UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

Tesi di Laurea Magistrale in Informatica

Progettazione di circuiti per l'implementazione di Quantum Physical Unclonable Functions (QPUF)

Relatore:
Prof.
Christiancarmine Esposito

Candidato:
Luca Boffa
Mat. 0522501521

Tutor:
Dott.
Franco Cirillo

ANNO ACCADEMICO 2024/2025

Indice

Introduzione

Negli ultimi decenni, la sicurezza informatica ha assunto un ruolo centrale in tutti i settori della società digitale, imponendo la necessità di soluzioni sempre più solide ed evolute per garantire l'autenticazione sicura di dispositivi e utenti. L'autenticazione costituisce infatti uno dei pilastri fondamentali della sicurezza dei sistemi informatici: in assenza di una corretta identificazione delle entità coinvolte in una comunicazione, qualsiasi sistema risulta esposto a rischi significativi, quali frodi, accessi non autorizzati e attacchi su larga scala.

Tradizionalmente, i meccanismi di autenticazione si sono basati su informazioni segrete memorizzate, come password o chiavi crittografiche, rientrando nel paradigma di sicurezza noto come "ciò che si sa". Tuttavia, l'efficacia di questi metodi è stata messa progressivamente in discussione dalla crescente sofisticazione delle tecniche di attacco, incluse la clonazione di dispositivi, il reverse engineering, il furto di credenziali e le vulnerabilità software. In risposta a tali criticità, il settore si trova oggi in una fase di transizione significativa, in cui l'attenzione si sta spostando verso modelli di autenticazione basati su caratteristiche fisiche o biometriche – il cosiddetto approccio "ciò che si è" – che possano garantire maggiore resistenza alle compromissioni.

In questo contesto emergono le *Physical Unclonable Functions* (PUF), primitive hardware che sfruttano le inevitabili variazioni fisiche introdotte durante il processo di fabbricazione dei circuiti integrati per generare risposte uniche e difficilmente replicabili. Le PUF si distinguono per la loro capacità di produrre risposte deterministiche ma non prevedibili a fronte di specifiche sfide di input, offrendo un'alternativa promettente rispetto ai tradizionali segreti digitali.

Nonostante il potenziale, anche le PUF classiche presentano limiti rilevanti. Tra le principali vulnerabilità si annoverano gli attacchi di modellazione, in cui un avversario riesce a predire le risposte tramite tecniche di apprendimento automatico, e gli attacchi fisici volti a ricostruire la struttura interna del dispositivo.

Per superare queste limitazioni, è stata proposta una nuova generazione di primitive, le *Quantum Physical Unclonable Functions* (QPUF), che si basano su principi fondamentali della meccanica quantistica, quali l'indeterminazione di Heisenberg e la non clonabilità degli stati quantistici. Le QPUF introducono una nuova dimensione di sicurezza, poiché impiegano stati quantistici, rendendo la clonazione del dispositivo teoricamente impossibile anche per un avversario dotato di capacità computazionali illimitate.

Nonostante le promesse offerte da questa tecnologia, l'implementazione delle QPUF pone numerose sfide pratiche. Tra le più rilevanti si annoverano la complessità dei sistemi ottici e quantistici necessari, la sensibilità ai disturbi ambientali e le difficoltà legate all'integrazione in dispositivi embedded o a basso costo. Inoltre, rimangono aperte questioni legate alla scalabilità, all'affidabilità in condizioni operative reali e alla potenziale emergenza di nuove tipologie di attacchi quantistici, ancora poco esplorate.

L'obiettivo di questo lavoro è analizzare criticamente le potenzialità delle QPUF come futura soluzione per l'autenticazione sicura in ambito hardware. A partire da un'analisi approfondita dello stato dell'arte, si è cercato di ampliare lo studio progettando nuovi circuiti quantistici, con l'intento di esplorare architetture alternative e contribuire allo sviluppo di soluzioni più robuste ed efficienti. Questo ha permesso di individuare le principali direzioni di ricerca nell'ambito delle QPUF.

Capitolo 1

Background teorico

1.1 PUF: principi e classificazioni

Le Physical Unclonable Functions (PUF) sono primitive di sicurezza hardware che sfruttano le variazioni fisiche inevitabili e non replicabili presenti nei processi di produzione dei circuiti integrati(IC). Queste variazioni rendono ogni istanza hardware unica e imprevedibile, anche se realizzata con lo stesso progetto e nella stessa fabbrica.

Principio di base

Una PUF implementa una funzione che, data una certa challenge (sfida), restituisce una determinata response (risposta). Tale risposta viene poi confrontata con un valore atteso o una referenza memorizzata per verificare l'autenticità del dispositivo o completare un'operazione crittografica. A differenza di una funzione matematica pura, una PUF può generare risposte diverse alla stessa sfida, a causa della sensibilità ai fattori ambientali e al rumore fisico. Per questo motivo, le PUF sono più correttamente descritte come funzioni probabilistiche, dove parte dell'input è rappresentata da una variabile casuale incontrollabile.

Una coppia challenge-response è detta CRP (Challenge-Response Pair), e l'insieme di relazioni tra sfide e risposte che una PUF implementa è definito comportamento CRP. Per alcune PUF, questo comportamento è deterministico, per altre è necessario specificare condizioni ambientali per garantire la ripetibilità.

Le PUF operano generalmente in due fasi:

- **Fase di enrollment (registrazione)**: si raccolgono numerose CRP da una PUF specifica, che vengono memorizzate in un database sicuro.
- **Fase di authentication (verifica)**: si applica una challenge casuale alla PUF e si confronta la risposta con quella attesa nel database.

Per valutare l'affidabilità di una PUF, si definiscono due metriche:

- **Intra-distance**: la distanza tra due risposte della stessa PUF con la stessa challenge; rappresenta l'instabilità (deve essere molto bassa).
- **Inter-distance**: la distanza tra le risposte di due diverse PUF alla stessa challenge; misura l'unicità (idealmente pari al 50% di Hamming distance).

Le PUF sono:

- Facili da valutare: è semplice ottenere la risposta a una sfida.
- Difficili da prevedere: è praticamente impossibile, anche conoscendo altre coppie challenge-response, prevedere la risposta a una nuova sfida.
- Non clonabili: la natura fisica della PUF non può essere replicata semplicemente, nemmeno con sofisticate tecniche di reverse engineering.

1.2 Fondamenti di computazione quantistica

La computazione quantistica rappresenta un paradigma computazionale radicalmente diverso rispetto all'informatica classica. Essa sfrutta le leggi della meccanica quantistica per elaborare le informazioni, offrendo potenzialità computazionali che superano i limiti dei computer tradizionali in specifici ambiti, come la fattorizzazione, la simulazione di sistemi fisici e la crittografia. In questa sezione si introducono i concetti fondamentali della computazione quantistica necessari alla comprensione e all'implementazione delle Quantum Physical Unclonable Functions (QPUF).

1.2.1 Qubit e sovrapposizione

Qubit, contrazione di quantum bit, è il termine coniato da Benjamin Schumacher per indicare il bit quantistico ovvero l'unità di informazione quantistica.

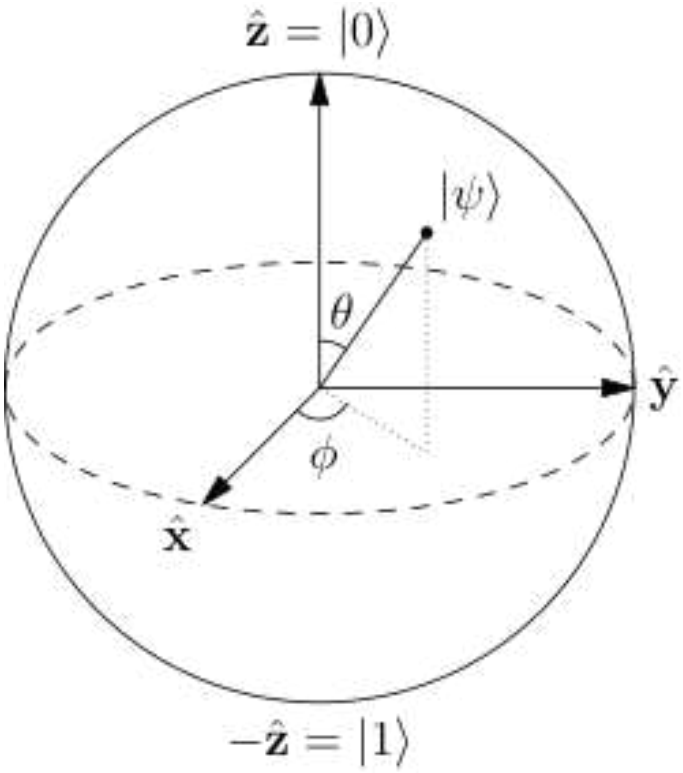

Fig. 1.1: Rappresentazione geometrica astratta di un qubit

1.2.2 Entanglement

Una caratteristica fondamentale che distingue i qubit dai bit classici è che più qubit possono mostrare **entanglement quantistico**; infatti, il qubit stesso può essere considerato un'espressione di entanglement quantistico. In questo contesto, l'entanglement quantistico è una proprietà locale o non locale di due o più qubit che consente a un insieme di qubit di manifestare correlazioni più elevate rispetto a quanto possibile nei sistemi classici.

Il sistema più semplice che manifesta entanglement quantistico è costituito da due qubit.
Consideriamo, ad esempio, due qubit entangled nello stato di Bell noto come

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

In questo stato, definito una sovrapposizione equiprobabile, la probabilità di misurare lo stato prodotto $|00\rangle$ o $|11\rangle$ è uguale e pari a $|1/\sqrt{2}|^2 = 1/2$. In altre parole, non è possibile determinare se il primo qubit abbia valore "0" o "1", e lo stesso vale per il secondo qubit.

Immaginiamo che questi due qubit entangled siano separati, uno affidato ad Alice e l'altro a Bob. Quando Alice misura il suo qubit, ottiene con uguale probabilità il valore $|0\rangle$ oppure $|1\rangle$, quindi ora conosce il valore del suo qubit. Grazie all'entanglement tra i qubit, Bob ottiene esattamente la stessa misura di Alice. Ad esempio, se Alice misura $|0\rangle$, allora anche Bob deve ottenere $|0\rangle$, poiché lo stato $|00\rangle$ è l'unico in cui il qubit di Alice è $|0\rangle$.

In sintesi, per questi due qubit entangled, qualunque risultato Alice ottenga alla misura, Bob ottiene un risultato perfettamente correlato, indipendentemente dalla distanza che li separa. Questa correlazione perfetta si verifica in qualsiasi base di misura, anche se né Alice né Bob possono predire con certezza il valore del proprio qubit prima della misura — un fenomeno sorprendente che non può essere spiegato dalla fisica classica.

1.3 Porte Quantistiche

Una **porta quantistica** (o *porta quantica*) è una porta logica basata sulla fisica quantistica, progettata per operare su circuiti composti da un numero limitato di qubit. Rappresentano l'analogo quantistico delle porte logiche digitali utilizzate nei computer classici.

Le porte logiche quantistiche sono descritte da **matrici unitarie**. Il numero di qubit in ingresso e in uscita da una porta quantistica deve essere identico. Una porta che agisce su n qubit è rappresentata da una matrice unitaria di dimensione $2^n \times 2^n$.

Gli stati quantistici su cui agiscono queste porte sono vettori in uno spazio vettoriale complesso di dimensione 2^n. I vettori di base rappresentano i possibili esiti della misura dello stato, mentre uno stato quantistico è una combinazione lineare (o sovrapposizione) di tali esiti.

Le porte quantistiche più comuni operano su spazi di uno o due qubit, analogamente alle porte logiche classiche, che operano su uno o due bit.

Gli stati quantistici sono generalmente rappresentati utilizzando la **notazione bra-ket** introdotta da Dirac. La rappresentazione vettoriale di un singolo qubit è la seguente:

$$|a\rangle = v_0|0\rangle + v_1|1\rangle \quad \rightarrow \quad \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$$

dove v_0 e v_1 sono le ampiezze di probabilità complesse associate allo stato del qubit. Tali valori determinano la probabilità di ottenere come risultato della misura il valore 0 oppure 1.

Gli stati base, detti anche stati computazionali, sono:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

1.3.1 Gate di Hadamard

La **porta di Hadamard** agisce su un singolo qubit e ha l'effetto di creare una sovrapposizione tra gli stati di base. In particolare, applicando la porta Hadamard

agli stati $|0\rangle$ e $|1\rangle$ si ottiene:

$$|0\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|1\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Questo significa che una misura dello stato risultante avrà la stessa probabilità di restituire 0 o 1, poiché la porta Hadamard trasforma uno stato base in una combinazione lineare equiprobabile degli stati computazionali. In termini fisici, rappresenta una rotazione di π intorno all'asse $\frac{\hat{x}+\hat{z}}{\sqrt{2}}$ sulla sfera di Bloch.

Fig. 1.2: Rappresentazione di una porta di Hadamard

La porta Hadamard è rappresentata dalla seguente matrice unitaria:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Dal momento che vale la relazione $HH^\dagger = I$, dove I è la matrice identità e $H^\dagger$ è l'aggiunta hermitiana di H, si conferma che la porta di Hadamard è effettivamente una **matrice unitaria**, come tutte le porte logiche quantistiche.

1.3.2 Gate di Pauli X

La **porta X di Pauli** agisce su un singolo qubit ed è considerata l'equivalente quantistico della porta `NOT` nei computer classici, relativamente alla base computazionale $\{|0\rangle, |1\rangle\}$. In questo contesto, tale base distingue la direzione dell'asse Z nella sfera di Bloch, in cui una misura con autovalore $+1$ corrisponde allo stato classico 1, mentre un autovalore -1 corrisponde a 0.

La porta X effettua uno scambio tra i due stati base:

$$|0\rangle \longrightarrow |1\rangle$$
$$|1\rangle \longrightarrow |0\rangle$$

Per questo motivo, è anche chiamata **bit-flip**. Geometricamente, la sua azione può essere interpretata come una rotazione di π radianti (ossia 180°) attorno all'asse X della sfera di Bloch.

La porta X è rappresentata dalla seguente matrice:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Questa è la prima delle tre matrici di Pauli, tutte matrici unitarie ed hermitiane fondamentali nel formalismo della meccanica quantistica.

Fig. 1.3: Rappresentazione di una porta X

1.3.3 Gate di Pauli Y

La **porta Y di Pauli** agisce su un singolo qubit ed è associata a una rotazione di π radianti attorno all'asse Y della sfera di Bloch.

Questa trasformazione ha il seguente effetto sugli stati base:

$$|0\rangle \longrightarrow i\,|1\rangle$$
$$|1\rangle \longrightarrow -i\,|0\rangle$$

L'effetto complessivo è quello di uno **scambio di bit** (come la porta X), ma accompagnato da una variazione di fase complessa. Per questo motivo, la porta Y è talvolta indicata anche come **bit-phase-flip**.

La porta Y è rappresentata dalla seguente matrice:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Questa è la seconda delle tre matrici di Pauli. Anche la matrice Y è **unitaria** e **hermitiana**, proprietà fondamentali delle trasformazioni quantistiche reversibili.

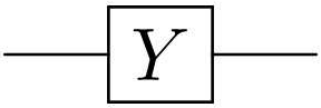

Fig. 1.4: Rappresentazione di una porta Y

1.3.4 Gate di Pauli Z

La **porta Z di Pauli** agisce su un singolo qubit ed equivale a una rotazione di π radianti attorno all'asse Z della sfera di Bloch. È un caso particolare della porta di *phase shift*, con angolo $\phi = \pi$.

La trasformazione opera nel seguente modo:

$$|0\rangle \longrightarrow |0\rangle$$
$$|1\rangle \longrightarrow -|1\rangle$$

Lo stato $|0\rangle$ resta invariato, mentre lo stato $|1\rangle$ acquisisce un fattore di fase negativa. Per questo motivo, la porta Z è talvolta denominata **phase-flip**.

La rappresentazione matriciale della porta Z è la seguente:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Fig. 1.5: Rappresentazione di una porta Z

Questa è la terza e ultima delle **matrici di Pauli**. Come le altre, anche la matrice Z è **unitaria** e **hermitiana**, caratteristiche essenziali per garantire la reversibilità e la conservazione della norma nello spazio degli stati quantistici.

1.3.5 Differenza tra gate di Pauli e rotazioni quantistiche

Le rotazioni quantistiche generalizzano i gate di Pauli consentendo una rotazione continua di un angolo arbitrario θ attorno a un asse specifico. La loro forma è espressa come $R_x(\theta) = e^{-i\theta X/2}$, $R_y(\theta) = e^{-i\theta Y/2}$ e $R_z(\theta) = e^{-i\theta Z/2}$. Questi operatori sono fondamentali in contesti dove è richiesto un controllo preciso dello stato quantistico, come negli algoritmi parametrizzati o nei circuiti variazionali. In sintesi, i gate di Pauli sono un caso particolare delle rotazioni, ottenibile ponendo $\theta = \pi$.

1.3.6 Gate CNOT

La *Controlled-NOT* (CNOT) è una porta logica quantistica a due qubit in cui il secondo qubit (detto *target*) viene invertito se e solo se il primo qubit (detto *controllo*) è nello stato $|1\rangle$. Essa rappresenta l'equivalente quantistico della porta XOR classica e può generare entanglement, risultando fondamentale nella costruzione di circuiti quantistici universali.

L'azione della porta può essere rappresentata dalla seguente matrice:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Quando applicata a uno stato generico $a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle$, la CNOT restituisce $a\,|00\rangle + b\,|01\rangle + c\,|11\rangle + d\,|10\rangle$.

La porta è sia unitaria che hermitiana ed è spesso utilizzata, in combinazione con rotazioni su singoli qubit, per implementare algoritmi quantistici fondamentali, come l'algoritmo di Deutsch–Jozsa o la creazione degli stati di Bell.

1.3.7 Gate CZ

La porta Controlled-Z (CZ) è una porta a due qubit. Essa agisce su una coppia di qubit, in cui uno funge da *controllo* e l'altro da *bersaglio* (*target*). Il comportamento fondamentale della CZ consiste nell'applicare una *inversione di fase*

(una rotazione di fase di π) al qubit bersaglio, ma solo quando il qubit di controllo si trova nello stato $|1\rangle$.

Formalmente la porta CZ può essere rappresentata dalla seguente matrice unitaria U_{CZ} nel sistema a due qubit (base computazionale $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$):

$$U_{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

L'effetto della porta CZ sugli stati base del sistema a due qubit può essere riassunto come segue:

$$\begin{aligned} \mathrm{CZ}\,|00\rangle &= |00\rangle \\ \mathrm{CZ}\,|01\rangle &= |01\rangle \\ \mathrm{CZ}\,|10\rangle &= |10\rangle \\ \mathrm{CZ}\,|11\rangle &= -\,|11\rangle \end{aligned}$$

Come si può notare, solo lo stato $|11\rangle$ subisce un'inversione di fase, confermando che l'operazione agisce condizionatamente allo stato del qubit di controllo.

In altre parole, la porta CZ lascia invariate le componenti associate a $|00\rangle$, $|01\rangle$ e $|10\rangle$, ma applica un segno negativo alla componente $|11\rangle$.

1.4 Quantum PUF (QPUF)

Le Quantum Physical Unclonable Functions (QPUF) sono una primitiva che genera un'impronta digitale univoca per un computer quantistico, sfruttando la casualità intrinseca dell'hardware quantistico, guidata dal principio della meccanica quantistica[1], [2] offrendo un livello di sicurezza superiore rispetto alle PUF classiche [3]. I concetti fondamentali di sovrapposizione ed entanglement, propri della meccanica quantistica, sono alla base dell'imprevedibilità e della non replicabilità delle risposte generate dalle QPUF [4].

Gli attuali computer quantistici sono soggetti a vari tipi di errori quantistici [5], derivanti da imperfezioni costruttive, errori di controllo e interazioni ambientali. Questi stessi errori vengono sfruttati dalle QPUF per generare risposte uniche per ciascun dispositivo. Le QPUF offrono vantaggi significativi grazie al teorema di non-clonazione della meccanica quantistica [6], che garantisce l'impossibilità di duplicare l'informazione quantistica. Sfruttando questo principio, le QPUF si distinguono per l'elevata entropia e la resistenza agli attacchi di clonazione, risultando particolarmente adatte per applicazioni crittografiche in cui l'integrità e l'autenticità dei dati sono fondamentali.

1.5 Stato dell'arte

In letteratura sono già stati condotti diversi studi rilevanti, che verranno analizzati qui di seguito.

Nel lavoro [7] si parla di un'architettura di sicurezza per ambienti industriali basata su Quantum Physical Unclonable Functions (QPUF), utilizzando hardware quantistico reale e accessibile via cloud. L'obiettivo è rafforzare la sicurezza dei dispositivi IIoT (Industrial Internet of Things), del firmware e delle comunicazioni in rete tramite un sistema di autenticazione e controllo accessi basato su QPUF. Nei Cyber-Physical Systems industriali (I-CPS), le QPUF permettono di generare impronte digitali quantistiche uniche e non clonabili, garantendo autenticità e integrità nelle comunicazioni tra dispositivi, sensori e sistemi SCADA. Il modello cloud-based adottato assicura scalabilità e facile integrazione con infrastrutture esistenti, in linea con l'attuale disponibilità di hardware quantistico tramite cloud. L'architettura proposta segue il principio della Security by Design (SbD), esteso al contesto quantistico (QSbD), incorporando la sicurezza già in fase di progettazione.

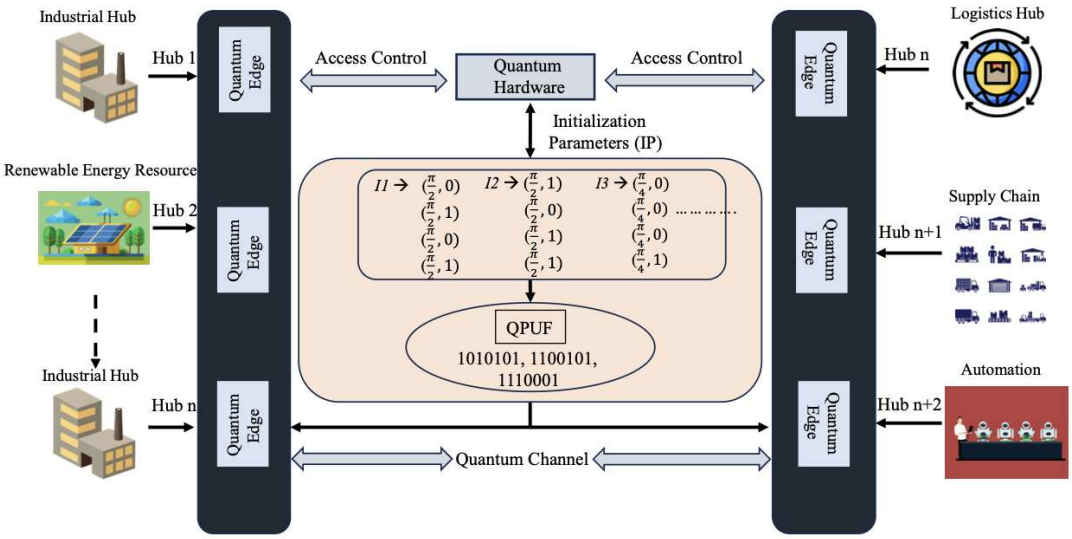

Fig. 1.6: Esempio di architettura IIoT basata sul principio QSbD

In [1] è stato osservato che il fenomeno del crosstalk nei qubit transmon superconduttivi può influenzare lo stato quantistico di un qubit, introducendo effetti indesiderati sulla coerenza del sistema. Partendo da questa osservazione,

gli autori propongono un metodo di generazione della firma per una QPUF (Quantum Physical Unclonable Function) basato sull'esperimento di Ramsey, il quale consente di determinare la frequenza risonante assoluta di un qubit. Il crosstalk genera rumore quantistico che altera le frequenze risonanti dei qubit adiacenti, introducendo così una caratteristica fisicamente non clonabile che può essere sfruttata a fini di autenticazione hardware nel contesto quantistico.

Successivamente, in [8] viene proposta una nuova architettura di QPUF che sfrutta la decoerenza quantistica e il fenomeno dell'entanglement per generare una sequenza binaria unica e casuale composta da zeri e uno. L'architettura è progettata utilizzando porte quantistiche standard come Ry, CNOT, Pauli-X e Hadamard, dimostrando sperimentalmente l'affidabilità nella generazione delle risposte della QPUF. Questo approccio apre la prospettiva per l'integrazione di QPUF in sistemi cyber-physical sicuri, orientati al paradigma della security-by-design.

