ μελετάμε σχήματα που σκόπο έχουν να μπορεί να ζητηθεί η αποτίμηση μιας συνάρτησης f με επαληθεύσιμο τρόπο.

Θα επιτρέψουμε κάθε παίχτης (δυνητικά κακόβουλος) να μπορεί:

1. Να δημιουργήσει τις παραμέτρους που θα επιτρέψουν τον δημόσιο υπολογισμό και την δημόσια επαλήθευση για μια συνάρτηση f της επιλογής του.
2. Να ζητήσει υπολογισμούς για κάποια είσοδο. (*public delegetability*)
3. Να εκτελέσει υπολογισμούς και να αποδείξει την ορθότητα τους.
4. Να ελέγξει την ορθότητα μιας εξόδου (για μια δεδομένη είσοδο). (*public verifiability*)

Το γεγονός πως αυτά επιτρέπονται για όλους τους παίχτες σημαίνει πως οι παίχτες δεν μπορούν να ορίσουν εκ των προτέρων ιδιωτικά κλειδιά ή εκ των προτέρων να υπάρχει οτιδήποτε αξιόπιστο.

3 Τυπικός Ορισμός και Επιθυμητές Ιδιότητες Δημόσια Επαληθεύσιμου Υπολογισμού

3.1 Ορισμός σχήματος

Τα σχήματα δημόσια επαληθεύσιμων υπολογισμών, ορίζονται με βάση τους ακόλουθους 4 PPT αλγόριθμους, που αντιστοιχούν ακριβώς στις 4 λειτουργίες που περιγράφηκαν προηγουμένως (παρακάτω θεωρούμε κ την παράμετρο ασφαλείας):

1. $Setup(1^\kappa, f) \to (param, PK_f, EK_f)$

   Τον εκτελεί ο χρήστης που θέλει να δημιουργήσει τις παραμέτρους για τον δημόσιο υπολογισμό και την δημόσια επαλήθευση μιας συνάρτησης f της επιλογής τους. Οι παράμετροι $param$ θεωρούμε παρακάτω πως δίνονται σε όλους (είναι ενδεικτικά επιλέγμενη ομάδα, γεννήτορας, κ.λπ.). Το PK_f είναι το δημόσιο κλειδί, και χρησιμοποιείται παρακάτω για επαληθεύσεις. Το EK_f είναι το κλειδί για παραγωγή πιστοποιητικών που θα χρειαστούν στις επαληθεύσεις. Και το PK_f και το EK_f θεωρούνται δημόσια.

2. $ProbGen(x, PK_f) \to (\sigma_x, VK_x)$

   Τον εκτελεί ο χρήστης που θέλει ζητήσει υπολογισμούς για κάποια είσοδο x. Το σ_x είναι μια κατάλληλη κωδικοποίηση της εισόδου που στην συνέχεια θα δοθεί στον χρήστη που θα εκτελέσει τον υπολογισμό και το VK_x είναι το δημόσιο κλειδί επαλήθευσης που θα δοθεί δημόσια σε όποιον θέλει να επαληθεύσει το αποτέλεσμα. Και εδώ τα σ_x, VK_x θεωρούνται δημόσια.

3. $Compute(\sigma_x, EK_f) \to \sigma_y$

   Τον εκτελεί ο χρήστης που αναλαμβάνει να εκτελέσει τον ζητούμενο υπολογισμό για είσοδο x. Η έξοδος σ_y είναι μια κωδικοποίηση της εξόδου $y = f(x)$ (θα περιλαμβάνει συνήθως κάποιο πιστοποιητικό για την επαλήθευση).

4. $Verify(\sigma_y, VK_x) \to out_y$

   Τον εκτελεί ο χρήστης που θέλει να επαληθεύσει ένα αποτέλεσμα. Ο αλγόριθμος επιστρέφει $out_y = y = f(x)$ αν το σ_y βρεθεί επαληθεύσιμο, και διαφορετικά, αν δηλαδή βρεθεί πως το αποτέλσμα που λήφθηκε δεν είναι το σωστό, επιστρέφει $out_y = \perp$.

3.2 Ορθότητα (Correctness)

Αρχικά, προκειμένου, ένα σχήμα δημόσια επαληθεύσιμου υπολογισμού να είναι χρήσιμο, θα πρέπει να εξασφαλίζει την *ορθότητα (correctness)*, που διαισθητικά σημαίνει πως όταν ο χρήστης εκτελέσει τίμια τον υπολογισμό, δηλαδή τρέξει την $Compute$, τότε θα πρέπει η $Verify$ να επιστρέφει (πάντα) $y = f(x)$. Το ορίζουμε τυπικότερα.

Ορισμός 3.1. *Ένα σχήμα δημόσια επαληθεύσιμου υπολογισμού ορίζεται ότι προσφέρει ορθότητα (correctness) για μια οικογένεια συναρτήσεων $\mathcal{F}$ αν για κάθε $f \in \mathcal{F}$ και κάθε $x \in D_f$ (όπου D_f το πεδίο ορισμού της f) ισχύει ότι:*

Αν $(param, PK_f, EK_f) = Setup(1^\kappa, f)$, $(\sigma_x, VK_x) = ProbGen(x, PK_f)$ και $\sigma_y = Compute(\sigma_x, EK_f)$ τότε:

$$\mathbb{P}\left[Verify(\sigma_y, VK_x) = f(x)\right] = 1$$

3.3 Αξιοπιστία (Soundness)

Επιπρόσθετα, το σχήμα θα πρέπει να διασφαλίζει ότι η $Verify$ επιστρέψει πως ένας υπολογισμός είναι πετυχημένος (δηλαδή $out_y \neq \perp$), τότε θα ισχύει $out_y = f(x)$. Αυτός ο περιορισμός χαλαρώνεται ελάχιστα απαιτώντας να υπάρχει αμελητέα, ως προς την παράμετρο ασφάλειας κ, πιθανότητα, να μην ισχύει $out_y = f(x)$. Την ιδιότητα αυτή θα αποκαλούμε *αξιοπιστία (soundness)*[1]. Προσέχουμε ότι η αξιοπιστία θέλουμε να ισχύει για κάθε συνάρτηση f. Το ορίζουμε τυπικότερα.

Όπως συνήθως, ορίζουμε ένα Τυχαίο Πείραμα Αξιοπιστίας για μια συγκεκριμένη συνάρτηση f, έναν αντίπαλο $\mathcal{A}$ και μια παράμετρο ασφαλείας κ:

1. $(param, PK_f, EK_f) = Setup(1^\kappa, f)$
2. Δίνουμε τα $param, PK_f, EK_f$ στον $\mathcal{A}$ και μας επιστρέφει ένα σημείο $x \in D_f$.
3. $(\sigma_x, VK_x) = ProbGen(x, PK_f)$
4. Δίνουμε το σ_x στον $\mathcal{A}$ και μας επιστρέφει σ_y.
5. $out_y = Verify(\sigma_y, VK_x)$ και επιστρέφουμε out_y.

Θα λέμε ότι ο αντίπαλος $\mathcal{A}$ επιτυγχάνει στο Τυχαίο Πείραμα αξιοπιστίας αν $out_y \neq \perp$ και $out_y \neq f(x)$.

Συμβολίζουμε $\Pi_{\mathcal{A},f}(\kappa)$ την πιθανότητα επιτυχίας αντιπάλου $\mathcal{A}$ στο Τυχαίο Πείραμα Αξιοπιστίας για μια συνάρτηση f και παραμέτρο ασφάλειας κ.

Ορισμός 3.2. *Ένα σχήμα δημόσια επαληθεύσιμου υπολογισμού ορίζεται ότι προσφέρει αξιοπιστία (soundness) για μια οικογένεια συνάρτησεων $\mathcal{F}$ αν αν για κάθε συνάρτηση $f \in \mathcal{F}$ και για κάθε PPT αντίπαλος $\mathcal{A}$ είναι:*

$$\Pi_{\mathcal{A},f}(\kappa) \leq negl(\kappa)$$

Σημείωση: Στο αρχικό paper, στον ορισμό της *αξιοπιστίας (soundness)* ο αντίπαλος $\mathcal{A}$ δεν αναφέρεται πως είναι PPT, όμως αυτό στην πραγματικότητα υπονοείται, αφού η αξιοπιστία θα στηριχθεί σε υπολογιστική δυσκολία προβλημάτων.

