Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Taxonomy of Transport Layer Security (TLS) Revocation Failure Scenarios

Naresh Charugundla

Independent Researcher, USA

ARTICLE INFO

Received: 31 March 2026

Accepted: 6 April 2026

ABSTRACT

Certificate revocation is one of the most operationally fragile components of the Transport Layer Security ecosystem. While TLS provides robust cryptographic guarantees for data in transit, those guarantees depend on the continued validity of the certificates underpinning them. When revocation mechanisms fail, trust decisions become unreliable, and the security posture of TLS-secured communications degrades in ways that are often invisible to the relying party. Existing discussions of revocation failure tend to conflate the underlying failure conditions with the validation responses they produce, obscuring the structural causes of trust breakdown and limiting the precision of incident analysis. This article addresses that gap by proposing a formal taxonomy of TLS revocation failure scenarios, classifying failures along four orthogonal and implementation-agnostic dimensions: failure origin, temporal validity of revocation signals, scope of impact, and trust determinism. Each dimension captures a distinct aspect of revocation failure behavior that cannot be derived from the others. The taxonomy is designed to remain applicable across the full diversity of TLS deployment environments, from constrained embedded systems and IoT infrastructure to large-scale public web deployments and intermediary-mediated architectures. If failure classification is separated from validation behavior, then the taxonomy can provide a neutral analytical foundation for reasoning about revocation reliability, which may support a deeper understanding of trust behavior under failure. It can further lay groundwork for future architectural and operational analysis.

Keywords: Transport Layer Security, Certificate Revocation, Public Key Infrastructure, Trust Determinism, Failure Taxonomy

1. Introduction

Transport Layer Security (TLS) serves as the foundational protocol for transmitting sensitive data across untrusted networks. It secures financial transactions, healthcare record exchanges, authentication flows, and virtually every class of application that depends on confidentiality and integrity over the public Internet [1]. This is achieved by TLS, as it establishes a protected channel between client and server that relies on Public Key Infrastructure (PKI) to authenticate identities through cryptographic certificates [2]. However, TLS certificates provide security guarantees that are as strong as the certificates underpinning them. When a relying party continues to accept a certificate that has been compromised, misissued, or revoked, it constitutes a direct failure of trust. Certificate revocation exists to close this gap, yet its operational reliability remains one of the most persistent and underexamined weaknesses in the TLS ecosystem [12]. Mechanisms such as Certificate Revocation Lists (CRLs) and the Online Certificate Status Protocol (OCSP) provide the infrastructure for communicating revocation events, but their effectiveness depends on timely distribution, reachable endpoints, and consistent client-side interpretation [5]. When any of these conditions break down, revocation fails and trust decisions become unreliable. This article proposes a formal taxonobreak that classifies TLS revocation failure scenarios along four analytical dimensions: failure origin, temporal validity of revocation signals, scope of impact, and trust determinism. The objective is not to prescribe

enforcement strategies but to provide a structured vocabulary for reasoning about how revocation failures arise, propagate, and affect trust outcomes across heterogeneous TLS deployments.

2. Background: TLS, PKI, and the Role of Certificate Revocation

2.1 The TLS Protocol and Its Security Properties

The Transport Layer Security (TLS) protocol adds a security layer on top of a reliable transport protocol, including the Transmission Control Protocol (TCP), to secure higher layer protocols, such as the Hypertext Transfer Protocol (HTTP) and the Internet Message Access Protocol (IMAP). TLS provides confidentiality, integrity, and authentication. Confidentiality and integrity are provided by symmetric encryption, which ensures that nobody is able to read the data being forwarded. Authentication verifies the identity of the communicating parties via certificates and cryptographic signatures [8].

It consists of two layers, the TLS Record Protocol and five subprotocols. The TLS Record Protocol sits directly atop TCP. Its responsibility is to fragment, optionally compress, authenticate, and encrypt the higher-level data before transmitting it over the connection [15]. The TLS Handshake Protocol runs over the Record Protocol, negotiating security settings, authenticating the peers, and determining a shared secret from which the session keys are derived [13]. A ClientHello message is sent from the client to the server to indicate which protocols, cipher suites, compression algorithms, and extensions the client supports. The server responds with a ServerHello, followed by a Certificate message containing the server's X.509 certificate chain [15]. The security of the session is then based on the chosen algorithms and the continued validity of the certificates received. An attacker can impersonate a legitimate server using a seemingly valid yet revoked certificate without raising protocol-level alerts [9].

The TLS protocol has had several variants based on the issues found in the previous version. An example would be TLS 1.0, based on SSL 3.0, to address previous incompatibilities and security issues [3]. TLS 1.1 added support for explicit IVs to reduce some CBC attacks and changed the handling of padding errors to reduce timing attacks [4]. TLS 1.3 removed the RSA key transport option and mandates the use of ephemeral key agreement cipher suites to provide perfect forward secrecy. This means that even if the server's long-term private key is later compromised, previously captured sessions remain indecipherable. All versions still rely on public key infrastructure (PKI) for authentication; if any certificate in the chain of trust is invalidated, the integrity of the entire authenticated session is affected. As more websites use TLS 1.3 (over 70% of surveyed websites validate to this standard by mid-2024), the volume of TLS-encrypted traffic has increased rapidly, making it more likely that issues will happen if the certificate validation path is broken [11].

