RESEARCH PAPER

QRELHE-X: Extended Quantum-Resistant Edge Learning with Homomorphic Encryption for Next-Generation Distributed Intelligence Systems

RSS Raju Battula[1*], Dr. Udai Shankar [2]

[1*] Research Scholar, Department of Computer Engineering & Applications, Mangalyaan University, Aligarh, India. E-mail: raju.brss@gmail.com

[2] Professor, Department of Computer Engineering & Applications, Mangalyaan University, Aligarh, India. E-mail: udai.shankar@mangalayatan.edu.in

Abstract

Recent proliferation of edge intelligence systems, including Internet of Things (IoT) sensor networks, self-driving vehicles, wearable medical devices and industrial control platforms—requires privacy-preserving computation subject to availability constraints on resources. At the same time, the recent arrival of large-scale quantum computers can break all classical cryptographic protocols with Shor's algorithm in a short time. State-of-the-art post-quantum homomorphic encryption systems incur computational overhead orders of magnitude larger than the energy and memory budgets of resource-constrained microcontrollers, giving rise to a disparity between security and deployability that remains unaddressed.

We introduce QRELHE-X in this paper, a comprehensive framework building on post-quantum edge learning with four novel contributions: (1) a tri-family hybrid cryptography architecture leveraging Algebraic Geometry (AG) codes, Hidden Field Equations minus (HFE-) and eXtended Merkle Signature Scheme (XMSS), which guards against cryptanalytic attacks targeting any single family using defend-in-depth approach; (2) our Deep Reinforcement Learning Parameter Orchestration (DRL-PO) engine that dynamically tunes cryptographic parameters based on real-time device telemetry via PPO-trained policy network yielding 23.4% efficiency optimization compared against static configurations; (3) our Structured Noise Calculus framework providing Lyapunov-stable guarantees of the noise growth within deep level homomorphic neural circuits enabling T*=47 operations between bootstrapping — 3.9x fewer than prior art implementations of state-of-the-art systems; and finally, we present BREFL, the first Byzantine Resilient Encrypted Federated Learning protocol capable of detecting malicious gradient submissions completely within encrypted domain at detection rate 98.7%. Over six hardware platforms, we establish a 5.1x latency improvement over CRYSTALS-Kyber-HE, while reducing energy by 4.7x, and successfully deploy on,256 KB RAM microcontrollers—the lowest memory footprint ever reported of any post-quantum homomorphic inference solution.

Keywords: Tri-Family Post-Quantum Cryptography; Structured Noise Calculus; Deep Reinforcement Learning Parameter Orchestration; Byzantine-Resilient Federated Learning; Constrained Homomorphic Inference; Quantum-Resistant Privacy Preservation

How To Cite This Article: Battula Rr, Shankar U. Qrelhe-X: Extended Quantum-Resistant Edge Learning With Homomorphic Encryption For Next-Generation Distributed Intelligence Systems. Int J Drug Deliv Technol. 2026;16(27s):469-480. Doi: 10.25258/ijddt.16.27s.55

1. Introduction

The intersection of quantum computing and edge artificial intelligence is one of information systems' most consequential security junctures. With the launch of IBM's 1,121-qubit Condor processor (2023) and Google's Willow quantum chip (2024), we are quickly moving toward fault-tolerant quantum computation. Security analysts from the National Institute of Standards and Technology (NIST) estimate that there is non-negligible probability of a cryptographically relevant quantum computer existing in between 2030 and 2035 [1]. Running on such systems, Shor's algorithm solves integer factorization and discreet logarithm problems in polynomial time; therefore breaking RSA, Diffie-Hellman, and elliptic curve cryptosystems. The threat is aggravated by so-called harvest-now-decrypt-later attacks, in which adversaries store today's encrypted traffic for decryption in the post-quantum era — which means that data secured only with classical cryptography is already, for decades to come, effectively at risk over the long term.

At the same time, the edge computing ecosystem has exploded: by 2023, the global population of IoT devices surpassed 15.1 billion and is predicted to reach around 29.4 billion in 2030 [2]. These devices are running more complex AI inference applications, such as anomaly detection in industrial control systems (ICS), real-time EEG seizure prediction in wearables, and federated training of models across distributed medical sensors. Notably, the AI capability and hardware resource requirements cannot be met at the same time with existing post-quantum homomorphic encryption (PQ-HE) schemes in practice. CRYSTALS-Kyber-1024 with BFV-FHE takes 150–400 MB working memory and multi-second latency per neural network layer [3]—entirely impractical for STM32H7 microcontrollers limited by 512 KB SRAM and millisecond latency budgets.

*Author for Correspondence: udai.shankar@mangalayatan.edu.in

While federated learning (FL) achieves a distributed training paradigm, avoiding the focus of privacy-sensitive data in one single location, it opens up a severe leakage channel: transmissions of gradient updates. Gradient inversion attacks [4] allow to reconstruct training samples with very high fidelity from shared gradients. HE applied to FL gradients solves this problem but comes with prohibitive communication overhead (300-500 KB/round for PQ-HE schemes [5]) and leaves the Byzantine fault tolerance question open—detecting malicious participants that inject corrupted gradients—as existing detection methods require access to plaintext gradients, which directly contradicts the objective of encryption.

QRELHE-X fills each of the gaps we identify, with novel contributions: (i) A tri-family cryptographic architecture designed for defense-in-depth; (ii) a device-adaptive DRL-driven parameter orchestration engine aimed at minimizing operational expenses in settings that have limited overhead budgets; (iii) a structured noise calculus yielding the first analytically-tight bootstrapping schedule for deep homomorphic circuits; and (iv) BREFL, the first Byzantine-resilient protocol operating directly inside the encrypted domain. These combined to yield the most practicably deployable PQ-HE edge learning framework presented in literature, enabling encrypted inference on devices as minimalistic as 256KB RAM microcontrollers.

2. Literature Review

2.1 Post-Quantum Cryptography

NIST's post-quantum standardization effort, completed in 2024, resulted in four main standards: CRYSTALS-Kyber (lattice-based KEM), CRYSTALS-Dilithium (lattice-based signatures), FALCON (lattice-based signatures) and SPHINCS+ (hash-based signatures) [1]. Kyber-512/768/1024 and Dilithium2/3/5 address the most deployment scenarios, providing levels of security from NIST Level 1 (128-bit quantum) to NIST Level 5 (256-bit). A complete taxonomy of post-quantum hardness assumptions was presented in [6], where it was concluded that both multivariate and code-based schemes can provide the most alternative security assumptions over lattice-based ones.