4 Υπόθεση υπολογιστικής δυσκολίας t-Strong Diffie Hellman (t-SDH)

Αρχικά, εφεξής, αναφερόμαστε στα bilinear pairings με λίγο τροποποιημένο ορισμό, λίγο πιο γενικό, επιτρέποντας τα δύο στοιχεία για τα οποία υπολογίζεται το pairing να είναι απο διαφορετικές ομάδες, δηλαδή το bilinear pairing είναι συνάρτηση $e : G_1 \times G_2 \mapsto G_T$ όπου G_1, G_2, G_3 κυκλικές ομάδες της ίδιας τάξης p. Κατά τα λοιπά, τα bilinear pairings ορίζονται όμοια με πριν.

Σημαντική σημείωση: Σε αντίθεση με προηγουμένως, παρακάτω συμβολίζουμε και τις τρεις G_1, G_2, G_T με πολλαπλασιαστικό συμβολισμό και e το ουδέτερο στοιχείο τους (συμπεραίνεται από τα συμφραζόμενα σε ποιας ομάδας το ουδέτερο στοιχείο αναφερόμαστε).

Ορίζουμε την υπόθεση t-SDH.

[1] η μετάφραση στα ελληνικά δεν είναι η συνήθης, όμως έχουμε και τους δύο αγγλικούς όρους *correctness* και *soundness* οπότε οι μεταφράσεις έπρεπε να διαφοροποιηθούν. Μια καλύτερη μετάφραση, που θα λάμβανε υπόψη το context θα ήταν ίσως η *πληρότητα* για το *correctness*, ωστόσο απέχει σημαντικά από την λέξη *correctness* και έτσι δεν την επέλεξα.

Ορισμός 4.1. *Έστω G_1, G_2, G_T κυκλικές ομάδες με την ίδια τάξη $p \in \mathbb{P}$. Θα λέμε ότι η υπόθεση t-Strong Diffie Hellman (t-SDH) ισχύει αν με είσοδο $(g, g^a, h, h^a, \cdots, h^{a^t}) \in G_1^2 \times G_2^{t+1}$ $(g \in G_1,\ h \in G_2)$ κάθε PPT αντίπαλος έχει αμελητέα ως προς κάποια παράμετρο ασφάλειας κ (οι ομάδες ορίζονται βάσει της κ) πιθανότητα να υπολογίσει ένα ζεύγος $(\beta, h^{(a+\beta)^{-1} \pmod p}) \in (\mathbb{Z}_p - \{-a\}) \times G_2$*

5 Δήμοσια Επαληθεύσιμη Αποτίμηση Πολυωνύμου

5.1 Εισαγωγή

Στην συνέχεια θα περιγράψουμε ένα σχήμα δημόσιου υπολογισμού (με βάση του προηγούμενους ορισμούς) που θα επιτρέπει την αποτίμηση ενός πολυωνύμου ορισμένο σε ένα πεπερασμένο πεδίο.

Διαισθητικά, για την αποτίμηση ενός πολυωνύμου $A(x)$ θα επιλέξουμε ένα πολυώνυμο 2ου βαθμού $B(x)$, θα υπολογίσουμε την διαίρεση $A(x) = Q(x)B(x) + R(x)$ και θα δημοσιεύσουμε τα $Q(x), B(x), R(x)$ με καμουφλαρισμένο τρόπο. Θα βάλουμε τον χρήστης που υπολογίζει να μας υπολογίσει και το $A(x)$ και με καμουφλαρισμένο τρόπο το $Q(x)$. Έπειτα στην επαλήθευση θα ελέγξουμε αν ισχύει $A(x) = Q(x)B(x) + R(x)$. Με τον καμουφλαρισμένο τρόπο εννοούμε ότι θα μπορεί να γίνουν οι υπολογισμοί και η επαλήθευση, χωρίς κανένας χρήστης να μπορεί να φτιάξει αποτέλεσμα $A(x), hidden[Q(x)]$ που θα αποδεχθεί η επαλήθευση.

5.2 Αναλυτικός ορισμός

Υποθέτουμε ότι η οικογένεια συναρτήσεων $\mathcal{F}$ είναι πολυώνυμα βαθμού d σε ένα πεπερασμένο πεδίο $\mathbb{F}_p$ όπου $p \in \mathbb{P}$. Ειδικότερα, τα πολυώνυμα αυτά ορίζονται ως $A(x) = \sum_{i=0}^{d} a_i x^i$ όπου $a_i \in \mathbb{F}_p$ (οι αριθμητικές πράξεις είναι του $\mathbb{F}_p$).

Ορίζουμε στην συνέχεια τους αλγόριθμους του σχήματος.

1. $\mathbf{Setup(1^{\varkappa}, A)}$

   Κατασκευάζονται 3 κυκλικές ομάδες G_1, G_2, G_T με τάξη πρώτο p για τις οποίες ορίζεται ένα bilinear pairing e και για τις οποίες ισχύει η υπόθεση $\lfloor \frac{d}{2} \rfloor$-SDH.

   Επιλέγονται επίσης στοιχεία $g \neq e \in G_1$, $h \neq e \in G_2$ (αφού $\mathrm{ord}\,(G_1), \mathrm{ord}\,(G_2) \in \mathbb{P}$ τα g, h είναι γεννήτορες).

   Σημειώνεται πως ο τρόπος που θα γίνει αυτή η δημιουργία αφήνεται ελεύθερος από τον αλγόριθμο και μπορεί να επιλέγει οποιαδήποτε κατασκευή που ικανοποιεί τις προϋποθέσεις που αναφέρονται. Ενδεικτικά, μπορεί να γίνει η κατασκευή με τις Ελλειπτικές Καμπύλες που περιγράφηκε παραπάνω.

   Οι δημόσιες παράμετροι $param$ ορίζονται ως $param = (p, G_1, G_2, G_T, e, g, h)$.

   Στην συνέχεια επιλέγεται ένα ομοιόμορφα τυχαίο $b_0 \xleftarrow{R} \mathbb{F}_p^*$ τέτοιο ώστε το πολύωνυμο $B(x) = x^2 + b_0$ να μην διαιρεί το $A(x)$ (αν βρουμε ότι δεν ισχύει, δοκιμάζουμε με νέο b_0).

   Εκτελούμε την διαίρεση του $A(x)$ με το $B(x)$ οπότε προκύπτουν $Q(x), R(x)$ ώστε να ισχύει η ταυτότητα της διαίρεσης $A = QB + R$ και επίσης το R είναι το πολύ 1ου βαθμού και του Q το πολύ $d-2$ βαθμού.

   Έστω ότι $Q(x) = \sum_{i=0}^{d-2} q_i x^i$ και $R(x) = r_1 x + r_0$.

   Υπολογίζω για κάθε $0 \leq i \leq d-2$ τα $\tilde{q_i} = h^{q_i}$ και επίσης τα $\tilde{b_0} = g^{b_0}, \tilde{r_1} = h^{r_1}, \tilde{r_0} = h^{r_0}$.

   Θέτω $PK_A = (\tilde{b_0}, \tilde{r_1}, \tilde{r_0}) = (g^{b_0}, h^{r_1}, h^{r_0})$.

   Θέτω $EK_A = (A, \tilde{q_0}, \tilde{q_1}, \cdots, \tilde{q_{d-2}}) = (A, h^{q_0}, h^{q_1}, \cdots, h^{q_{d-2}})$.

   (όπου A νοούνται οι $d+1$ πλήθους συντελεστές a_i του πολυωνύμου A)

   Επιστρέφω $(param, PK_A, EK_A)$.

2. $\mathbf{ProbGen(x, PK_A)}$

   Θέτω $\sigma_x = x$.

   Έχω λάβει $PK_A = (\tilde{b_0}, \tilde{r_1}, \tilde{r_0})$.

   Υπολογίζω $VK_{x,B} = \tilde{b_0} g^{x^2}$, $VK_{x,R} = \tilde{r_1}^x \tilde{r_0}$.

   Θέτω $VK_x = (VK_{x,B}, VK_{x,R})$.

   Επιστρέφω (σ_x, VK_x).

3. $\mathbf{Compute}(\boldsymbol{\sigma_x}, \mathbf{EK_A})$

   Για απλότητα συμβολισμού, συμβολίζω $x = \sigma_x$, όπως ισχύει στην τίμια περίπτωση.

   Έχω λάβει $EK_A = (A, \tilde{q_0}, \tilde{q_1}, \cdots, \tilde{q_{d-2}})$.