Nel lavoro [9] si utilizza un circuito quantistico progettato appositamente da utilizzare come QPUF, integrato in un protocollo basato su meccanismi di challenge-response, al fine di garantire un'autenticazione sicura ed efficiente dei dispositivi quantistici. L'approccio è stato valutato sperimentalmente sull'hardware quantistico fornito da IBM, analizzando parametri chiave quali l'instabilità, la casualità e l'unicità delle QPUF.

Nel capitolo successivo ci focalizziamo sul precedente lavoro per analizzare alcuni spunti di miglioramento.

Capitolo 2

Analisi dello studio "QPUF: Quantum Physical Unclonable Functions"

In questo paper, le Quantum Physical Unclonable Functions (QPUF) rappresentano una soluzione promettente per l'autenticazione tramite dispositivi quantistici, sfruttando gli errori e le imperfezioni intrinseche presenti nei dispositivi quantistici di tipo NISQ (Noisy Intermediate-Scale Quantum). Tali errori quantistici, solitamente considerati un limite, vengono invece utilizzati come risorsa per generare risposte uniche e non prevedibili, strettamente legate alle caratteristiche fisiche specifiche dell'hardware quantistico.

Attraverso il paradigma Challenge-Response, viene sottoposta una sfida al dispositivo quantistico, il quale genera una risposta determinata dalle proprie imperfezioni e errori quantistici unici. Questa risposta può essere successivamente utilizzata per autenticare il dispositivo [10] e verificarne l'integrità all'interno di un sistema sicuro.

2.1 Architettura del circuito QPUF proposto

Inizialmente, ogni qubit viene inizializzato nello stato zero. Successivamente, ciascun qubit subisce rotazioni sequenziali su tutti i piani (X, Y, Z), tramite appositi gate di rotazione con angoli parametrizzati nell'intervallo $[0,2\pi]$. Poi, ogni

qubit viene messo in entanglement con il suo predecessore o successore utilizzando gate Controlled-Z (CZ). Questi due passaggi vengono ripetuti una seconda volta per garantire l'entanglement tra tutti i qubit. Infine, tutti i qubit vengono misurati.

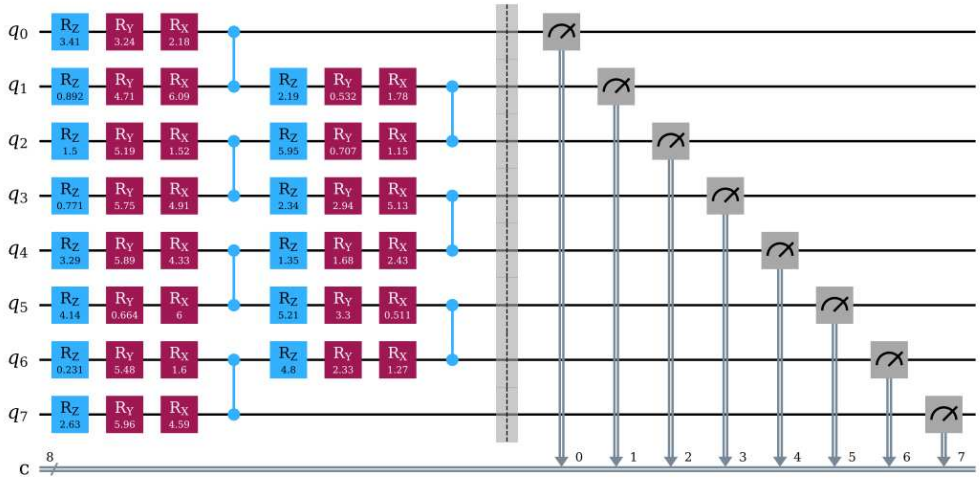

Fig. 2.1: Circuito proposto in questo studio

All'interno di questo schema QPUF, le sfide si presentano come vettori che includono tutti i parametri (gli angoli) per ogni gate in un'istanza del circuito. Ad esempio, in un circuito da 8 qubit come illustrato nella 2.1, una sfida è definita da angoli per ciascuno dei 3 tipi di gate, con valori compresi nell'intervallo $[0,2\pi]$.

La risposta, invece, consiste nei risultati delle misurazioni di tutti i qubit dopo aver eseguito il circuito un numero predeterminato di volte. In termini pratici, la risposta è un dizionario in cui ogni possibile combinazione dei valori dei qubit è associata alla percentuale di occorrenza (valore tra 0 e 1), con la somma totale dei valori pari a 1.

Idealmente, i qubit dovrebbero mostrare una distribuzione di probabilità tra gli stati 0 e 1, dipendente dai parametri dei gate, ovvero dagli angoli di rotazione. Tuttavia, la distribuzione reale tende a essere sbilanciata verso uno dei due stati a causa degli errori presenti, diventando così una firma unica del dispositivo.

2.2 Risultati Sperimentali

Sono stati condotti esperimenti su hardware quantistico IBM (`ibm_brisbane`, `ibm_kyiv`, `ibm_sherbrooke`), utilizzando circuiti QPUF a 8 qubit con 20.000 shot per esecuzione.

Nel test di Instabilità con 5 challenge e 5 runs, i valori si attestano intorno al 5%, con un massimo del 7%, mostrando una riduzione all'aumentare degli shot.
La Randomness varia tra il 35% e il 75%, con una media del 55%.
La Uniqueness, calcolata tra i tre dispositivi, si colloca tra il 15% e il 25%, senza mai scendere sotto il 12%.

2.3 Limiti e spunti per l'espansione

Una delle principali limitazioni di questo studio risiede nell'utilizzo di un singolo circuito quantistico progettato ad hoc; di conseguenza, non sono disponibili risultati sperimentali provenienti da altri circuiti per un confronto approfondito.

Un'ulteriore limitazione riguarda l'utilizzo di un numero ridotto di qubit, pari a otto. Questa restrizione impedisce di valutare come il sistema QPUF si comporti all'aumentare del numero di qubit, limitando così l'analisi della scalabilità e dell'efficacia del metodo in scenari più complessi.

L'idea, pertanto, è quella di estendere lo studio, progettando e testando circuiti quantistici differenti al fine di analizzarne il comportamento in relazione ai tipi di porte logiche utilizzate, al numero di qubit coinvolti e al numero di challenge applicate.

Capitolo 3

Implementazione di nuovi circuiti QPUF

Nel corso dello sviluppo della presente tesi, sono stati progettati diversi circuiti quantistici con l'obiettivo di implementare una QPUF. Ogni circuito ideato è stato sottoposto a una serie di test variando il numero di qubit, il numero di challenge e il numero di shots per analizzare l'impatto delle dimensioni del circuito sulle prestazioni e sulla robustezza delle metriche adottate.

3.1 Circuiti proposti

Per ciascuna configurazione circuitale, sono state provate implementazioni con 8, 10 e 12 (solo per alcuni 16 qubit). Queste configurazioni sono state selezionate per analizzare come la dimensione del circuito influisca sulla stabilità delle risposte e sulla qualità delle metriche ottenute. Inoltre, per ogni combinazione di qubit è stato variato sia il numero di *challenge* (tra 10 e 20), sia il numero di *shots* (5.000, 10.000 e 20.000), in modo da valutare l'impatto di questi parametri sulla qualità delle risposte generate.

L'idea è stata di creare varie configurazioni diverse in modo da testarle sulle macchine IBM (sia simulate che reali) per valutarne le differenze attraverso le metriche proposte.

3.1.1 Nomenclatura dei circuiti

Per facilitare la classificazione e il richiamo dei circuiti quantistici sviluppati nel presente lavoro, si è adottato un sistema di nomenclatura ispirato ai nomi dei pianeti del sistema solare in lingua inglese. Questa scelta nasce dall'idea di avere una denominazione chiara e semplice da ricordare.

I nomi scelti sono:

- Earth
- Mars
- Venus
- Mercury
- Saturn
- Jupiter
- Uranus
- Uranus Titania

3.2 Descrizione delle configurazioni testate

Per ogni circuito diverso ci sarà una breve descrizione, l'immagine del circuito e delle considerazioni progettuali.

Per tutti i circuiti proposti in questo lavoro verranno mostrate esclusivamente le immagini relative alla variante a 8 qubit. Le versioni con 10, 12 e 16 qubit non sono incluse poiché seguono esattamente la stessa struttura: si limitano ad aumentare il numero di qubit e l'estensione della catena di entanglement, senza introdurre modifiche alla logica generale del circuito.

3.2.1 Earth

Il circuito **Earth** deriva da una modifica diretta del circuito di partenza. In particolare, è stato ridefinito l'ordine di applicazione dei gate di Pauli, eseguendo prima il gate RX, successivamente il gate RY e infine il gate RZ.

Al termine di queste operazioni, è stato inserito un ulteriore layer finale di rotazioni RX prima della fase di misurazione, con l'obiettivo di introdurre una maggiore variabilità nelle risposte.

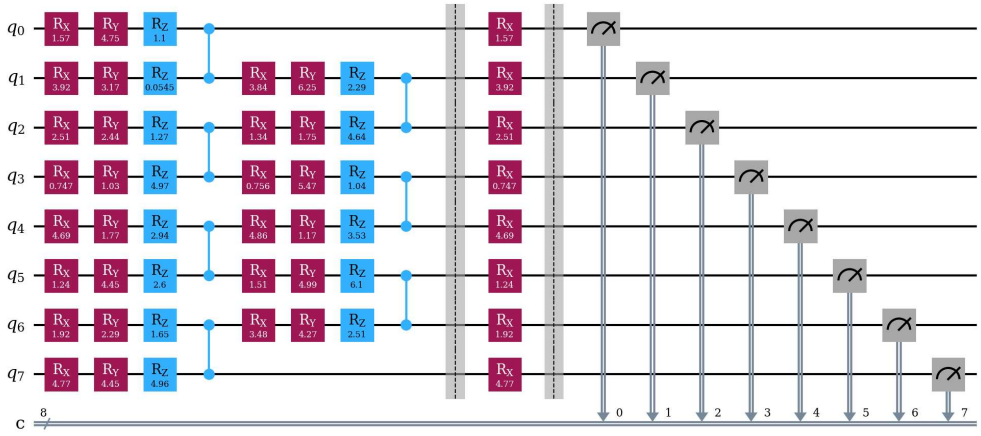

Fig. 3.1: Circuito Earth a 8 qubit

Considerazioni progettuali:

- L'alterazione dell'ordine dei gate di rotazione introduce differenti traiettorie nello spazio degli stati quantistici, modificando l'effetto delle rotazioni concatenate.
- L'aggiunta del layer finale di RX prima della misura consente una diversificazione ulteriore delle configurazioni finali, incrementando la sensibilità del circuito rispetto agli input.
- Il circuito mantiene una struttura relativamente semplice ma efficace per introdurre variabilità senza un eccessivo overhead di entanglement.

3.2.2 Mars

Il circuito **Mars** è un'evoluzione diretta del circuito *Earth*, che. introduce una sequenza più articolata di rotazioni sui qubit e un layer finale aggiuntivo prima della misurazione.

Struttura del circuito:

1. **Rotazioni iniziali:** Ogni qubit è sottoposto a una combinazione di RX, RY e RZ con challenge distinte, per generare sovrapposizioni ricche e diversificate.
2. **Entanglement regolare:** Viene applicata una topologia a coppie con gate CZ tra qubit adiacenti, per aumentare la correlazione controllata.
3. **Layer finale:** Prima della misurazione, si aggiungono rotazioni RZ seguite da porte Hadamard (H) alternate.
4. **Misurazione:** I qubit vengono infine misurati per ottenere l'output binario.

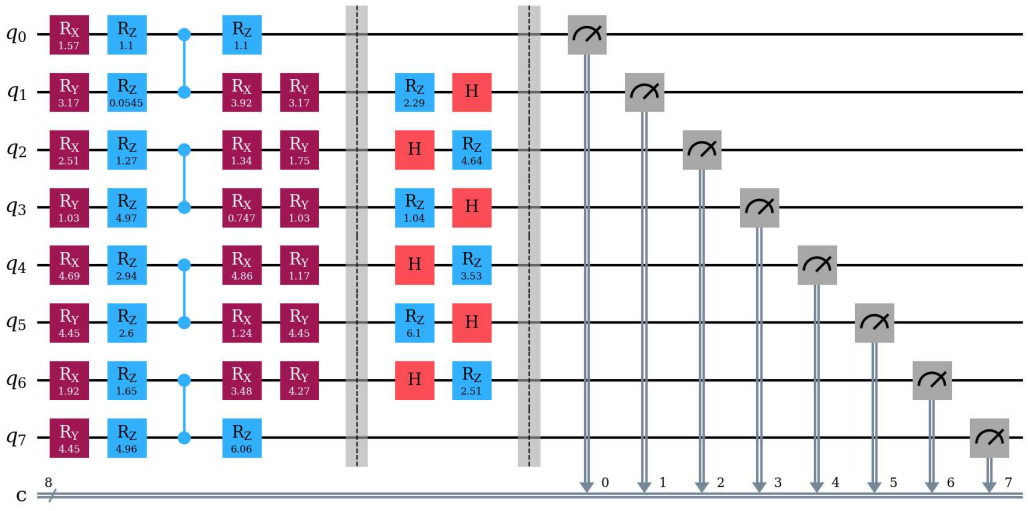

Fig. 3.2: Circuito Mars a 8 qubit

Considerazioni progettuali:

- Le rotazioni multiple per qubit aumentano la complessità.
- Il pattern regolare di CZ preserva la stabilità, garantendo al contempo una buona entanglement.
- Il layer RZ–H finale introduce ulteriore casualità, utile per la randomess.

3.2.3 Venus

Il circuito **Venus** si ispira al circuito della famiglia ansatz `EfficientSU2`, comunemente utilizzato in algoritmi variazionali (es. VQE) e machine learning quantistico. Questa architettura è apprezzata per la sua efficienza hardware e la buona espressività a fronte di una bassa profondità circuitale.
Il circuito può essere suddiviso in quattro fasi principali:

- **Rotazioni iniziali:** Ogni qubit è sottoposto a rotazioni parametriche RY e RZ, che introducono una diversificazione dello stato iniziale.
- **Entanglement 2-local:** Viene poi applicato uno strato di entanglement tramite gate CNOT disposti in maniera 2-local, accoppiando ciascun qubit con il qubit adiacente.
- **Rotazioni pre-misura:** Un secondo strato di rotazioni RY e RZ consente di raffinare ulteriormente la distribuzione degli output.
- **Misurazione:** I qubit vengono infine misurati per ottenere l'output binario.

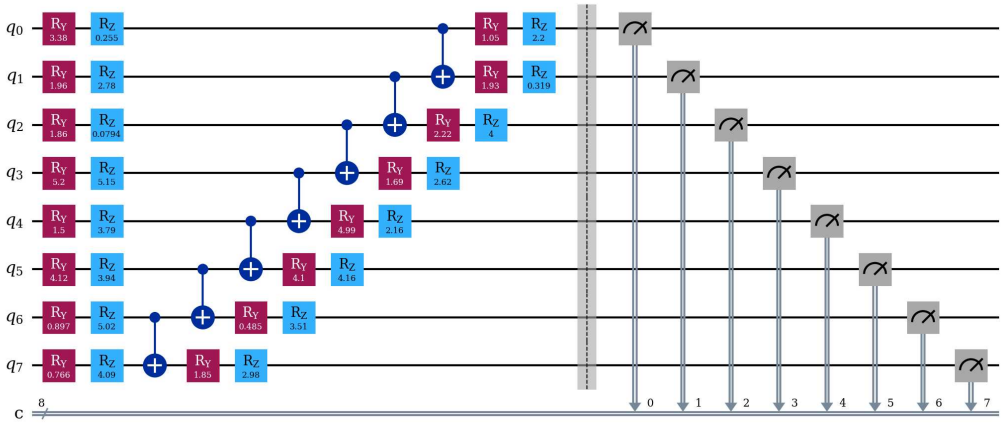

Fig. 3.3: Circuito Venus a 8 qubit

Considerazioni progettuali:

- Un architettura a bassa profondità aiuta a ridurre il rumore su hardware reale.
- La forma come `EfficientSU2` consente buon compromesso tra complessità e stabilità.

3.2.4 Mercury

Il circuito **Mercury** è stato progettato come tentativo di miglioramento rispetto al circuito Venus, con l'obiettivo di potenziare la diversità statistica degli output quantistici e amplificare la sensibilità alle variazioni della challenge di ingresso. Mentre Venus si basava sull'architettura `EfficientSU2`, Mercury introduce nuove strategie per aumentare la distanza tra gli output tra dispositivi diversi cercando di sfruttare al massimo il fenomeno dell'entanglement.

Il circuito può essere suddiviso in quattro fasi principali:

- **Preparazione dello stato:** Ogni qubit viene inizialmente posto in sovrapposizione tramite una porta Hadamard (H), seguita da rotazioni RY e RZ, con angoli specifici derivati dalla challenge.
- **Entanglement centrale:** Si applica una rete composta da:
  - Porte CNOT in catena inversa, come in `EfficientSU2`;
  - Porte CZ tra le porte CNOT, che introducono un livello di entanglement misto.
- **Rotazioni adattive e ulteriori entanglement:** Dopo una barriera di sincronizzazione, viene applicato un secondo strato di rotazioni RY e RZ, seguito da CZ tra i qubit dispari e i loro vicini sinistri.
- **Misurazione:** I qubit vengono infine misurati per ottenere l'output binario.

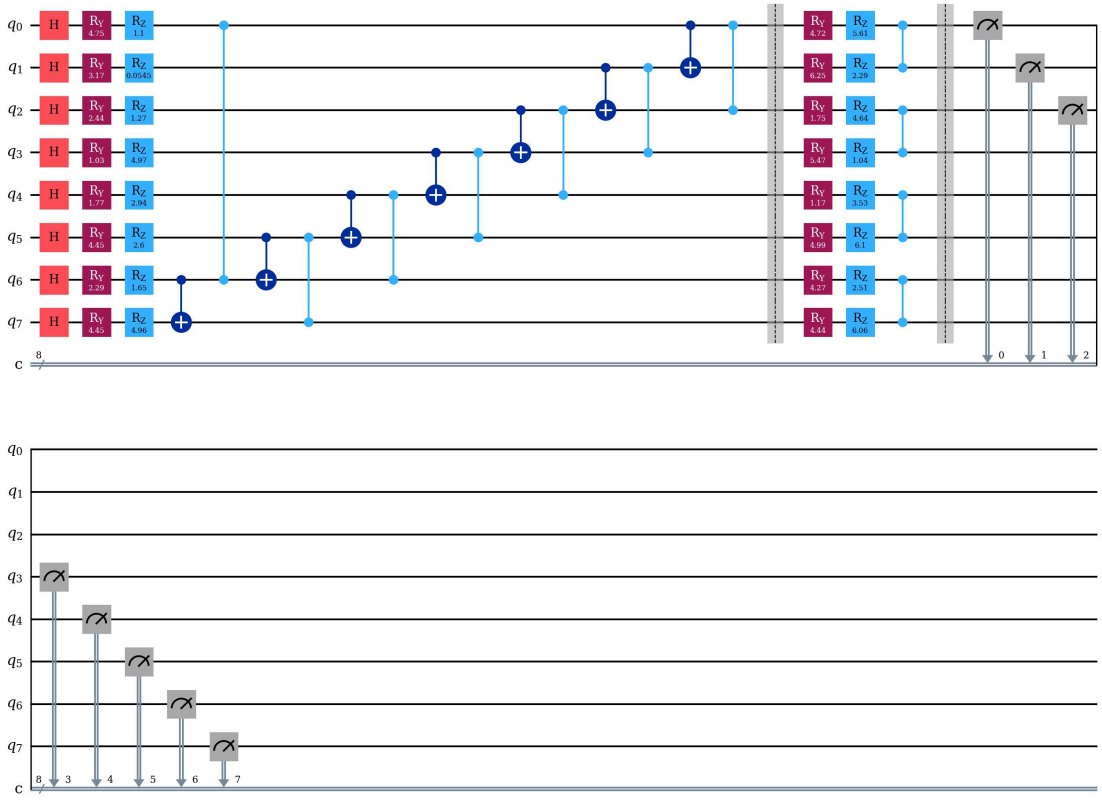

Fig. 3.4: Circuito Mercury a 8 qubit

Considerazioni progettuali:

- Creare un nuovo circuito basato sulla forma di `EfficientSU2`.
- Utilizzo di un entanglement misto con l'utilizzo di porte CZ e CNOT.

3.2.5 Saturn

Il circuito **Saturn** rappresenta un'evoluzione architetturale rispetto ai predecessori Venus e Mercury, con l'obiettivo di massimizzare l'unicità andando a creare uno "scivolo" di entanglement. Il design si ispira ancora ai principi di `EfficientSU2`, ma applicati in modo inverso e utilizzando solo il gate CZ.

Il circuito può essere suddiviso in quattro fasi principali:

- **Preparazione dello stato:** Ogni qubit viene inizialmente posto in sovrapposizione tramite porta Hadamard (H). Successivamente, in funzione della parità dell'indice i del qubit, vengono applicate rotazioni parametrizzate:

$$\text{Se } i \text{ è pari:} \quad H \to RX(\theta_{X_i}) \to RZ(\theta_{Z_i})$$
$$\text{Se } i \text{ è dispari:} \quad H \to RY(\theta_{Y_i}) \to RZ(\theta_{Z_i})$$

  Dove θ_{X_i}, θ_{Y_i} e θ_{Z_i} sono angoli specifici determinati dalla challenge.
  Subito dopo i qubit pari vengono messi in entanglement con il successivo qubit mediante un gate CZ.

- **Fase intermedia di perturbazione:** Tutti i qubit ricevono rotazioni condizionate:
  - Agli estremi ($j = 0$ e $j = n-1$), viene applicata una sola $RZ(\theta_{Z_i})$;
  - Negli altri casi, vengono introdotte $RX(\theta_{X_i})$ e $RY(\theta_{Y_i})$ in cascata, seguite da un gate CZ con il successivo qubit, espandendo lo scivolo di entanglement.

- **Fase di ulteriore confusione:** Si applica una strategia ibrida alternata per i qubit interni ($1 \leq j \leq n-2$):
  - Per j dispari, si applica $RZ(\theta_{Z_i}) \to$ Hadamard $\to CZ$ con il vicino a sinistra;
  - Per j pari, l'ordine è invertito: Hadamard $\to RZ(\theta_{Z_i}) \to CZ$ con il vicino a destra.

- **Misurazione:** I qubit vengono infine misurati per ottenere l'output binario.

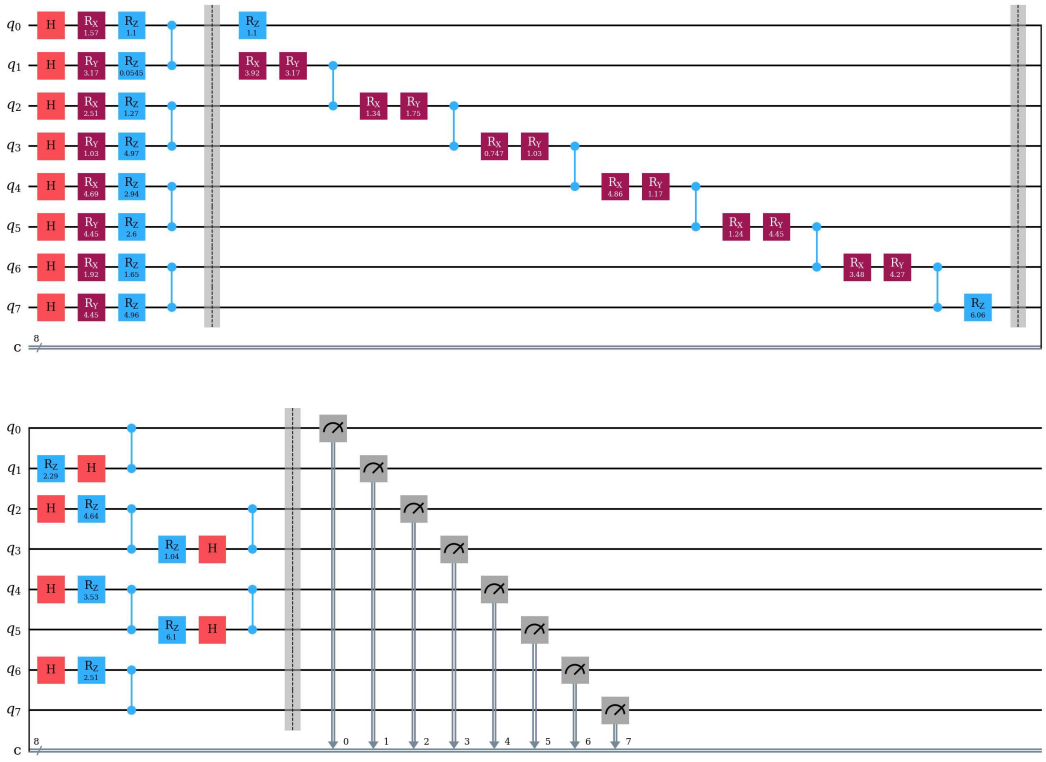

Fig. 3.5: Circuito Saturn a 8 qubit

Considerazioni progettuali:

- L'idea è stata quella di cercare correlazioni nei risultati basandoci sulla qualità delle risposte in relazione all'utlizzo di questa forma di entanglement.
- L'alternanza di rotazioni e porte Hadamard, combinata con lo scivolo di entanglement CZ, rende il circuito estremamente sensibile a variazioni minime della challenge.

