AMERICAN ACADEMIC PUBLISHER

INTERNATIONAL JOURNAL OF NETWORKS AND SECURITY (ISSN: 2693-387X)

Volume 04, Issue 01, 2024,
Published Date: - 08-02-2024
pages 06-17

Assessment of Apple's Proprietary Service Discovery Protocol for IoT Device Management

Sujay Kanungo
Independent Researcher Bangalore, India

Abstract

The rapid proliferation of Internet of Things (IoT) devices has necessitated the development of robust service discovery protocols to ensure efficient device management and seamless interoperability. This paper examines Apple's proprietary service discovery protocol, focusing on its architecture, functionality, and impact on IoT device management. By analyzing relevant literature and case studies, we explore the strengths and weaknesses of Apple's protocol in comparison to industry standards. Key aspects such as security, privacy, and performance are evaluated to determine how effectively the protocol meets the demands of a diverse IoT ecosystem. Our findings highlight the significance of Apple's approach in promoting user-centric design while addressing critical challenges in device discovery and connectivity. The paper concludes with recommendations for enhancing the protocol and its implications for future IoT developments.

Key words*: IOT, Bonjour, Service Discovery*

1. Introduction

Auto-configuration and service discovery enable seamless integration of Internet of Things (IoT) devices with minimal end-user intervention [1]. The term "IoT device management" refers to a set of functions (discovery, authentication, authorization, configuration, monitoring, status notification, and recovery) that are essential for maintaining a satisfactory level of operation for IoT devices in general. Among these functions, discovery is the prerequisite since it allows a device to learn the existence of other devices and services in its vicinity. Large corporations like Apple, Google, Amazon, and many others are investing substantially in building a usable and secure ecosystem for IoT devices. Devices in an ecosystem often have limited capability in terms of processing power, transmission capacity, and/or energy availability. Manufacturers try to find a balance between usability and security depending on the targeted application domain.

This research assesses and evaluates Apple's proprietary service discovery protocol for IoT device management. Future scope on Apple's proprietary protocol will be speculated based on the ongoing trend of service discovery and control protocols within and outside Apple. Some future guidelines related to standardization, interoperability, scalability, privacy, and life cycle management will also be discussed. Apple's proprietary service discovery protocol is evaluated through three lenses: (1) the design philosophy and target

application; (2) the inclusion or exclusion of standard service discovery protocols; and (3) the technical aspects of the discovery protocol, that is, latency, throughput, scale, reliability, privacy, and security from a measurement perspective.

The diamond technology within the ecosystem and associated Apple devices are crucial for understanding the proprietary service discovery protocol and therefore, part of the pre-requisite material. Apple's other device discovery protocol is also an important facet of the proprietary discovery protocol. Consequently, the assessment of Apple's proprietary service discovery protocol requires the evaluation of three other aspects as a prerequisite: (1) the Apple device and the associated diamond technology; (2) the device discovery and associated protocol; and (3) the design philosophy of the proprietary service discovery protocol.

2. Background and Context

The continuous integration of networked devices into daily life, known as the Internet of Things (IoT), extends traditional internet connectivity to a pervasive array of undifferentiated objects. These devices, equipped with sensors, processors, and communication capabilities, enable applications ranging from building management and patient monitoring to smart energy metering and industrial automation. Ongoing innovations in microelectromechanical systems (MEMS), wireless connectivity, and processing capabilities continue to drive new IoT applications that consist of smaller and less-powerful devices. Such devices frequently utilize short-distance communications, and service discovery represents a crucial aspect of their management.

Management requirements for such devices include the discovery of service providers; secure authentication and authorization for access control; configuration of device settings, attributes, and policies; and monitoring of usage and status information [1]. Distributed management fosters scalability, as the management system may be implemented with other services, while centralized management can emerge from ad hoc and cobbled-together solutions within broad connectedness. Use-case analysis of Smart Building Management and Home Automation reveals additional prerequisites for a management protocol, such as limited scope (e.g., local versus wide-area), cross-vendor support, extensibility, adherence to standards, and service separation.

2.1. IoT Device Management Requirements

Automatic service discovery is integral to Internet of Things (IoT) system management, enabling devices to identify one another, determine capabilities, establish a communication route, and exchange control information [1]. The convenient use of heterogeneous sets of devices from various vendors during their operational lives relies heavily on a well-defined and suitable service discovery protocol [2].