   Υπολογίζω το $y = A(x) = \sum_{i=0}^{d} a_i x^i$.

   Υπολογίζω το πιστοποιητικό $\pi = \prod_{i=0}^{d-2} \tilde{q_i}^{x^i}$.

   Επιστρέφω $\sigma_y = (y, \pi)$.

4. $\mathbf{Verify}(\boldsymbol{\sigma_y}, \mathbf{VK_x})$

   Έχω λάβει $\sigma_y = (y, \pi)$ και $VK_x = (VK_{x,B}, VK_{x,R})$.

   Ελέγχω αν ισχύει ότι:
   $$e(g, h^y) = e(VK_{x,B}, \pi)e(VK_{x,R})$$
   (όπου e είναι η συνάρτηση bilinear pairing που ορίστηκε παραπάνω στις δημόσιες παραμέτρους $param$)

   Αν ισχύει επιστρέφω $out_y = y$ αλλιώς επιστρέφω $out_y = \perp$.

5.3 Ορθότητα (Correctness)

Θεώρημα 5.1. *Το παραπάνω σχήμα δημόσιου υπολογισμού για την αποτίμηση πολυωνύμου ικανοποιεί την ορθότητα (correctness).*

Απόδειξη. Πρέπει να αποδείξουμε ότι αν οι $Setup$, $ProbGen$, $Compute$ τρέξουν όπως προβλέπουν οι (παραπάνω) ορισμοί του σχήματος, και επίσης ειδικότερα ο $ProbGen$ έχει κληθεί με όρισμα x, τότε η $Verify$ επιστρέφει $out_y = A(x)$.

Ισχύει ότι:
$$\pi = \prod_{i=0}^{d-2} \tilde{q_i}^{x^i} = \prod_{i=0}^{d-2} (h^{q_i})^{x^i} = \prod_{i=0}^{d-2} h^{q_i x^i} = h^{\sum_{i=0}^{d-2} q_i x^i} = h^{Q(x)}$$

Επίσης είναι $y = A(x)$ (αναφερόμαστε στο εξαγόμενο y από το σ_y).

Τελικά ισχύει ότι:
$$\begin{aligned}
& e(g, h^y) \\
&= e(g, h)^y \\
&= e(g, h)^{A(x)} \\
&= e(g, h)^{Q(x)B(x)+R(x)} \\
&= e(g, h)^{Q(x)B(x)} \qquad e(g, h)^{R(x)} \\
&= e(g^{B(x)}, h^{Q(x)}) \qquad e(g, h^{R(x)}) \\
&= e(g^{x^2+b_0}, \pi) \qquad e(g, h^{r_1 x + r_0}) \\
&= e(g^{b_0} g^{x^2}, \pi) \qquad e(g, (h^{r_1})^x h^{r_0}) \\
&= e(\tilde{b_0} g^{x_2}, \pi) \qquad e(g, \tilde{r_1}^x \tilde{r_0}) \\
&= e(VK_{x,B}, \pi) \qquad e(g, VK_{x,R})
\end{aligned}$$

Συνεπώς η $Verify$ θα επιστρέψει το $y = A(x)$, δηλαδή τελικά $out_y = y = A(x)$. □

5.4 Αξιοπιστία (Soundness)

Θεώρημα 5.2. *Το παραπάνω σχήμα δημόσιου υπολογισμού για την αποτίμηση πολυωνύμου ικανοποιεί την αξιοπιστία (soundness).*

Απόδειξη. Η απόδειξη βασίζεται σε αναγωγή του (υπολογιστικά δυσκόλου) προβλήματος $\lfloor \frac{d}{2} \rfloor$-SDH στο πρόβλημα επιτυχίας του Τυχαίου Πειράματος Αξιοπιστίας.

Έστω λοιπόν ένας PPT αντίπαλος $\mathcal{A}$ για το τυχαίο πείραμα αξιοπιστίας.

Κατασκευάζω έναν PPT αλγόριθμο $\mathcal{B}$ για το πρόβλημα $\lfloor \frac{d}{2} \rfloor$-SDH.

Δίνεται ως είσοδος ομάδες G_1, G_2, G_T, τάξης $p \in \mathbb{P}$, ένα bilinear pairing e και η πλειάδα $(g, g^v, h, h^v, h^{v^2}, \cdots, h^{v^{\lfloor \frac{d}{2} \rfloor}})$. Καλούμαστε να υπολογίσουμε μια τιμή $h^{(v+\beta)^{-1}}$ όπου β της επιλογής μας, αλλά θα πρέπει να επιστραφεί και αυτό στην έξοδο.

Το v συμβολίζεται με a συνηθέστερα, όμως, για να μην μπλεχτούν οι συμβολισμοί με τους συντελεστές a_i του $A(x)$, εδώ το συμβολίζουμε v.

Σημειώνεται πως η αξιοπιστία απαιτεί η ιδιότητά της να ικανοποιείται για κάθε συνάρτηση $f \in \mathcal{F}$, συνεπώς το ίδιο θα πρέπει να ισχύσει και εδώ, οπότε θεωρούμε το πολυώνυμο $A(x)$ ως άγνωστο, ώστε τελικά τα συμπεράσματα να ισχύουν για κάθε πολύωνυμο $A(x)$.

Στόχος είναι να προσομοιώσουμε μια εκτέλεση του Τυχαίου Πειραμάτος Αξιοπιστίας ώστε αν η εκτελούνταν όπως προβλέπει το σχήμα, να προέκυπταν αποτελέσματα με ίδια κατανομή πιθανότητας. Η μόνη, μη σταθερή από το σχήμα, επιλογή που υπάρχει είναι η τιμή b_0 που πρέπει να είμαι μια ομοιόμορφη τυχαία μεταβλητή. Θεωρόυμε $b_0 = v$. (στην είσοδο του SDH, το v θεωρείται μια ομοιόμορφη τυχαία μεταβλητή.)

Προσομοιώνουμε στην συνέχεια, το Τυχαίο Πειράμα Αξιοπιστίας, ακριβώς, όπως είναι ορισμένο, αλλά για $b_0 = v$.

Για την $Setup$:

Θέτουμε ως δημόσιες παράμετρους $param = (p, G_1, G_2, G_T, e, g, h)$, όπως μας δόθηκαν ως είσοδος του SDH.

Θα βρούμε τα πολυώνυμα $Q(x)$, $R(x)$,.

Είναι:

$$A(x) = Q(x)B(x) + R(x)$$

$$\iff \sum_{i=0}^{d} a_i x^i = \left(\sum_{i=0}^{d-2} q_i x^i\right)(x^2 + v) + r_1 x + r_0 = \left(\sum_{i=2}^{d} q_{i-2} x^i\right) + \left(\sum_{i=0}^{d-2} v q_i x^i\right) + r_1 x + r_0$$

$$\iff \begin{cases} a_d = q_d \\ a_{d-1} = q_{d-1} \\ a_i = q_{i-2} + v q_i \quad \forall\ 2 \leq i \leq d-2 \\ a_1 = r_1 + v q_1 \\ a_0 = r_0 + v q_0 \end{cases}$$

$$\iff \begin{cases} q_d = a_d \\ q_{d-1} = a_{d-1} \\ q_i = a_{i+2} - v q_{i+2} \quad \forall\ 0 \leq i \leq d-4 \\ r_1 = a_1 - v q_1 \\ r_0 = a_0 - v q_0 \end{cases}$$

Επιλύοντας την αναδρομική σχέση για τα q_i λαμβάνουμε ισοδύναμα:

$$q_{d-2-i} = \sum_{j=0}^{\lfloor \frac{i}{2} \rfloor} a_{d-i+2j} (-1)^j v^j$$

και

$$r_0 = \sum_{j=0}^{\lfloor \frac{d}{2} \rfloor} a_{2j} (-1)^j v^j$$

$$r_1 = \sum_{j=0}^{\lfloor \frac{d-1}{2} \rfloor} a_{2j+1} (-1)^j v^j$$

Ισχύει τότε:

$$\begin{aligned} \widetilde{q_{d-2-i}} &= h^{q_{d-2-i}} \\ &= h^{\sum_{j=0}^{\lfloor \frac{i}{2} \rfloor} a_{d-i+2j} (-1)^j v^j} \\ &= \prod_{j=0}^{\lfloor \frac{i}{2} \rfloor} h^{a_{d-i+2j} (-1)^j v^j} \\ &= \prod_{j=0}^{\lfloor \frac{i}{2} \rfloor} \left(h^{v^j}\right)^{a_{d-i+2j} (-1)^j} \end{aligned}$$

Όμοια προκύπτει ότι ισχύει:

$$\tilde{r_0} = h^{r_0} = \prod_{j=0}^{\lfloor \frac{d}{2} \rfloor} \left(h^{v^j}\right)^{a_{2j}(-1)^j}$$

$$\tilde{r_1} = h^{r_1} = \prod_{j=0}^{\lfloor \frac{d-1}{2} \rfloor} \left(h^{v^j}\right)^{a_{2j+1}(-1)^j}$$

Τα $h^{v^j},\ 0 \leq j \leq \lfloor \frac{d}{2} \rfloor$ είναι γνωστά από την είσοδο του SDH.