3.2.6 Jupiter

Il circuito **Jupiter** prende il nome dal più grande pianeta del sistema solare, a sottolineare la sua architettura imponente.

La progettazione del circuito si articola in quattro fasi chiave:

- **Preparazione dello stato:** Ogni qubit viene inizialmente posto in sovrapposizione tramite porta Hadamard (H). Successivamente, in funzione della parità dell'indice i del qubit, vengono applicate rotazioni parametrizzate:

$$\begin{aligned} \text{Se } i \text{ è pari:} \quad & H \to RZ(\theta_{Z_i}) \to RX(\theta_{X_i}) \\ \text{Se } i \text{ è dispari:} \quad & H \to RZ(\theta_{Z_i}) \to RY(\theta_{Y_i}) \end{aligned}$$

  Dove θ_{X_i}, θ_{Y_i} e θ_{Z_i} sono angoli specifici determinati dalla challenge.

- **Entanglement strutturato:** Su qubit a distanza 2 viene applicato $CNOT(j, j+1)$ seguito da una rotazione $RZ(\theta_{Z_j})$ sul target, introducendo correlazioni non lineari.

- **Propagazione e randomizzazione:** Viene alternata $RY(\theta_{Y_j})$ per qubit pari e $RX(\theta_{X_j}) + CZ(j, (j+1) \bmod n)$ per qubit dispari.

- **Misurazione selettiva:** La misura finale adotta un pattern periodico per rompere la regolarità:

$$j \bmod 3 = \begin{cases} 0 & \Rightarrow H(j) \text{ e poi misurazione} \\ 1 & \Rightarrow RZ(\theta_{Z_j}) \text{ e poi misurazione} \\ 2 & \Rightarrow \text{solo misurazione} \end{cases}$$

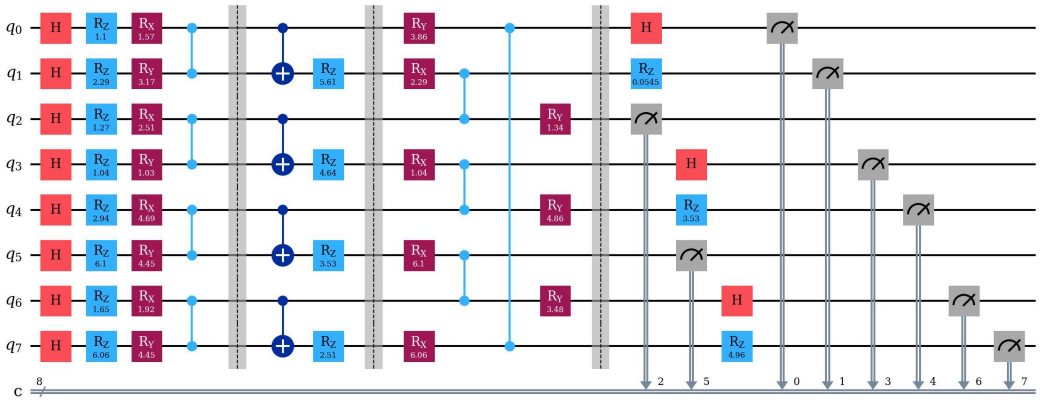

Fig. 3.6: Circuito Jupiter a 8 qubit

Considerazioni progettuali:

- La struttura stratificata consente di combinare entanglement locale e globale, aumentando la complessità dello stato finale.
- L'utilizzo di entrambi i gate CZ e CNOT su livelli diversi tende ad aumentare la randomicità.
- La misurazione selettiva rompe la simmetria introducendo una componente casuale controllata.

3.2.7 Uranus

Il circuito **Uranus** si caratterizza per una struttura ibrida che combina entanglement controllato e rotazioni parametrizzate basate sulla challenge ricevuta. La progettazione di questo circuito mira ad ottenere un entanglement misto alterando i gate CZ e CNOT su uno "scivolo".

Il circuito è suddiviso in quattro fasi operative principali:

- **Preparazione dello stato:** Ogni qubit viene inizialmente posto in sovrapposizione tramite porta Hadamard (H). Successivamente, in funzione della parità dell'indice i del qubit, vengono applicate rotazioni parametrizzate:

$$\begin{aligned} \text{Se } i \text{ è pari:} \quad & H \to RX(\theta_{X_i}) \to RZ(\theta_{Z_i}) \\ \text{Se } i \text{ è dispari:} \quad & H \to RY(\theta_{Y_i}) \to RZ(\theta_{Z_i}) \end{aligned}$$

  Dove θ_{X_i}, θ_{Y_i} e θ_{Z_i} sono angoli specifici determinati dalla challenge.

- **Entanglement controllato:** L'entanglement tra i qubit è ottenuto tramite porte controllate, selezionate in base alla parità dei qubit coinvolti:
  - Se il primo qubit è **pari**: applicazione di una porta CZ;
  - Se il primo qubit è **dispari**: applicazione di una porta CNOT.

- **Rotazioni e entanglement addizionale:** Per aumentare ulteriormente l'espressività del circuito e diversificare lo spazio degli stati, viene applicata una seconda fase di rotazioni su ciascun qubit.
  Infine, per i qubit con indice pari $j < n-1$, viene applicata una porta CNOT addizionale per incrementare la propagazione delle correlazioni quantistiche:

$$\text{Se } j \equiv 0 \pmod 2 \quad \text{applica } CNOT(q_j, q_{j+1})$$

- **Misurazione:** I qubit vengono infine misurati per ottenere l'output binario.

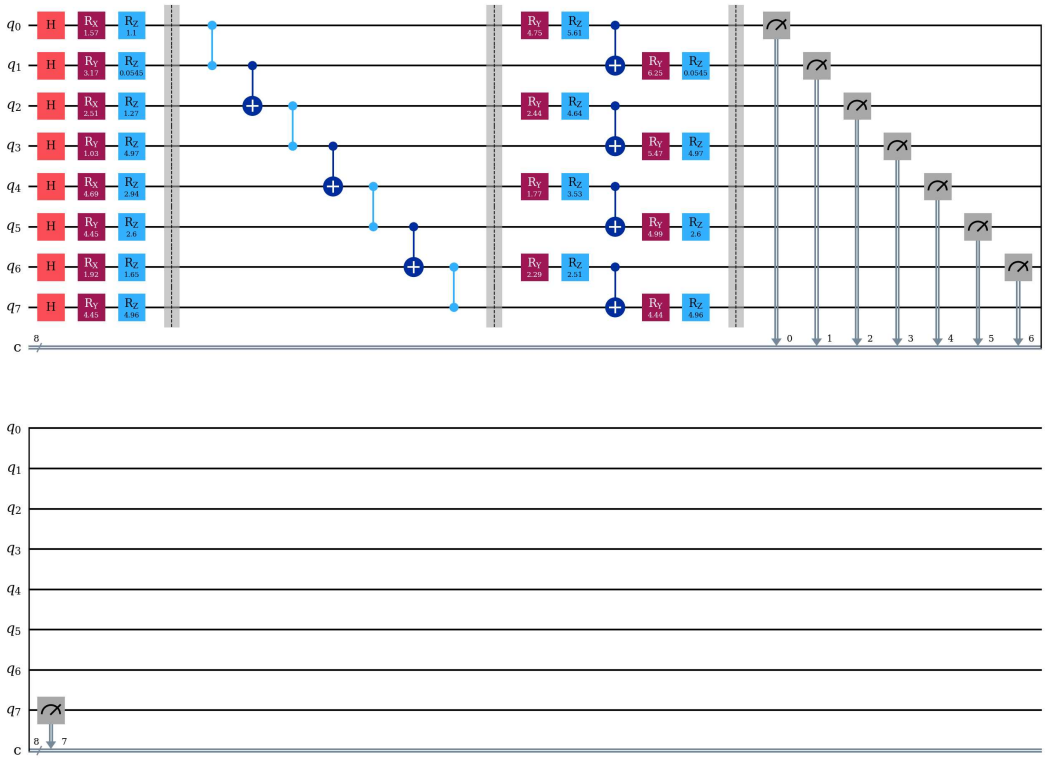

Fig. 3.7: Circuito Uranus a 8 qubit

Considerazioni proggettuali:

- La catena mista di entanglement di CNOT e CZ derivante del circuito ansatz `EfficientSU2` ci garantisce una buona distribuzione delle correlazioni tra i qubit.
- L'ulteriore livello finale aggiunge un semplice entanglement e rotazioni che potrebbero ulteriormente migliorare le risposte delle challenge.

3.2.8 Uranus Titania

Il circuito **Uranus Titania** prende il nome da *Titania*, uno dei principali satelliti di Urano. Questa denominazione non è casuale: il circuito infatti mira a migliorare l'architettura Uranus andandola a fondere la struttura di entanglement tipica del circuito Venus, basato su `EfficientSU2`. Questo circuito è il primo ad essere derivato in modo ibrido.

Il circuito è strutturato in quattro fasi principali:

- **Sovrapposizione diversificata** Ogni qubit è inizialmente posto in sovrapposizione mediante porta Hadamard (H). Successivamente:

  $$\text{Se } j \text{ è pari: } RX(\theta_{X_i}) \rightarrow RZ(\theta_{Z_i})$$
  $$\text{Se } j \text{ è dispari: } RY(\theta_{Y_i}) \rightarrow RZ(\theta_{Z_i})$$

  Dove θ_{X_i}, θ_{Y_i} e θ_{Z_i} sono angoli specifici determinati dalla challenge.

- **Entanglement controllato** L'entanglement tra i qubit è ottenuto tramite porte controllate, selezionate in base alla parità dei qubit coinvolti:
  - Se il primo qubit è **pari**: applicazione di una porta CZ;
  - Se il primo qubit è **dispari**: applicazione di una porta CNOT.

- **Catena in stile `EfficientSU2`** Dopo la prima fase di entanglement, si applica un ulteriore catena di entanglement ovvero quella che caratterizza il circuito ansatz `EfficientSU2`.

- **Misurazione** I qubit vengono infine misurati per ottenere l'output binario.

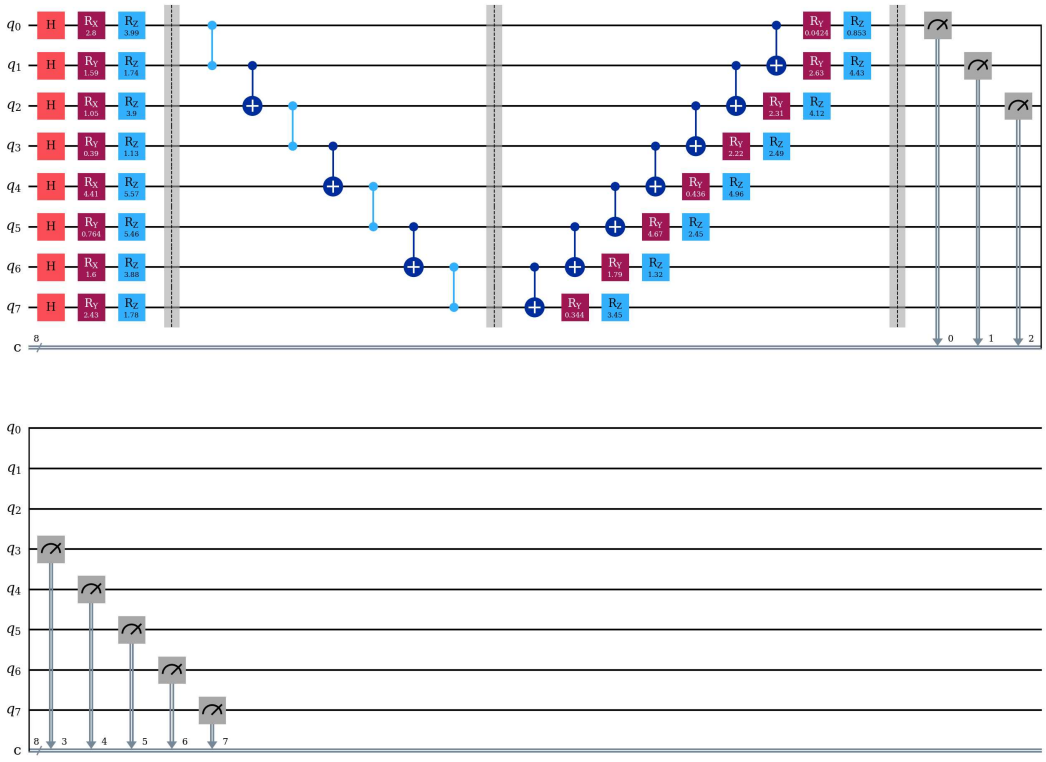

Fig. 3.8: Circuito Uranus Titania a 8 qubit

Considerazioni proggettuali:

- Primo circuito costruito da un approccio totalmente ibrido, andando ad unire 2 parti di configurazioni precedenti.
- Possiamo capire dai risultati quale delle catene è più impattante andando a confrontare i risultati direttamente con i circuiti Venus e Uranus.

3.3 Ambiente di test: IBMQ

IBM Quantum è un'iniziativa di IBM volta allo sviluppo e alla promozione dell'informatica quantistica. Attraverso una combinazione di hardware avanzato, software dedicato e accesso cloud, IBM offre una piattaforma per esplorare, progettare e testare algoritmi quantistici. L'obiettivo principale è quello di accelerare la scoperta scientifica e l'innovazione industriale, rendendo l'informatica quantistica una risorsa accessibile anche al di fuori dei laboratori di ricerca.

Il sito `https://www.ibm.com/quantum` fornisce una panoramica dettagliata sulle tecnologie quantistiche sviluppate da IBM, tra cui i servizi offerti tramite IBM Quantum Experience. Gli utenti possono accedere a computer quantistici reali tramite IBM Cloud, sviluppare algoritmi quantistici utilizzando Qiskit – l'open-source SDK di IBM – e consultare materiale formativo di alto livello pensato sia per studenti che per ricercatori.

3.3.1 Uso dei simulatori IBM

Oltre all'impiego dei simulatori reali messi a disposizione da Qiskit, durante il lavoro sperimentale sono stati utilizzati anche i simulatori di rumore forniti per riprodurre in maniera più fedele il comportamento dei dispositivi fisici reali di IBM. In particolare, sono stati caricati i profili di rumore delle macchine `ibmq_kyiv`, `ibmq_sherbrooke` e `ibmq_brisbane`, al fine di verificare la robustezza e l'affidabilità dei circuiti implementati in condizioni prossime a quelle hardware.

Tuttavia, i risultati ottenuti su tali simulatori real-device-oriented hanno mostrato valori molto più bassi per la metrica di *uniqueness*, indicando una significativa perdita di capacità discriminante nei confronti delle challenge caricate. A fronte dei risultati non soddisfacenti riscontrati sui simulatori rumorosi, si è scelto di abbandonare tale approccio intermedio, orientando l'analisi direttamente verso l'esecuzione sperimentale sui dispositivi quantistici reali messi a disposizione da IBM Quantum, così da valutare le prestazioni dei circuiti in condizioni operative effettive.

3.3.2 Uso di macchine reali IBM

Durante i test sono state impiegate le macchine quantistiche disponibili nel piano gratuito:

- `ibmq_kyiv`
- `ibmq_sherbrooke`
- `ibmq_brisbane`

Tutte le macchine utilizzate sono basate su IBM Eagle R3.
IBM Quantum Eagle R3 è una QPU quantistica da 127 qubit, parte della famiglia Eagle sviluppata da IBM. Rispetto alle generazioni precedenti, integra tecnologie di packaging più scalabili, come il passaggio dei segnali attraverso più strati del chip, permettendo una maggiore densità di I/O senza compromettere le prestazioni.
La revisione R3, rilasciata a dicembre 2022, mantiene l'architettura e le funzionalità della R1, ma offre proprietà di coerenza migliorate, rendendola più adatta a operazioni quantistiche complesse e di lunga durata. Supporta porte quantistiche native come ECR, RZ, SX, X, oltre a operazioni come measure, reset.

3.3.3 Qiskit

Qiskit SDK è un *framework open-source* sviluppato da IBM per programmare e simulare computer quantistici. Fornisce strumenti avanzati per costruire, manipolare e ottimizzare **circuiti quantistici** sia statici che dinamici.

Le sue funzionalità principali includono:

- `qiskit.circuit`: per creare e gestire circuiti, registri, porte e strutture di controllo.
- `qiskit.circuit.library`: una vasta libreria di porte e circuiti pronti all'uso.
- `qiskit.quantum_info`: strumenti per analizzare stati quantistici e operatori, utili per verificare la qualità delle computazioni.

- `qiskit.transpiler`: per adattare e ottimizzare i circuiti in base alla topologia dell'hardware quantistico.
- `qiskit.primitives`: definizioni di base per eseguire stime e campionamenti (*Sampler*, *Estimator*) su circuiti quantistici.

A complemento c'è **Qiskit Runtime**, un servizio cloud che consente di eseguire programmi quantistici su hardware IBM Quantum in modo più efficiente, con tecniche di *mitigazione degli errori* (come decoupling dinamico, ZNE e readout mitigation) e diverse modalità di esecuzione: **Job**, **Session** e **Batch**.

Una delle caratteristiche distintive di Qiskit è la sua capacità di connettersi direttamente al cloud IBM Quantum, consentendo l'esecuzione remota di algoritmi su hardware quantistico fisico distribuito globalmente.

3.4 Strategie di testing e tecnologie utilizzate

L'accesso alle macchine IBM è avvenuto attraverso l'uso delle *API key* fornite da IBM agli utenti registrati nel piano gratuito. Tuttavia, questo piano prevedeva una limitazione significativa: un tetto massimo di 10 minuti di tempo computazionale al mese per ogni account. Tale restrizione si è rivelata un ostacolo rilevante, considerato l'elevato numero di test necessari per valutare le varie configurazioni di circuiti e parametri.

Per ovviare a tale limitazione e garantire una copertura sperimentale sufficientemente ampia, è stato necessario ricorrere alla creazione e gestione di più account gratuiti. Questo ha permesso di estendere il tempo computazionale complessivo disponibile e portare a termine con successo la raccolta delle risposte su hardware quantistico reale, assicurando una maggiore affidabilità nella valutazione delle metriche ottenute.

Le esecuzioni su macchine reali hanno rappresentato un passaggio fondamentale per confrontare i risultati ottenuti in simulazione con quelli derivanti dall'effettiva implementazione fisica dei circuiti, consentendo di rilevare effetti pratici quali errori

di gate, rumore e instabilità tipiche dei dispositivi quantistici attuali.

Per l'avvio dei test è stato sviluppato un apposito *Jupyter Notebook* in grado di gestire l'intero processo in modo modulare. Questo notebook consente all'utente di selezionare, per ogni sessione di test, la macchina quantistica IBM desiderata, il tipo di circuito da testare, e tutti i parametri associati, tra cui: il numero di qubit, il numero di *shots*, il numero di *challenge* e il numero di *runs*. Tale modularità ha facilitato l'esecuzione ordinata e ripetibile degli esperimenti, permettendo un elevato grado di personalizzazione per ciascuna configurazione.

La raccolta automatizzata delle risposte è stata anch'essa integrata nello stesso notebook, che si occupa di recuperare i risultati dalle esecuzioni, elaborare i dati grezzi e salvarli in file in formato `.csv`. Questo approccio ha garantito una gestione strutturata e replicabile dell'intero processo sperimentale, semplificando le successive analisi.

Il codice sorgente sviluppato per questa tesi è disponibile nella mia repository GitHub personale, al seguente indirizzo:
`https://github.com/secLuk3/QuantumPUF-MasterDegreeThesis`

Capitolo 4

Analisi dei risultati

La metodologia di analisi si è basata sulla valutazione sistematica delle prestazioni dei circuiti quantistici, concentrandoci su parametri specifici che abbiamo ritenuto fondamentali per caratterizzare il comportamento dei PUF:

- **Dimensione del circuito:** Abbiamo testato configurazioni a 8, 10, 12 e 16 qubit per valutare come la complessità del circuito influisca sulle metriche di prestazione. Questa gamma di dimensioni ci ha permesso di esaminare il comportamento del sistema dalla scala ridotta fino a configurazioni più complesse.

- **Numero di shots:** Per ogni configurazione, abbiamo eseguito 5.000, 10.000 e 20.000 misurazioni (shots), al fine di analizzare la convergenza statistica delle risposte e l'impatto della ripetizione sulla stabilità dei risultati. Questo approccio ci ha consentito di determinare il numero ottimale di shots necessari per ottenere misurazioni affidabili.

- **Set di challenge:** Inizialmente abbiamo utilizzato un set composto da 10 challenge diverse. Dopo una fase di testing iniziale, analizzando attentamente i risultati ottenuti con diversi numeri di shots, abbiamo osservato che un numero maggiore di shots produceva metriche più stabili e affidabili. Questa osservazione ci ha portato a decidere di ampliare il nostro studio, aumentando il numero di challenge da 10 a 20 per le configurazioni più promettenti. Questo ci ha permesso di ottenere una caratterizzazione più robusta e statisticamente significativa delle prestazioni dei circuiti.

Per ogni challenge, abbiamo eseguito 5 runs indipendenti per garantire la validità statistica dei nostri risultati e valutare la consistenza delle metriche ottenute. In ogni esperimento, abbiamo misurato con precisione i valori di instabilità, casualità e unicità, registrando sistematicamente i risultati in grafici comparative che ci hanno permesso di identificare pattern significativi e tendenze nelle prestazioni dei circuiti al variare dei parametri operativi.

4.1 Nomenclatura delle Configurazioni

Per facilitare l'identificazione e il riferimento alle diverse configurazioni sperimentali, abbiamo adottato la seguente nomenclatura nei nomi dei file e nei riferimenti all'interno di questo studio:

Simbolo	**Descrizione**
nq	Numero di qubit (8, 10, 12, 16)
nc	Numero di challenge (10, 20)
s	Numero di shots (5.000, 10.000, 20.000)
nr	Numero di runs (5)

Table 4.1: Legenda per la nomenclatura delle configurazioni sperimentali

Ad esempio, una configurazione indicata come "nomeCircuito_8nq_20nc_s20000_5nr" si riferisce a un esperimento condotto con 8 qubit, 20 challenge, 20.000 shots e 5 runs.

4.2 Metriche utilizzate

Al fine di valutare la qualità di un circuito QPUF proposto, sono state analizzate le seguenti metriche:

- **Instabilità** (Instability)
- **Casualità** (Randomness)
- **Unicità** (Uniqueness)

4.2.1 Instabilità

L'instabilità si riferisce alla variabilità e all'imprevedibilità delle risposte generate da un QPUF su uno stesso dispositivo, dovute a imperfezioni di fabbricazione, errori di controllo, gradienti termici e interazioni ambientali. Tali instabilità possono compromettere l'affidabilità e la sicurezza del sistema.

Per quantificare l'instabilità, si utilizza la *Normalized Absolute Probabilistic Distance* (NAPD) tra ogni coppia di risposte. Sia $r^k_{n,q}$ la risposta associata alla q-esima combinazione di risultati dei qubit, generata dal dispositivo k in risposta alla sfida n, su un totale di $Q = 2^n$ combinazioni. La NAPD tra due risposte è definita come:

$$\text{NAPD}(r^k_n, r^h_m) = \frac{1}{2} \sum_{q=1}^{Q} \left| r^k_{n,q} - r^h_{m,q} \right| \tag{4.1}$$

Il fattore di normalizzazione $\frac{1}{2}$ garantisce che il valore della distanza sia compreso nell'intervallo $[0, 1]$.

