Connectivity-Aware Hybrid Access Control for Low-Power IoT Devices

Hyeon-Yeong Yoon and Ki-Hyung Kim
Department of Cybersecurity
Ajou University, Suwon, Republic of Korea
Email: gusd0423@ajou.ac.kr, kkim86@ajou.ac.kr

***Abstract*—Low-power IoT devices such as ESP32-class microcontrollers increasingly control security-critical resources while operating over unstable networks. Existing access control architectures for IoT typically assume either a permanently online, cloud-centric model or a purely offline model, and rarely treat connectivity itself as a first-class design parameter. This paper presents a connectivity-aware hybrid access control architecture for constrained IoT devices that integrates verifiable credentials (VCs), hybrid RBAC/ABAC policies, the Open Policy Agent (OPA), and WebAssembly (Wasm). We propose a three-tier policy model—RBAC fast path, offline ABAC, and online-extended ABAC—driven by a local connectivity state machine (online, offline, intermittent) and TTL-based freshness tracking of revocation and status data. We implement the design on an ESP32-class device using OPA-to-Wasm compilation and qualitatively evaluate its behavior through five representative attack scenarios. The analysis shows how connectivity-aware tiering and signed, updateable policies help the device maintain or tighten its security posture under adversarial network conditions.**

***Index Terms*—IoT security, access control, verifiable credentials, Open Policy Agent, Connectivity-Aware Authorization.**

I. Introduction

Microcontroller-class IoT devices are now responsible for controlling door locks, building access, lighting, and industrial actuators. These devices are inexpensive and power efficient, but they run with tight CPU, memory, and storage budgets and are frequently deployed behind unstable Wi-Fi or gateway links. Despite these constraints, their authorization decisions have direct physical impact and must remain correct even when the cloud is unreachable.

Most existing IoT authorization architectures assume a cloud-centric model: the device forwards tokens or context to a remote Policy Decision Point (PDP) that hosts RBAC or ABAC policies. When connectivity is unavailable, devices either fail closed—disabling critical functionality—or resort to ad-hoc caching of previous decisions. Such caching can silently weaken security, as it typically ignores how old or incomplete the cached information is. At the opposite extreme, purely offline designs perform all checks locally but struggle to incorporate revocation, cross-device anomaly signals, or centrally updated policies.

In parallel, verifiable credentials (VCs) and self-sovereign identity (SSI) have been adopted as a way to express signed claims about users and devices, and recent work shows that even constrained hardware can verify VCs locally. Policy-as-code frameworks such as the Open Policy Agent (OPA) allow rich attribute-based access control (ABAC) policies to be expressed in a declarative language and evaluated close to applications, including on microcontrollers via WebAssembly (Wasm) backends. However, these building blocks are typically studied in isolation, and most systems still treat connectivity as a binary assumption rather than as a dynamic, attackable dimension.

This work asks: *How can a constrained IoT device make secure authorization decisions across the full range of connectivity conditions—from fully offline, through intermittent, to stably online?*

Rather than assuming an always-online or always-offline setting, we take the viewpoint of a device that must by design support all three regimes. At the core of our architecture, an ESP32-class device acts as a verifier for VCs and as a local policy engine; this core must be able to operate completely offline, but it is also expected to exploit revocation and risk information whenever connectivity is available. Around this core we design a connectivity-aware, tiered decision model that treats connectivity and data freshness as explicit inputs to authorization.

In summary, this paper makes three contributions. First, we propose a connectivity-aware three-tier policy model with a Tier 0 RBAC fast path, a Tier 1 offline ABAC core using only local context, and a Tier 2 online-extended ABAC layer that exploits revocation status, risk scores, and other server-side information across offline, intermittent, and online operation. Second, we design a simple connectivity state machine combined with TTL-based freshness tracking for status and policy data so that connectivity and freshness jointly determine which tiers may be used and enforce fail-safe degradation when data becomes stale. Third, we implement a device-side decision algorithm and prototype on an ESP32 using OPA-to-Wasm compilation, and qualitatively evaluate the architecture through five representative attacks: credential theft and replay, revocation bypass while offline, forced offline downgrade, malicious policy update, and MITM/replay.

By framing connectivity and data freshness as first-class policy inputs from the outset, our architecture aims to make IoT access control robust to the messy, adversarial networks in which these devices actually operate.

II. Background and Problem Statement

A. Verifiable Credentials and SSI for IoT

Verifiable Credentials (VCs) provide a standard format for expressing and verifying signed claims about a subject. A typical ecosystem involves issuers, holders (wallets), and verifiers [1]. Issuers sign credentials containing attributes such as role or organizational affiliation; holders later present verifiable presentations (VPs) to verifiers, who validate the signature chain and validity period. Revocation can be implemented via status lists or accumulators; recent proposals use bitstring-based lists that compactly encode revocation state for many credentials [2].