Συνεπώς τα $\tilde{q_i}, \tilde{r_0}, \tilde{r_1}$ υπολογίζονται με τις τελευταίες σχέσεις.

Επίσης το $\tilde{b_0} = g^{b_0} = g^v$ είναι και αυτό γνωστό από την είσοδο του SDH.

Συνεπώς έχοντας υπολογίσει τα τελευταία, επιστρέφω $PK_A = (\tilde{b_0}, \tilde{r_1}, \tilde{r_0})$ και $\mathrm{EK_A} = (A, \tilde{q_0}, \cdots, \tilde{q_{d-2}})$.

Συνεχίζοντας στο Τυχαίο Πείραμα Αξιοπιστίας, μετά την $Setup$ εκτελούμε ακριβώς το πείραμα κατά τα προβλεπόμενα, τρέχοντας τον $\mathcal{A}$, κλπ.

Στο τέλος ο $\mathcal{A}$ κερδίζει αν και μόνο αν η $Verify$ δώσει $out_y \neq \perp$, ισοδύναμα αν ο έλεγχος της $Verify$ πετύχει και αν $out_y \neq f(x)$.

Έστω ότι ο $\mathcal{A}$ κερδίζει.

Για το $\pi \in G_2$ που επιστράφηκε από τον $\mathcal{A}$ υπάρχει w ώστε $\pi = h^w$, αφού h γεννήτορας της G_2.

Τότε:

$$e(g, h^y) = e(VK_{x,B}, \pi)e(g, VK_{x,R})$$

Εκτελούμε την αρχική $Compute$, λαμβάνουμε ένα $\sigma_y = (y, \pi_*)$ ώστε επίσης η $Verify$ να αποδέχεται (με βάση την παραπάνω απόδειξη *Ορθότητας (Soundness)*):

$$e(g, h^{A(x)}) = e(VK_{x,B}, \pi^*)e(g, VK_{x,R})$$

Διαιρώντας κατά μέλη τις δύο σχέσεις λαμβάνουμε:

$$\begin{aligned} e(g, h^y)e(g, h^{A(x)})^{-1} &= e(VK_{x,B}, \pi)e(VK_{x,B}, \pi_*)^{-1} \\ \implies e(g, h)^{y-A(x)} &= e(VK_{x,B}, \pi\pi_*^{-1}) \\ \implies e(g, h)^{y-A(x)} &= e(g^{x^2+v}, \pi\pi_*^{-1}) \\ \implies e(g, h)^{y-A(x)} &= e(g, \pi\pi_*^{-1})^{x^2+v} \end{aligned}$$

Αφού το h γεννήτορας της G_2 υπάρχει w ώστε $\pi\pi_*^{-1} = h^w$.

Επίσης αφού τα g,h δεν είναι τα ουδέτερα στοιχεία, τότε ούτε και το $e(g, h)$ είναι το ουδέτερο στοιχείο της G_T, οπότε αφού $\mathrm{ord}\,(G_T) \in \mathbb{P}$ τότε το $e(g, h)$ είναι γεννήτορας της G_T.

Συνεπώς:

$$\begin{aligned} &e(g, h)^{y-A(x)} = e(g, \pi\pi_*^{-1})^{x^2+v} \\ \implies &e(g, h)^{y-A(x)} = e(g, h^w)^{x^2+v} = e(g, h)^{w(x^2+v)} \\ &\overset{\text{Βλ. }2}{\implies} y - A(x) = w(x^2 + v) \\ &\overset{y \neq A(x),\ \mathrm{ord}(G_2) \in \mathbb{P}}{\implies} (x^2 + v)^{-1} = w(y - A(x))^{-1} \end{aligned}$$

Τότε:

$$h^{(x^2+v)^{-1}} = h^{w(y-A(x))^{-1}} = \left(h^w\right)^{(y-A(x))^{-1}} = \left(\pi\pi_*^{-1}\right)^{(y-A(x))^{-1}}$$

Συνεπώς υπολογίζω το $\left(\pi\pi_*^{-1}\right)^{(y-A(x))^{-1}}$ (όλα τα επιμέρους γνωστά), αυτό είναι ίσο με $h^{(x^2+v)^{-1}}$ και επιστρέφω τελικά $(x^2, h^{(x^2+v)^{-1}})$ (δηλαδή $\beta = x^2$).

[2] Η ισότητα ισχύει $\mod \mathrm{ord}\,(G_T)$ αλλά συμβολίζουμε ισότητες τις ισοτιμίες.

Συνεπώς, αν ο $\mathcal{A}$ κερδίζει, κερδίζει και ο $\mathcal{B}$.

Όμως από την υπόθεση υπολογιστικής δυσκολίας του $\lfloor \frac{d}{2} \rfloor$-SDH ο $\mathcal{B}$ κερδίζει με αμελητέα ως προς κ (διότι $\mathrm{ord}\,(G_1) = \mathrm{ord}\,(G_2) = \mathrm{ord}\,(G_T) = n = \Theta(2^\kappa)$) πιθανότητα.

Συνεπώς και ο $\mathcal{A}$ κερδίζει με αμελήτεα ως προς κ πιθανότητα, δηλαδή $\Pi_{\mathcal{A},f} \leq negl(\kappa)$ για κάθε PPT αντίπαλο $\mathcal{A}$ και κάθε συνάρτηση $f \in \mathcal{F}$.

Συνεπώς το σχήμα δημόσιου υπολογισμού προσφέρει *αξιοπιστία (soundness).* □

5.5 Σύνοψη των επιδόσεων

Ο αλγόριθμος $Setup$ απαιτεί:

1. 1 τυχαίο στοιχείο
2. d πολλαπλασιασμούς στο $\mathbb{F}_p$
3. 1 ύψωση σε δύναμη στην G_1
4. $d + 1$ υψώσεις σε δυνάμεις στην G_2

Ο αλγόριθμος $ProbGen$ απαιτεί:

1. 1 πολλαπλασιασμό στο $\mathbb{F}_p$
2. 1 ύψωση σε δύναμη και 1 πολλαπλασιασμό στην G_1
3. 1 ύψωση σε δύναμη και 1 πολλαπλασιασμό στην G_2

Ο αλγόριθμος $Compute$ απαιτεί (θεωρώντας ότι η αποτίμηση του πολυωνύμου γίνεται με τον *Κανόνα Horner*):

1. $2d - 3$ πολλαπλασιασμούς (και $d - 1$ προσθέσεις) στο $\mathbb{F}_p$
2. $d - 1$ υψώσεις σε δύναμη και $d - 2$ πολλαπλασιασμούς στην G_2

Ο αλγόριθμος $Verify$ απαιτεί:

1. 1 ύψωση σε δύναμη και 1 υπολογισμό αντιστρόφου στην G_2
2. 2 bilinear pairings

Το σημαντικό είναι ότι το $Verify$ είναι ταχύτερο από το να γίνει επανυπολογισμός του $A(x)$, που θα ήθηλε d πολλαπλασιασμούς και d προσθέσεις στο $\mathbb{F}_p$.

6 Υπόθεση υπολογιστικής δυσκολίας co-computational Diffie-Hellman (*co*-CDH)

Ορίζουμε την υπόθεση *co*-CDH.