Sia ora $r^k_{n,i}$ la risposta ottenuta alla i-esima esecuzione, tra le D totali, per la sfida n sul dispositivo k. L'instabilità è stimata come:

$$\text{Instabilità}(k) = \frac{1}{N} \sum_{n=1}^{N} \frac{2}{D(D-1)} \sum_{i=1}^{D} \sum_{j=i+1}^{D} \text{NAPD}(r^k_{n,i}, r^k_{n,j}) \tag{4.2}$$

Idealmente, l'instabilità dovrebbe essere minima, tendendo a zero, indicando che le risposte sono consistenti nel tempo.

4.2.2 Casualità (Randomness)

La casualità misura il grado di distinzione tra risposte a sfide differenti. Sia r^k_n la risposta del dispositivo k alla sfida n, allora la metrica di casualità è data da:

$$\text{Casualità}(k) = \frac{2}{N(N-1)} \sum_{n=1}^{N} \sum_{m=n+1}^{N} \text{NAPD}(r^k_n, r^k_m) \tag{4.3}$$

Valori elevati di casualità (prossimi a 1) indicano che le risposte a sfide diverse sono altamente distinguibili, come desiderato in un buon sistema QPUF.

4.2.3 Unicità

L'unicità assicura che dispositivi diversi producano risposte differenti alla stessa sfida. Sia r_n^k la risposta del dispositivo k alla sfida n, la metrica di unicità è definita come:

$$\text{Unicità} = \frac{2}{K(K-1)} \sum_{k=1}^{K} \sum_{h=k+1}^{K} \frac{1}{N} \sum_{n=1}^{N} \text{NAPD}(r_n^k, r_n^h) \tag{4.4}$$

Valori ideali per l'unicità sono prossimi a 1, denotando che i dispositivi sono tra loro facilmente distinguibili sulla base delle risposte generate.

In breve si misura quanto le risposte siano distintive tra diversi dispositivi che eseguono istanze della stessa PUF. Un'alta unicità garantisce che due dispositivi, anche se identici nella progettazione, generino firme significativamente diverse.

4.3 Risultati

In questa sezione vengono presentati e analizzati i risultati ottenuti per ciascun circuito proposto. Per ogni configurazione, sono riportati:

- una tabella che riassume i valori medi delle principali metriche (*instability*, *randomness*, *uniqueness*) su più run di test, fornendo così un quadro generale del comportamento del circuito;
- il grafico relativo al circuito che ha ottenuto le migliori prestazioni secondo i valori riportati in tabella, per una visualizzazione immediata dell'andamento delle metriche;
- una sezione di considerazioni, in cui vengono discussi i risultati osservati e vengono evidenziati eventuali trend o comportamenti rilevanti.

Questa struttura consente di avere una panoramica chiara e sintetica dei risultati per ciascuna configurazione. Nelle fasi successive dell'analisi verranno inoltre presi in considerazione ulteriori fattori per una valutazione più approfondita delle prestazioni dei circuiti.

4.3.1 Earth

Test con 8 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.11	0.60	0.24
10.000	0.07	0.59	0.24
20.000	0.05	0.62	0.28

Table 4.2: Configurazione a 8 qubit, 10 challenge e 5 runs

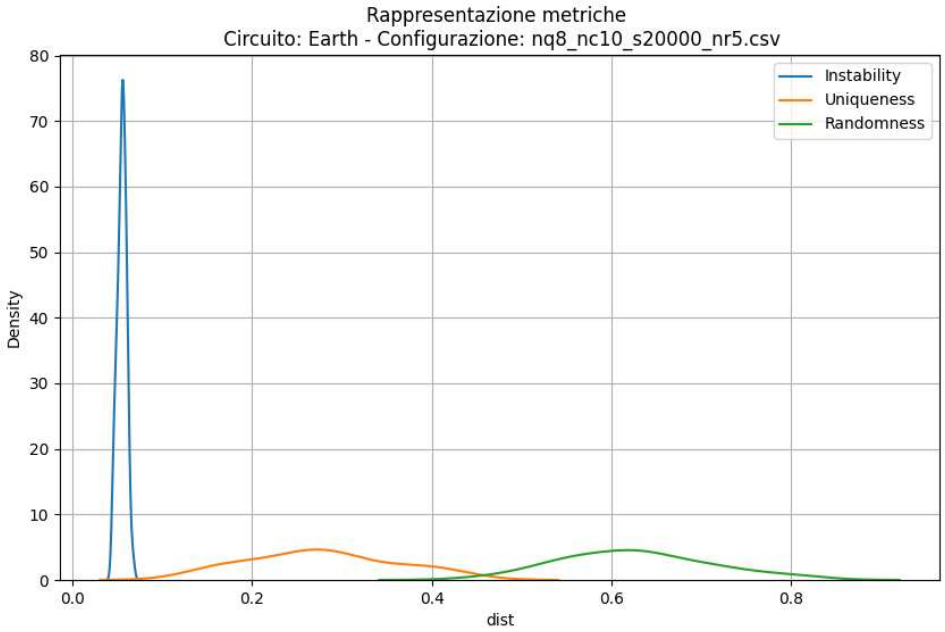

Fig. 4.1: Metriche per la configurazione nq8 nc10 s20.000 e nr5

Considerazioni: questo test è molto interessante visto che al calare della metrica dell'instability crescono le metriche della randomness e della uniqueness. Questo comportamento può portare a un ottimo risultato vendendo già basso il valore dell'instability.

Test con 10 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.20	0.69	0.30
10.000	0.15	0.68	0.25
20.000	0.11	0.67	0.23

Table 4.3: Configurazione a 10 qubit, 10 challenge e 5 runs

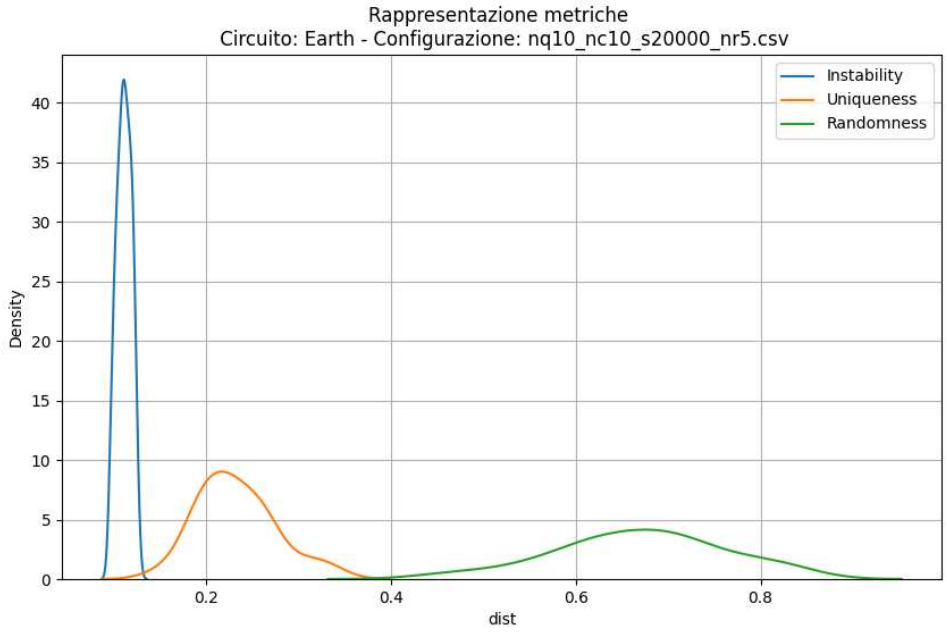

Fig. 4.2: Metriche per la configurazione nq10 nc10 s20.000 e nr5

Considerazioni: con 10 qubit notiamo una buona convergenza sullo stesso valore per quanto riguarda instability e randomness. Sarebbe interessante aumentare il numero di runs e shots.

Test con 12 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.33	0.81	0.39
10.000	0.26	0.77	0.35
20.000	0.19	0.76	0.30

Table 4.4: Configurazione a 12 qubit, 10 challenge e 5 runs

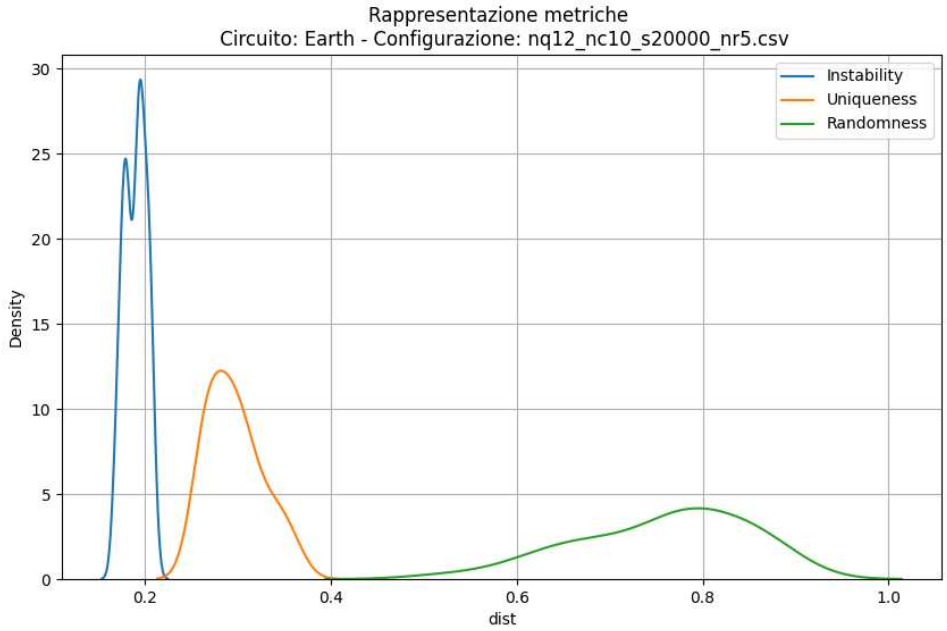

Fig. 4.3: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: durante questo test si nota un buon drop dell'instability al crescere degli shots, purtroppo non è stato possibile aumentarli per via delle limitazioni imposte dalla piattaforme di test.

Test con 16 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.71	0.95	0.72
10.000	0.61	0.92	0.63
20.000	0.50	0.89	0.52

Table 4.5: Configurazione a 16 qubit, 10 challenge e 5 runs

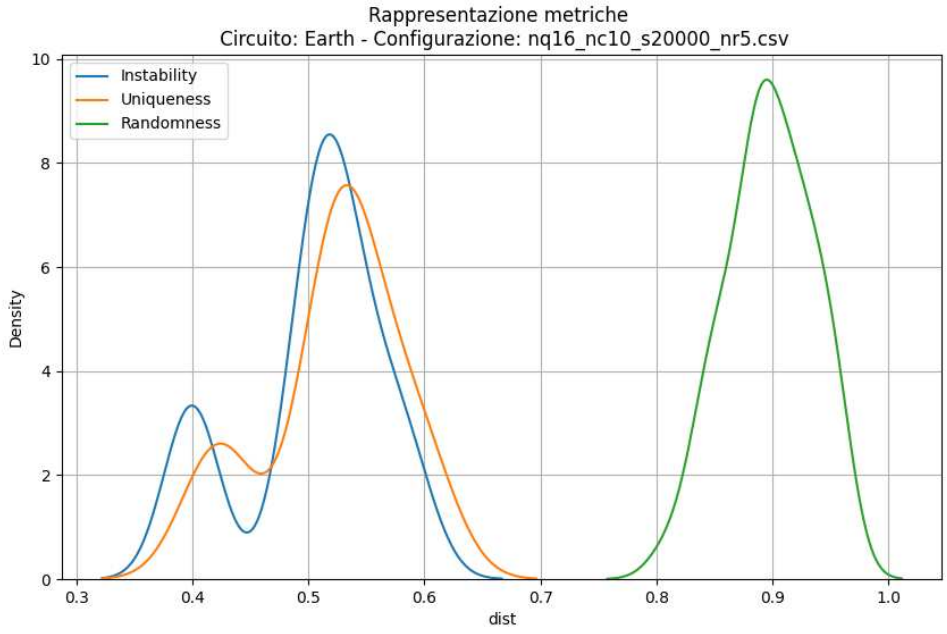

Fig. 4.4: Metriche per la configurazione nq16 nc10 s20.000 e nr5

Considerazioni: questo test è stato condotto per vedere come si comportava il circuito con un numero di qubit sempre più alto. I risultati sono stati bassi per la metrica di instability dati il numero di shots e runs in confronto con gli altri circuiti.

Test con 20 challenge, 20.000 shots e 5 runs

qubit	Instability	Randomness	Uniqueness
8	0.06	0.59	0.26
10	0.11	0.63	0.28
12	0.19	0.70	0.33

Table 4.6: Configurazione a 20 challenge, 20.000 shots e 5 runs

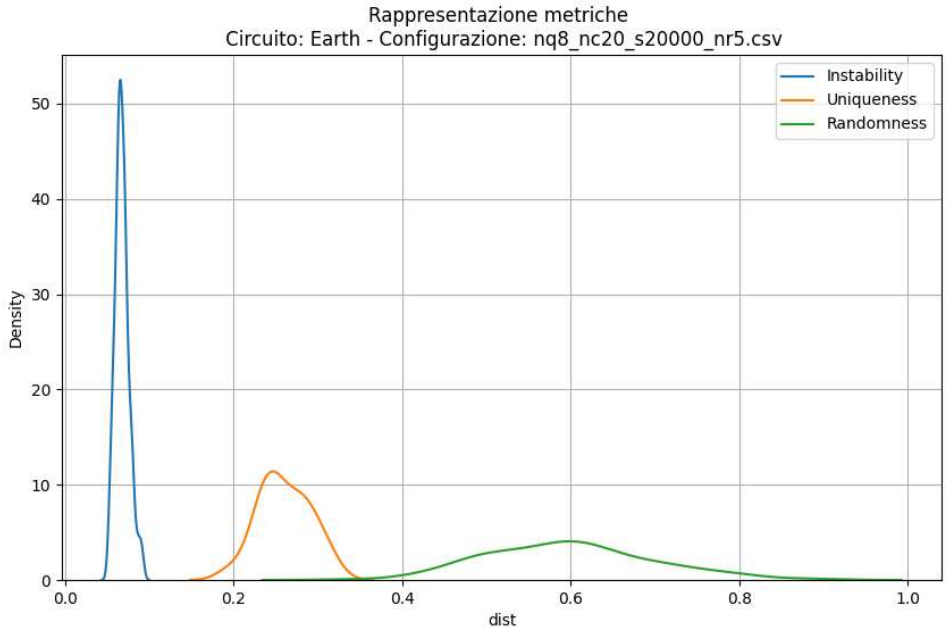

Fig. 4.5: Metriche per la configurazione nq8 nc20 s20.000 e nr5

Considerazioni: portando il numero di challenge a 20 notiamo un buon comportamento del circuito basato su 8 qubit, tuttavia si mantengono in linea con la loro controparte a 10 challenge non mostrando miglioramenti sostanziali.

4.3.2 Mars

Test con 8 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.09	0.76	0.17
10.000	0.07	0.73	0.16
20.000	0.05	0.74	0.15

Table 4.7: Configurazione a 8 qubit, 10 challenge e 5 runs

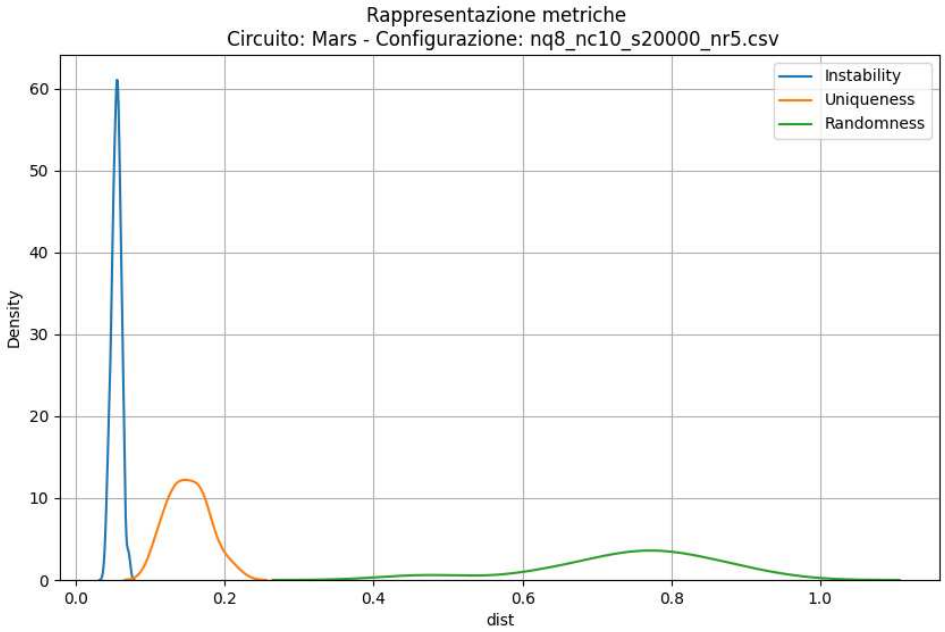

Fig. 4.6: Metriche per la configurazione nq8 nc10 s20.000 e nr5

Considerazioni: da questo test emerge che l'incremento del numero di shot ha un impatto più marcato sulla metrica di instability rispetto alle altre due metriche considerate, con una riduzione di circa 4 punti contro i 2 punti osservati per randomness e uniqueness.

Test con 10 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.16	0.81	0.27
10.000	0.12	0.80	0.24
20.000	0.09	0.79	0.24

Table 4.8: Configurazione a 10 qubit, 10 challenge e 5 runs

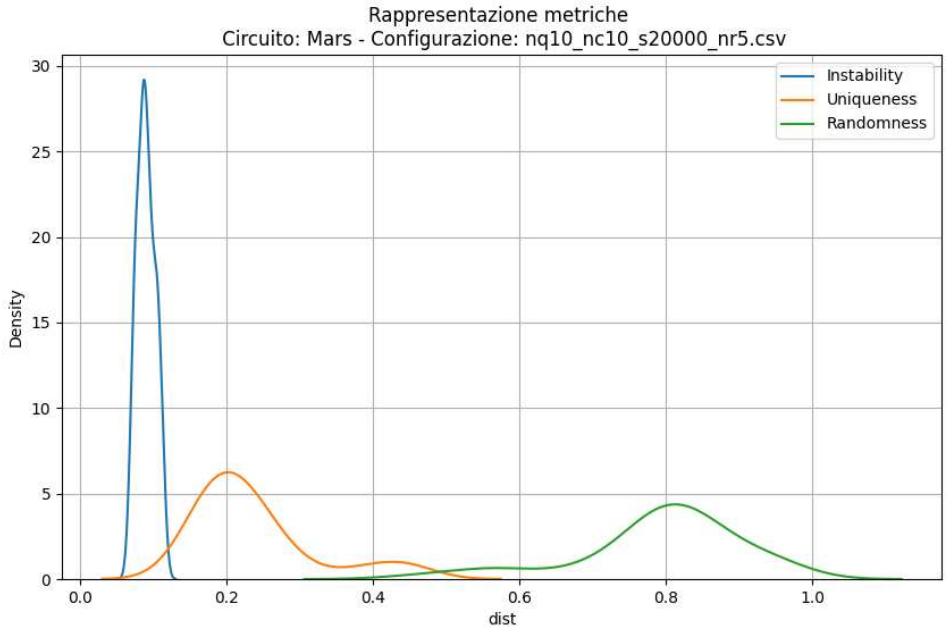

Fig. 4.7: Metriche per la configurazione nq10 nc10 s20.000 e nr5

Considerazioni: rispetto al test precedente, si osserva un incremento significativo della metrica di uniqueness, pari a circa 10 punti percentuali, a fronte però di un lieve peggioramento della instability.

Test con 12 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.25	0.90	0.34
10.000	0.19	0.88	0.31
20.000	0.14	0.88	0.27

Table 4.9: Configurazione a 12 qubit, 10 challenge e 5 runs

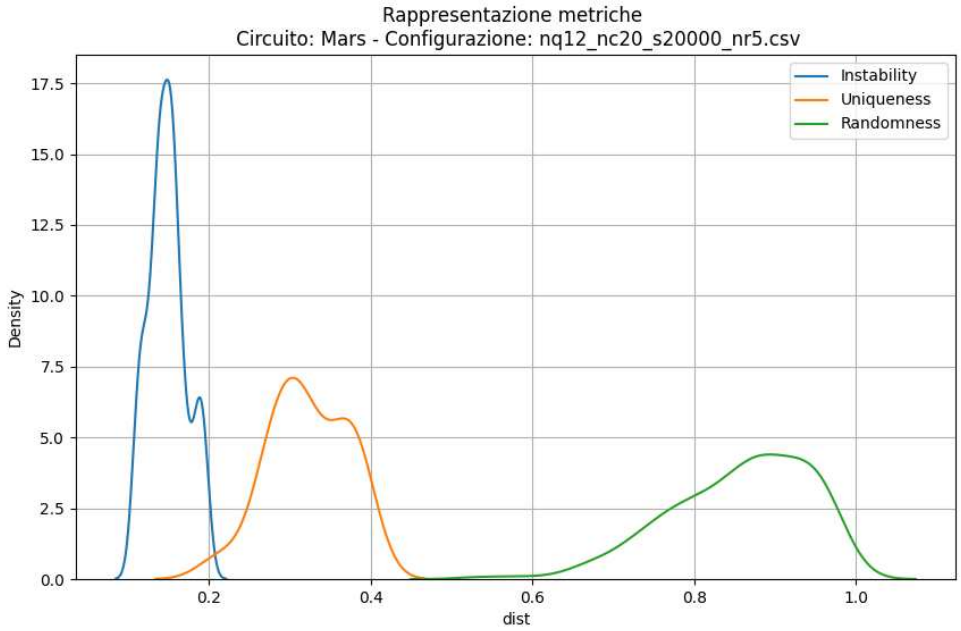

Fig. 4.8: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: in questo test notiamo valori alti per la metrica di randomness, tuttavia notiamo un peggioramento maggiore per l'instability e un lieve guadagno per la uniqueness, che denota un cattivo comportamento all'alzare il numero di qubit.

Test con 20 challenge, 20.000 shots e 5 runs

qubit	Instability	Randomness	Uniqueness
8	0.05	0.75	0.30
10	0.09	0.78	0.33
12	0.15	0.86	0.32

Table 4.10: Configurazione a 20 challenge, 20.000 shots e 5 runs

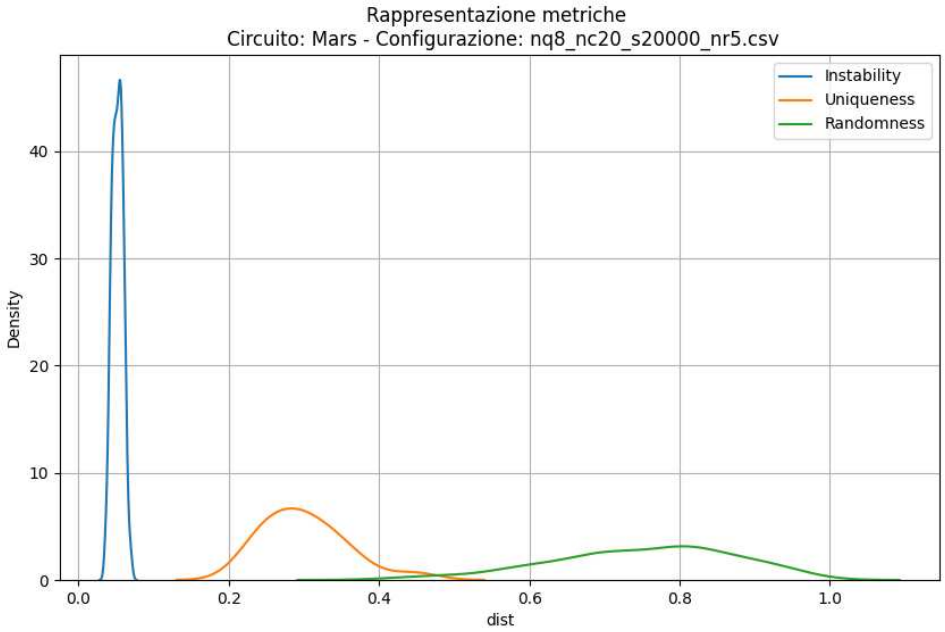

Fig. 4.9: Metriche per la configurazione nq8 nc20 s20.000 e nr5

Considerazioni portando il numero di challege a 20 notiamo un buon comportamento del circuito basato su 8 qubit, che mostra lo stesso livello di instability ma raddoppia il valore di uniqueness portandolo a 30. Gli altri 2 ciruiti testati (10 e 12 qubit) mostrano anche loro questo comportamento ma la uniqueness rimane un pò piu bassa ripetto al raddoppio del circuito a 8 qubit.