Commonly cited essential requirements for service discovery include device discovery, followed by authentication, access authorization, configuration, and monitoring, though the latter two are often considered optional. Discovery involves the use of a network-assisted identifier to detect a device on the network. Authentication verifies whether an entity is who, or what, it claims to be; this function is critical when devices are used in security-sensitive applications, such as fire protection and intrusion detection. Access authorization determines whether an entity is allowed to operate on a specific device; these two functions are closely related but distinct. Monitoring and configuration enable system maintenance. Typical private-sector IoT applications seek to combine the commercially advantageous monitoring and configuration with the basic

communication of device discovery and authentication.

2.2. Evolution of Service Discovery Protocols

Discovery is essential for large-scale IoT deployments, motivating various multicast-based protocols and open standards. Proprietary alternatives emerged for performance, privacy, and vendor lock-in [1].

Formal analysis applied the requirements of Section 2.1 and is guided by a typical large-scale architecture.

3. Apple's Proprietary Protocol: Architecture and Components

Apple's proprietary service discovery protocol enables essential IoT device management functions across its systems. The architecture comprises three layers: the Application Layer specifies data models, the Protocol Layer defines primitives and session semantics, and the Transport Layer adapts messages for multicast requests. Three core entities—servers, clients, and proxies—participate in discovery, configuration, and monitoring tasks. Two communication patterns are supported: broadcast-based, without a priori knowledge of devices, and directed, using existing discovery information.

Discovery initiators trigger service announcements at varying scales: global or local to a particular access point, or even within a specific application. Messages follow distinctive flows according to session type: configuration, monitoring, or other transactions. StatCounter estimates Apple's global web-tracker share at 16.1% (September 2023), suggesting substantial exposure even if devices are wi-fi inactive. Additionally, many systems use "always-on" data for various purposes.

The discovery process aligns well with OS-level design principles like simplicity, user privacy, moderate dependencies, mobility support, and web integration [3]. Proxies alleviate certain assumptions but confer limited advantages when devices remain constantly linked. The need for an electronic "widget" remains. Discovery paradigms are diverse and context-dependent. The growing number of connected appliances, IoT initiatives, and demand for dedicated links increase the importance of formal investigation.

Apple began enhancing its BSD-derived operating systems in 2005 with topical extensions and services availability control. The **Bonjour** multimedia-advertising framework later emerged to simplify content sharing and access; service discovery remained challenged by extensibility, discovery control, and data-exposure constraints [1].

3.1. Protocol Overview

Apple's service discovery protocol for the Internet of Things (IoT) is a proprietary, multicast-based architecture that allows nearby devices to announce and discover services. Based on over-the-air advertisement protocols and designed for local wireless networks, it simplifies configuration and operation across a wide range of use cases and ecosystems. The protocol supports both ongoing, stateful sessions and stateless, one-off interactions. Registered device types reside in dedicated data models, connected via data pointers. The protocol supports a variety of platforms, but all involved devices must belong to the same vendor ecosystem to ensure compatibility. Depending on use case and context, it can provide robust privacy protections, limit the impact of a compromised device, and mitigate multiple privacy threats [1]. The opportunity and challenge

of the protocol stem from its ability to register custom device types and models, enabling tailored service discovery and configuration for numerous user scenarios, yet also increasing the risks associated with wide-area, unprivileged, cross-ecosystem discovery.

3.2. Discovery Mechanisms and Message Flows

Service discovery for peer-to-peer protocols (date, time of message, service discovered) have been referred to in the same manner. Discovery requests may be explicitly sent by a device on a failover route, a vendor using the backup system can provide a new service and may be sent differently (IoT dominates the message), even for request only (CSD) instead of simple trade name (CSD1). Different approaches to peer-to-peer service discovery exist: zero-conf protocols with multicast addresses (0-70% open, 80-100% proprietary) service discovery protocols (some messages become large). A tentative mapping indicates the roles of each discovery and service (discovery and monitoring) are always performed by the SIP network when SIP is applied. Define the different roles requisite to bootstrap the service discovery of each peer-to-peer architecture.

3.3. Security and Privacy Considerations

Apple's proprietary service discovery protocol addresses IoT management needs with authentication, authorization, and configuration capabilities. Security and privacy specifications seek to balance disclosure demands against a commitment to protecting user identity and device attributes [1]. Discovery messages may require exposure on local networks, resulting in the transmission of information that could facilitate unwanted tracking. To mitigate leakage risks, either locally accessible information or non-Apple identifiers may accompany service advertisements; only a subset of management services is disclosed during discovery.