Ορισμός 6.1. *Έστω G_1, G_2, G_T κυκλικές ομάδες με την ίδια τάξη $p \in \mathbb{P}$. Θα λέμε ότι η υπόθεση co-computational Diffie-Hellman (co-CDH) ισχύει αν με είσοδο $g, g^a \in G_1$ και $h, h^b \in G_2$ (όπου τα a, b επιλεγμένα ομοιόμορφα τυχαία $a, b \xleftarrow{R} \mathbb{F}_p^*$) κάθε PPT αντίπαλος έχει αμελητέα ως προς μια παράμετρο ασφάλειας κ πιθανότητα να υπολογίσει την τιμή g^{ab}.*

7 Δημόσια Επαληθεύσιμος Πολλαπλασιασμός Πίνακα-Διανύσματος

7.1 Εισαγωγή

Σκοπός εδώ είναι ο δημόσια επαληθεύσιμος υπολογισμός ενός πολλαπλασιασμού πίνακα-διανύσματος όπου η αριθμητική ορίζεται σε ένα πεπερασμένο πεδίο $\mathbb{F}_p$. Σημειώνεται πως ο ζητούμενος υπολογισμός δεν περιλαμβάνει στον ορισμό του διαίρεση, οπότε υπό την προϋπόθεση πως οι αριθμοί δεν ξεπερνούν το p (δεν συμβαίνει δηλαδή overflow), η αριθμητική μπορεί να θεωρηθεί η συνήθης αριθμητική ακεραίων.

Σημειώνεται ότι στο αρχικό paper, ακολουθείται συμβολισμός με indexes και αθροίσματα, εδώ θα χρησιμοποιηθεί διανυσματικός συμβολισμός για καλύτερη διαίσθηση και κομψότητα. Αυτό, όπως θα δούμε θα μας βοηθήσει να προτείνουμε μια δική μας βελτίωση της επίδοσης σε ένα σημείο.

Για διανυσματικά μεγέθη $\vec{x}$ συμβολίζω με x_i την i-οστή συντεταγμένη τους. Όμοια για πίνακες.

7.2 Αναλυτικός ορισμός

Η οικογένεια συναρτήσεων $\mathcal{F}$ ορίζεται ως $\{\vec{x} \mapsto M\vec{x} \mid M \in \mathbb{F}_p^{n\times m}\}$.

Ο πίνακας M καθορίζεται στην αρχή κατά το *Setup* και στην συνέχεια οι χρήστες μπορούν να ζητήσουν υπολογισμούς για περισσότερες εισόδους $\vec{x}$.

Ορίζουμε στην συνέχεια τους αλγόριθμους του σχήματος.

1. $\mathbf{Setup}(\mathbf{1}^{\boldsymbol{\varkappa}}, \mathbf{M})$

   Επιλέγονται δύο κυκλικές ομάδες G_1, G_2 με τάξη $p \in \mathbb{P}$ και μια ακόμα ομάδα G_T ώστε να ορίζεται ένα bilinear pairing $e : G_1 \times G_2 \mapsto G_T$. Οι ομάδες επιλέγονται ώστε να ισχύει η υπόθεση *co*-CDH για τις (G_1, G_2).

   Επιλέγεται γεννήτορας h της G_2 και μια ομοιόμορφα τυχαία τιμή $\delta \overset{R}{\leftarrow} \mathbb{F}_p^*$. Υπολογίζεται το $\tilde{h} = h^\delta$.

   Επιλέγεται γεννήτορας g της G_1 και ομοιόμορφα τυχαίο διάνυσμα $\vec{\lambda} \overset{R}{\leftarrow} {\mathbb{F}_p^*}^n$. Υπολογίζεται το $\vec{g} = g^{\vec{\lambda}}$ (όπου η $\vec{x} \mapsto g^{\vec{x}}$ θεωρείται πως ορίζεται στοιχείο-στοιχείο).

   Επιλέγεται ομοιόμορφα τυχαίος πίνακας $R \overset{R}{\leftarrow} \mathbb{F}_p^{n\times m}$.

   Υπολογίζεται ο πίνακας N με στοιχεία $N_{ij} = g_i^{\delta M_{ij} + R_{ij}} = g^{\delta \lambda_i M_{ij} + \lambda_i R_{ij}}$. Ανακύπτει το άμεσο ερώτημα αν ο N μπορεί να οριστεί με μια μητρωική έκφραση. Όπως θα δούμε, με βάση τον τρέχοντα ορισμό αυτό δεν γίνεται, όμως στην πραγματικότητα *(δική μας συνεισφορά θα εξηγηθεί αργότερα)* ο πίνακας N θα μπορούσε να οριστεί απλούστερα ως το διάνυσμα $N_j = \prod_{i=0}^n N_{ij} = g^{\sum_{i=0}^n \delta\lambda_i M_{ij} + \lambda_i R_{ij}}$ που ισοδύναμα γράφεται $\vec{N} = g^{\delta M^T \vec{\lambda} + R^T \vec{\lambda}} = g^{(\delta M + R)^T \vec{\lambda}}$ (αρχικά παρουσιάζουμε το σχήμα με τον πίνακα N και στο τέλος περιγράφουμε τις προτεινόμενες τροποποιήσεις).

   Θέτουμε το κλειδί αποτίμησης ως $EK_M = (M, N)$.

   Έπειτα υπολογίζουμε το δημόσιο κλειδί $PK_j = e\left(\prod_{i=1}^n g_i^{R_{ij}}, h\right)$ ή ισοδύναμα $\vec{PK} = e\left(g^{R^T\vec{\lambda}}, h\right)$ (όπου το bilinear pairing επεκτείνεται σε διανυσματικό ορισμό ως στοιχείο-στοιχείο).

   Τελικά, επιστρέφουμε αρχικά τις δημόσιες παραμέτρους $params = (p, G_1, G_2, G_T, e, \vec{g}, h, \tilde{h})$ (προσέχουμε ότι δεν δίνεται το g), το κλειδί επαλήθευσης EK_M και το δημόσιο κλειδί VK_x.

2. $\mathbf{ProbGen}(\vec{\mathrm{x}}, \mathbf{PK_M})$

   Υπολογίζεται το $VK_x = \prod_{j=1}^m PK_j^{x_j}$.

   Θέτουμε και $\sigma_x = \vec{x}$.

   Επιστρέφονται τα σ_x και VK_x.

3. $\mathbf{Compute}(\boldsymbol{\sigma}_\mathbf{x}, \mathbf{EK_M})$

   Θεωρούμε $x = \sigma_x$ και $(M, N) = EK_M$.

   Αρχικά υπολογίζεται το $\vec{y} = M\vec{x}$.

Έπειτα υπολογίζεται το πιστοποιητικό:

$$\Pi = \prod_{i=1}^{n}\prod_{j=1}^{m} N_{ij}^{x_j}$$

Θέτουμε $\sigma_y = (\vec{y}, \Pi)$ και το επιστρέφουμε.

4. $\mathbf{Verify}(\boldsymbol{\sigma}_\mathbf{y}, \mathbf{VK_x})$

   Θεωρούμε $(\vec{y}, \Pi) = \sigma_y$.

   Εξετάζουμε αν ισχύει:

$$e(\Pi, h) = e\left(\prod_{i=1}^{n} g_i^{y_i}, \tilde{h}\right) VK_x$$

   Αν ισχύει επιστρέφουμε $\vec{y}$ αλλιώς $\perp$.

7.3 Ορθότητα (Correctness) + Διανυσματική Ερμηνεία

Θεώρημα 7.1. *Το παραπάνω σχήμα δημόσιου υπολογισμού ικανοποιεί την Ορθότητα (Correctness).*

Απόδειξη. Ισχύει ότι:

$$\begin{aligned} VK_x &= \prod_{j=1}^{m} e\left(g^{\left(R^T\vec{\lambda}\right)_j}, h\right)^{x_j} \\ &= e\left(g^{\sum_{j=1}^{m}\left(R^T\vec{\lambda}\right)_j x_j}, h\right) \\ &= e\left(g^{\left(R^T\vec{\lambda}\right)\vec{x}}, h\right) \\ &= e\left(g^{(R\vec{x})\vec{\lambda}}, h\right) \end{aligned}$$

Σημειώνεται πως χρησιμοποιήθηκε το παρακάτω λήμμα:

Λήμμα 7.1. *Έστω πίνακας A και διανύσματα $\vec{x}$, $\vec{y}$. Ισχύει ότι: $(A\vec{x})\,\vec{y} = (A^{\mathrm{T}}\vec{y})\,\vec{x}$*

Απόδειξη. Χρησιμοποιούμε τον μητρωικό συμβολισμό εσωτερικού γινομένου $\vec{x}\vec{y} = x^T y = y^T x$ (όπου τα διανύσματα θεωρούνται ως διανύσματα-στήλες.).