4.3.3 Venus

Test con 8 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.11	0.64	0.31
10.000	0.08	0.64	0.30
20.000	0.06	0.63	0.29

Table 4.11: Configurazione a 8 qubit, 10 challenge e 5 runs

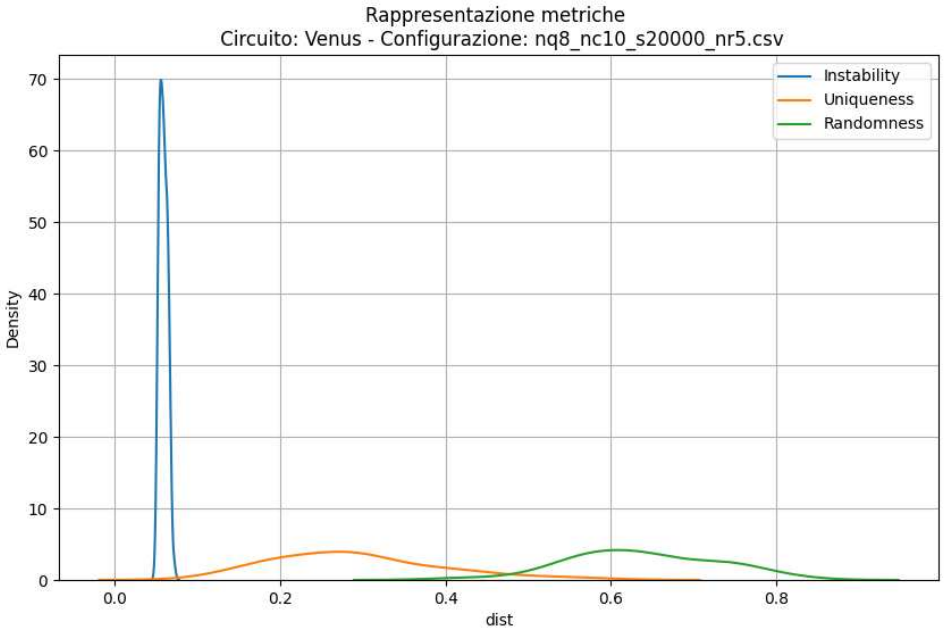

Fig. 4.10: Metriche per la configurazione nq8 nc10 s20.000 e nr5

Considerazioni: in questo test abbiamo ottimi valori per la metrica di instability e un basso drop per la metrica di uniqueness all'aumentare del numero degli shot.

Test con 10 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.20	0.69	0.38
10.000	0.15	0.68	0.39
20.000	0.10	0.67	0.36

Table 4.12: Configurazione a 10 qubit, 10 challenge e 5 runs

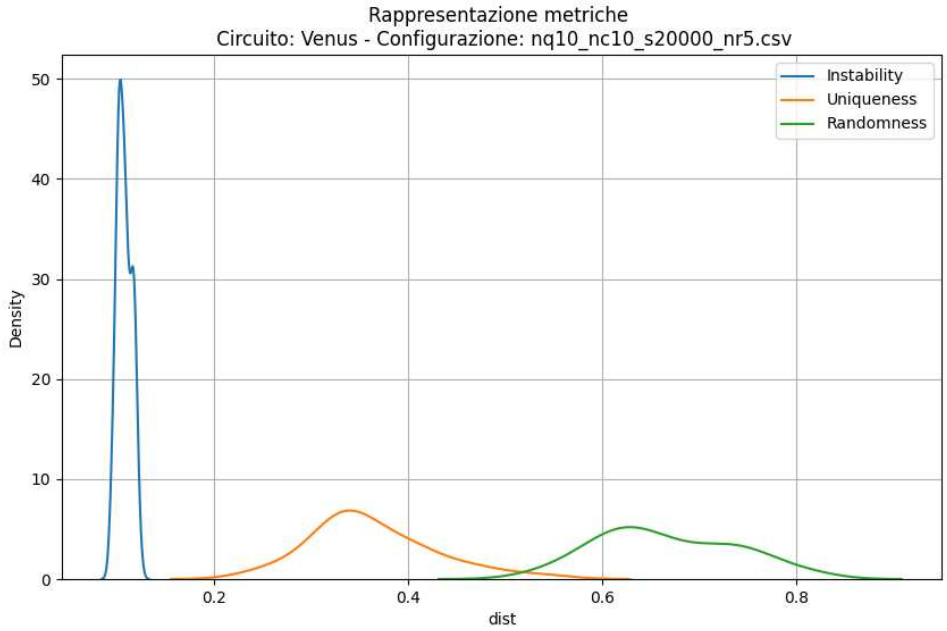

Fig. 4.11: Metriche per la configurazione nq10 nc10 s20.000 e nr5

Considerazioni: in questo test notiamo che portando il numero di qubit a 10 abbiamo un buon incremento per la metrica di uniqueness al netto di un leggero peggioramento dell'instability.

Test con 12 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.35	0.79	0.53
10.000	0.26	0.74	0.47
20.000	0.19	0.72	0.43

Table 4.13: Configurazione a 12 qubit, 10 challenge e 5 runs

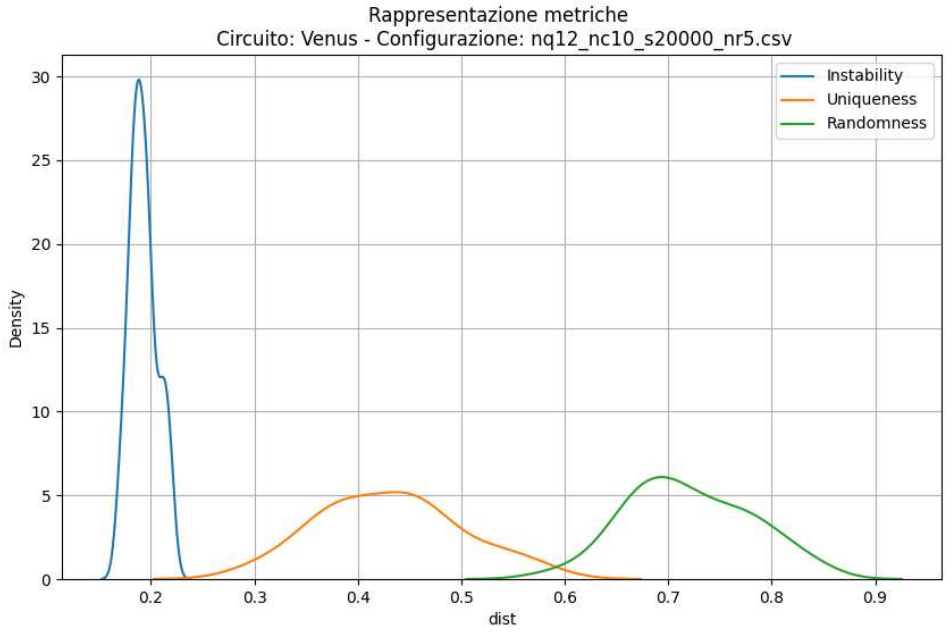

Fig. 4.12: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: portando il numero dei qubit a 12 notiamo un incremento delle metriche di randomness e uniqueness ma anche un netto peggioramento dell'instability.

Test con 16 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.71	0.95	0.84
10.000	0.62	0.92	0.78
20.000	0.51	0.88	0.73

Table 4.14: Configurazione a 16 qubit, 10 challenge e 5 runs

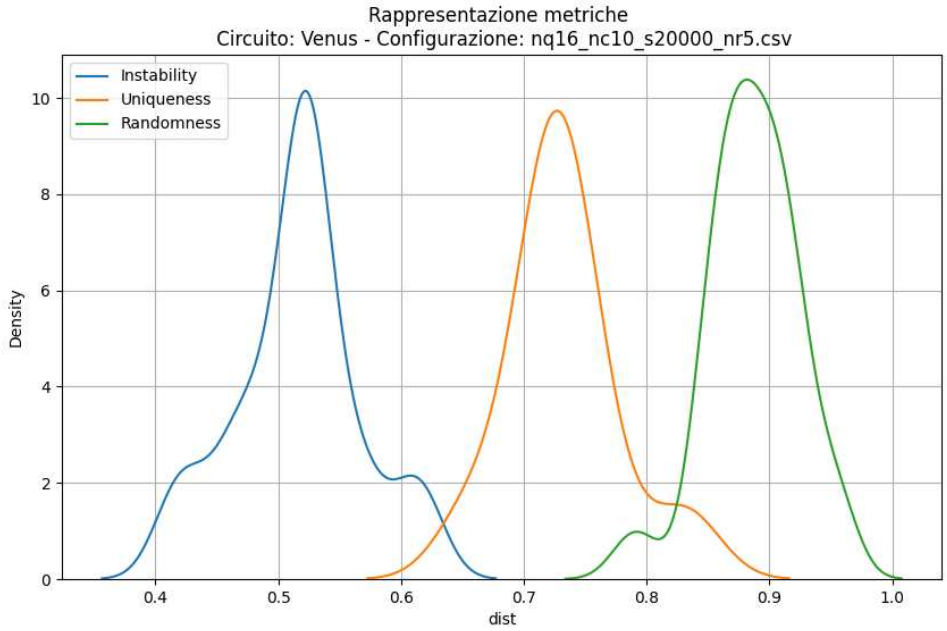

Fig. 4.13: Metriche per la configurazione nq16 nc10 s20.000 e nr5

Considerazioni: in questo test notiamo valori alti per la metrica di randomness, tuttavia notiamo un peggioramento maggiore per l'instability e un lieve guadagno per la uniqueness, che denota un cattivo comportamento all'alzare il numero di qubit.

Test con 20 challenge, 20.000 shots e 5 runs

qubit	Instability	Randomness	Uniqueness
8	0.06	0.56	0.37
10	0.11	0.62	0.41
12	0.20	0.68	0.46

Table 4.15: Configurazione a 20.000 shots, 20 challenge e 5 runs

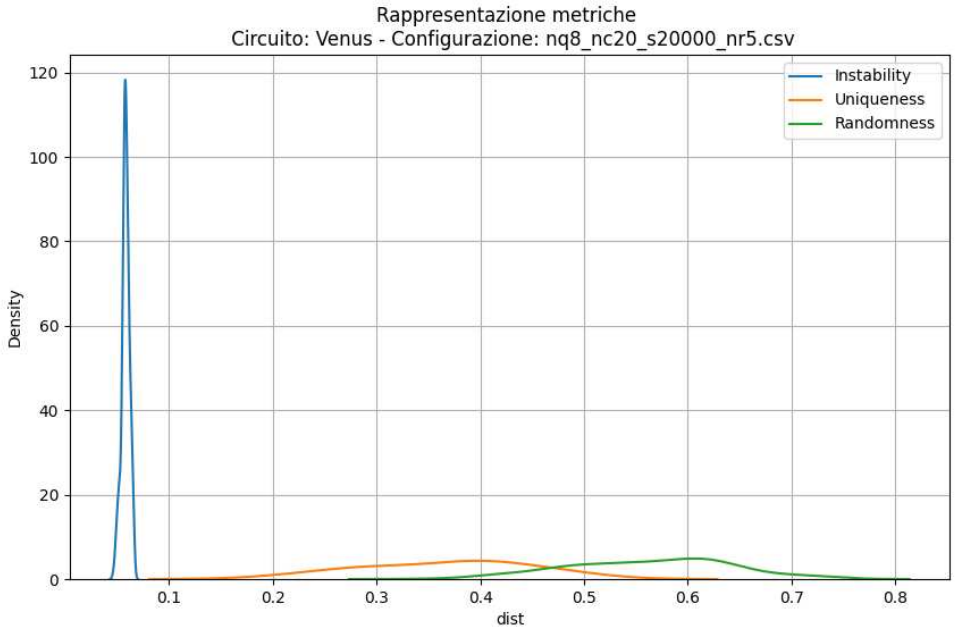

Fig. 4.14: Metriche per la configurazione nq8 nc20 s20.000 e nr5

Considerazioni: in questo test notiamo che rispetto agli altri circuiti, le metriche di instability e randomness rimangono quasi invariate mentre notiamo un ottimo aumento per la metrica di uniqueness. Notiamo dal grafico che è presente una spike molto alta per l'instability che mostra un convergenza verso uno specifico valore.

4.3.4 Mercury

Test con 8 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.34	0.48	0.39
10.000	0.33	0.44	0.37
20.000	0.32	0.43	0.36

Table 4.16: Configurazione a 8 qubit, 10 challenge e 5 runs

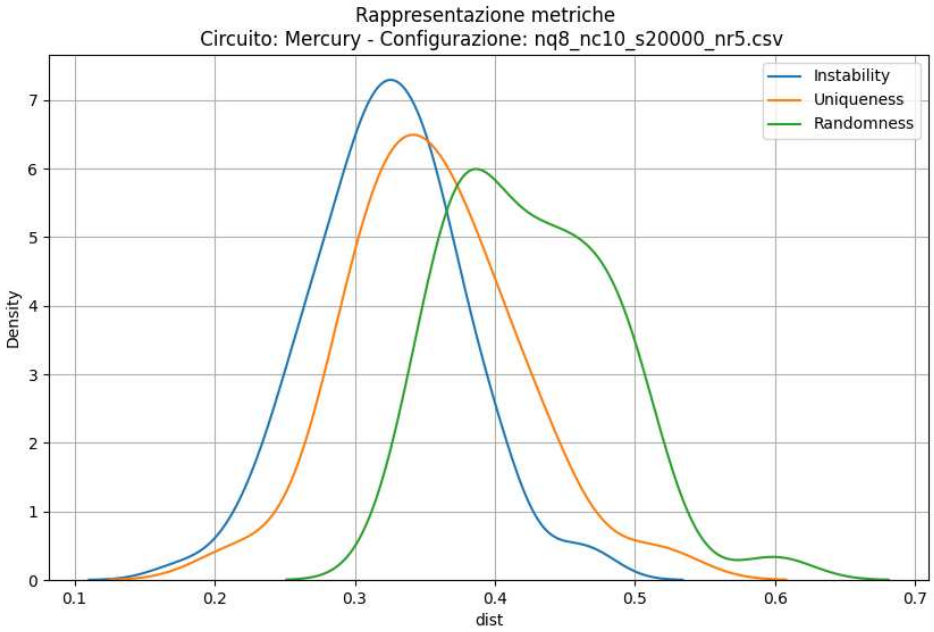

Fig. 4.15: Metriche per la configurazione nq8 nc10 s20.000 e nr5

Considerazioni: utilizzando questo circuito si nota subito un cattivo comportamento, soprattutto per un basso calo della metrica dell'instability che di base comunque è alta. Purtroppo le curve sono troppo vicine il che mostra un risultato non ottimale.

Test con 10 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.39	0.51	0.42
10.000	0.35	0.46	0.37
20.000	0.35	0.45	0.34

Table 4.17: Configurazione a 10 qubit, 10 challenge e 5 runs

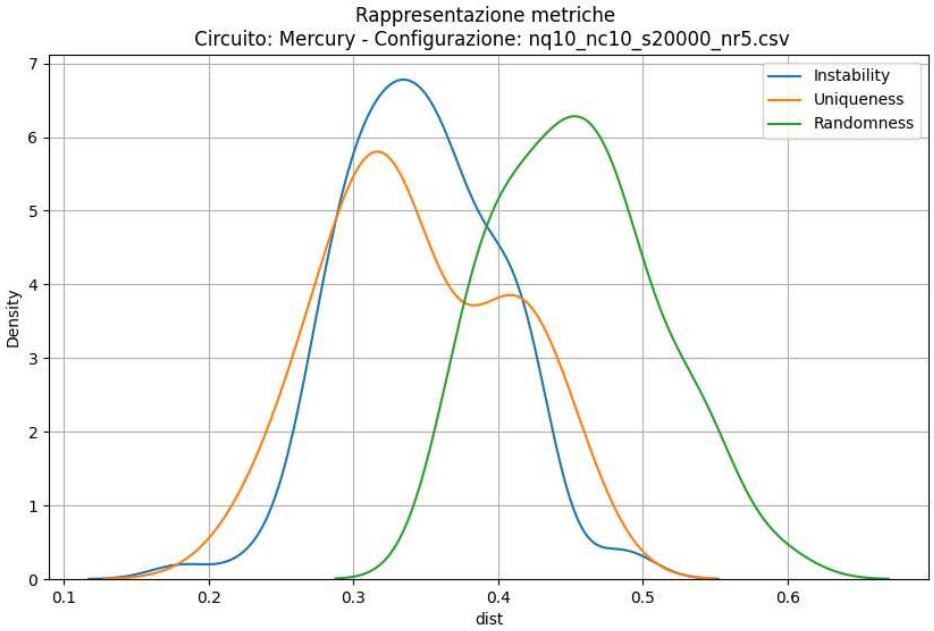

Fig. 4.16: Metriche per la configurazione nq10 nc10 s20.000 e nr5

Considerazioni: aumentando il numero di qubit il comportamento non cambia, l'instability rimane alta a scende di poco, mentre per quanto riguarda le metriche di randomness e uniqueness calano in modo più marcato.

Test con 12 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.51	0.62	0.54
10.000	0.41	0.56	0.46
20.000	0.35	0.50	0.42

Table 4.18: Configurazione a 12 qubit, 10 challenge e 5 runs

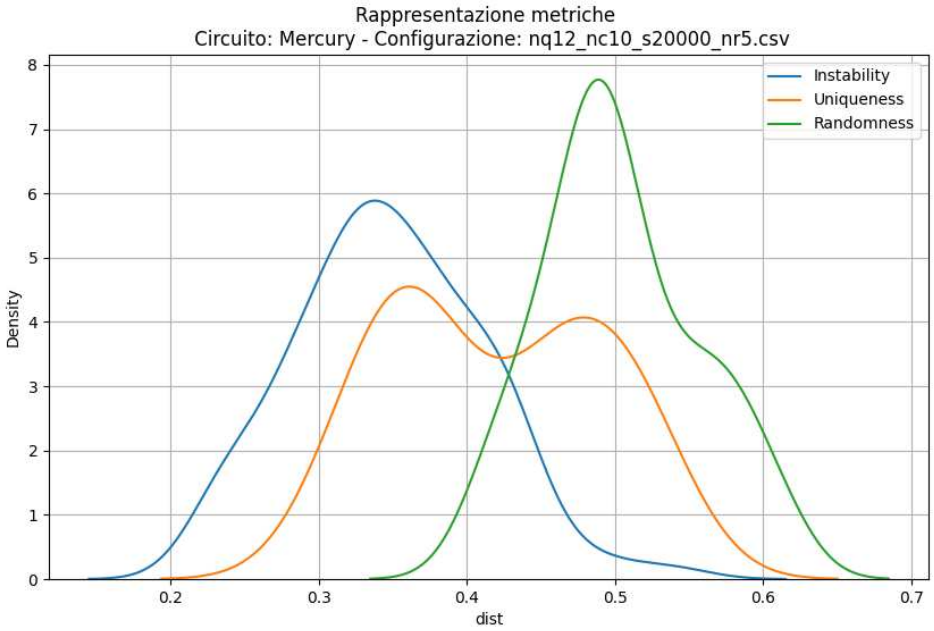

Fig. 4.17: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: il comportamento con un numero maggiore di qubit appare leggermente migliorato, in quanto l'instability si riduce sensibilmente all'aumentare degli shot, accompagnata da un incremento delle metriche di randomness e uniqueness. Tuttavia, l'instability rimane comunque elevata, rendendo necessario un ulteriore approfondimento tramite un numero maggiore di shot e run per confermare la stabilità delle misure.

Test con 20 challenge, 20.000 shots e 5 runs

qubit	Instability	Randomness	Uniqueness
8	0.31	0.42	0.31

Table 4.19: Configurazione a 20.000 shots, 20 challenge e 5 runs

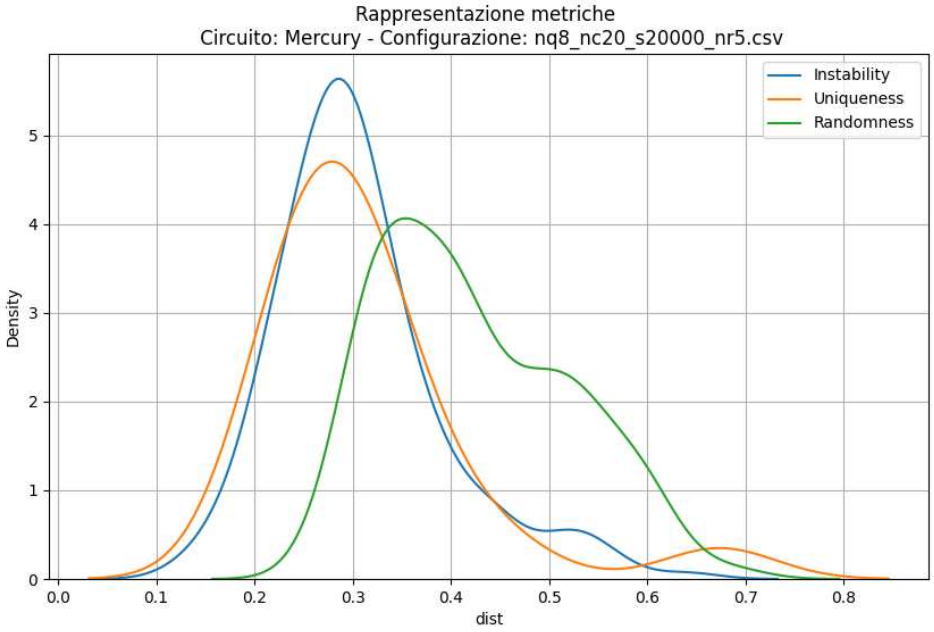

Fig. 4.18: Metriche per la configurazione nq8 nc20 s20.000 e nr5

Considerazioni: in questo test si evidenza come aumentando il numero di challenge a 20 non cambia il comportamento medio del circuito mostrando sempre un alto valore di instability e valori peggiori per le altre metriche di randomness e uniqueness.

4.3.5 Saturn

Test con 8 qubit, 10 challenge e 5 runs

Table 4.20: Configurazione a 8 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.12	0.55	0.30
10.000	0.09	0.56	0.28
20.000	0.07	0.57	0.27

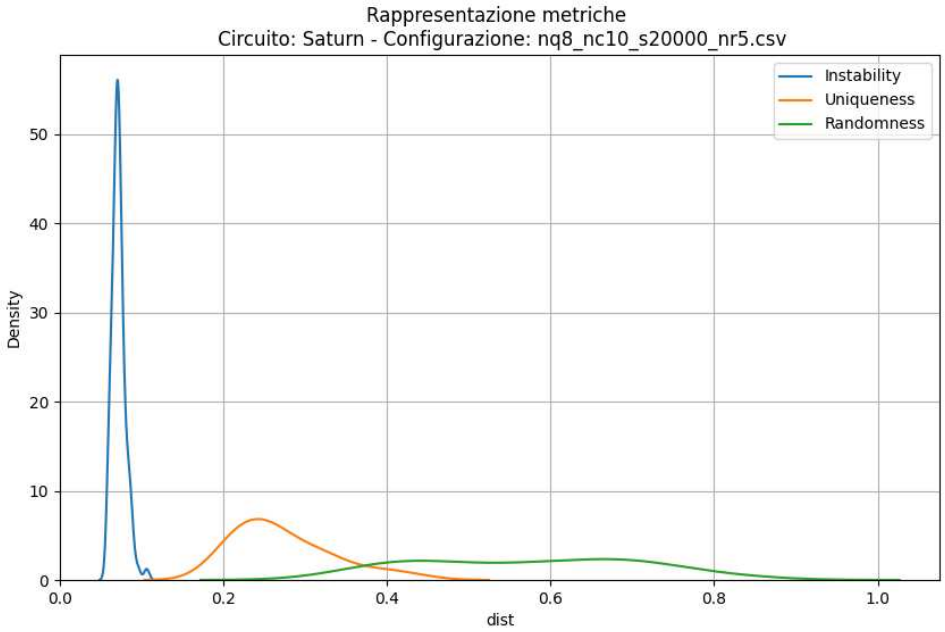

Fig. 4.19: Metriche per la configurazione nq8 nc10 s20.000 e nr5

Considerazioni: in questo test si evidenzia come abbiamo valori accettabili di instability al netto di valori discreti per la randomness e uniqueness.