Several works explore using VCs and SSI for proximity-based or offline access control [3], [4]. They demonstrate that constrained devices can verify credentials without a permanent backend connection, which is attractive for IoT. However, these systems largely focus on the credential verification flow and treat revocation freshness and connectivity changes as separate operational concerns.

B. Hybrid RBAC/ABAC and Policy-as-Code

RBAC assigns permissions to roles and roles to subjects; it is simple and widely used but often too coarse for IoT deployments where context such as time, location, or device state matters. ABAC generalizes RBAC by allowing policies over arbitrary attributes of subject, object, action, and environment. Hybrid RBAC/ABAC models combine a fast role-based path with richer attribute checks and have been studied in smart-home and IoT settings [7].

Policy-as-code frameworks such as the Open Policy Agent (OPA) evaluate declarative policies written in Rego over structured input data. OPA is commonly deployed as a sidecar PDP in cloud or edge systems, and its policies can be compiled to Wasm for portable execution. When coupled with lightweight Wasm runtimes, this enables policy evaluation on microcontrollers while maintaining strong isolation between policy code and device firmware.

C. Connectivity as a Security Dimension

Surveys of IoT access control catalog cloud-centric, capability-based, and edge-based models [5], [6]. Cloud-centric PDPs centralize decisions but assume reliable links; edge IAM systems move logic closer to devices, but still often assume that connectivity outages are rare and short. In many designs, the behavior of the system when status or policy data becomes stale is left implicit.

In practice, connectivity is both unreliable and adversarial: an attacker can jam wireless links, reboot gateways, or selectively delay traffic. If the access control strategy does not explicitly account for this, the system may either fail open (by continuing to rely on stale state) or fail closed in ways that are unacceptable for safety-critical deployment (e.g., door locks that simply stop working).

This motivates an architecture in which connectivity state and data freshness are explicit inputs to authorization decisions, not just background assumptions.

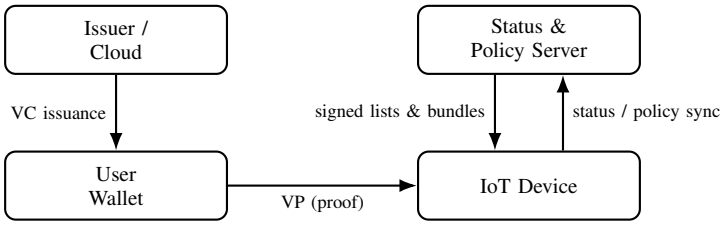

Fig. 1. High-level system model and credential flow between issuer, wallet, IoT device (ESP32-class), and status/policy server. The server and device exchange status lists and policy bundles over the status/policy sync channel.

III. Connectivity-Aware Hybrid Architecture

In this section we present the proposed architecture, combining the system model, connectivity state machine, three-tier policy model, and device-side algorithm.

A. System Model

The system consists of four logical roles:

- **Issuer / Cloud**: issues VCs to subjects (users, administrators, devices). Issuers manage public keys and DID documents and publish revocation information through status lists.
- **User Wallet**: a mobile wallet that stores VCs and generates VPs. It communicates with the IoT device over BLE, NFC, or a local network.
- **IoT Device**: an ESP32-class microcontroller controlling a physical resource, such as a lock or relay. It acts as both verifier and local PDP/PEP, running VC verification logic and a Wasm runtime for policy evaluation.
- **Status & Policy Server**: a backend service that publishes revocation status lists and signed policy bundles. IoT devices synchronize with this server opportunistically when they consider themselves online.

The overall architecture and credential flow are illustrated in Fig. 1. We assume standard cryptographic primitives (digital signatures, TLS) are correctly implemented and that issuers' private keys are not compromised. Full physical compromise of the device and fully malicious administrators remain outside strict scope; we discuss their impact in the evaluation.

B. Connectivity State Machine and Freshness

The IoT device maintains a local connectivity state machine with three states:

- **Online**: recent health checks and synchronization attempts succeeded; remote status and policy services are usable.
- **Offline**: repeated attempts to contact the server failed; the device must assume that remote services are unavailable.
- **Intermittent**: the device has recently experienced both successes and failures; it may sometimes synchronize but cannot rely on per-request connectivity.