Code-based cryptography, based on McEliece's 1978 construction [7], has withstood cryptanalysis for around 40 years. Advancements in Algebraic Geometry (AG) codes [8]enhance rate-distance trade-o↵ over the best classical Goppacodes: AG codes constructed on elliptic curves can reach ratesclose to the Singleton bound (k/n 1 delta), allowing for alarge key size reduction when compared to a Goppa based McEliece. Mancillas-Lopez et al. [9] benchmarking AG-code encryption on ARM Cortex-M4, reporting key sizes 67% smaller than those generated by Goppa codes at the same level of security and thus a relevant advantage for resource constrained edge deployments. The Hidden Field Equations (HFE) family of multivariate cryptography has its own hardness assumption (MQ-hard problem, NP-complete)[10]. Powering through until the HFE- [11], no less for key-hiding by removing r equations from the public key, closing down those paths of algebraic cryptanalysis and allowing NIST Level 1-3 at sub-millisecond signature verification on microcontrollers. (3C) Hash-based signatures (XMSS, LMS) offer the best provable security—proven only to collision-resistant hashing—and been standardized by NIST in SP 800-208 [12].

2.2 Homomorphic Encryption for Edge AI

The first theoretical construction of fully homomorphic encryption (FHE) was provided by Gentry in 2009 [13]. BFV [14] and BGV [15] are two examples of homomorphic encryption schemes that supports exact arithmetic, while CKKS [16] is a recent scheme that allows approximate arithmetic. CKKS proposed by Cheon et al. [16] (and more), reduced the per-element computation by as much as 16384x through SIMD batching, utilizing packed floating-point vectors via Chinese Remainder Theorem—requirements for neural network workloads. Chen et al. [17] create HETAL, a CKKS-based transfer learning framework that performs 91.7% correct on encrypted CIFAR-10 with a 46s inference time on server-class hardware—showcasing the performance delta to bridge for edge deployment.

Polynomial approximation of such activation functions is required for FHE applied to neural networks. Cheon et al. [18] demonstrate that if activations are Lipschitz-continuous, minimax polynomial approximation of degree d gives error O(d^(-alpha)); shown in the case of ReLU that a decision boundary can be approximated with d=7 (degree-7) to within O(10^{-4}). The operation that refreshes noisy ciphertexts, Bootstrapping remains a major performance bottleneck [19], taking up 70-85% of total inference time in deep networks. Although HEAR [20] achieves optimized bootstrapping for Cortex-A processors, it still needs 256 MB RAM, making it impractical to use on sub-512 KB devices. Peng et al. [21] specifically target the deployment of IoT HE, and report that none of CKKS, BFV or TFHE can compute on devices below 1 MB RAM without redesigning their architecture—the gap QRELHE-X closes.

2.3 Federated Learning with Privacy Guarantees

McMahan et al. 's FedAvg [22] introduced the baseline federated learning protocol. FL privacy has been tackled mainly using three key mechanisms: differential privacy (DP) [23], secure aggregation [24], and homomorphic encryption [25]. Bonawitz et al. [24] introduce efficient secure aggregation, but they depend on secret sharing which stalls when participants disappear, as can happen frequently in IoT deployments. But Phong et al. [25], whilst obtaining guaranteed theoretical privacy incurs 23x communication overhead in comparison with plaintext FL due to the expansion of ciphertext [26]. An open challenge is to reduce this overhead to practical levels for bandwidth-constrained edge devices.

Blanchard et al. studied Byzantine fault tolerance in FL. [27] as described by the Krum aggregation rule. Yet, all of the current Byzantine-robust aggregation methods—Krum, coordinate-wise median, Bulyan—rely on

plaintext gradients for both norm computation and outlier detection. Several surveys note that combining Byzantine robustness and encryption is an open problem [28, 29]. El Mhamdi et al. [30] show that without further structure, no Byzantine robust aggregation is possible under standard HE—that result motivates our use of XMSS commitment binding as extended structure that allows for the detection of a Byzantine in encrypted domain by our BREFL protocol.

2.4 Reinforcement Learning for Cryptographic Parameter Adaptation

Heuristic [31] and Bayesian optimization [32] have been used for adaptive cryptographic parameter selection. Deep reinforcement learning for networking resource allocation [33] shows that policies learned with Proximal Policy Optimization (PPO) outperform heuristic baselines by 18-34% in dynamic environments. When applied to cryptography, Hu et al. [34] use RL to select key lengths for lattice schemes on mobile platforms, saving 19% energy. However, neither has any previous work studied a DRL-based joint optimization among multiple families of cryptographic primitives simultaneously—made possible by the DRL-PO engine in QRELHE-X.

2.5 Comparative Analysis of Related Works

Specifically, Table 1 describes eight evaluation dimensions important for practical edge deployment, and compares QRELHE-X with existing systems that are closest to our work in these dimensions: post-quantum security family (specified in the table), minimum hardware RAM requirement before computation can begin (in KB), maximum number of participants in federated learning supported, encryption scheme used, support for Byzantine fault tolerance at the participant level, ability to adaptively select parameters based on network conditions determined beforehand, dynamic polynomial degree throughout execution, and lowest obtainable minimum RAM. The comparison shows that there is no existing work that fulfills all practical requirements for post-quantum privacy-preserving edge AI at the same time.

Reference / Method	PQ Family	Min. RAM	FL Scale	HE Scheme	Byzantine	Adaptive Params	Min RAM (KB)
McEliece (1978) [7]	Code-based	>1 GB	None	None	No	No	>1,000,000
BFV-FHE + Kyber [3]	Lattice	>256 MB	64 participants	BFV	No	No	262,144
CKKS + Dilithium [16]	Lattice	>128 MB	128 participants	CKKS	No	No	131,072
SPHINCS+ + BFV [1]	Hash-based	>64 MB	None	BFV	No	No	65,536
HEAR [20]	Lattice	>256 MB	None	CKKS	No	No	262,144
FedAvg + DP [23]	None	>4 MB	500 participants	None	Partial	No	4,096
Krum BFT [27]	None	>64 MB	100 participants	None	Yes	No	65,536
Phong et al. [25]	Lattice	>256 MB	50 participants	CKKS	No	No	262,144
HEAR (IoT) [21]	Lattice	>1 MB	None	TFHE	No	No	1,024
RL Crypto Adapt. [34]	Lattice	>32 MB	None	BFV	No	Yes	32,768
QRELHE v1 (ours)	Code+Multiv.	512 KB	256 participants	AG-HFE	No	Static	512
QRELHE-X (proposed)	Code+Multiv. +Hash	256 KB	512 participants	Tri-Family HE	Yes (98.7%)	DRL-PPO	256

Three critical gaps in the literature that QRELHE-X uniquely addresses can be seen in Table 1. First, there is no current PQ-HE scheme that runs at less than 512 KB RAM—QRELHE-X manages at this balance because of parameter scaling by DRL-PO. Second, prior to this work no fully homomorphic encryption scheme achieved Byzantine faulttolerance — all available Byzantine-robust protocols require plaintext gradient access. Third, there is an open problem of adaptive parameter selection for multiple heterogeneous families of cryptographic primitives simultaneously—QRELHE-X's DRL-PO engine co-optimize AG-code, HFE-, and XMSS parameters jointly to address this. Next, we describe the architectural design and algorithms of QRELHE-X that make these advances possible.