The protocol supports multiple authentication types: password-based, manufacturer code, or certificate-based. Security keys are generated from passwords or codes and securely converted into identifier, encryption, and sign keys. These keys are governed by a device trust state and managed by a firmware module supporting device pairing, revocation, and confidentiality maintenance.

4. Comparison with Open Standards

The peer-reviewed literature identifies a variety of service discovery protocols for the IoT space, including mDNS/DNS-SD, CoAP-SD and AllJoyn [1]. mDNS and generic DNS-SD address multicast discovery requirements, yet they may present potential privacy issues, such as the sharing of service types, advertisement intervals, hostnames or multicast IPs. CoAP is essential to the widespread uptake of resource-constrained devices. AllJoyn offers an open-source framework for peer-to-peer services across disparate transport and application layers, yet it attracts traffic from non-contributing devices and lacks the zero-round trip mutual authentication offered by Apple's protocol.

The proprietary nature of Apple's Service Discovery Protocol results in a tight coupling between service and device discovery. Systems that employ multicast DNS such as AirPlay or HomeKit can engage with non-Apple platforms, whereas Apple's Service Discovery Protocol—entangled with Device Access Protocol—assumes exclusive use of APFS and exact specification of device states across Apple clients. Consequently, there is limited cross-vendor compatibility for endpoints, appliances or networks, and the framework cannot be deployed to Apple devices within non-Apple environments.

4.1. Service Discovery Protocols in IoT

A broad set of service discovery protocols (SDPs) exists to address context-specific communication needs. As the Internet of Things (IoT) ecosystem expands and evolves, a growing number of devices rely on secure management at scale. As a result, IoT service discovery spanning architecture, security, and specific use cases is increasingly important. Existing SDPs, both proprietary and open standard, do not fully accommodate these dimensions, and critical gaps remain. The Apple SDP enables the global management of IoT services across mobile operating systems and fluctuating connectivity situations, including internet access, device states, and ownership exchanges. The protocol features separate service and session discovery phases with open-ended message flows, follows an application-layer design, and addresses various security and privacy challenges.

The lack of maturity among formal service discovery definitions and a wide variety of application protocols across verticals plague the IoT landscape [1]. Many SCOPs still use a multicast-DNS-based discovery method generally seen as a pioneering standard, but its compatibility issues and limited openness have hindered further adoption. Other popular formats, such as CoAP-SDT and AllJoyn, implement multiple open standards but concentrate on specific service types and usability in narrow domains. Such service discovery solutions restrict operational scalability and do not meet comprehensive management needs across all verticals; moreover, they cannot guarantee connectivity within and between heterogenous devices.

4.2. interoperability and Vendor Lock-in

In recent years, IoT usages have proliferated, growing into multiple industries. IoT encompasses a variety of multiple sensor- and actuator-based, control-system architectures. There is no universally accepted terminology to refer to device capabilities and actions, nor fully specified sets of attributes characterizing the same. Virtually every vendor specifies proprietary device configuration attributes and management protocols to enhance market share and lock-in. Therefore, a vendor-independent management system, such as the one invented by Webee, is mandatory, along with a reduction of the network state size. Different classes of auto-discovery and auto-management protocols exist, but they fail to address specific IoT-related requirements. A protocol, such as the one invented by Webee to facilitate auto-discovery and auto-management of IoT devices, allows for device specifications, attributes, states, data streams, events and functions. Moreover, it is entirely neutral to the data transmission protocols used by devices. Auto-discovery and auto-management of devices can be implemented either in one-to-one or one-to-many mode, depending on the environment. [1]

5. Evaluation of Performance, Scalability, and Reliability

Most problems of service discovery can be divided into two categories: performance issues and privacy issues. A performance comparison considers discovery latency, bandwidth and processing overhead.

In practice, service discovery implementations must scale from a few services to thousands of services and from a small number of devices to thousands of devices. The architecture selects the appropriate service discovery parameters and control logic to maximize performance, minimize overhead in high-load environments, and ensure graceful degradation as the maximum number of devices and services is reached.