Transitions are triggered by simple probes (e.g., periodic HTTPS requests) and timeouts. Independently of this state, the device records the time of last successful synchronization for each of:

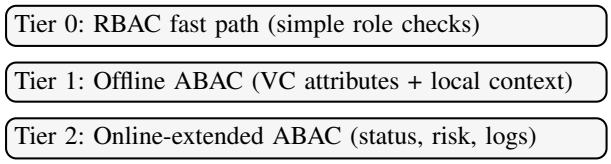

Fig. 2. Connectivity-aware three-tier policy model. Offline decisions use Tier 0 and Tier 1; Tier 2 is only evaluated when the device is online and status/policy data are fresh.

- revocation status lists, and
- policy bundles (OPA packages compiled to Wasm).

Each artifact is associated with a Time To Live (TTL); if the current time minus the last-sync time exceeds the TTL, the artifact is considered stale. Freshness is not only a transport-level property but a semantic signal used by policy.

C. Three-Tier Policy Model

On top of the connectivity state we define a three-tier policy model:

- **Tier 0: RBAC Fast Path**. Tier 0 contains simple role-based rules such as "role == admin" or "role == emergency operator". These rules are evaluated with minimal overhead and are intended to capture very coarse-grained, high-privilege paths. In many deployments, Tier 0 is used for management operations and is always available.
- **Tier 1: Offline ABAC**. Tier 1 contains attribute-based rules that depend only on local data: VC attributes, the current time, device mode, local sensor readings (e.g., door state, motion sensor), or other configuration stored on the device. Tier 1 constitutes a minimum safety policy that must hold even in the absence of connectivity. For example, a resident may be allowed to open a door only during defined hours and when the alarm is disarmed.
- **Tier 2: Online-Extended ABAC**. Tier 2 extends Tier 1 with online-only information: revocation status, risk scores derived from cross-device logs, anomaly detection flags, or centrally maintained deny-lists. Tier 2 is enabled only if the device is online and both status and policy artifacts are fresh. Tier 2 can tighten decisions but is not allowed to grant access that would be denied by Tier 1.

The resulting three-tier structure is summarized in Fig. 2. Tier 0 and Tier 1 form the offline core, while Tier 2 is an online-only tightening layer. Policies are written in Rego with multiple `allow` rules corresponding to these tiers. Guards such as `input.conn_state == "ONLINE"` and `input.status.fresh` are used to enable Tier 2 rules only under suitable conditions.

D. Device-Side Decision Algorithm and Implementation

For each incoming request and VP, the device executes the following steps:

1) **Verify VP**. Validate the VP's signature chain, proof purpose, and expiry. Failure results in an immediate hard deny.
2) **Tier 0 evaluation**. If a Tier 0 RBAC rule matches (e.g., the VC includes an admin role), the request is allowed and the decision is logged.
3) **Compute state and freshness**. Determine the current connectivity state, and compute freshness flags for status and policy artifacts based on their TTLs.
4) **Select tiers**. Tier 1 is always evaluated; Tier 2 is evaluated only if the state is online and all required artifacts are fresh.
5) **Evaluate policy**. Invoke the Wasm-compiled Rego policy with an input document containing VC claims, local context, connectivity state, and freshness flags. If any tier returns a hard deny, the request is denied; otherwise, if at least one tier returns allow, the request is allowed.
6) **Default deny**. If no rule allows the request, it is denied by default.

In our prototype the ESP32 runs a compact Wasm engine and a VC verification library. OPA policies are compiled to a single Wasm module and embedded as a firmware resource. The status and policy caches occupy a small, fixed amount of flash and RAM. Qualitatively, we observe that public-key verification for VCs dominates the cost of a request; Rego evaluation itself adds modest overhead relative to this base cost, which suggests that the three-tier model is feasible for low-rate control operations such as door access. A full quantitative performance evaluation is left as future work.

IV. Security Evaluation

We qualitatively evaluate the architecture using five attack scenarios derived from the threat model. For each scenario we highlight how the three-tier model and connectivity awareness change the device behavior compared to typical cloud-centric or SSI-only designs.

A. Credential Theft and Replay

An attacker acquires a copy of a victim's VC or VP (for example by compromising the mobile wallet or photographing a QR code) and tries to replay it from another device or location. In our system VPs are bound to a holder key, so a bare credential file is insufficient. Tier 1 ABAC then checks local context such as time, device mode, or physical state, making repeated or out-of-pattern use easier to detect. When online, Tier 2 further incorporates risk information derived from cross-device logs, turning simple replay attacks into anomalies that can be denied or throttled.

B. Revocation Bypass While Offline

A revoked user may attempt to keep the device permanently offline so that revocation lists are never updated. Traditional designs often leave the behavior under stale revocation data implicit, so devices continue to rely on outdated state. In our architecture each status list has an explicit TTL. Policies for high-risk operations require fresh status; once the TTL expires, those operations are blocked or heavily restricted until a successful synchronization occurs. This does not eliminate the revocation window, but it makes the window explicit and configurable instead of accidental.