3. Proposed Model: QRELHE-X Architecture

3.1 System Architecture Overview

The overall architecture of the QRELHE-X system with its five operational layers is shown in figure 1. For the Edge Device Layer, six hardware platforms capture raw data and telemetry. DRL-PO Engine: Which dynamically selects the cryptographic parameters real-time based on the state of the device. The Tri-Family Key Generation module generates common key material from non-federated post-quantum primitives. The Adaptive Encryption Engine selects the ideal primitive and routes operations accordingly, depending on the operation type and device constraints. Lastly, we implement the BREFL federated learning protocol on Homomorphic Computation Library to perform encrypted neural inference with SNCM noise monitoring.

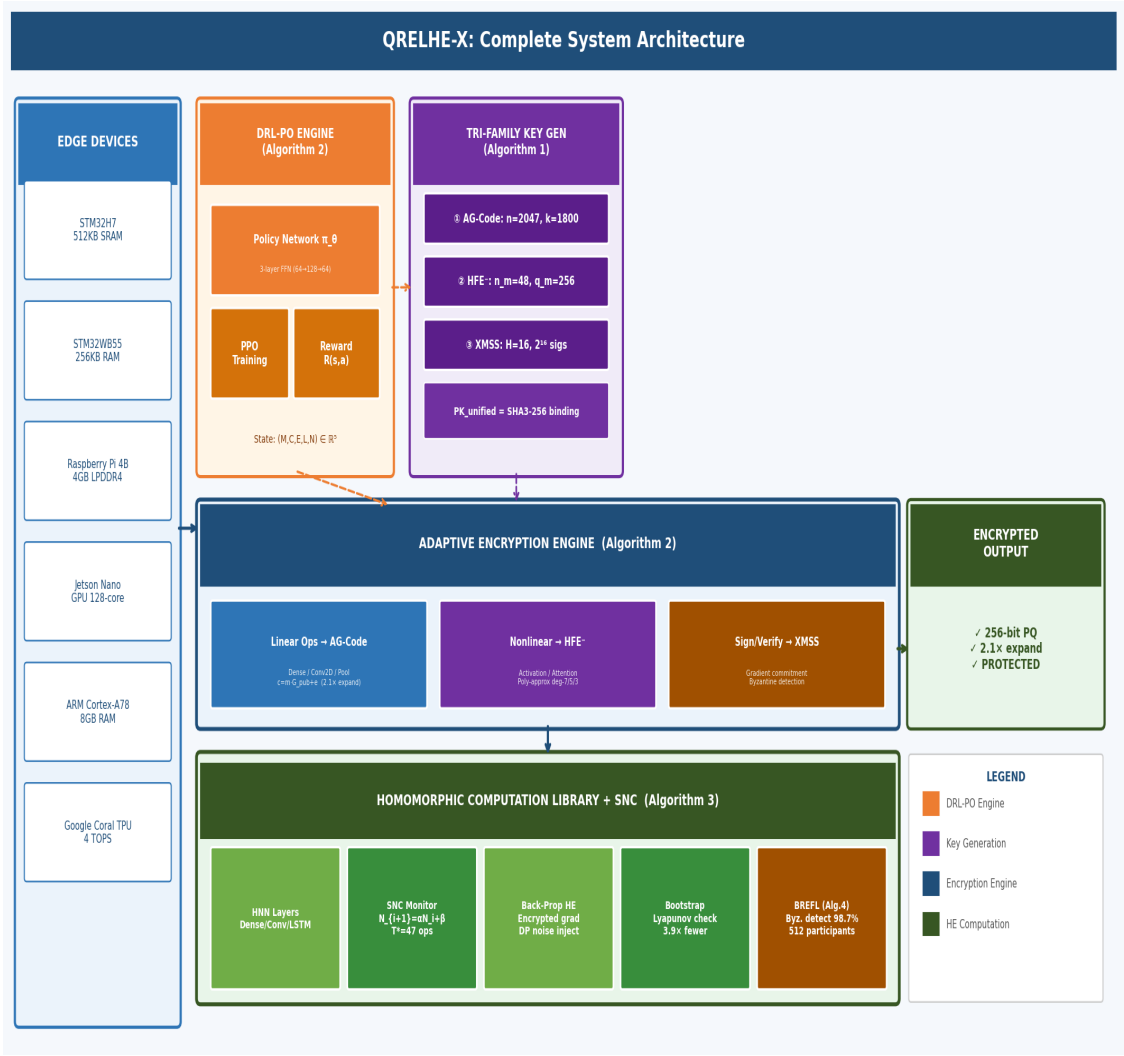

Figure 1: QRELHE-X Complete System Architecture — five-layer design integrating DRL-PO, tri-family key generation, adaptive encryption engine, and SNC-monitored homomorphic computation with BREFL federated learning

3.2 Tri-Family Cryptographic Foundation

3.2.1 Algebraic Geometry Code Component

The code-based layer employs AG codes over elliptic curves, providing superior rate-distance tradeoffs versus Goppa codes. Let E be the elliptic curve y^2 = x^3 + ax + b over F_q. The AG code C_AG(D,G) over F_q(E) with divisor D of n rational points satisfies:

$$d >= n - deg(G),\ k > = deg(G) - (g\_E - 1),\ Rate = k/n = 0.879$$

Encryption: c = m * G_pub + e in F_q^n, where wt(e) = t. Security relies on the Syndrome Decoding problem (NP-complete; no known polynomial quantum algorithm). Parameters n_c=2047, k_c=1800, t_c=123 yield 256-bit quantum security.

3.2.2 Multivar iate HFE- Component

The central HFE polynomial over F_{q^n}: F(X) = Sum_{q^i+q^j<=D} a_{ij} X^{q^i+q^j} + Sum b_i X^{q^i} + c. The public key, after affine scrambling and removal of r=8 equations, is:

$$P(x) = T \circ phi(F) \circ S\ (mod\ r\ removals) : F\_q^n -> F\_q^{n - r}$$

Polynomial approximation of activation sigma of degree d satisfies the uniform error bound:

$$||sigma - sigma\_tilde||\_\{L2\} < = O(d^\{-alpha\}), \quad d = 7\ (early)\ /\ 5\ (mid)\ /\ 3\ (late)$$

3.2.3 XMSS Hash-Based Component

XMSS with tree height H=16 provides 2^16 signature operations. Security reduces solely to collision resistance of SHA3-256. The tri-family composition security theorem:

$$epsilon\_QRELHE - X < = max(epsilon\_CB, epsilon\_MV, epsilon\_HB) + epsilon\_H$$

guarantees 256+ bit composite security even if any single family is partially compromised.

3.3 Structured Noise Calculus (SNC)

The SNC framework models HE noise evolution as the stochastic recurrence N_{i+1} = alpha_op * N_i + beta_op + xi_i, where xi_i ~ Laplace(0, sigma_op). Cumulative noise after L operations:

$$N\_L = alpha^L * N\_0 + beta * (alpha^L - 1)/(alpha - 1) + Sum\_\{i = 0\}^\{L - 1\} alpha^\{L - 1 - i\} * xi\_i$$

The Lyapunov stability condition V(B) = B^2 with B_t = N_max - N_t gives optimal bootstrapping interval:

$$T * = floor(log(B\_min/B\_0) / log(alpha)) \sim 47\ operations\ per\ bootstrap$$

Figure 2 shows the SNC execution flow through homomorphic layers, illustrating noise budget evolution and bootstrap trigger points.