Τότε:

$$(A\vec{x})\,\vec{y} = (Ax)^T y = (x^T A^T) y = x^T (A^T y) = (A^T \vec{y})\,\vec{x}$$

□

Τότε:

$$\begin{aligned} \Pi &= \prod_{i=1}^{n}\prod_{j=1}^{m} N_{ij}^{x_j} \\ &= \prod_{i=1}^{n}\prod_{j=1}^{m} g^{\delta\lambda_i M_{ij} x_j + \lambda_i R_{ij} x_j} \\ &= g^{\sum_{i=1}^{n}\sum_{j=1}^{m} \delta\lambda_i M_{ij} x_j + \lambda_i R_{ij} x_j} \\ &= g^{\sum_{i=1}^{n} \delta\lambda_i M\vec{x} + \lambda_i R\vec{x}} \\ &= g^{\delta(M\vec{x})\vec{\lambda} + (R\vec{x})\vec{\lambda}} \\ &= g^{((\delta\mathrm{M}+R)\vec{x})\vec{\lambda}} \end{aligned}$$

Επίσης:

$$
\begin{aligned}
& e\left(\prod_{i=1}^{n} g_i^{y_i}, \tilde{h}\right) VK_x \\
&= e\left(g^{\sum_{i=1}^{n}\lambda_i y_i}, h^{\delta}\right) e\left(g^{(R\vec{x})\vec{\lambda}}, h\right) \\
&= e\left(g^{\vec{\lambda}\vec{y}}, h^{\delta}\right) e\left(g^{(R\vec{x})\vec{\lambda}}, h\right) \\
&= e\left(g^{\delta\vec{\lambda}\vec{y}+R\vec{x}}, h\right) \\
&= e\left(g^{((\delta\mathrm{M}+R)\vec{x})\vec{\lambda}}, h\right) \\
&= e\left(\Pi, h\right)
\end{aligned}
$$

Συνεπώς η $Verify$ επιστρέφει $\vec{y} = \mathrm{M}\vec{x}$. □

7.4 Αξιοπιστία (Soundness)

Θεώρημα 7.2. *Το παραπάνω σχήμα δημόσιου υπολογισμού ικανοποιεί την Αξιοπιστία (Soundness).*

Απόδειξη. Έστω ένας PPT αντίπαλος $\mathcal{A}$ για το τυχαίο πείραμα αξιοπιστίας για το παραπάνω σχήμα δημόσιου υπολογισμού και για έναν συγκεκριμένο πίνακα M.

Κατασκευάζω έναν PPT αλγόριθμο $\mathcal{B}$ για το πρόβλημα co-CDH.

Δίνονται ως είσοδος ομάδες G_1, G_2, bilinear pairing e και στοιχεία $g', {g'}^a \in G_1$ και $h', {h'}^b \in G_2$. Ζητείται ο υπολογισμός του ${g'}^{ab}$.

Αρχικά προσομοιώνουμε την $Setup$. Επιλέγονται $g = {g'}^a$, $h = h'$ και $\delta = \delta' \cdot b$ όπου το $\delta' \xleftarrow{R} \mathbb{F}_p^*$.

Τότε υπολογίζω: $\tilde{h} = h^{\delta} = \left({h'}^b\right)^{\delta'}$.

Όπως στην $Setup$, επιλέγεται ομοιόμορφα τυχαίο $\vec{\lambda} \xleftarrow{R} {\mathbb{F}_p^*}^n$ και υπολογίζεται το $\vec{g} = g^{\vec{\lambda}}$.

Επιλέγεται ομοιόμορφα τυχαίος πίνακας $N \xleftarrow{R} G_1^{n\times m}$.

Υπολογίζω το δημόσιο κλειδί:

$$PK_j = \frac{e\left(\prod_{i=1}^{n} N_{ij}, h\right)}{e\left(\prod_{i=1}^{n} g_i^{M_{ij}}, \tilde{h}\right), \tilde{h}}$$

Θέτω κλειδί επαλήθευσης $EK_M = (M, N)$.

Θέτω δημόσιες παραμέτρους $param = (p, G_1, G_2, G_T, e, \vec{g}, h, \tilde{h})$.

Επιστρέφω $param, PK_M, EK_M$.

Λήμμα 7.2. *Η έξοδος της προσομοιωμένης Setup έχει ίδια κατανομή πιθανότητας με την έξοδο μιας εκτέλεσης Setup.*

Απόδειξη. Ως προς τις δημόσιες παραμέτρους $param$, αυτό ισχύει λόγω της τυχαιότητας του $\vec{\lambda}$ και του δ (προκύπτει λόγω τυχαιότητας του δ').

Αφού g γεννήτορας g_1 υπάρχουν μοναδικά n_{ij} τέτοια ώστε $N_{ij} = g^{n_{ij}}$. Τότε ισχύει ισοδύναμα:

$$PK_j = e\left(g^{\sum_{i=1}^{n} n_{ij} - \delta\lambda_i M_{ij}}, h\right)$$

Έστω πίνακας $R_{ij} = n_{ij}\lambda_i^{-1} - \delta M_{ij}$.

Τότε: $PK_j = e\left(g^{\sum_{i=1}^{n} R_{ij}\lambda_i}, h\right)$, ισοδύναμα $\vec{PK} = e\left(g^{R^T\lambda}, h\right)$.

Επίσης: $n_{ij} = R_{ij}\lambda_i + \delta\lambda_i M_{ij}$, οπότε $N_{ij} = g^{R_{ij}\lambda_i + \delta\lambda_i M_{ij}}$.

Συνεπώς αν η $Setup$ είχε επιλέξει τον συγκεκριμένο R, όταν επέλεξε τυχαίο πίνακα R, τότε θα προέκυπταν τα αποτέλεσματα που προκύπτουν.

Ο R είναι ομοιόμορφα τυχαίος λόγω της τυχαιότητας των n_{ij} (προκύπτει λόγω της τυχαιότητας των N_{ij}). Συνεπώς μπορεί ισοδύναμα να επιλέγει ο τυχαίος R και προκύπτει ως η ίδια συνάρτηση τα EK_M, PK_M. Συνεπώς η προσομοιωμένη $Setup$ δίνει αποτελέσματα με ίδια κατανομή πιθανότητας με μια κανονική εκτέλεση της $Setup$ όπως προβλέπει το σχήμα.

Σημείωση: Παρακάτω αναφερόμαστε στον πίνακα R που προκύπτει μοναδικά από τον N. Πουθένα δεν χρειάζεται ο υπολογισμός του, μόνο για απλότητα των αποδείξεων, και προκειμένου να έχουμε ίδιο συμβολισμό με την Ορθότητα (Correctness). □

Στην συνέχεια εκτελούμε το Τυχαίο Πείραμα Αξιοπιστίας ακριβώς όπως προβλέπεται.

Έστω ότι ο $\mathcal{A}$ κερδίζει το Τυχαίο Πείραμα Αξιοπιστίας.

Στο τέλος ο $\mathcal{A}$ έχει επιστρέψει x και $(\vec{y}, \Pi) = \sigma_y$ ώστε $Verify(\sigma_y, VK_x) = \vec{y} \neq \perp$ και $\vec{y} \neq \vec{y_*}$ όπου $\vec{y_*} = M\vec{x}$.

Αφού $Verify(\sigma, y, VK_x) \neq \perp$ τότε:

$$\begin{aligned} e(\Pi, h) &= e\left(\prod_{i=1}^{n} g_i^{y_i}, \tilde{h}\right) VK_x \\ \implies e(\Pi, h) &= e\left(g^{\sum_{i=1}^{n} g^{\lambda_i y_i}}, h^{\delta}\right) e\left(g^{(R\vec{x})\vec{\lambda}}, h\right) \\ &= e\left(g^{\delta\vec{\lambda}\vec{y}+(R\vec{x})\vec{\lambda}}, h\right) \\ &= e\left(g^{(\delta\vec{y}+R\vec{x})\vec{\lambda}}, h\right) \end{aligned}$$

Παραπάνω χρησιμοποιήσαμε ότι $VK_x = e\left(g^{(R\vec{x})\vec{\lambda}}, h\right)$ που αποδείχθηκε εντός της απόδειξης ορθότητας (correctness).