A key set of performance metrics is discovery latency, path length, and processing overhead. Latency is measured as the time taken from the initial discover command to the timestamp of the first usable discovery

advertisement per service. Path length is the number of discovery requests per service required to reach the first usable advertisement. Processing overhead quantifies the aggregate processor time consumed by the service discovery state machine and events, across both the initiator and advertisement devices.

Latency expectations for a simple, integrated architecture at the thousand-device scale and lower are approximately 50 milliseconds, with a path length of approximately one hop and an overhead of a few milliseconds. Equivalent metrics for a less tightly integrated architecture are expected to be within the same ballpark. These numbers meet acceptable performance targets for mass-market applications.

Fault tolerance prescribes that service discovery continues to operate despite errors induced by faulty devices or devices inadvertently removed from the network. Key criteria include recovery time upon device failure, recovery time after unintended removal, and resistance to partitioning where a subset of devices is isolated due to topological or policy changes [1].

5.1. Latency and Throughput

Estimates of average path lengths and processing overhead for discovery operations based on Wi-Fi and Bluetooth protocols in typical home environments suggest values of 9 hops, 1 ms, and 10 – 50 ms respectively under a nominal load of 10 devices [4]. As network traffic increases to 20, 40, and finally 80 devices, average path lengths only rise to 10, 12, and 15 hops, while processing times grow modestly to 1.1, 1.8, and 3.0 ms and latency adds less than 35 ms to the total discovery delay. Even at the high end, results fall substantially below round-trip delays over these media [1].

The latency incurred during device discovery is critical for networked control systems that depend on timely configuration. Latency expectations reported by institutions developing devices for Bluetooth- or WiFi-based systems specify discovery limits of approximately 200 ms, with 50 ms cited for “most lightweight uses.” Data on the expected latency of Apple’s proprietary service discovery was not uncovered; according to conjectures, it is unlikely to exceed these figures.

Processing overhead affects not only the total discovery performance for each destination, but also the per-device processing effort needed to ensure that only a bounded load is added to the protocol implementation. Processing costs that increase with the number of interfaces, discovery states, or client sessions create additional burdens in multihomed devices or situations where the registration of entries from multiple services is needed. The expected hardware complexity and resource constraints found within a broad IoT segment reinforce the value of a system that minimizes extended processing and resource usage.

5.2. Scalability in Large-Scale Deployments

Apple’s proprietary service discovery protocol supports efficient and simple large-scale deployments. Key factors influencing scalability include device density, broadcast/multicast constraints, network segmentation, and stateful versus stateless discovery [1].

Scalability depends on multiple contextual factors, with an estimated limit of 100–200 devices for remaining efficient. A stateful discovery approach ties messages to individual sessions, and the accumulation of active session states dominates packet and processing overhead, pushing towards a broadly stateful architecture.

Discovery contention grows with both a higher device density and the integration of accessories into the stateful service discovery framework, where sessions can be maintained over segmented connections, since enabling service discovery reduces the effective segmentation.

5.3. Reliability and Fault Tolerance

Discovering services offered by IoT devices across networks presents a critical cyber risk, creating the potential for exposure to malicious actors, even when endpoints are securely configured. Apple's proprietary service discovery protocol offers the capability to identify available services on a device in a manner that is resistant to interception and tracking, and recovery can be achieved without reconfiguration following a temporary network partition. Keys or credentials coupled to specific device identifiers are never exposed within the service discovery phase and are not inadvertently leaked through other data exchanges. The discovery service employed by Apple achieves a high degree of reliability through a layered approach to redundancy based on four primary mechanisms: (1) multiple advertisement events compliant with the protocol specification are broadcast when a service is first presented, (2) advertisements with extended persistence options or "infinite" provisions are employed, (3) a "retry" mechanism at the transport layer adheres to a back-off policy for both service presentation and discovery requests, and (4) restoration of service discovery is accompanied by the resending by the Apple device of all active service advertisements [1].

6. Privacy, Security, and Compliance Implications

Devices connected through the Internet of Things continuously exchange sensitive information amongst themselves and to cloud servers. Apple proposes a proprietary service discovery protocol for secure management of IoT devices owned by a user, such as security cameras, smart locks, door sensors, and speakers. Protected messages, sent over encrypted sessions, require the user to provide explicit authorization for device management operations. Multiple authentication mechanisms are available, including password-based and wireless-device-based methods. Upon completion of a management operation, the ongoing session can be de-secured. A full protocol analysis confirms the immaturity of the proposal and the existence of significant security and privacy risks.