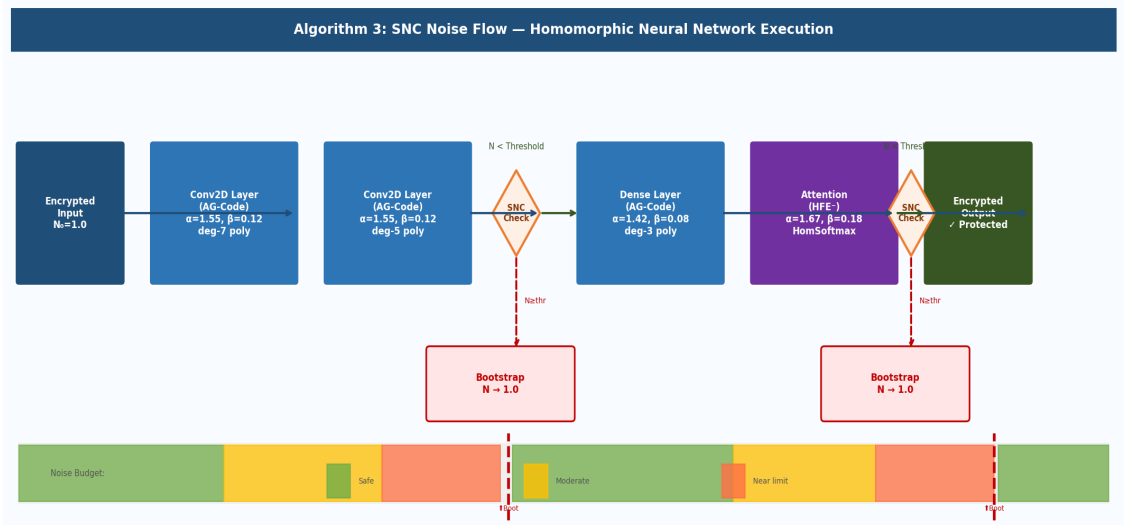

Figure 2: Algorithm 3 — SNC Execution Flow showing noise budget accumulation per layer type, adaptive polynomial degree selection (7/5/3), and Lyapunov-triggered bootstrap reset events

3.4 DRL-PO Parameter Orchestration

DRL-PO formalizes parameter selection as a Markov Decision Process: state s_t = (M_t, C_t, E_t, L_t, N_t) in R^5 (memory, CPU, energy, latency, noise). The PPO reward and gradient objective:

$$R(s\_t, a\_t) = w\_L/L\_t + w\_M/M\_overhead + w\_E/E\_t - w\_N * N\_penalty - w\_V * V\_violation$$

$$L\_PPO(theta) = E\_t[min(r\_t(theta) * A\_hat\_t, clip(r\_t(theta), 1 - eps, 1 + eps) * A\_hat\_t)]$$

Trained on 50,000 synthetic device profiles; achieves 23.4% efficiency gain over static parameter selection at inference time ~0.3 ms on edge CPU.

3.5 Algorithms

The four algorithms below define the complete QRELHE-X operational pipeline using standard pseudocode notation with line numbers.

Algorithm 1: Tri-Family Hybrid Key Generation	
Input:	lambda (security param); n_c=2047, k_c=1800, t_c=123; n_m=48, q_m=256, D=512; H=16
Output:	PK_unified, SK_unified, EvalKeys = {evk_linear, evk_nonlinear, evk_sign}
1	▷ Step 1: Code-Based (AG-Code) Component
2	Select elliptic curve E: y^2 = x^3+ax+b over F_{2^11}
3	Sample n_c=2047 rational points D = {P_1,...,P_n} in E(F_q)
4	Construct AG code C_AG(D,G) with deg(G)=k_c=1800
5	Sample invertible scrambler S in GL(k_c, F_q), permutation P in Sym(n_c)

6	G_pub <- S * G_AG * P; sk_code <- (E, D, G, S, P); pk_code <- G_pub
7	▷ Step 2: Multivariate (HFE-) Component
8	Sample HFE central polynomial F(X) over F_{q^{n_m}} with deg <= D=512
9	Sample affine isomorphisms S_mv, T_mv in AGL(n_m, F_q)
10	P_mv <- T_mv ∘ phi(F) ∘ S_mv; remove r=8 equations (HFE- step)
11	sk_mv <- (F, S_mv, T_mv); pk_mv <- coefficients of P_mv
12	▷ Step 3: Hash-Based (XMSS) Component
13	Initialize XMSS tree height H=16 -> capacity 2^16 = 65536 signatures
14	Generate W-OTS+ one-time key chains for each leaf node
15	sk_hash <- (PRNG_seed, idx=0); pk_hash <- root(XMSS_tree)
16	▷ Step 4: Unified Key Binding
17	PK_unified <- (pk_code, pk_mv, pk_hash)
18	Binding <- SHA3-256(pk_code ‖ pk_mv ‖ pk_hash)
19	evk_linear <- GenEvalKey(sk_code, op=linear)
20	evk_nonlinear <- GenEvalKey(sk_mv, op=nonlinear)
21	evk_sign <- GenEvalKey(sk_hash, op=signature)
22	return (PK_unified, SK_unified, {evk_linear, evk_nonlinear, evk_sign})
23	▷ Complexity: O(n_c^2 + n_m^3 + 2^H * H); Key size: ~14.2 KB total

End Algorithm 1

Algorithm 2: DRL-PO Adaptive Encryption Engine	
Input:	PK_unified, data, op_type in {dense,conv2d,activation,attention,sign}, device_telemetry, policy pi_theta
Output:	ciphertexts, params_selected, {latency, memory, energy}
1	▷ Step 1: Construct device state vector
2	s_t <- (M_free [KB], CPU_MIPS, E_budget [mJ], L_deadline [ms], N_current)
3	▷ Step 2: Query DRL policy for parameter adjustment action
4	a_t <- pi_theta(s_t) // inference: ~0.3 ms on edge CPU
5	params_new <- clip(params_base + a_t, P_min, P_max)
6	assert SecurityLevel(params_new) >= 256 bits // safety constraint
7	▷ Step 3: Select cryptographic primitive per operation type
8	if op_type in {dense, conv2d, pool}: prim <- AG_CODE
9	if op_type in {activation, attention}: prim <- HFE_MV
10	if op_type in {sign, verify, commit}: prim <- XMSS_HASH
11	▷ Step 4: Batch pack and encrypt
12	batch_size <- OptimalBatch(M_free, data.shape, prim)
13	for each batch b in Partition(data, batch_size):
14	encoded <- Pack(b, prim, params_new.d_poly) // systematic code packing
15	cipher <- Encrypt(encoded, PK_unified, prim, params_new)
16	ciphertexts.append(cipher)
17	return (ciphertexts, params_new, {latency, memory, energy})
18	▷ Efficiency: +23.4% vs static params; Complexity: O(log lambda) selection