C. Forced Offline Downgrade

By jamming Wi-Fi or rebooting gateways, an attacker can deliberately degrade connectivity in order to weaken authorization checks. In many cloud-centric or edge IAM setups, offline mode effectively means "use any cached token or decision", which is close to fail-open behavior. Here offline mode is deliberately more restrictive: Tier 2 is disabled, and certain sensitive actions are allowed only when Tier 2 can be evaluated. As the connectivity state moves from online to intermittent to offline, the available decision surface shrinks, turning connectivity manipulation into denial-of-service rather than privilege escalation.

D. Malicious Policy Update

A network adversary may try to inject or modify policies by impersonating the policy server or performing a man-in-the-middle attack during updates. Policy bundles in our design are signed artifacts, optionally modeled as verifiable presentations issued under administrator credentials. The device accepts an update only if both the transport layer (e.g., TLS) and the application-level signatures validate, and if the issuer is authorized as an administrator. Policies then execute inside a Wasm sandbox with a narrow host interface, limiting damage from logic bugs even when a legitimate administrator deploys a flawed policy.

E. MITM and Network Replay

An attacker may replay older status lists or policy bundles to roll back the device to a weaker configuration. Status and policy artifacts carry version numbers and issuance times; the device discards any artifact older than the currently stored one or beyond its TTL. When freshness cannot be established, Tier 2 is disabled and policies for high-risk operations become more conservative. This prevents attackers from silently downgrading the security posture by replaying stale but syntactically valid data.

F. Discussion and Limitations

Across these scenarios, connectivity-aware tiering and freshness-aware updates reduce the attack surface introduced by unstable networks. The architecture still has limitations: a determined attacker with physical access may compromise device keys, and a fully malicious administrator can intentionally install unsafe policies. Choosing TTL values also involves a trade-off between availability and the length of the revocation window. These aspects must be tuned per deployment and are left for future quantitative evaluation.

V. Conclusion and Future Work

This paper presented a connectivity-aware hybrid access control architecture for low-power IoT devices. The design integrates verifiable credentials, hybrid RBAC/ABAC policies, OPA, and WebAssembly into a single stack that runs entirely on an ESP32-class microcontroller. A three-tier policy model and a simple connectivity state machine allow the device to adapt its decision strategy as links to the cloud appear and disappear, using TTL-based freshness of status and policy data to prevent silent security downgrades.

Through five attack scenarios we argued that the architecture strengthens resilience against credential theft, revocation bypass, forced offline downgrade, malicious policy updates, and network replay relative to common cloud-centric, SSI-only, and edge-IAM designs. The focus of this work is conceptual and architectural; a full-scale empirical evaluation remains future work.

Immediate extensions include (i) measuring latency, CPU, and energy overhead of Tier 0/1/2 decisions under realistic workloads; (ii) using formal methods to verify safety properties of the connectivity state machine and key policies; and (iii) exploring multi-device deployments in which several IoT devices share limited risk information while respecting bandwidth and privacy constraints. More broadly, we believe that treating connectivity and data freshness as first-class policy inputs is essential for making IoT access control robust to the messy, adversarial networks in which these devices actually operate.

Acknowledgment

This research was supported in part by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2025-RS-2021-II211835) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2021-0-00590, RS-2021-II210590, Decentralized High Performance Consensus for Large-Scale Blockchains)

References

[1] M. Sporny *et al.*, "Verifiable Credentials Data Model v2.0," W3C Recommendation, 2025.

[2] M. Sporny *et al.*, "Bitstring Status List v1.1," W3C Editor's Draft, 2025. [Online]. Available: https://w3c.github.io/vc-bitstring-status-list/

[3] A. H. Enge, A. Satybaldy, and M. Nowostawski, "An offline mobile access control system based on self-sovereign identity standards," *Computer Networks*, vol. 219, 2022.

[4] D. Lagutin, Y. Kortesniemi, N. Fotiou, and V. A. Siris, "Enabling decentralized identifiers and verifiable credentials for constrained IoT devices using OAuth-based delegation," in *Proc. NDSS Workshop on Decentralized IoT Security and Standards*, 2019.

[5] K. Ragothaman, Y. Wang, B. Rimal, and M. Lawrence, "Access control for IoT: A survey of existing research, dynamic policies and future directions," *Sensors*, vol. 23, no. 4, 2023.

[6] S. Pal and Z. Jadidi, "Protocol-based and hybrid access control for the IoT: Approaches and research opportunities," *Sensors*, vol. 21, no. 20, 2021.

[7] S. Ameer, J. Benson, and R. Sandhu, "Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT," *IEEE Trans. Dependable and Secure Computing*, vol. 20, no. 5, 2023.