Test con 10 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.21	0.62	0.44
10.000	0.15	0.63	0.40
20.000	0.12	0.61	0.39

Table 4.21: Configurazione a 10 qubit, 10 challenge e 5 runs

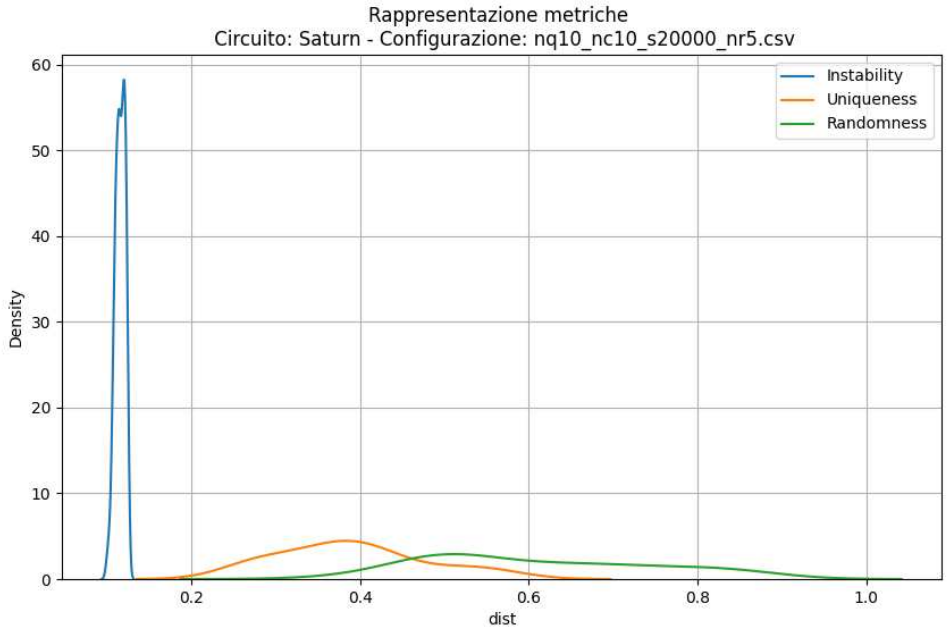

Fig. 4.20: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: aumentando a 10 il numero di qubit notiamo che c'è un leggero peggioramento per la metrica di instability ma un netto miglioramento per le metriche di randomness e uniqueness. 10 quibit sembra essere un buon trade-off per questo circuito.

Test con 12 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.40	0.68	0.58
10.000	0.30	0.63	0.50
20.000	0.22	0.60	0.47

Table 4.22: Configurazione a 12 qubit, 10 challenge e 5 runs

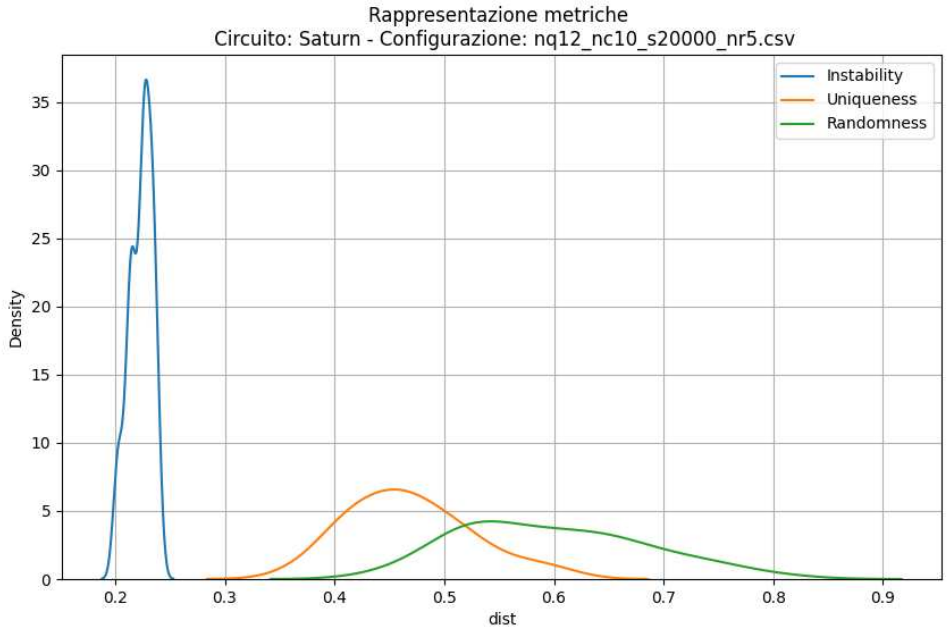

Fig. 4.21: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: in questo test si evidenzia come portando a 12 il numero dei qubit si hanno valori più alti per la metrica di instability che però potrebbe scendere andando ad aumentare il numero di shot e/o runs.

Test con 20 challenge, 20.000 shots e 5 runs

qubit	Instability	Randomness	Uniqueness
8	0.06	0.43	0.36
10	0.11	0.42	0.41
12	0.22	0.56	0.46

Table 4.23: Configurazione a 20.000 shots, 20 challenge e 5 runs

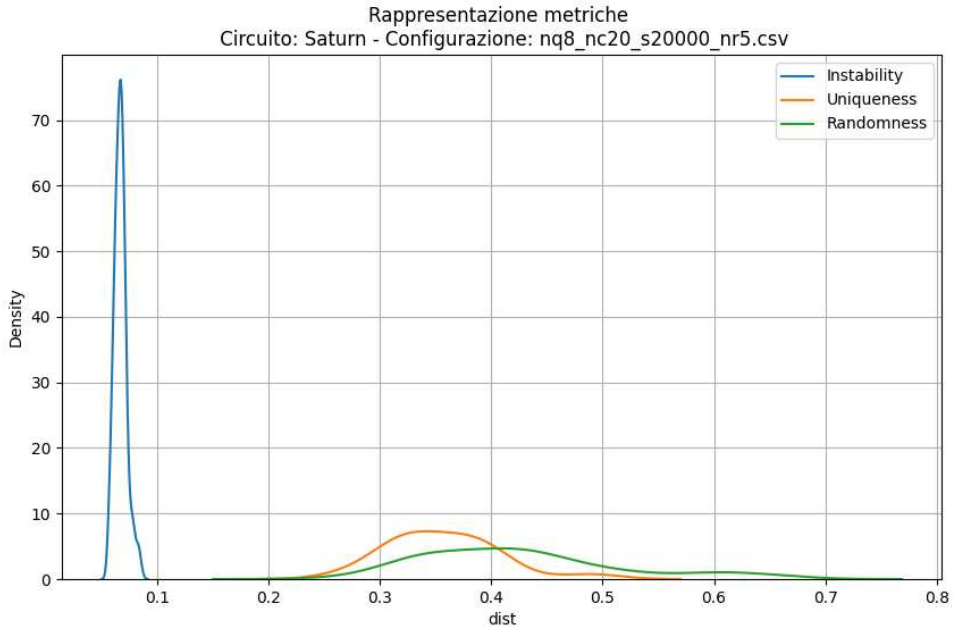

Fig. 4.22: Metriche per la configurazione nq8 nc20 s20.000 e nr5

Considerazioni: in questo test si evidenza come aumentando il numero di challenge a 20 cambia il comportamento medio del circuito mostrando valori simili per la metrica di instability, ma valori più alti per le altre metriche.

4.3.6 Jupiter

Test con 8 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.27	0.71	0.30
10.000	0.26	0.70	0.31
20.000	0.23	0.70	0.23

Table 4.24: Configurazione a 8 qubit, 10 challenge e 5 runs

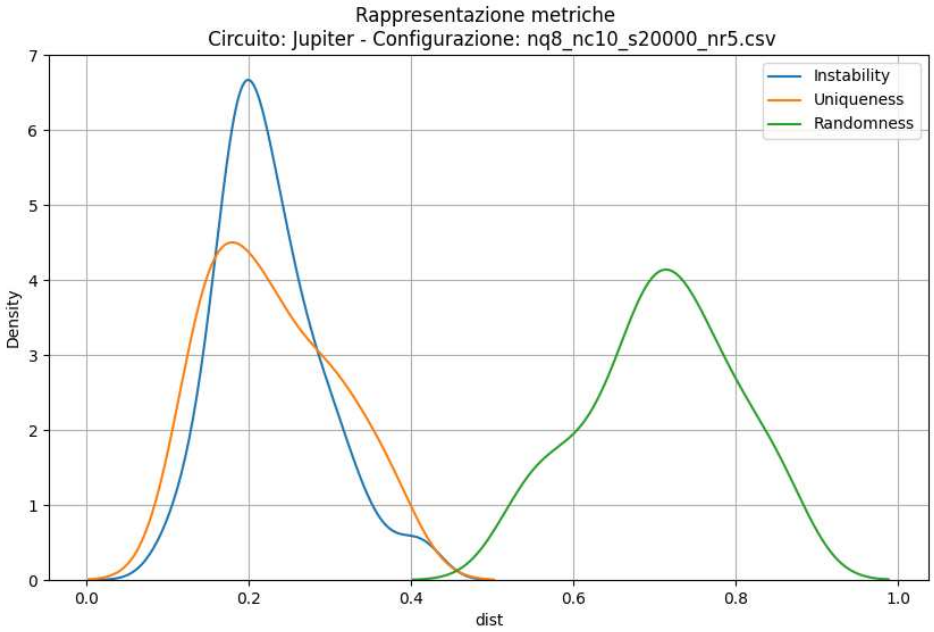

Fig. 4.23: Metriche per la configurazione nq8 nc10 s20.000 e nr5

Considerazioni: in questo test si nota che questo circuito mostra valori alti per la metrica dell'instability e che non si discosta dalla curva che rappresenta la metrica di uniqueness.

Test con 10 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.38	0.74	0.37
10.000	0.35	0.72	0.40
20.000	0.32	0.71	0.33

Table 4.25: Configurazione a 10 qubit, 10 challenge e 5 runs

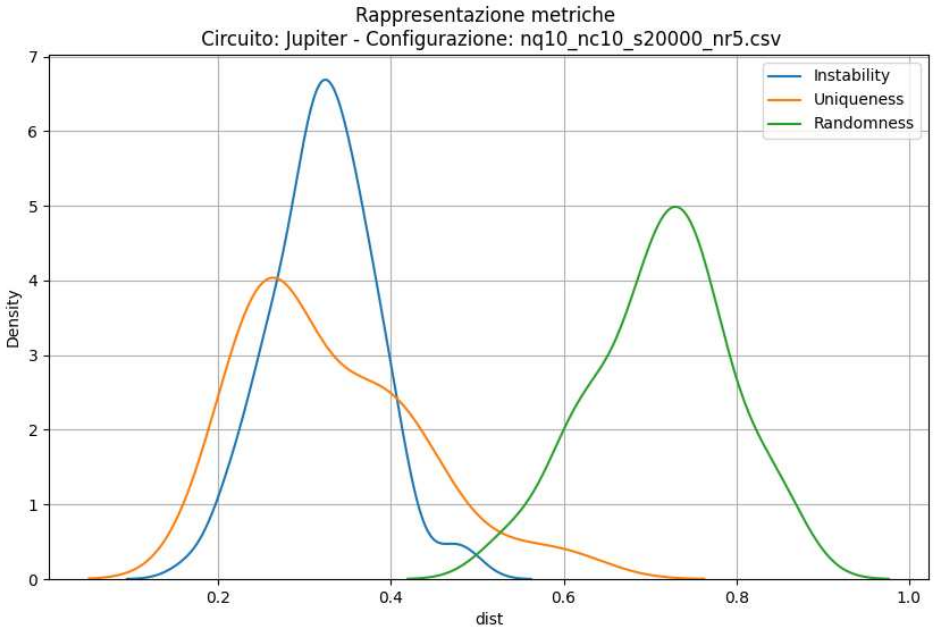

Fig. 4.24: Metriche per la configurazione nq10 nc10 s20.000 e nr5

Considerazioni: in questo test si nota un comportamento che è uguale per il circuito a 8 qubit.

Test con 12 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.30	0.85	0.33
10.000	0.24	0.83	0.29
20.000	0.18	0.81	0.24

Table 4.26: Configurazione a 12 qubit, 10 challenge e 5 runs

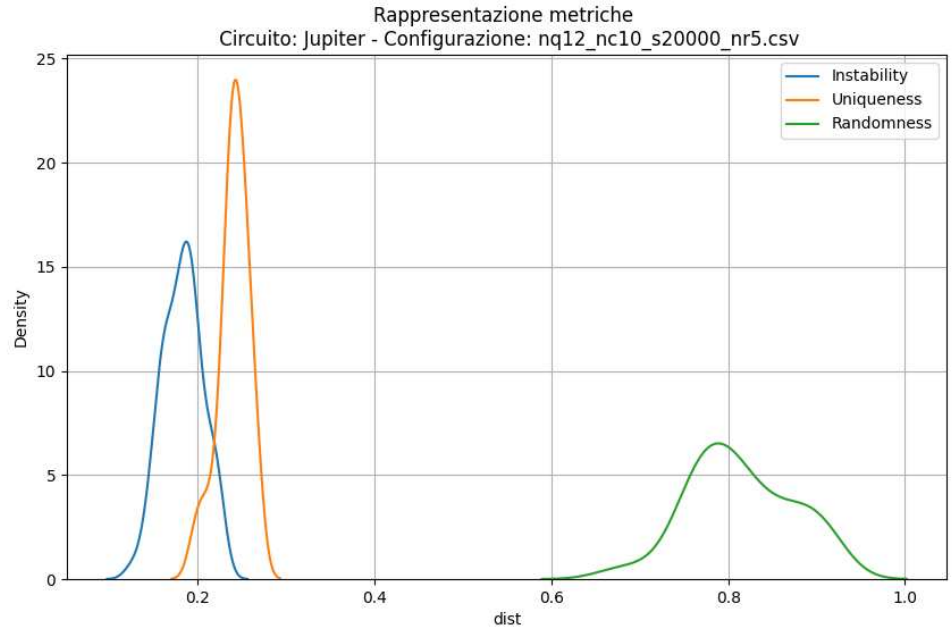

Fig. 4.25: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: in questo test si evidenza un leggero miglioramento rispetto ai 2 test precedenti. Tuttavia le curve non si discostano abbastanza per ritenerlo un circuito valido per ulteriori test futuri.

Test con 20 challenge, 20.000 shots e 5 runs

qubit	Instability	Randomness	Uniqueness
8	0.27	0.69	0.27
10	0.35	0.70	0.36
12	0.19	0.80	0.24

Table 4.27: Configurazione a 20.000 shots, 20 challenge e 5 runs

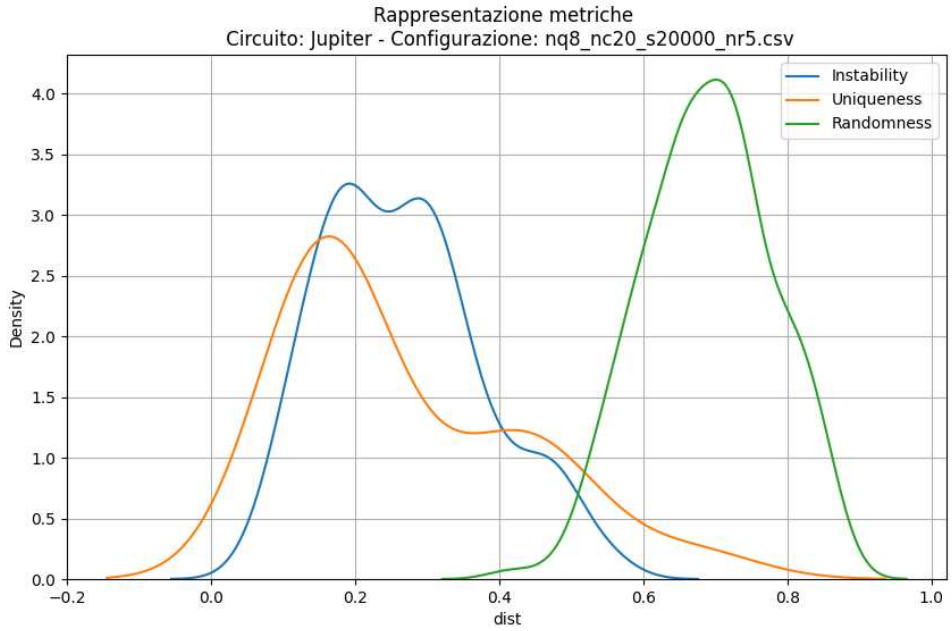

Fig. 4.26: Metriche per la configurazione nq8 nc20 s20.000 e nr5

Considerazioni: in questo test si evidenza come aumentando il numero di challenge a 20 non cambia il comportamento medio del circuito mostrando valori simili per le 3 metriche metriche.

4.3.7 Uranus

Test con 8 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.11	0.60	0.37
10.000	0.08	0.58	0.37
20.000	0.06	0.60	0.38

Table 4.28: Configurazione a 8 qubit, 10 challenge e 5 runs

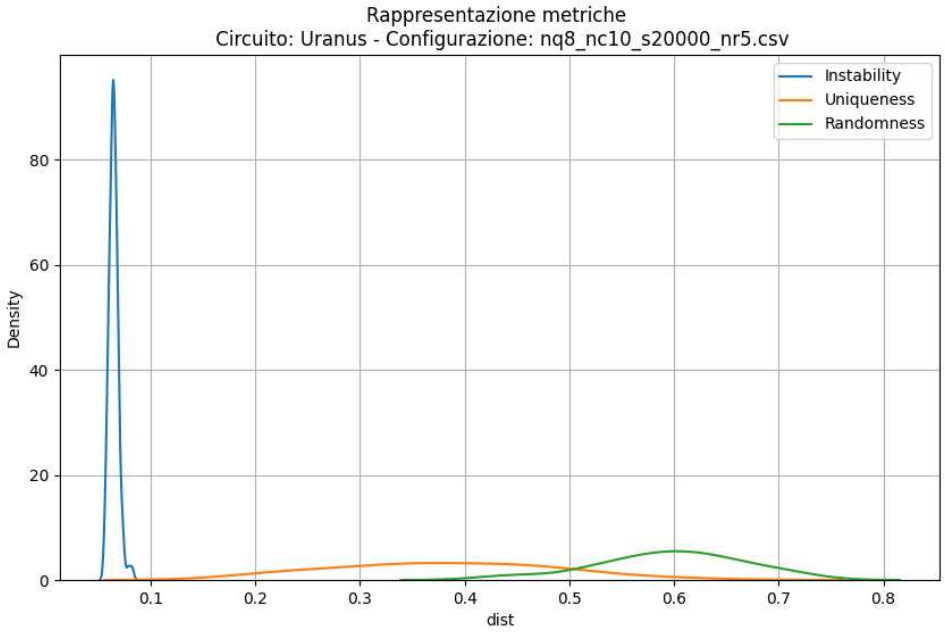

Fig. 4.27: Metriche per la configurazione nq8 nc10 s20.000 e nr5

Considerazioni: in questo test si nota che questo circuito mostra valori ottimi per la metrica dell'instability e allo stesso tempo buoni valori di uniqueness.

Test con 10 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.21	0.64	0.47
10.000	0.15	0.63	0.46
20.000	0.11	0.61	0.44

Table 4.29: Configurazione a 10 qubit, 10 challenge e 5 runs

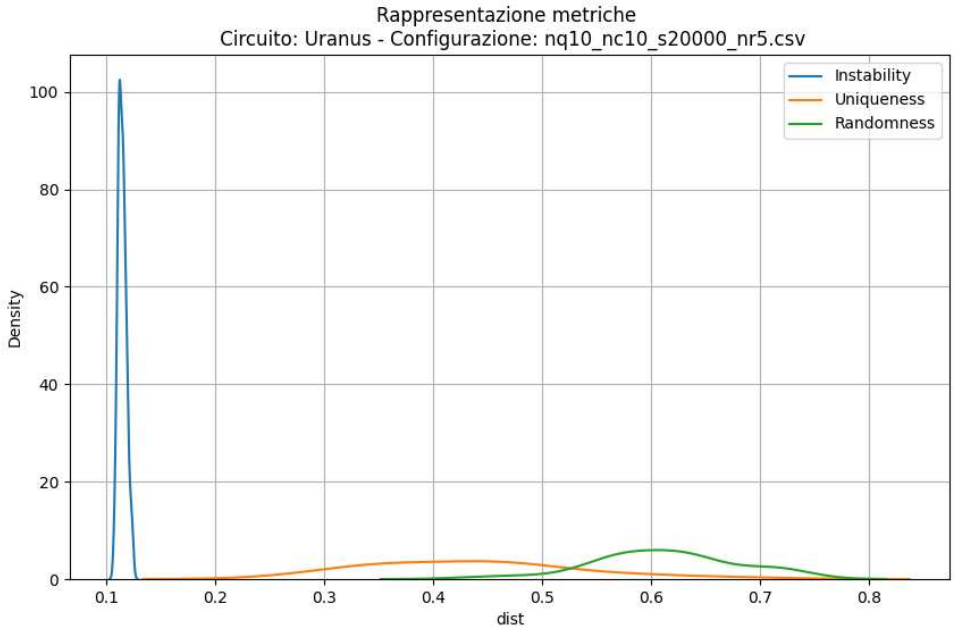

Fig. 4.28: Metriche per la configurazione nq10 nc10 s20.000 e nr5

Considerazioni: in questo test si nota che aumentando il numero di qubit su questo circuito si hanno valori in linea con il test precedente.

Test con 12 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.37	0.72	0.59
10.000	0.28	0.68	0.53
20.000	0.21	0.65	0.50

Table 4.30: Configurazione a 12 qubit, 10 challenge e 5 runs

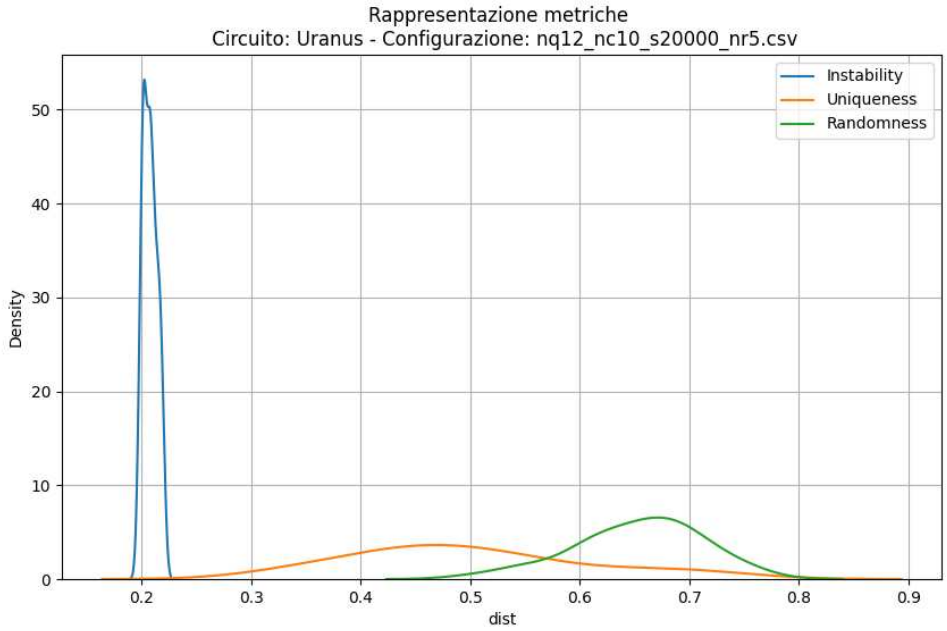

Fig. 4.29: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: in questo test, l'aumento del numero di qubit a 12 non ha portato benefici alla metrica di instability, che risulta peggiorata di circa il doppio rispetto alla configurazione con 10 qubit.