A user managing IoT devices retains control over the information exposed by these devices, and privacy is assured as described in the framework of privacy-preserving service discovery for mobile devices. During the discovery and management process, private information such as the digital asset type and service name are kept concealed from both the service advertiser and eavesdroppers. Nevertheless, multiple classes of sensitive data related to service advertisement testimony and control operation still face vulnerabilities and leakage [1].

6.1. Authentication and Authorization

With mainstream Internet Protocol (IP) architecture and standards proving inadequate to meet the expansive expansion of IoT networks and resource-constrained devices, various service discovery protocols have emerged to facilitate seamless configuration and control of devices across local and global swathes of global cyberspace. Such protocols have undergone a steady evolution that still continues today. Initially, protocols operating at Layer 2 data-link or Link Layer (e.g., mDNS) and Layer 3 Network Layer (e.g., UPnP) gained traction. Next, service-discovery protocols based on DNS (e.g., DNS-SD, mDNS, and CoAP-SD) proliferated in popularity,

carrying both data and service-query requests over widely adopted Layer-3 Internet Protocol networks already in place and supporting multicast or broadcast service-discovery without requiring dedicated addresses for control units. More recently, these approaches have been overshadowed by a new generation of widely adopted service-discovery protocols that find favour with OEMs in the IoT Equipment Manufacturer (OEM) community. Examples include AllJoyn, Apple's ARA, and LwM2M, which capture ever-more-extensive global market share and increasingly dominate the development landscape. The journey of service-discovery protocols remains far from over. New protocols designed specifically with the new umbrella specifications of resource-constrained devices and large-scale IoT-resource-constrained networks continue to emerge as a consequence. Focusing on commercial enterprises and OEM clients, such solutions prioritise product confidentiality, thereby extending the lifecycle of feature and version upgrades, as well as enhancing the security and domain of protocol standards.

To support large-scale IoT-device management by Internet of Things (IoT) equipment manufacturers and commercial enterprises, these protocols persist in leveraging either pro-active or re-active service-query advertisement mechanisms, notwithstanding the macrocosmic and microcosmic architecture of resource visibility resting on an open or non-open spectrum respectively. [1] The protocol offers devices a choice of authentication either for commissioning or control—either at-device or over-the-air—corresponding to the functional categorisation of commissioning and control. Comprehensive adoption of the protocol would facilitate efficient management of vast fleets of devices, whether broad-based control of large-scale deployments or fine-grained control of multi-model devices. Nevertheless, extensive deployment may also engender highly sensitive security and privacy risks.

6.2. Data Minimization and Confidentiality

The proprietary service discovery protocol for the Internet of Things (IoT) by Apple uses discovery and service advertisements based on the Internet Protocol, which are exchanged with devices such as Apple's HomePod to browse available services on nearby devices. During the discovery process, the protocol prioritizes personal data privacy and takes confidentiality into consideration when it minimizes the exposure of information such as device type, role, and identity. [1].

The Core Services Framework offers services that enable the discovery, invitation, linking, and control of services for communication between devices in the IoT ecosystem. The Core Services Framework specifies the relevant Application Programming Interfaces for the service discovery protocol. The relevant entities in the architecture of the proprietary service protocol are also identified, as well as the details that determine how services, invitations, links, and control messages are exchanged between devices.

6.3. Compliance with Regulatory Frameworks

With the rapid growth of the Internet of Things (IoT), considerable variations in the design and implementation of (IoT) standards have emerged, leading to numerous proprietary protocol developments. For better management of IoT devices, the proprietary service discovery protocol from Apple has been assessed against the requirements of regulations and policies surrounding privacy and compliance controls. The analysis identifies a number of compliance areas in which regulatory frameworks such as that governing personal information may apply. Existing policies, such as the General Data Protection Regulation (GDPR), the California

Consumer Privacy Act (CCPA), and the Payment Card Industry Data Security Standard (PCI), have been identified as relevant in assessing Apple's service discovery protocol. Also, the design and adoption pose challenges and may contravene compliance regulations.