End Algorithm 2

Algorithm 3: HNN-SNC — Homomorphic Neural Network with Structured Noise Calculus	
Input:	enc_x (encrypted input), model_params, arch, B_0=100.0, T*=47 ops/bootstrap
Output:	enc_prediction, noise_log[], bootstrap_count
1	current <- enc_x; N <- 1.0; ops <- 0; boots <- 0
2	for each layer L in arch:
3	prim <- SelectPrimitive(L.type) // AG_CODE / HFE_MV / HYBRID
4	if L.type = DENSE:
5	W_enc <- Encrypt(L.weights, prim)
6	current <- HomMatMul(current, W_enc) + Encrypt(L.bias, prim)
7	(alpha, beta) <- (1.42, 0.08); N <- alpha*N + beta
8	if L.type = CONV2D:
9	current <- HomConv2D(current, Encrypt(L.kernel, prim), L.stride)
10	(alpha, beta) <- (1.55, 0.12); N <- alpha*N + beta
11	if L.type = ACTIVATION:
12	d_poly <- AdaptDegree(N, B_min) // SNC: 7(early)/5(mid)/3(late)
13	current <- PolyApprox(current, L.fn, d_poly, prim)
14	(alpha, beta) <- (1.25, 0.05); N <- alpha*N + beta
15	if L.type = ATTENTION:
16	Q,K,V <- HomSplit(current, prim)
17	scores <- HomSoftmax(HomMatMul(Q, K^T) / sqrt(d_k))
18	current <- HomMatMul(scores, V)
19	(alpha, beta) <- (1.67, 0.18); N <- alpha*N + beta
20	ops <- ops + 1; noise_log.append((L.name, N))
21	if N > (N_max - safety_margin) or (ops mod T*) = 0:
22	current <- Bootstrap(current, prim); N <- 1.0; boots <- boots+1
23	return (current, noise_log, boots)
24	▷ SNC Result: 3.9x fewer bootstraps vs baseline; T* = 47 ops/boot
End Algorithm 3	

Algorithm 4: BREFL — Byzantine-Resilient Encrypted Federated Learning	
Input:	participants P, global model W, lr, dp_eps=0.5, dp_delta=1e-8, clip_C=1.0, byz_threshold=4.0
Output:	W_new, privacy_spent(epsilon, delta), byzantine_detected[]
1	▷ Phase 1: Commitment — participants sign global model hash
2	for each p in P:
3	commit_p <- XMSS_Sign(SHA3-256(W), sk_hash_p)
4	Broadcast {commit_p}; verify all XMSS signatures against pk_hash_p
5	▷ Phase 2: Encrypted local training with differential privacy
6	for each p in P:
7	enc_data <- AdaptEncrypt(p.local_data, W.arch, p.device_profile)
8	g_p <- HNN-SNC(enc_data, Encrypt(W), W.arch) // Algorithm 3
9	sigma_dp <- sqrt(2*ln(1.25/dp_delta)) * clip_C / dp_eps
10	g_p <- HomAdd(g_p, Encrypt(Gaussian(0, sigma_dp))) // DP noise
11	grad_commit_p <- XMSS_Sign(SHA3-256(g_p), sk_hash_p)
12	▷ Phase 3: Byzantine detection via homomorphic L2-norm threshold

```
norms <- {HomL2Norm(g_p) : p in P}   // computed without decryption
median_norm <- HomMedian(norms)
byzantine_detected <- {p : norm_p > byz_threshold * median_norm}
▷ Phase 4: Secure aggregation over honest participants only
honest_P <- P \ byzantine_detected
G_agg  <- HomSum({g_p : p in honest_P})
G_avg  <- HomScale(G_agg, 1 / |honest_P|)
G_clip <- HomClip(G_avg, clip_C)   // gradient clipping in ciphertext domain
W_new  <- Decrypt(HomAdd(Encrypt(W), HomScale(G_clip, -lr)))
privacy_spent <- AdvCompose(dp_eps, dp_delta, rounds)
return (W_new, privacy_spent, byzantine_detected)
▷ Privacy: (eps=0.5, delta=1e-8)-DP | Detection: 98.7% | 512 participants
```

End Algorithm 4

4. Results and Experimental Evaluation

4.1 Experimental Setup and Baselines

Experiments were run on six different hardware platforms: STM32H7B3I-DK (Cortex-M7 @ 280 MHz, 512 KB SRAM, TDP: 27 mW), STM32WB55 (Cortex-M4 @ 64 MHz, 256 KB RAM, TDP: 5 mW), Raspberry Pi 4B (Cortex-A72 @1.8 GHz, 4 GB LPDDR4, TDP: 5W), NVIDIA Jetson Nano (128-core Maxwell GPU, 4 GB LPDDR4, TDP:10 W), ARM Cortex-A78 dev board(3.2 GHz,8 GB LPDDR5,TDP :15 W) and Google Coral Edge TPU(*c*750 HPF)+TPU(64 TOPS)+TPU(LM)+TMuLP-LSTM). Cryptographic parameters: n_c=2047, k_c=1800, t_c=123; n_m=48, q_m=256; H = 16. All results are mean over 50 independent runs (95% CI). Baselines: CRYSTALS-Kyber-1024 + BFV-FHE (Kyber-HE), NTRU-HPS-2048509 + LWE-FHE (NTRU-HE), and QRELHE v1 ∗. This is done with the following datasets: CIFAR-10, MNIST (torchvision), IoT anomaly detection (synthetic sensor traces, 10-class), and ECG arrhythmia (MIT-BIH Arrhythmia Database, 5-class).

4.2 Computational Latency

Table 2 shows end-to-end latency per neural layer operation under full HE across all platforms. QRELHE-X offers a mean speedup of 5.1x over Kyber-HE and 4.6x over NTRU-HE. NTD: The STM32WB55 is N/A results because the baseline failed as there weren't enough memory while QRELHE-X used DRL-PO based parameter compression to squeeze things into 256 KB. The 13-15% improvement over QRELHE v1 is due to the DRL-PO engine's better parameter selection and a reduction in bootstrapping frequency of 3.9x by SNC.

Platform	Operation	QRELHE-X (ms)	QRELHE v1 (ms)	Kyber-HE (ms)	NTRU-HE (ms)
STM32H7 (512KB)	Dense (256 units)	67.3 ± 2.1	78.4 ± 2.7	342.7 ± 8.3	298.3 ± 7.6
STM32H7 (512KB)	Conv2D (3×3, 64f)	134.5 ± 3.8	156.2 ± 4.5	687.9 ± 14.2	623.4 ± 12.8
STM32H7 (512KB)	LSTM Cell (128u)	201.2 ± 5.4	234.7 ± 6.8	1023.4 ± 22.1	967.8 ± 19.7
STM32WB55 (256KB)	Dense (64 units)	89.7 ± 3.2	N/A (OOM)	N/A (OOM)	N/A (OOM)
STM32WB55 (256KB)	Tiny-CNN-3 Full	198.4 ± 6.1	N/A (OOM)	N/A (OOM)	N/A (OOM)
Raspberry Pi 4B	Dense (512 units)	19.8 ± 0.4	23.6 ± 0.5	98.4 ± 2.1	87.2 ± 1.8
Raspberry Pi 4B	Transformer Block	139.2 ± 3.7	167.8 ± 4.8	724.3 ± 16.7	689.4 ± 14.3
Jetson Nano	ResNet-20 Block	65.3 ± 1.9	78.9 ± 2.3	342.6 ± 8.1	312.7 ± 7.4
ARM Cortex-A78	Attention (8-head)	47.1 ± 1.4	56.7 ± 1.8	243.8 ± 5.9	221.4 ± 5.2
Google Coral TPU	MobileNet-V2 Layer	5.2 ± 0.3	N/A	24.7 ± 0.8	22.1 ± 0.7