Αφού g γεννήτορας G_1 υπάρχει $w \in \mathbb{F}_p$ ώστε $\Pi = g^w$. Άρα:

$$e(g,h)^w = e(g^w, h) = e(\Pi, h) = e\left(g^{(\delta\vec{y}+R\vec{x})\vec{\lambda}}, h\right) = e(g,h)^{(\delta\vec{y}+R\vec{x})\vec{\lambda}}$$

Επειδή g, h γεννήτορες τότε $e(g,h) \neq e$ (λόγω της 2ης ιδιότητας των bilinear pairings) και τότε επειδή $\operatorname{ord}(G_T) \in \mathbb{P}$ το $e(g,h)$ είναι γεννήτορας της G_T.

Συνεπώς: $w = (\delta\vec{y} + R\vec{x})\vec{\lambda}$.

Άρα: $\Pi = g^w = g^{(\delta\vec{y}+R\vec{x})\vec{\lambda}}$.

Στην συνέχεια εκτελώ την $\sigma_{y*} = Compute(\vec{x}, EK_M)$. Επειδή τα αποτελέσματα της προσομοιωμένης $Setup$ έχουν ίδια κατανομή πιθανότητας με μια κανονική εκτέλεση της $Setup$, τότε από την απόδειξη ορθότητας (correctness) προκύπτει $Verify(\sigma_{y*}, VK_x) = y_* = M\vec{x}$.

Έστω τότε $\sigma_{y*} = (\vec{y_*}, \Pi_*)$.

Όπως αποδείχθηκε στην απόδειξη ορθότητας:

$$\Pi_* = g^{((\delta M+R)\vec{x})\vec{\lambda}}$$

Τότε:

$$\begin{aligned} \Pi\Pi_*^{-1} &= g^{(\delta\vec{y}+R\vec{x})\vec{\lambda}-((\delta M+R)\vec{x})\vec{\lambda}} \\ &= g^{\delta(\vec{y}-M\vec{x})\vec{\lambda}} \\ &= g^{\delta(\vec{y}-\vec{y_*})\vec{\lambda}} \end{aligned}$$

Αν $\vec{y}\vec{\lambda} = \vec{y_*}\vec{\lambda}$ δηλώνουμε αποτυχία στο σημείο αυτό. Θα αποδείξουμε στην συνέχεια σε επόμενο θεώρημα μετά την απόδειξη αυτή πως αυτό συμβαίνει μόνο με αμελητέα πιθανότητα ως προς την παράμετρο ασφάλειας κ.

Συνεπώς $(\vec{y} - \vec{y_*})\vec{\lambda} \neq 0$, οπότε ορίζεται ο $\left((\vec{y} - \vec{y_*})\vec{\lambda}\right)^{-1} \pmod{\operatorname{ord}(G_1)}$ διότι $\operatorname{ord}(G_1) = p \in \mathbb{P}$.

Επίσης, στην αρχή είχαμε θέσει $\delta = \beta\delta' \implies \delta\delta'^{-1} = \beta$ (διότι $\delta' \in \mathbb{F}_p^*$ και $p \in \mathbb{P}$).

Συνεπώς υπολογίζω:

$$\left(\Pi\Pi_*^{-1}\right)^{\delta'^{-1}\left((\vec{y}-\vec{y_*})\vec{\lambda}\right)^{-1}} = g^b = \left(g'^a\right)^b = g'^{ab}$$

Βρήκα το g'^{ab}, δηλαδή την λύση του co-CDH και την επιστρέφω.

Έστω ότι ο $\mathcal{A}$ κερδίζει με μη αμελητέα ως προς κ πιθανότητα. Τότε ο $\mathcal{B}$ κερδίζει και αυτός με μη αμελητέα ως προς κ πιθανότητα (αφού δηλώνει αποτυχία αν κερδίσει ο $\mathcal{A}$ με αμελητέα ως προς κ πιθανότητα).

Συνεπώς ο αλγόριθμος $\mathcal{B}$ είναι ένας PPT αλγόριθμος που κερδίζει το co-CDH με μη αμελητέα ως προς κ πιθανότητα, που είναι άτοπο λόγω της co-CDH υπόθεσης.

Συνεπώς, το σχήμα δημόσιου υπολογισμού ικανοποιεί την αξιοπιστία (soundness). □

Παραπάνω, υποθέσαμε το παρακάτω θεώρημα το οποίο τώρα αποδεικνύουμε.

Θεώρημα 7.3. *Κάθε PPT αντίπαλος $\mathcal{A}$ για το Τυχαίο Πείραμα Αξιοπιστίας για το δημόσιο σχήμα υπολογισμού για κάθε πίνακα M, όταν κερδίζει το πείραμα, έχει αμελητέα πιθανότητα να επιστρέψει $(\vec{y}, \Pi) = \sigma_y$ ώστε να ισχύει $\vec{y}\vec{\lambda} = \vec{y_*}\vec{\lambda}$, όπου $\vec{y_*} = M\vec{x}$ και $\vec{x}$ το σημείο $\vec{x}$ που είχε δώσει αρχικά ο $\mathcal{A}$.*

Απόδειξη. Θα απόδειξω ότι ισχύει με αναγωγή του προβλήματος διακριτού λογαριθμού. Ειδικότερα, έστω ότι υπάρχει PPT αντίπαλος $\mathcal{A}$ που έχει μη αμελητέα πιθανότητα να δώσει τελικά $\vec{y}\vec{\lambda} = \vec{y_*}\vec{\lambda}$.

Κατασκευάζω τότε PPT αλγόριθμο $\mathcal{B}$ για το πρόβλημα διακριτού λογάριθμου στην ομάδα G_1.

Δίνονται $g', g'^a \in G_1$ και πρέπει να υπολογίσουμε το $a \pmod{\operatorname{ord}(G_1)}$.

Έστω ότι ο $\mathcal{A}$ κερδίζει το Τυχαίο Πείραμα Αξιοπιστίας για πίνακα M.

Ακολουθούμε την ίδια προσομοίωση της $Setup$ όπως στην παραπάνω απόδειξη Αξιοπιστίας (Soundness), με την μόνη διαφορά πως εδώ χρησιμοποιούμε τον γεννήτορα $g = g'$, τον γεννήτορα h τον επιλέγουμε ομοιόμορφα τυχαία και επίσης τροποποιούμε τον ορισμό του διανύσματος $\vec{\lambda}$.

Επιλέγουμε έναν ομοιόμορφα τυχαίο αριθμό $k \stackrel{R}{\leftarrow} \mathbb{Z}_n$.

Θέτουμε $\lambda_k = a$ και τα υπόλοιπα λ_i τα λαμβάνουμε ομοιόμορφα τυχαία όπως πριν.

Σημειώνεται ότι χρειαζόμαστε μόνο το $g_k = g^{\lambda_k} = {g'}^a$ που έχει δοθεί και όχι το ίδιο το λ_k.

Τελικά ο $\mathcal{A}$ επιστρέφει $\vec{x}, \vec{y}, \Pi$ και έστω ότι ο $\mathcal{A}$ επιτυγχάνει.

Αν ο $\mathcal{A}$ αποτύχει δηλώνω αποτυχία στο σημείο αυτό – με αμελητέα πιθανότητα ως προς κ.

Ισχύει τότε ότι:

$$\begin{aligned}
\vec{y}\vec{\lambda} &= \vec{y_*}\vec{\lambda} \\
\implies \sum_{i=1}^{n} y_i \lambda_i &= \sum_{i=1}^{n} y_{*i} \lambda_i \\
\implies (y_k - y_{*k})\lambda_k &= \sum_{i=1, i \neq k}^{n} (y_{*i} - y_i)\lambda_i \\
\implies (y_k - y_{*k})a &= \sum_{i=1, i \neq k}^{n} (y_{*i} - y_i)\lambda_i
\end{aligned}$$

Αν $y_k = y_{*k}$ δηλώνω αποτυχία στο σημείο αυτό.

Ωστόσο το διάνυσμα λ είναι ομοιόμορφα τυχαίο και ανεξάρτητο του k διότι στο στιγμιότυπο του DLog το a θεωρείται και αυτό ομοιόμορφα τυχαίο. Συνεπώς η είσοδος του $\mathcal{A}$ είναι ανεξάρτητη του k.

Επειδή ο $\mathcal{A}$ πέτυχε $y \neq y_*$ οπότε υπάρχει τουλάχιστον ένα k' ώστε $y_{k'} \neq y_{*k'}$.