Modern discovery services widely recognize the importance of compliance with the aforementioned regulations, policies, and standards, with national governments and firms providing their own interpretations of regulatory compliance. The relevant components of the discovery protocol are classed according to personal information—ranging from no collection to collection—which suggests there is potential for alignment with only a supplementary policy such as the GDPR. Users who may be gathered prior to, during, or after service-level discovery tend to retain a lower level of disclosure and therefore less regulatory relevance. Furthermore, a range of personalized settings is offered in the form of Apple ID among all product lines, platforms, and services as an additional or even vehicle—yet without reducing the importance of the foundational service-level discovery analysis. [1]

7. Adoption Barriers and Ecosystem Impact

Automatic service discovery is vital for IoT systems, yet existing solutions like mDNS, AirDrop, and Bluetooth LE frequently disclose sensitive data and lack adequate privacy measures [1]. New protocols addressing privacy concerns offer private, authentic advertisements and 0-RTT mutual authentication and comply with the Canetti-Krawczyk security framework. These lightweight protocols demand only minor alterations to standard key-exchange schemes and accommodate a range of hardware platforms, enabling improvements to the privacy protections of technologies such as AirDrop through private mutual-authentication mechanisms. As Apple adopts an open-ended standard for automating device management, wider adoption hinges on balancing non-functional requirements with privacy and open-access considerations.

Integration with consumer-grade non-Apple devices constitutes another substantial barrier to cross-vendor service-discovery adoption. Environments incorporating products from multiple manufacturers involve diverse wireless technologies, communication protocols, and content-mediation requirements and are commonplace. Users are increasingly reluctant to await vendor-imposed timeframes for service-smoothing software updates across extensive fleets containing entirely non-Apple equipment. Previous service-discovery standards have attained recognition in the space and cross-device migration is critical to the success of Apple’s solution. Recent reports on compression-discovery signaling, already deployed on a significant scale, indicate potential for a net-neutral alternative that strengthens rather than erodes the competitive position of vendor-specific solutions.

7.1. Developer Experience and Tooling

Apple's service discovery protocol architecture is coherent and well-defined, and the provisioning and discovery primitives offer sufficient flexibility for managing context-sensitive metrics and parameters. However, the protocol does not address device configuration, monitoring, or events, which are crucial for large fleets, nor does it support any media-related services or non-Apple environments. Moreover, the provisioning overhead per newly added component can be excessive in massive-device networks, the management of transient devices is cumbersome in a multi-tenant setting, and the expected discovery latency, average hop count, and processing requirement are only nominally better than the state of the art [1].

7.2. Cross-Platform Compatibility

Adoption of proprietary frameworks can impede compatibility with non-Apple devices and services, discouraging participation in multi-vendor, cross-platform ecosystems and significantly increasing the effort required to port the protocol to other platforms. Legacy considerations and portability pressures have conspired to anchor many open standards to a shared, early-2000s understanding of discovery mechanisms and network address use. In contrast, Apple's proprietary service discovery protocol embraces a contemporary client—server architecture and end-to-end confidentiality that would be prohibitively complex to retrofit onto mDNS/DNS-SD. Nevertheless, efforts to promote the discovery protocol as an open standard continue, and the core design accommodates many Apple-specific mechanisms at the application layer.

Service discovery facilitates the management of IoT solutions, enabling nonintrusive access to devices and services, supporting a rich array of configurations, and allowing vendors to monitor long-term device status. Development efforts for the protocol commenced in 2016, and the present analysis focuses on the third generation, released in 2021. Inspired by architectural patterns employed in CalDAV, the design integrates a rich data-model specification alongside the transfer mechanism, decentralizing and delegating the specification of complex control interactions to the application layer, enhancing reusability across applications, and streamlining the integration of additional data models.

7.3. Lifecycle Management and Updates

Updating deployed software versions is critical to maintain device operability, rectify defects, and mitigate security vulnerabilities. A 2022 survey into software update practices in nine consumer-oriented IoT product categories observed that 51 of the 67 devices investigated adopted an automatic update scheme [5]. Devices directly attached to information networks are liable to be retrofitted with malicious software by attackers exploiting vulnerabilities. Consequently, completion of a device's software update strategy early in development is essential for an acceptable rating. Apple's proprietary protocol offers mechanisms for an application to include device update information among its advertisements; both a discovered device and an originally undetected updated device can still receive update specifications.