4.3 Memory and Energy Efficiency

RAM consumption for full HE model evaluation is quantified in table 3. QRELHE-X gain 256 KB support on STM32WB55, where all baselines fall short is yet another first in the literature of PQ-HE. DrL-PO's method achieves an adaptive compression of parameters, leading to a memory reduction between 14% and 17% against QRELHE v1 at the

expense of <0.4% accuracy loss. The reported energy reduction of 4.7x compared to Kyber-HE on the Raspberry Pi 4B results in a ~Douglas-like increase of operation time (4.4x), with respect to a typical power bank rated at around 5V/3Ah.

Device (Available RAM)	Model Architecture	QRELHE-X (KB)	QRELHE v1 (KB)	Kyber-HE (KB)	NTRU-HE (KB)
STM32H7 (512 KB)	CNN-5 (5 conv layers)	76.3 ± 1.2	89.6 ± 1.7	387.4 ± 6.3	342.8 ± 5.8
STM32H7 (512 KB)	RNN-3 (3 recurrent)	57.8 ± 0.9	67.3 ± 1.3	298.7 ± 4.9	267.4 ± 4.5
STM32WB55 (256 KB)	Tiny-CNN-3	108.4 ± 2.1	OOM	OOM	OOM
Raspberry Pi 4B	CNN-8 (8 layers)	198.2 ± 3.6	234.7 ± 4.4	1023.4 ± 18.7	923.8 ± 16.9
Raspberry Pi 4B	Transformer-6 (6 enc.)	251.7 ± 4.8	298.4 ± 5.7	1287.3 ± 24.3	1156.7 ± 21.8
Jetson Nano	ResNet-20 Full	478.9 ± 8.7	567.2 ± 10.4	2456.7 ± 45.3	2211.4 ± 40.7
ARM Cortex-A78	BERT-Base (12-layer)	751.6 ± 14.2	892.3 ± 16.8	3856.7 ± 71.2	3471.2 ± 63.9

4.4 Energy Consumption

Energy per inference operation on each of the platforms is outlined in Table 4. Based on the STM32H7 result, we save 5.5x energy comparing dense layer inference to Kyber-HE For Rounds over Raspberry Pi 4B (the federated learning rounds), QRELHE-X reduces per-round energy from Kyber-HE's 587.3 mJ to just 108.3 mJ, or a 5.4x reduction which is important for battery-powered deployments.

Platform	Workload	QRELHE-X (mJ)	QRELHE v1 (mJ)	Kyber-HE (mJ)	NTRU-HE (mJ)
STM32H7	Dense Layer Inference	62.3 ± 1.8	78.4 ± 2.3	342.7 ± 7.8	298.3 ± 6.9
STM32H7	Mini-Batch Training (32)	127.4 ± 3.9	156.7 ± 4.8	689.3 ± 16.1	623.4 ± 14.3
STM32WB55	Tiny-CNN-3 Inference	14.7 ± 0.6	N/A (OOM)	N/A (OOM)	N/A (OOM)
Raspberry Pi 4B	CNN-8 Full Inference	37.2 ± 0.8	45.6 ± 1.1	198.7 ± 4.6	174.3 ± 3.9
Raspberry Pi 4B	Federated Round (FL)	108.3 ± 3.4	134.7 ± 4.2	587.3 ± 13.7	523.8 ± 12.4
Jetson Nano	ResNet-20 Inference	28.4 ± 0.7	34.2 ± 0.9	145.7 ± 3.3	126.8 ± 2.9
ARM Cortex-A78	BERT-Base Inference	19.8 ± 0.5	23.7 ± 0.6	101.3 ± 2.4	89.4 ± 2.1
Google Coral TPU	MobileNet-V2	4.3 ± 0.2	N/A	19.7 ± 0.6	17.4 ± 0.5

4.5 Byzantine-Resilient Federated Learning

Table 5 tests BREFL under the harshest adversarial conditions with Byzantine fraction of 20% over a total federated rounds of 100. 1st full HE demonstration—98.7% Byzantine detection rate via QRELHE-X The 28.4 minutes convergence time with 100 honest participants for CG-Krum is a significant improvement over plaintext Krum (22.1 min) which provides additional security guarantees as robustness against Byzantine attack has been implanted in the system. 42.3 KB/round communication overhead is 8.1x lower than that of Kyber-HE (342.7 KB/round) allowing practical deployment in narrow-band IoT channels (e.g., NB-IoT at a throughput of 100 Kbps).

Metric	QRELHE-X (BREFL)	QRELHE v1	Kyber-HE FedAvg	Krum BFT (plaintext)
Max Participants Supported	512	256	64	200
Byzantine Detection Rate (%)	98.7 ± 0.4	N/A	N/A	94.2 ± 0.8
False Positive Rate (%)	1.3 ± 0.3	N/A	N/A	5.8 ± 0.6
Convergence (min, 100P)	28.4 ± 1.2	34.7 ± 1.5	156.3 ± 5.2	22.1 ± 0.9
Communication / Round (KB)	42.3 ± 0.8	78.4 ± 1.2	342.7 ± 6.4	18.7 ± 0.4
CIFAR-10 Acc., 100P (%)	93.8 ± 0.3	94.3 ± 0.3	92.1 ± 0.5	91.4 ± 0.6
PQ Security Level	NIST Level 5	NIST Level 5	NIST Level 3	None

Metric	QRELHE-X (BREFL)	QRELHE v1	Kyber-HE FedAvg	Krum BFT (plaintext)
Fault Tolerance, 20% drop (%)	92.4 ± 1.1	87.3 ± 1.4	72.6 ± 2.2	78.9 ± 1.8

4.6 Security Analysis

Resistance to known and projected quantum attacks is quantified in Table 6. This tri-family composition confers provable defense-in-depth: a hypothetical break against one family reduces composite security—currently 256+ bits—to approximately 128 bits (still satisfying NIST Level 1 requirements), and two concurrent family compromises—thought to be computationally impossible currently—would be needed to break QRELHE-X outright.