Test con 20 challenge, 20.000 shots e 5 runs

qubit	Instability	Randomness	Uniqueness
8	0.06	0.56	0.48
10	0.11	0.64	0.46
12	0.20	0.67	0.53

Table 4.31: Configurazione a 20.000 shots, 20 challenge e 5 runs

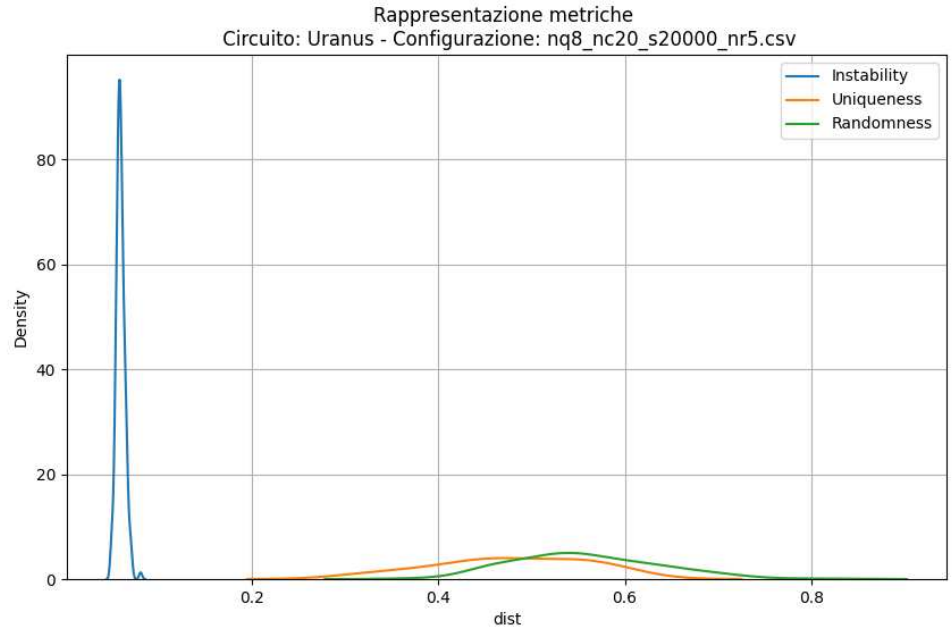

Fig. 4.30: Metriche per la configurazione nq8 nc20 s20.000 e nr5

Considerazioni: in questo test si nota che portando il numero di challenge a 20 notiamo un ottimo risultato per quanto riguarda il circuito a 8 qubit che mostra un basso valore di instability e quasi il 50% di uniqueness. Per gli altri 2 circuiti con 10 e 12 qubit si hanno valori simili alla controparte con 10 challenge .

4.3.8 Uranus Titania

Test con 8 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.12	0.43	0.39
10.000	0.83	0.42	0.36
20.000	0.06	0.4	0.36

Table 4.32: Configurazione a 8 qubit, 10 challenge e 5 runs

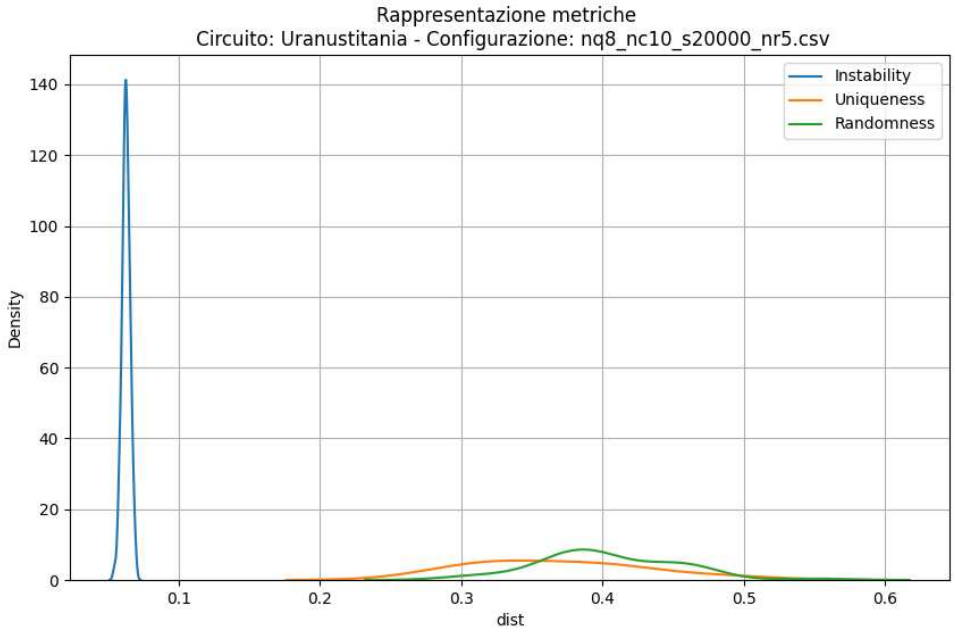

Fig. 4.31: Metriche per la configurazione nq8 nc10 s20.000 e nr5

Considerazioni: in questo test si osservano risultati molto positivi per quanto riguarda le metriche di instability e uniqueness. Inoltre, si registra un miglioramento nella distanza tra le curve, con un conseguente incremento delle performance anche nei casi peggiori.

Test con 10 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.23	0.49	0.44
10.000	0.16	0.45	0.43
20.000	0.12	0.42	0.38

Table 4.33: Configurazione a 10 qubit, 10 challenge e 5 runs

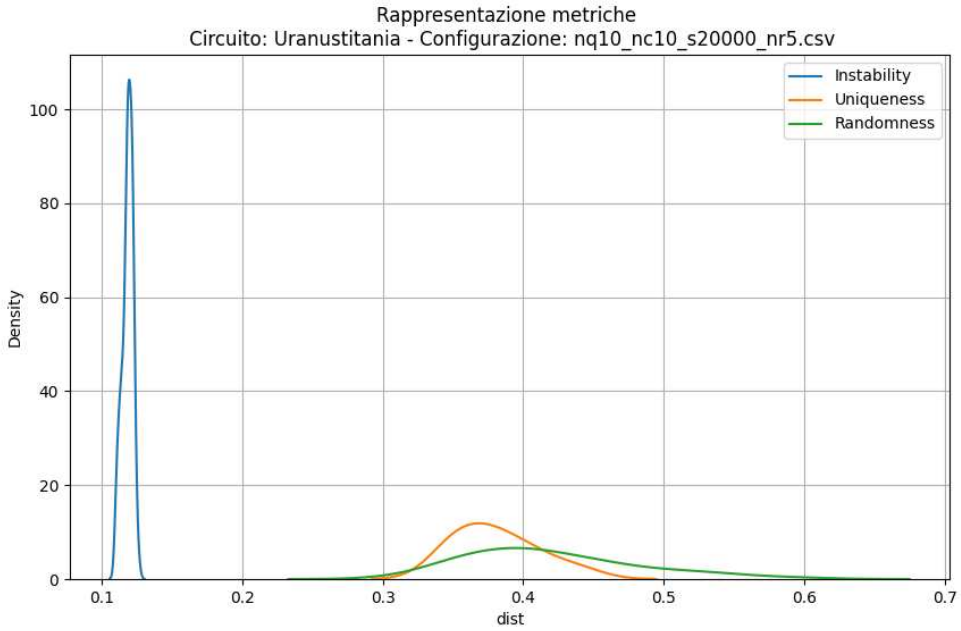

Fig. 4.32: Metriche per la configurazione nq10 nc10 s20.000 e nr5

Considerazioni: all'aumentare del numero di qubit a 10, il comportamento osservato risulta complessivamente simile a quello del test condotto con 8 qubit. Tuttavia, si rileva un lieve peggioramento della instability, compensato da un leggero miglioramento delle metriche di randomness e uniqueness.

Test con 12 qubit, 10 challenge e 5 runs

Shots	Instability	Randomness	Uniqueness
5.000	0.42	0.65	0.61
10.000	0.32	0.53	0.50
20.000	0.23	0.47	0.47

Table 4.34: Configurazione a 12 qubit, 10 challenge e 5 runs

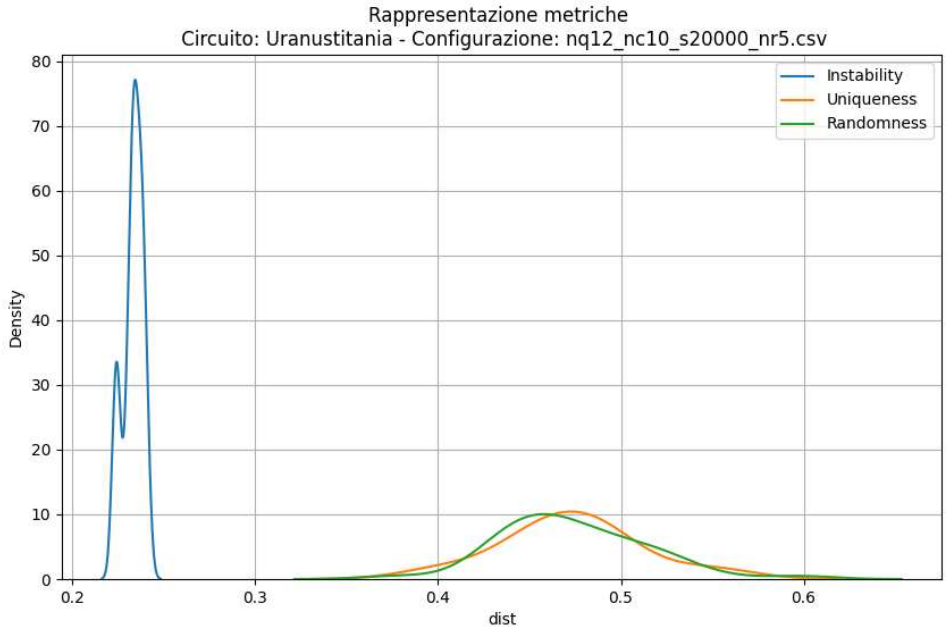

Fig. 4.33: Metriche per la configurazione nq12 nc10 s20.000 e nr5

Considerazioni: con l'incremento a 12 qubit, si osserva un ulteriore aumento dell'instability, come evidente dallo spostamento della curva verso valori più alti. Tuttavia, tale peggioramento è controbilanciato da un miglioramento progressivo sia nella randomness che nella uniqueness. In particolare, le curve mostrano una maggiore compattezza e simmetria nella distribuzione, segno di una maggiore regolarità e coerenza del comportamento del circuito.

Test con 20 challenge, 20.000 shots e 5 runs

qubit	Instability	Randomness	Uniqueness
8	0.06	0.36	0.38
10	0.12	0.39	0.40
12	0.23	0.46	0.45

Table 4.35: Configurazione a 20.000 shots, 20 challenge e 5 runs

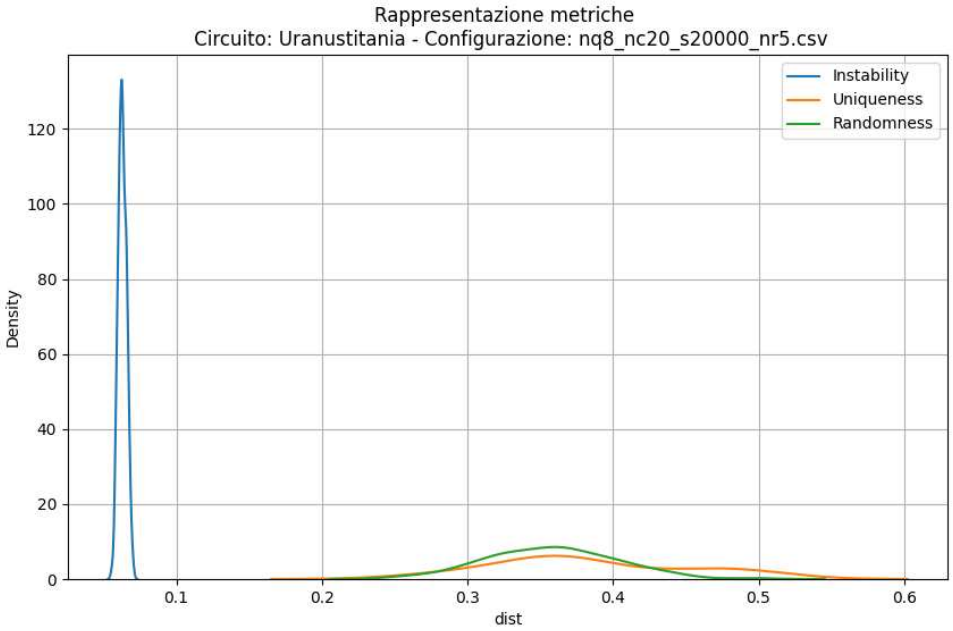

Fig. 4.34: Metriche per la configurazione nq8 nc20 s20.000 e nr5

Considerazioni: rispetto ai test a 10 challenge si osservano miglioramenti per la metriche di instability e uniqueness per i circuiti a 8 e 10 qubit. Il circuito a 12 qubit invece mostra un comportamento in linea con quello testato con 10 challenge.

4.4 Classifica dei migliori risultati

Le configurazioni riportate in Tabella 4.36 rappresentano le dieci combinazioni di parametri con la **massima differenza** tra *unicità* e *instabilità*. Per ogni configurazione testata, la differenza è stata calcolata secondo la seguente formula:

$$\Delta = \min(\text{Uniqueness}) - \max(\text{Instability})$$

Questo criterio identifica i casi peggiori di ogni configurazione, dove l'instabilità è massima e l'unicità minima, condizioni critiche per l'affidabilità delle PUF. A differenza della media, valuta la massima distanza tra le due metriche, penalizzando le configurazioni con sovrapposizioni rischiose e premiando quelle che garantiscono una netta separazione tra alta unicità e bassa instabilità. In questo modo si valorizzano le configurazioni in grado di mantenere una chiara separazione tra i comportamenti desiderabili (alta unicità) e quelli da evitare (alta instabilità).

I risultati dello studio precedente [10], mostravano un'instabilità intorno al 5% (con un massimo del 7%) e un'unicità che si collocava tra il 15% e il 25% (senza mai scendere sotto il 12%). Il Δ in questo caso è uguale al 5%.

Table 4.36: Top 10 configurazioni con maggiore differenza globale (min uniqueness - max instability)

#	Configurazione	nq	nc	s	nr	Differenza	Guadagno(%)
1	uranus	8	20	20.000	5	0.227	354%
2	uranusTitania	10	10	20.000	5	0.218	336%
3	uranusTitania	8	10	20.000	5	0.217	334%
4	uranusTitania	10	20	20.000	5	0.204	308%
5	uranusTitania	8	20	20.000	5	0.198	296%
6	uranusTitania	8	20	20.000	5	0.198	296%
7	uranus	12	20	20.000	5	0.197	294%
8	uranusTitania	8	10	10.000	5	0.193	286%
9	uranusTitania	10	10	10.000	5	0.193	286%
10	uranusTitania	8	10	5.000	5	0.175	254%

Dalla classifica emerge che le configurazioni migliori, secondo questo criterio più conservativo, si concentrano prevalentemente sul sistema Uranus Titania, anche se Uranus continua a offrire risultati di rilievo. Si osserva inoltre che configurazioni con un numero elevato di challenge (nc) e di shot (s) tendono a produrre una differenza più marcata tra le metriche di uniqueness e instability, contribuendo a ridurre il rischio di collisioni nei casi limite. Rispetto allo studio originale abbiamo riscontrato che le prime 10 configurazioni hanno raggiunto risultati migliori.

Qui di seguito vengono mostrate le tre configurazioni che presentano le prestazioni migliori in termini di differenza tra unicità e instabilità.
Per ciascuna configurazione, sono stati analizzati i valori medi della distanza associati alle due metriche su un insieme di challenge. L'asse delle ascisse rappresenta l'indice della challenge (*n-th Challenge*), mentre l'asse delle ordinate riporta il valore medio della distanza calcolata.

1) Uranus-nq8-nc20-s20.000 Differenza minima = 0.227

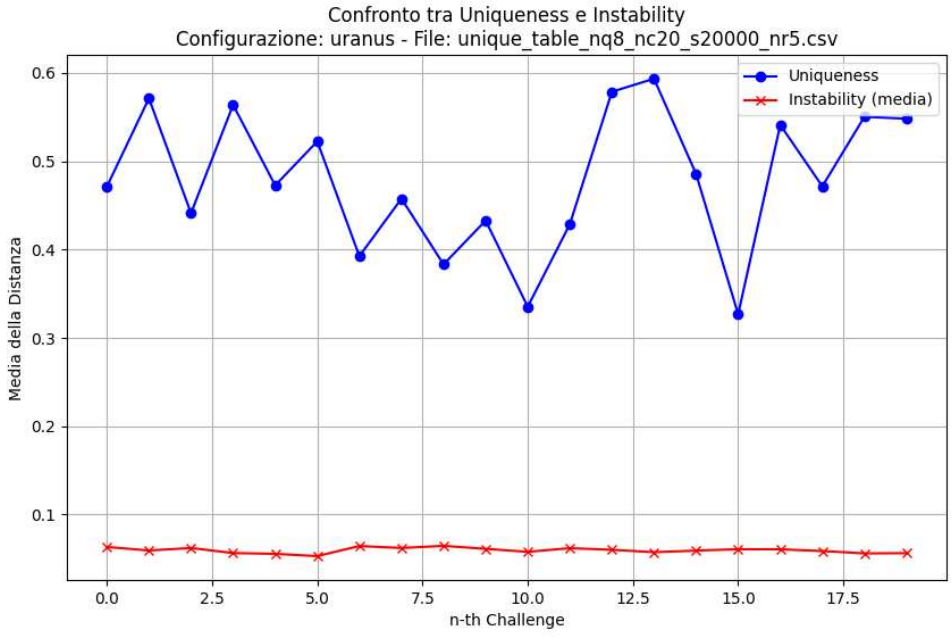

Fig. 4.35: Differenza delle metriche per Uranus nq8 - nc20 - s20.000

2) UranusTitania-nq10-nc10-s20.000 Differenza minima = 0.218

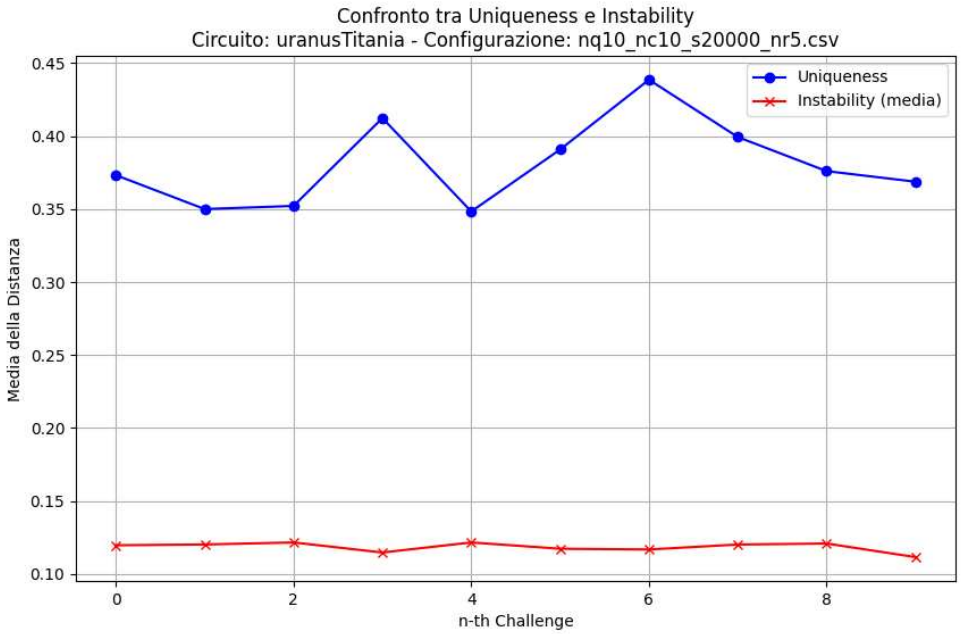

Fig. 4.36: Differenza delle metriche per Uranus Titania nq10 - nc10 - s20.000

3) UranusTitania-nq8-nc10-s20.000 Differenza minima = 0.217

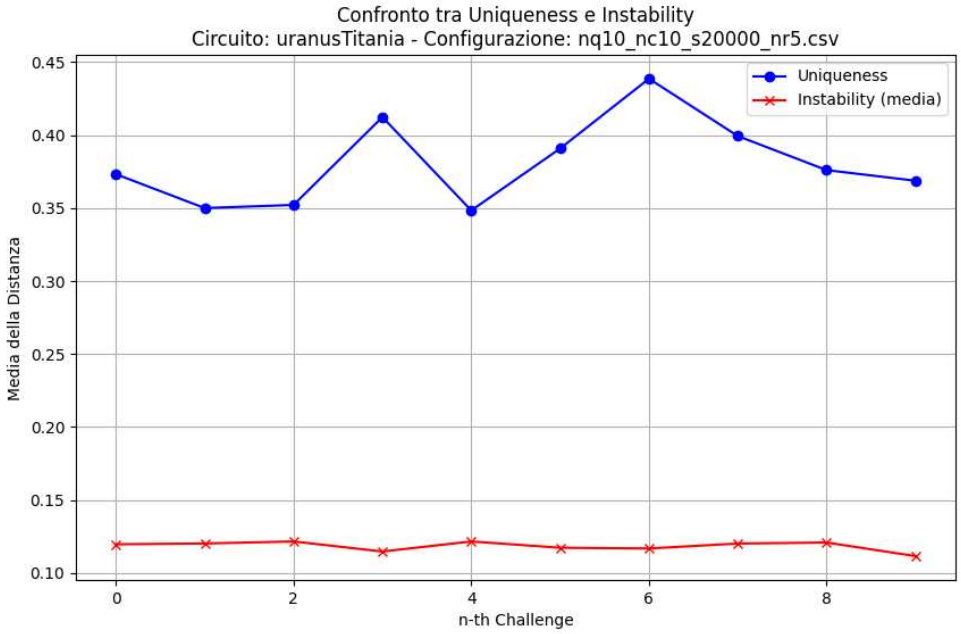

Fig. 4.37: Differenza delle metriche per Uranus Titania n8 - nc10 - s20.000

Dopo aver analizzato le tre configurazioni con le migliori prestazioni in termini di differenza tra unicità e instabilità, vengono di seguito presentate le configurazioni ottimali individuate per ciascun circuito realizzato riportando anche la posizione della classifica globale.