Η πιθανότητα να ισχύει $k' = k$ είναι $\frac{1}{n}$ διότι το k είναι ομοιόμορφα τυχαίο και ανεξάρτητο του k' αφού το k' τελικά προκύπτει από την είσοδο του $\mathcal{A}$ που είναι ανεξάρτητη του k.

Άρα με πιθανότητα $\frac{1}{n}$ δεν δηλώνω αποτυχία. Η πιθανότητα αυτή είναι μη αμελητέα ως προς την παράμετρο ασφαλείας κ. Συνεπώς με αμελητέα ως προς κ πιθανότητα συνεχίζω χωρίς να δηλώσω αποτυχία.

Τότε έχω $y_k - y_{*k} \neq 0$, οπότε ορίζεται ο αντίστροφος του, $\bmod \operatorname{ord}(G_1)$, διότι $\operatorname{ord}(G_1) \in \mathbb{P}$.

Συνεπώς:

$$a = (y_k - y_{*k})^{-1} \sum_{i=1, i \neq k}^{n} (y_{*i} - y_i)\lambda_i$$

Άρα τελικά ο $\mathcal{B}$ κερδίζει το πρόβλημα διακριτού λογάριθμου στην G_1 με αμελητέα ως προς κ πιθανότητα, γεγονός που είναι άτοπο διότι το πρόβλημα διακριτού λογάριθμο έχει υποτεθεί υπολογιστικά δύσκολο στην G_1.

Συνεπώς κάθε PPT αντίπαλος $\mathcal{A}$ δίνει $\vec{y}\vec{\lambda} = \vec{y}_*\vec{\lambda}$ με αμελητέα ως προς κ πιθανότητα. $\square$

7.5 Βελτίωση της επίδοσης – *δική μου συνεισφορά*

Τα παρακάτω είναι προτεινόμενη βελτίωση από τον γράφοντα την παρούσα εργασία.

Όπως είχε ήδη αναφερθεί παραπάνω, όταν επιλέχθηκε ο πίνακας N αυτός δεν είχε κάποια μητρωική έκφραση γεγονός που εξάρχης αξιολογήθηκε περίεργο διότι όλο το υπόλοιπο σχήμα είναι στήμενο και ορισμένο με μητρωικές εκφράσεις.

Παρατηρούμε πως η μόνη χρήση του πίνακα N είναι στον υπολογισμό του πιστοποιητικού Π, που αποδείχθηκε στην απόδειξη ορθότητα πως τελικά είναι ίσο με $\Pi = g^{((\delta\mathrm{M}+R)\vec{x})\vec{\lambda}}$.

Προβληματιζόμαστε ισχυρά καθώς το Π έχει μητρωική έκφραση, οπόε ίσως θα μπορούσαμε να δώσουμε και τον N μητρωικά ώστε να έχουμε μια μητρωική έκφραση για το Π ως συνάρτηση του N όπως δηλαδή υπολογίζει η $Compute$.

Ωστόσο, με βάση το Λήμμα που αποδείχθηκε παραπάνω πως $(A\vec{x})\vec{y} = (A^T\vec{y})\vec{x}$ παρατηρούμε ότι:

$$\Pi = g^{((\delta\mathrm{M}+R)\vec{x})\vec{\lambda}} = g^{((\delta\mathrm{M}+R)^T\vec{\lambda})\vec{x}} = g^{\sum_{i=1}^{n}((\delta\mathrm{M}+R)^T\vec{\lambda})_i x_i} = \prod_{i=1}^{n}\left(g^{(\delta\mathrm{M}+R)^T\vec{\lambda}}\right)_i^{x_i}$$

Προβληματιζόμαστε λοιπόν μήπως θα μπορούσαμε να δίνουμε το $\vec{N} = g^{(\delta M+R)^T\vec{\lambda}}$ στο κλειδί επαλήθευσης αντί του πίνακα N ώστε να υπολογίζουμε το Π με βάση την τελευταία έκφραση.

Παρατηρούμε ότι τότε θα ήταν:

$$N_j = \left(g^{(\delta M+R)^T\vec{\lambda}}\right)_j = g^{\sum_{i=1}^{n}(\delta M_{ij}+R_{ij})\lambda_i} = \prod_{i=1}^{n} g^{\delta M_{ij}+R_{ij})\lambda_i} = \prod_{i=1}^{n} N_{ij}$$

Συνεπώς δεν χρειάζεται όλος ο πίνακας N παρά μόνο τα γινόμενα όλων των στοιχείων κάθε στήλης του N, δηλαδή το παραπάνω διάνυσμα $\vec{N} = \prod_{i=1}^{n} N_{ij} = g^{(\delta M+R)^T\vec{\lambda}}$.

Μάλιστα, η αλλαγή του πίνακα N στο διάνυσμα $\vec{N}$ δεν βλάπτει την αξιοπιστία (soundness), καθώς εξαρχής τον υπολογισμό των γινομένων των στοιχείων κάθε στήλης θα μπορούσε να τον είχε κάνει κάθε αντίπαλος αν δίναμε τον πίνακα N και έχουμε ήδη δείξει την αξιοπιστία όταν δημοσιοποιείται ο πίνακας N.

Επίσης, οι παραπάνω σχέσεις δείχνουν πως η αλλαγή δεν βλάπτει ούτε την ορθότητα (correctness) αφού τελικά υπολογίζεται το ίδιο Π, που είναι και η μοναδική χρήση του πίνακα $\vec{N}$ στο σχήμα δημόσιου υπολογισμού.

Συνοψίζοντας, μπορούμε ισοδύναμα αντί να υπολογίσουμε και να δημοσιοποιήσουμε τον πίνακα N, να δημοσιοποιήσουμε (στον αλγόριθμο $Setup$) το διάνυσμα $\vec{N} = g^{(\delta M+R)^T\vec{\lambda}}$ και έπειτα στον αλγόριθμο $Compute$ να υπολογίσουμε το πιστοποιητικό Π ως $\Pi = \prod_{i=1}^{n}\left(g^{(\delta\mathrm{M}+R)^T\vec{\lambda}}\right)_i^{x_i}$.

Έτσι πέραν του αναγκαίου M,τα πρόσθετα δεδομένα για την επαληθευσιμότητα που χρειάζεται να μεταδοθούν από τον αλγόριθμο $Setup$ μειώνονται σε $\Theta(m)$ από $\Theta(mn)$ και επίσης στον αλγόριθμο $Compute$, πέραν του αναγκαιού υπολογισμού $M\vec{x}$ η χρονική πολυπλοκότητα των πρόσθετων υπολογισμών για την επαληθευσιμότητα μειώνεται σε $\Theta(m)$ από $\Theta(mn)$!

Αναφορές

[1] Kaoutar Elkhiyaoui et al. "Efficient Techniques for Publicly Verifiable Delegation of Computation". In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ASIA CCS '16. ACM, May 2016. DOI: 10.1145/2897845.2897910. URL: http://dx.doi.org/10.1145/2897845.2897910.

[2] Alfred Menezes. *An introduction to pairing-based cryptography*. 2009. DOI: 10.1090/conm/477/09303. URL: http://dx.doi.org/10.1090/conm/477/09303.

[3] Antoine Joux. "A One Round Protocol for Tripartite Diffie–Hellman". In: *Journal of Cryptology* 17.4 (June 2004), pp. 263–276. ISSN: 1432-1378. DOI: 10.1007/s00145-004-0312-y. URL: http://dx.doi.org/10.1007/s00145-004-0312-y.

[4] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing". In: *Advances in Cryptology — ASIACRYPT 2001*. Springer Berlin Heidelberg, 2001, pp. 514–532. ISBN: 9783540456827. DOI: 10.1007/3-540-45682-1_30. URL: http://dx.doi.org/10.1007/3-540-45682-1_30.

[5] Dan Boneh and Matthew Franklin. "Identity-Based Encryption from the Weil Pairing". In: *SIAM Journal on Computing* 32.3 (Jan. 2003), pp. 586–615. ISSN: 1095-7111. DOI: 10.1137/s0097539701398521. URL: http://dx.doi.org/10.1137/S0097539701398521.

[6] Paulo S. L. M. Barreto and Michael Naehrig. "Pairing-Friendly Elliptic Curves of Prime Order". In: *Selected Areas in Cryptography*. Springer Berlin Heidelberg, 2006, pp. 319–331. ISBN: 9783540331094. DOI: 10.1007/11693383_22. URL: http://dx.doi.org/10.1007/11693383_22.