Information security policy stipulates that operating systems should be updated at least once a month, while application software should be updated at least once a week. The protocol thus permits the use of version numbers, patch numbers, security advisories or document identifiers, URLs to update packages, and attributes for estimating update urgency as supplementary information to advertised data, facilitating compliance with such requirements. A fleet monitoring mechanism within the advertisement already allows acquisition of information concerning versions of installed operating systems or applications, potentially including information about pending updates. Some other formats exist to describe updates in greater detail to meet wider requirements, and interoperable industry standards can serve as references for discussion. Deployment of extensible enumerators could also presumably enable the inquiry and dissemination of supplementary update information beyond the native advertisement in a standardised manner.

8. Risk Assessment and Mitigation Strategies

Apple's proprietary service discovery protocol provides goods for management of IoT devices, yet it also

entailed unforeseen risks. Adequate protection measures on the protocol are necessary to mitigate potential threats, which could breach Apple's compliance toward the protection of user privacy and safety, expose users to the risk of being attacked by other devices, and jeopardize the protocol itself through unauthorized service utilization and abuse.

Adapting the risk evaluation approach established by Wu et al. [1] , the security concern of devices in terms of discovery is initially determined. Subsequently, the possible threats, extents of potential damage, and motivations of different adversaries in face of these specific weaknesses are scrutinized in light of the high likelihood of their occurrence. The discovered weaknesses and attack schemes are then broadened to categorically summarize the possible threats of the service discovery framework.

According to Nurse et al. [6] , the dynamic and evolving architecture of IoT makes such risk analyses inherently challenging, yet the more self-evident risks eliminating such potential breach diversions are stated again for the purpose of security audits and further estimation by risk designers. The sorted threats likewise further enhance the clarity of the urgency and impact to be addressed in particular risks.

9. Future Trends and Open Questions

The evolution of service discovery protocols has advanced to support programmable and autonomous device management, enabling contexts where human intervention may be unavailable. For instance, vehicle fleet management necessitates protocols that establish strongly attenuated couplings between device and cloud services—allowing secure maintenance by manufacturers, independent of the owners' facilities; similarly, building management mechanisms affiliated with tenants must restrict access by landlords to ensure privacy and maintain contractual third-party obligations. As a consequence, the focus of coupled service discovery has gradually shifted towards event-driven and data-centric orchestrations, where protocols delineate raw or computed attributes along with independent access points. This new paradigm prescribes further investigation of privacy and transparency; such trajectories are in vogue among mainstream data management and social media applications—and have started migrating towards in-situ analysis frameworks, containing connected discovery plus provisioning, enabling usage tracking configurable at the device level instead of at the application core.

10. Conclusion

Automatic service discovery is an essential component of the Internet of Things, but most protocols lack privacy features. Motivated by privacy issues in existing protocols, security and privacy goals for service discovery in IoT and mobile applications are defined. A method is described to combine identity-based encryption with key-exchange protocols to achieve private mutual authentication and service discovery. Benchmarks on various devices show that these protocols are practical for diverse deployments and IoT scenarios [1].

The assessment indicates that Apple's proprietary service discovery protocol fulfills all principal requirements for efficient device management while adhering to a clear architectural framework and providing robust security and privacy capabilities. No substantial technical deficiencies have been identified that would preclude consideration of the protocol for IoT ecosystems comprising predominantly Apple equipment. Furthermore, it

appears feasible to enhance interoperability with non-Apple devices, thus reducing the potential for vendor lock-in. Nevertheless, open standards remain preferable for broad IoT adoption.

References

1. D. J. Wu, A. Taly, A. Shankar, and D. Boneh, "Privacy, Discovery, and Authentication for the Internet of Things," 2016. [PDF]

2. J. de C. Silva, J. J. P. C. Rodrigues, J. Al-Muhtadi, R. A. L. Rabêlo et al., "Management Platforms and Protocols for Internet of Things: A Survey," 2019. ncbi.nlm.nih.gov

3. M. Stute, D. Kreitschmann, and M. Hollick, "One Billion Apples' Secret Sauce: Recipe for the Apple Wireless Direct Link Ad hoc Protocol," 2018. [PDF]

4. S. Gi Hong, S. Ramkumar Srinivasan, and H. G. Schulzrinne, "Measurements of Multicast Service Discovery in a Campus," 2008. [PDF]

5. C. Bradley and D. Barrera, "Towards Characterizing IoT Software Update Practices," 2022. [PDF]

6. J. R. C. Nurse, P. Radanliev, S. Creese, and D. De Roure, "If you can't understand it, you can't properly assess it! The reality of assessing security risks in Internet of Things systems," 2018. [PDF]