12) Saturn-nq8-nc20-s20.000 Differenza minima = 0.162

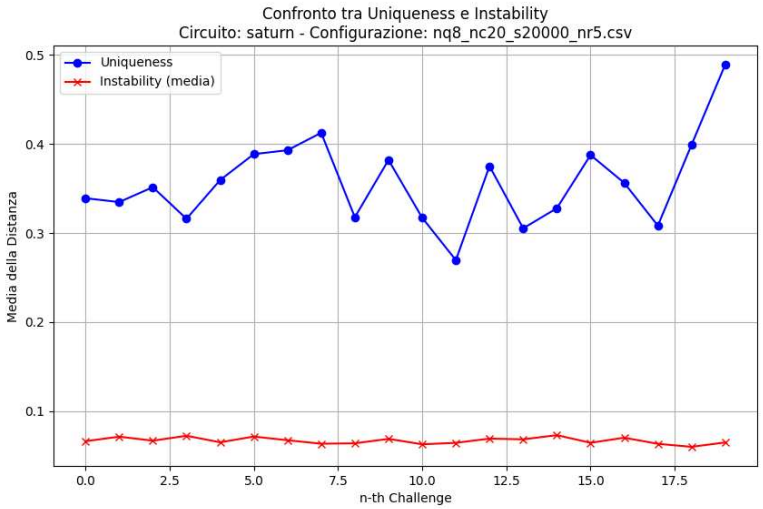

Fig. 4.38: Differenza delle metriche per Saturn nq8 - nc20 - s20.000

18) Venus-nq8-nc20-s20.000 Differenza minima = 0.134

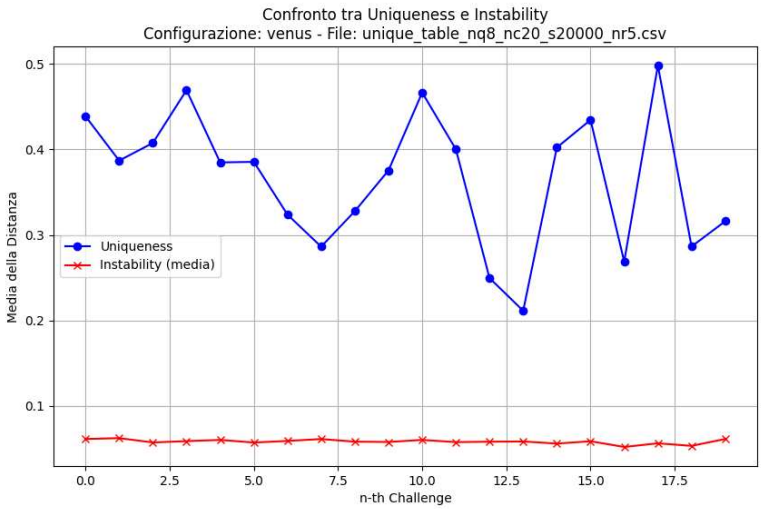

Fig. 4.39: Differenza delle metriche per Venus nq8 - nc20 - s20.000

20) Mars-nq8-nc20-s20.000 Differenza minima = 0.131

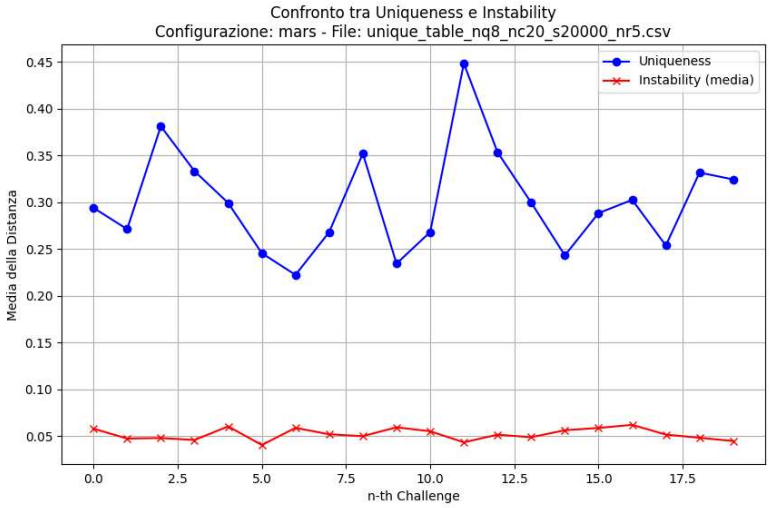

Fig. 4.40: Differenza delle metriche per Mars nq8 - nc20 - s20.000

31) Earth-nq10-nc20-s20.000 Differenza minima = 0.102

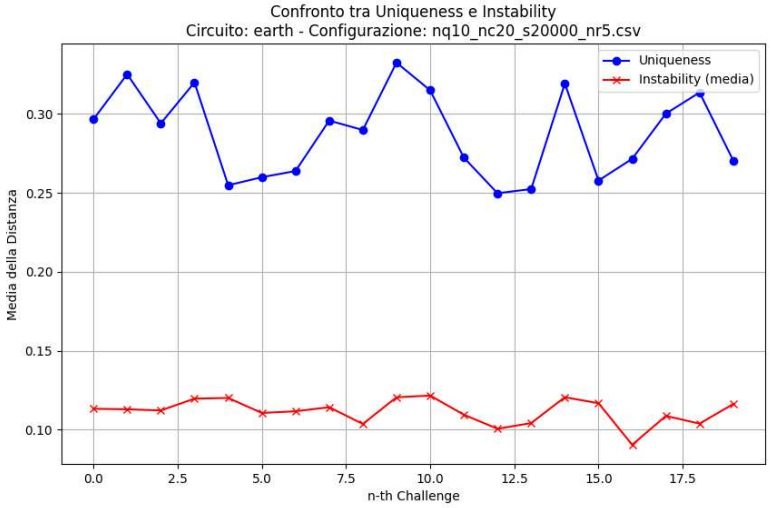

Fig. 4.41: Differenza delle metriche per Earth nq10 - nc20 - s20.000

79) Jupiter-nq12-nc10-s20.000 Differenza minima = -0.096

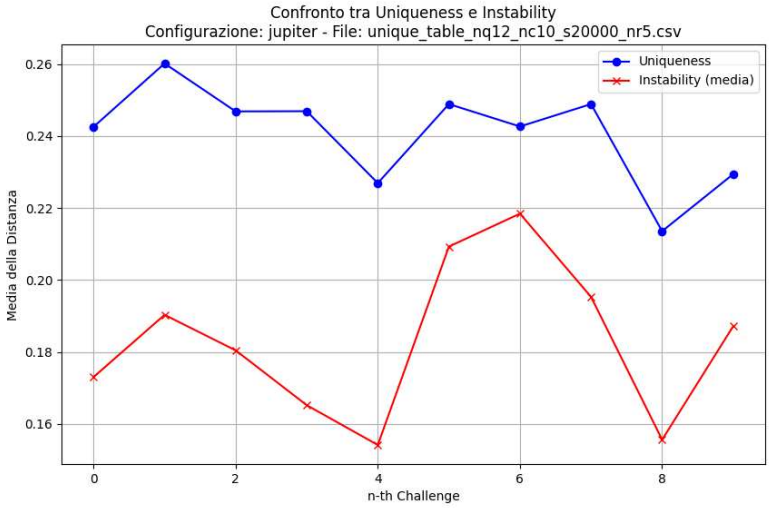

Fig. 4.42: Differenza delle metriche per Jupiter nq12 - nc10 - s20.000

87) Mercury-nq12-nc10-s10.000 Differenza minima = -0.224

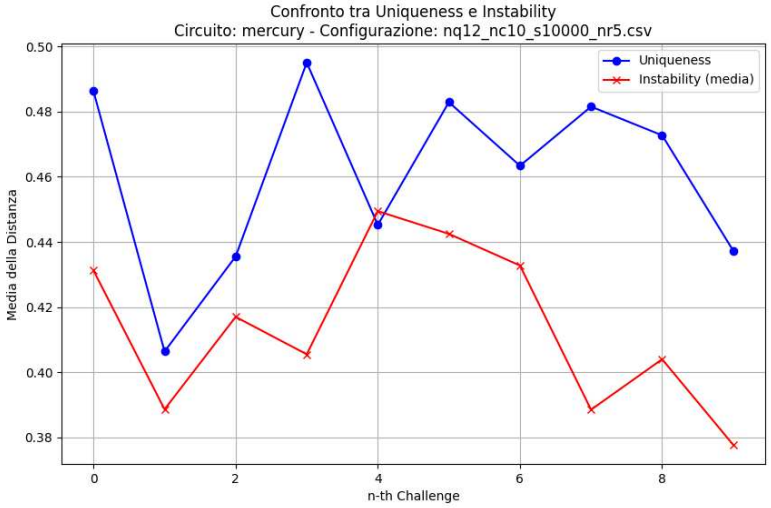

Fig. 4.43: Differenza delle metriche per Mercury nq12 - nc10 - s10.000

4.5 Effetto delle variazioni circuitali

Le implementazioni a 16 qubit (e superiori) sono state successivamente scartate a causa delle scarse performance rilevate e dei valori anomali riscontrati nelle tre metriche adottate per la valutazione: *instability*, *randomness* e *uniqueness*. In particolare, tali anomalie hanno evidenziato una marcata instabilità e una significativa perdita di affidabilità, rendendo queste configurazioni inadatte agli scopi del presente studio. Infatti le architetture più promettenti si sono rivelate quelle con 8, 10 e 12 qubit, le quali hanno mostrato un buon bilanciamento tra complessità circuitale, tempo di esecuzione e qualità delle risposte. È stato osservato che l'incremento del numero di *shots* comporta un lieve miglioramento della stabilità delle metriche, mentre la variazione del numero di *challenge* consente una valutazione più robusta della metrica di uniqueness.

L'aumento del numero di challenge da 10 a 20 ha prodotto risultati migliori in termini di capacità discriminante tra le risposte: si è osservata una divergenza più marcata tra le curve relative all'instability e alla uniqueness. Tuttavia, sarà opportuno verificare tale comportamento con un numero maggiore di esecuzioni o di misure per confermarne la consistenza.

Dal punto di vista circuitale, è emerso che l'aumento del numero di gate entanglement, in particolare l'uso estensivo del gate `CNOT`, ha un impatto positivo sulla metrica di uniqueness. Circuiti caratterizzati da una maggiore interconnessione tra i qubit risultano più efficaci nel generare risposte distinguibili tra diversi dispositivi. In tal senso, le architetture che seguono una disposizione a catena per i gate `CNOT`, simile alla struttura dell'ansatz `EfficientSU2`, hanno mostrato prestazioni superiori.

Capitolo 5

Considerazioni sulla sicurezza

L'autenticazione dei dispositivi rappresenta un elemento cruciale per garantire la fiducia e la sicurezza all'interno di architetture distribuite, specialmente in contesti in cui l'integrità dell'identità del dispositivo è un requisito per l'accesso a risorse critiche. In ambienti classici, questo processo è tradizionalmente supportato da certificati digitali, token o chiavi crittografiche.

Nei sistemi IoT, diversi protocolli di comunicazione wireless permettono un'interazione fluida tra dispositivi. Tuttavia, questi dispositivi sono vulnerabili a numerose minacce e attacchi informatici. Un accesso malevolo anche a un solo dispositivo può compromettere l'intera infrastruttura industriale, portando a malfunzionamenti, interruzioni di sistema o manipolazioni nei meccanismi di controllo e nei dati dei sensori.

Nel contesto del calcolo quantistico, lo sviluppo di protocolli di autenticazione basati su QPUF (Quantum Physical Unclonable Functions) rappresenta sia un'opportunità che una sfida.

5.1 Possibili attacchi a PUF e sistemi quantistici

Studi recenti [11][12] hanno evidenziato come alcune implementazioni convenzionali di PUF presentino vulnerabilità di sicurezza, suscettibilità alla clonazione e vulnerabilità ad attacchi basati su tecniche di machine learning, sollevando dubbi sulla loro affidabilità nelle applicazioni reali.

Anche se ci spostiamo nel contesto quantum possiamo trovare alcune criticità. Nel sistemi quantistici NISQ (Noisy Intermediate-Scale Quantum), attualmente accessibili tramite piattaforme cloud, emerge un problema critico legato alla qualità non uniforme dei dispositivi e alla loro disponibilità limitata. Queste condizioni possono incentivare comportamenti malevoli da parte dei fornitori di servizi, come la sostituzione non autorizzata del dispositivo utilizzato o la falsificazione delle sue caratteristiche operative.

Per affrontare questo rischio, è stato proposto un meccanismo di fingerprinting dinamico [13], in grado di verificare l'autenticità del dispositivo utilizzato durante l'esecuzione remota di un circuito quantistico.
Tuttavia, anche il metodo del fingerprinting può risultare vulnerabile in scenari ostili, poiché un fornitore non affidabile potrebbe individuare i circuiti utilizzati per l'analisi e manipolare le risposte ottenute, compromettendo così la validità del processo di verifica e l'integrità complessiva del sistema.

5.2 Protocollo di autenticazione QPUF

Il seguente protocollo descrive una procedura per implementare una **Quantum Physical Unclonable Function (QPUF)** su un dispositivo quantistico fornito tramite *provider cloud* (es. IBM Quantum). Il modello coinvolge tre attori principali:

- **Quantum Provider (QP):** il fornitore del servizio di calcolo quantistico (es. IBM, Amazon Braket, ecc.).
- **Client:** l'utente che richiede l'autenticazione del dispositivo.
- **Attestation Service (AS):** autorità indipendente che gestisce la registrazione e l'autenticazione del dispositivo.

Il protocollo è articolato in tre fasi: **Enrollment (Registrazione)**, **Attestation (Autenticazione)** e **Validazione**.

Fase 1: Enrollment – Registrazione del dispositivo
Durante la registrazione, il **Quantum Provider** invia all'**Attestation Service** l'identificativo del dispositivo da registrare.

1. L'**Attestation Service** genera un set di N challenge casuali $\{C_1, C_2, \ldots, C_N\}$. Ogni challenge definisce una configurazione di angoli di rotazione per un circuito quantistico.
2. Il **Provider** esegue i circuiti corrispondenti sul dispositivo fisico (es. tramite IBM Quantum) e restituisce gli output bitstring $\{R_1, R_2, \ldots, R_N\}$.
3. Le **Challenge-Response Pairs (CRP)** vengono memorizzate in un **database sicuro** gestito dall'**Attestation Service**.
4. Per evitare attacchi di precomputazione, l'**Attestation Service** introduce **casualità temporale** nell'invio delle challenge (random timing), rendendo impraticabile per il provider il caching dei risultati.

Fase 2: Attestation – Autenticazione del dispositivo

Questa fase verifica che le computazioni siano effettivamente eseguite su un **dispositivo autentico già registrato**.

1. Il **Client** richiede l'autenticazione per un certo dispositivo specificando il suo identificativo.
2. L'**Attestation Service** seleziona n challenge casuali (diversi da quelli usati in precedenti attestazioni).
3. Il **Client** inoltra le challenge al **Quantum Provider**, che le esegue sul dispositivo e restituisce i rispettivi n response.
4. Il **Client** invia all'**Attestation Service** le response ottenute, insieme alle challenge e all'ID del dispositivo.
5. L'**Attestation Service** confronta i nuovi response con le CRP archiviate nella fase di Enrollment, calcolando una **distanza media** (es. NAPD) tra le distribuzioni ottenute.

Fase 3: Validazione – Confronto con soglia di accettazione

- Se la **distanza media** è inferiore a una **soglia** λ predefinita, il dispositivo è **autenticato con successo**.
- La soglia λ viene determinata analizzando:
  - **Instability**: variazioni intra-dispositivo (stesso dispositivo, stesso challenge).
  - **Uniqueness**: variazioni inter-dispositivo (stesso challenge, dispositivi diversi).
- Nel contesto del nostro esperimento, un valore di soglia pari a $\lambda = 0{,}14$ risulta adeguato come punto di separazione efficace, considerando che l'*Instability* massima osservata è pari a 0,09 mentre la *Uniqueness* minima è pari a 0,32.

Schema del protocollo

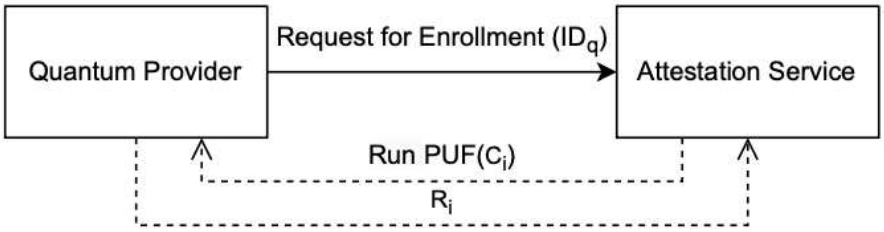

Fig. 5.1: Esempio di fase di Enrollment

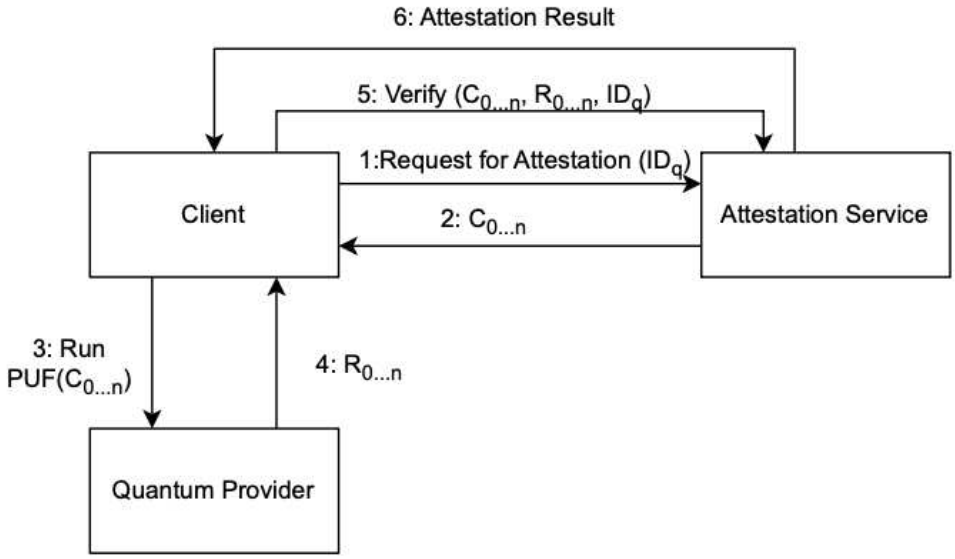

Fig. 5.2: Esempio di fase di Attestation

Sicurezza del protocollo

- Il sistema è **resiliente a manomissioni** da parte del provider: la randomizzazione delle challenge impedisce l'uso di CRP precomputate.
- L'utilizzo di **Quantum PUF** rende i response **dipendenti dall'hardware fisico**, impedendo la clonazione del dispositivo.

Capitolo 6

Conclusioni e sviluppi futuri

In conclusione, l'evoluzione delle tecnologie quantistiche e l'aumento dell'accessibilità a dispositivi NISQ (Noisy Intermediate-Scale Quantum) pongono le basi per un utilizzo sempre più concreto del calcolo quantistico in scenari reali e distribuiti. Tuttavia, questo progresso porta con sé nuove sfide in termini di sicurezza, autenticazione e integrità dei dispositivi e delle comunicazioni. In questo contesto, i Quantum Physical Unclonable Functions (QPUF) si configurano come una soluzione innovativa per rafforzare la sicurezza delle infrastrutture (come quelle IoT), grazie alla loro capacità di generare impronte digitali hardware uniche e non clonabili.

Il lavoro svolto ha dimostrato che le QPUF possono essere impiegate per realizzare meccanismi di autenticazione affidabili e compatibili con i limiti tecnologici attuali dei dispositivi NISQ, senza necessità di canali o memorie quantistiche persistenti. I risultati ottenuti superano i valori riportati in studi precedenti, confermando la validità dell'approccio adottato e aprendo nuove prospettive di sviluppo.

Tuttavia, esistono sicuramente margini di miglioramento, sia a livello progettuale che parametrico, che potrebbero ulteriormente aumentare la robustezza e la resistenza dei rusultati ottenuti. Guardando al futuro, con l'espansione dell'ecosistema quantistico e l'adozione sempre più diffusa di dispositivi quantistici su scala globale, le QPUF potranno giocare un ruolo chiave nella definizione di nuovi standard di sicurezza, abilitando l'autenticazione e la protezione dei nodi nell'Industria IoT.

6.1 Limiti della ricerca

Uno dei principali limiti incontrati durante la presente ricerca è stato rappresentato dalle restrizioni imposte dal provider IBM, in particolare la limitazione di tempo di utilizzo a 10 minuti mensili per ciascun account. Questa condizione ha fortemente vincolato le possibilità di sperimentazione, limitando il numero di esecuzioni (run) e rendendo di fatto impossibile l'esecuzione di test su larga scala per singole configurazioni.

Nel dettaglio, è stato osservato che superando la soglia dei 20.000 *shots* per ciascuna delle 5 run previste, i tempi di esecuzione per macchina richiedevano un intervallo minimo stimato tra i 12 e i 13 minuti. Tale durata eccede significativamente la quota mensile disponibile per un singolo account, rendendo impraticabile la raccolta di dati ad alta risoluzione anche solo per una singola configurazione circuitale quantistica.

Questa limitazione ha imposto la necessità di operare con compromessi metodologici, selezionando subset ridotti di configurazioni e limitando il numero complessivo di *shots* e *runs* per mantenersi entro i limiti temporali disponibili. Di conseguenza, pur garantendo una base solida per l'analisi comparativa tra le varie configurazioni, non è stato possibile esplorare in maniera esaustiva scenari di maggiore complessità o raccogliere dati statistici più approfonditi, il che rappresenta un margine di miglioramento per lavori futuri.

6.2 Possibili sviluppi futuri

Alla luce delle limitazioni riscontrate durante la sperimentazione, diversi spunti si prestano ad essere esplorati in lavori futuri per approfondire e potenziare l'analisi condotta.

Una prima direzione riguarda l'utilizzo di altri provider di calcolo quantistico che offrano maggiore disponibilità di tempo di esecuzione o un accesso più flessibile alle risorse hardware. Tra questi si possono citare Amazon Braket, Rigetti, Google Quantum AI e Azure Quantum, che offrono ambienti di esecuzione diversi da quelli adottati in questo lavoro. Il loro utilizzo permetterebbe di testare le soluzioni in contesti eterogenei e confrontare in modo più accurato le diverse infrastrutture quantistiche.

Un secondo ambito di approfondimento consiste nell'estendere il tempo dedicato a ciascuna configurazione, aumentando sia il numero di *shots* che quello di *runs*, al fine di valutare se le metriche di instabilità, casualità e unicità mostrano miglioramenti o convergenze significative con una base statistica più ampia. Questo tipo di indagine potrebbe chiarire meglio i limiti intrinseci delle configurazioni testate e fornire indicazioni più affidabili per applicazioni pratiche.

Infine, potrebbe essere opportuno considerare l'integrazione di tecniche di post-processing o algoritmi di ottimizzazione dei parametri per migliorare ulteriormente la qualità delle risposte ottenute, specialmente nei casi borderline dove le metriche risultano difficili da interpretare. Anche l'impiego di simulatori quantistici ad alta precisione in grado di simulare fedelmente il rumore di una specifica QPU, sebbene non perfettamente equivalenti all'hardware reale, potrebbe supportare l'analisi in fase preliminare e guidare la scelta delle configurazioni da testare effettivamente sui dispositivi quantistici reali, permettendo così un significativo risparmio di tempo di esecuzione su tali dispositivi.

Bibliografia

[1] Christopher Z Chwa, Leleia A Hsia, and Laurence D Merkle. "Quantum Crosstalk as a Physically Unclonable Characteristic for Quantum Hardware Verification". In: *NAECON 2023-IEEE National Aerospace and Electronics Conference*. IEEE. 2023, pp. 309–313.

[2] Jalil Morris et al. "Fingerprinting quantum computer equipment". In: *Proceedings of the Great Lakes Symposium on VLSI 2023*. 2023, pp. 117–123.

[3] Prasanna Ravi, Anupam Chattopadhyay, and Shivam Bhasin. "Security and quantum computing: An overview". In: *2022 IEEE 23rd Latin American Test Symposium (LATS)*. IEEE. 2022, pp. 1–6.

[4] Sreeja Chowdhury et al. "Physical security in the post-quantum era: A survey on side-channel analysis, random number generators, and physically unclonable functions". In: *Journal of Cryptographic Engineering* (2021), pp. 1–37.

[5] Simon J Devitt, William J Munro, and Kae Nemoto. "Quantum error correction for beginners". In: *Reports on Progress in Physics* 76.7 (2013), p. 076001.

[6] William K Wootters and Wojciech H Zurek. "A single quantum cannot be cloned". In: *Nature* 299.5886 (1982), pp. 802–803.

[7] Venkata KVV Bathalapalli et al. "Qpuf: Quantum physical unclonable functions for security-by-design of industrial internet-of-things". In: *Cryptography* 9.2 (2025), p. 34.

[8] Venkata KVV Bathalapalli et al. "QPUF 2.0: Exploring Quantum Physical Unclonable Functions for Security-by-Design of Energy Cyber-Physical Systems". In: *2025 IEEE 26th International Symposium on a World*

of Wireless, Mobile and Multimedia Networks (WoWMoM). IEEE. 2025, pp. 281–286.

[9] Franco Cirillo and Christian Esposito. “A QPUF-Based Scheme for Secure and Adaptable Quantum Device Attestation in NISQ Devices”. In: *2025 International Conference on Quantum Communications, Networking, and Computing (QCNC)*. IEEE. 2025, pp. 117–121.

[10] Franco Cirillo and Christian Esposito. “Practical Evaluation of a Quantum Physical Unclonable Function and Design of an Authentication Scheme”. In: *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 1. IEEE. 2024, pp. 1354–1363.

[11] Nils Wisiol et al. “Neural network modeling attacks on arbiter-PUF-based designs”. In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 2719–2731.

[12] Trevor Kroeger et al. “Cross-PUF attacks on arbiter-PUFs through their power side-channel”. In: *2020 IEEE International Test Conference (ITC)*. IEEE. 2020, pp. 1–5.

[13] Jindi Wu, Tianjie Hu, and Qun Li. “Detecting Fraudulent Services on Quantum Cloud Platforms via Dynamic Fingerprinting”. In: *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. 2024, pp. 1–8.