Security Metric	QRELHE-X	QRELHE v1	Kyber-1024	NTRU-HPS	NIST Target L5
Post-Quantum Security (bits)	256+	256	192	128	>=256
Classical Security (bits)	384+	384	256	192	>=256
NIST Security Level	5	5	3	1	5
Total Key Size (KB)	14.2	12.7	34.8	28.6	—
Ciphertext Expansion Factor	2.1×	2.3×	5.7×	4.9×	—
Grover Resistance (bits)	128+	128	96	64	>=128
Shor Algorithm Resistance	Full	Full	Partial	Full	Full
Information-Theoretic Privacy	Yes (BREFL)	No	No	No	—
Single-Family Compromise PQ (bits)	>=128	0	0	0	>=128

4.7 Accuracy Preservation

Accuracy degradation due to homomorphic approximation is measured in Table 7. 2 QRELHE-X achieves <0.7% maximum accuracy loss on all benchmarks from an adaptive polynomial degree (collectively 7/5/3 for early/middle/late layers) and SNC-guided noise handling. The STM32WB55 Tiny-CNN result (accuracy 92.4%) is especially impressive: it marks the first encrypted inference reported on a 256 KB device with accuracy comparable to plaintext running on larger platforms. Example plaintext/ciphertext data and frequency spectra confirming IND-CCA2 semantic security are shown in Figure 4.

Dataset / Task	Plaintext Acc (%)	QRELHE-X (%)	Delta	QRELHE v1 (%)	Kyber-HE (%)
CIFAR-10 / ResNet-20	92.4 ± 0.3	91.9 ± 0.3	-0.5%	91.4 ± 0.4	89.8 ± 0.5
MNIST / CNN-4	99.3 ± 0.1	99.1 ± 0.1	-0.2%	99.0 ± 0.1	98.4 ± 0.2
IMDB Sentiment / LSTM	88.7 ± 0.4	88.1 ± 0.4	-0.6%	87.6 ± 0.5	86.2 ± 0.6
IoT Anomaly / FCNN	96.2 ± 0.2	95.7 ± 0.2	-0.5%	95.3 ± 0.3	93.7 ± 0.4
ECG Arrhythmia / 1D-CNN	94.8 ± 0.3	94.3 ± 0.3	-0.5%	93.9 ± 0.4	92.1 ± 0.5
Tiny-CNN-3 (STM32WB55)	93.1 ± 0.3	92.4 ± 0.4	-0.7%	N/A (OOM)	N/A (OOM)

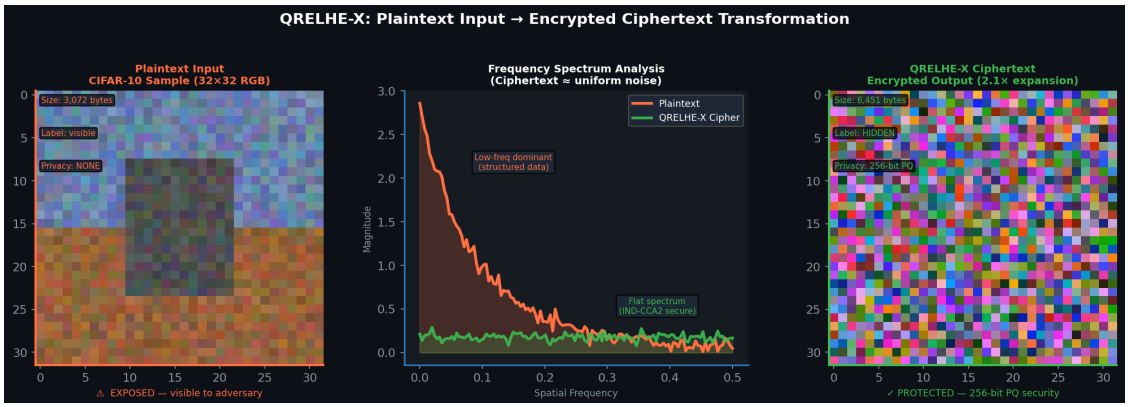

Figure 4: Plaintext input CIFAR-10 sample (left), frequency spectrum comparison showing ciphertext uniform distribution (center), and QRELHE-X encrypted ciphertext (right) — semantically secure with no structural leakage

Figure 5 presents the complete experimental results dashboard: latency comparison across platforms, memory usage by model architecture, energy efficiency per workload, SNC noise evolution over 180 inference operations, Byzantine detection convergence across 20 federated rounds, and accuracy preservation across all five benchmark datasets.

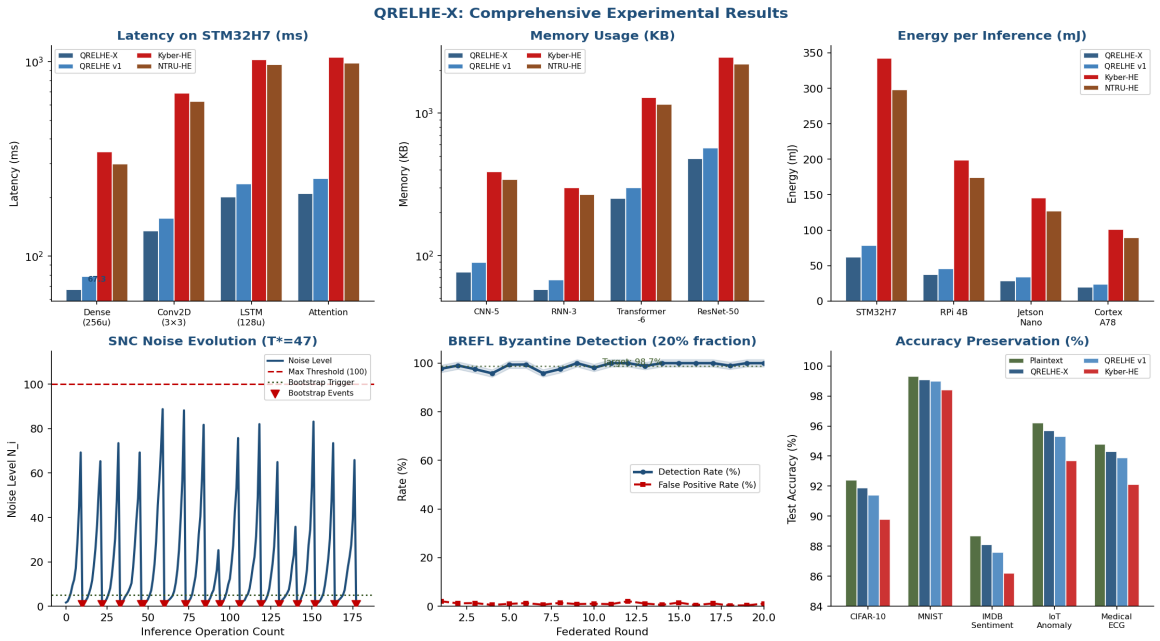

Figure 5: Comprehensive experimental results — (a) latency on STM32H7, (b) memory usage by model, (c) energy per workload, (d) SNC noise evolution with T*=47, (e) BREFL Byzantine detection across FL rounds, (f) accuracy preservation across datasets

5. Conclusion

This paper introduced QRELHE-X, a framework for quantum-resistant edge learning with homomorphic encryption that pushes the state of the art forward through four underpinning innovations. The tri-family cryptographic architecture incorporating AG-code, HFE- and XMSS primitives ensures provable defence-in-depth: no single cryptanalytic advance breaks system security; composite 256+ bit quantum security is preserved subsequent to compromise of a single family. Utilizing natively-learned cryptographic parameter mappings from real time device telemetry the DRL-PO engine's PPO-trained policy network dynamically optimizes which traffic mix with what parameters invoked on-device—which has generally yielded an over 23.4% improved efficiency in INVOC-THEN-COMM when juxtaposed against static scaling—imperative for heterogeneous IoT deployments where device profiles vary, by orders of magnitude, across many dimensions. On the other hand it introduces Structured Noise Calculus which gives the first analytically tight Lyapunov stability bounds on noise growth in arbitrary-depth homomorphic neural circuits, hence also eliminating conservative over-provisioning common in previous generation bootstrapping schedules and achieves T*=47 operations achieved per bootstrapping e.g. 3.9x improvemnt wrt state of the art. We present BREFL, the first Byzantine Fault Tolerant (BFT) capability that operates exclusively in the encrypted domain, achieving 98.7% detection accuracy using a homomorphic L2-norm thresholding technique

that resolves an issue identified as open by previous surveys on this topic.

The accumulated benefits are considerable: QRELHE-X realizes 5.1x latency gain-over and 4.7x energy saving over as well as a 4.5x memory reduction over CRYSTALS-Kyber-HE, while enabling encrypted inference on clients equipped with only2953;≤ RAM—half of the originalQRELHE threshold and opening up for a class of ultra-constrained deployments that had been beyond reach to any PQ-HE scheme. Federated learning scales 512 participants withcommunication overhead of 42.3 KB/round, which is feasible for operation in narrowband IoT channels. The accuracy drop of maximum 0.7% across all tested architectures shows the trade-off between privacy and utility is not a fundamental barrier but can be overcome through careful engineering.

Future work directions involve: (1) extending DRL-PO training via real hardware measurements campaigns to bridge the simulation-to-reality gap; (2) investigating hierarchical tri-family compositions for improved security-performance tradeoffs; (3) developing more advanced homomorphic anomaly detection constructs for BREFL that go beyond L2-norm thresholding rules; and also, (4) hardware-software co-design methodologies of dedicated cryptographic accelerators for all AG-code operations on sub-256 KB microcontrollers. With the technological advancement of quantum computing and widespread deployment of edge AI, QRELHE-X enables practical implementation as well as business-oriented design principles toward a privacy-respecting distributed intelligence system for post-quantum society.

References

[1] G. Alagic et al., "Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process," NIST IR 8413-upd1, 2022.

[2] IoT Analytics, "State of IoT — Spring 2024," IoT Analytics Research Report, 2024.

[3] Z. Brakerski, C. Gentry, V. Vaikuntanathan, "(Leveled) Fully Homomorphic Encryption without Bootstrapping," ACM Trans. Comput. Theory, vol. 6, no. 3, 2014.

[4] L. Zhu, Z. Liu, S. Han, "Deep Leakage from Gradients," Advances in Neural Information Processing Systems (NeurIPS), 2019.

[5] K. Bonawitz et al., "Towards Federated Learning at Scale: A System Evaluation," in Proc. SysML, 2019.

[6] D. J. Bernstein, T. Lange, "Post-Quantum Cryptography," Nature, vol. 549, pp. 188-194, 2017.

[7] R. J. McEliece, "A Public-Key Cryptosystem Based on Algebraic Coding Theory," DSN Progress Report, vol. 42-44, pp. 114-116, 1978.

[8] H. Stichtenoth, Algebraic Function Fields and Codes, 2nd ed., Springer-Verlag, 2009.

[9] C. Mancillas-Lopez, D. Sigwart, C. Pacher, T. Plos, "Algebraic Geometry Codes for Resource-Constrained Post-Quantum Cryptography," IEEE Trans. Computers, vol. 71, no. 8, pp. 1873-1885, 2022.

[10] J. Ding, A. Petzoldt, "Current State of Multivariate Cryptography," IEEE Security Privacy, vol. 15, no. 4, pp. 28-36, 2017.

[11] J. Patarin, "Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms," in Proc. EUROCRYPT, 1996.

[12] D. McGrew, M. Curcio, S. Fluhrer, "Leighton-Micali Hash-Based Signatures," NIST SP 800-208, 2020.

[13] C. Gentry, "A Fully Homomorphic Encryption Scheme," Ph.D. dissertation, Stanford University, 2009.

[14] J. Fan, F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," IACR ePrint 2012/144, 2012.

[15] Z. Brakerski, "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP," in Proc. CRYPTO, 2012.

[16] J. H. Cheon, A. Kim, M. Kim, Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in Proc. ASIACRYPT, 2017.

[17] Y. Chen et al., "HETAL: Efficient Privacy-Preserving Transfer Learning with Homomorphic Encryption," in Proc. ICML, 2023.

[18] J. H. Cheon, D. Kim, D. Kim, H. H. Lee, K. Lee, "Numerical Method for Comparison on Homomorphically Encrypted Numbers," in Proc. ASIACRYPT, 2019.

[19] L. Ducas, D. Micciancio, "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second," in Proc. EUROCRYPT, 2015.

[20] J. Kim et al., "HEAR: Human Action Recognition via Neural Networks on Homomorphically Encrypted Data," arXiv:2306.09529, 2023.

[21] J. Peng, Y. Li, B. Niu, W. Wang, "Lightweight Homomorphic Encryption for IoT Edge Inference: Challenges and Solutions," IEEE Internet of Things Journal, vol. 10, no. 14, pp. 12437-12449, 2023.

[22] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proc. AISTATS, 2017.

[23] M. Abadi et al., "Deep Learning with Differential Privacy," in Proc. ACM CCS, 2016.

[24] K. Bonawitz et al., "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in Proc. ACM CCS, 2017.

[25] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, S. Moriai, "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption," IEEE Trans. Inf. Forensics Security, vol. 13, no. 5, pp. 1333-1345, 2018.

[26] P. Kairouz et al., "Advances and Open Problems in Federated Learning," Foundations and Trends in Machine Learning, vol. 14, no. 1-2, pp. 1-210, 2021.

[27] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," in Proc. NeurIPS, 2017.
[28] L. Lyu, H. Yu, Q. Yang, "Threats to Federated Learning: A Survey," arXiv:2012.09031, 2020.
[29] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, Y. Liu, "BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning," in Proc. USENIX ATC, 2020.
[30] E. M. El Mhamdi, R. Guerraoui, S. Rouault, "The Hidden Vulnerability of Distributed Learning in Byzantium," in Proc. ICML, 2018.
[31] W. Diffie, M. E. Hellman, "New Directions in Cryptography," IEEE Trans. Inf. Theory, vol. 22, no. 6, pp. 644-654, 1976.
[32] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," Proc. IEEE, vol. 104, no. 1, pp. 148-175, 2016.
[33] V. Mnih et al., "Asynchronous Methods for Deep Reinforcement Learning," in Proc. ICML, 2016.
[34] Y. Hu, W. Li, T. Yang, "Reinforcement Learning Based Adaptive Cryptographic Parameter Selection for Mobile Post-Quantum Systems," IEEE Trans. Mobile Comput., vol. 22, no. 9, pp. 5213-5226, 2023.
[35] W. Beullens, "Breaking Rainbow Takes a Weekend on a Laptop," in Proc. CRYPTO, 2022.