2.2 Public Key Infrastructure and Certificate Lifecycle Management

Public key infrastructure (PKI) uses X.509 certificates as the trust model on which TLS certificates are issued, managed, and revoked. These bind a public key to an authenticated identity and are digitally signed by a Certificate Authority (CA) whose signature is trusted by all parties relying on the certificate [2]. Before establishing a TLS connection, the TLS client verifies the certificate provided by the TLS server to ensure it is signed by a trusted CA and that its binding is intact. Such verification is a prerequisite for the authentication guarantee of TLS [8].

Certificate lifecycle management, including the issuance, renewal, and revocation of certificates, is important when a certificate's private key has been compromised (e.g., the Heartbleed bug), a certificate has been wrongfully issued to an attacker (e.g., the DigiNotar incident), or a certificate is being used for phishing, man-in-the-middle, or other attacks [12]. A certificate owner or the issuing CA has the duty to notify relying parties of the revocation of a certificate if any of the conditions above occur prior to the expiration date. This is done through either a CRL or an OCSP response [5].

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

CRLs (Certificate Revocation Lists) are periodically published lists of revoked certificate serial numbers signed by the issuing CA [2], which are retrieved and cached by clients for certificate validation. OCSP avoids the need to download a whole list. With OCSP, instead, the OCSP client asks an OCSP responder the status of the certificate using its serial number, and receives a signed response indicating whether the certificate is still valid, revoked, or unknown [5]. Both OCSP and CRLs are dependent on external infrastructure and the freshness of data, which may not be available under the threat model described in the rest of this article. A closely related technology is Certificate Transparency (CT), which uses append-only public logs that collect all certificates issued by CAs, ensuring misissued and fraudulent certificates can also be detected rather than relying solely on revocation polling [12].

2.3 The TLS Handshake and Cryptographic Foundations

The TLS handshake protocol uses authenticated key exchange to create a shared secret, from which symmetric encryption keys are derived. Prior to TLS 1.3, RSA, finite field Diffie-Hellman (DH), or Elliptic Curve DH (ECDH) were used for key exchange. In TLS 1.3, the RSA option is no longer available, as the ephemeral DH variants (EC)DHE provide perfect forward secrecy. In the RSA-based key transport used in earlier protocol versions, the client generated a pre-master secret, encrypted it with the server's RSA public key, and transmitted it to the server, which decrypted it to derive the master secret. In a more efficient DH key exchange, the server sends the client a public key share signed with the server's long-term private key, and both parties derive a master secret from both public and private key shares [13].

TLS employs a range of cryptographic building blocks, including hash functions, digital signatures, message authentication codes, key exchange protocols, and symmetric encryption schemes [14]. Cipher suites encode the combination of these algorithms in a single identifier. Currently, 352 cipher suites are officially supported, targeting different applications and security levels. A suite name expresses the key exchange method, the authentication approach, the encryption algorithm, and the MAC construction. For example, TLS_ECDH_RSA_WITH_RC4_128_SHA specifies that ECDH will be used for key exchange, RC4 for encryption, and SHA for the MAC [15]. The choice of cipher suite directly affects the security properties of the session. Suites that lack authentication or encryption are retained only for specific constrained use cases and are not recommended for general use. The handshake also carries extensions through which clients advertise supported signature algorithms, DH groups, and key shares for each supported group [14]. The integrity of the handshake, and specifically the certificate exchanged within it, depends on the revocation infrastructure operating correctly at the moment of validation.

2.4 Structural Weaknesses in Revocation Distribution

Despite the existence of these protocols, revocation in practice is difficult. CRLs are often large. Their size introduces bandwidth overhead and retrieval latency that can be significant, particularly for clients that must fetch them frequently [2]. OCSP reduces this overhead for single queries but requires the OCSP responder to be reachable, while clients must decide whether to accept the certificate (even though its status can't be validated) or to reject the certificate and close the connection [12].

The rise of traffic interception intermediaries further complicates the problem. For Content Delivery Networks (CDNs) to serve content over HTTPS, it is common for content providers to share their certificate's private key with the CDN or allow the CDN to obtain certificates for their domain. This changes the notion of the "end" in end-to-end encrypted TLS sessions, pushing the trust boundary outward and introducing additional parties whose certificate management practices affect revocation reliability [13]. Enterprise network gateways that intercept HTTPS traffic by terminating and re-creating TLS sessions introduce similar complications: the certificate presented to the end client is issued by the intercepting intermediary rather than the origin server, and the revocation status of that intermediary's certificate becomes a separate dependency in the validation chain [8].

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

These challenges become even more complex due to the diversity of TLS deployment environments. Consumer browsers on unrestricted networks operate under very different conditions than embedded controllers, network appliances, or enterprise clients behind egress-filtering firewalls [1]. TLS is application-independent and is used to protect communications ranging from web browsing to virtual private networks to IoT device management and energy management systems [15]. Each environment places different demands on revocation infrastructure and interprets revocation failures differently. A timeout waiting for an OCSP response may be handled permissively by a browser and strictly by a regulated financial application, even when the underlying failure condition is identical. This variability in response behavior makes it essential to classify revocation failures by their intrinsic properties rather than by the validation outcomes they happen to produce in specific environments [12].

3. Motivation for Taxonomy-Based Analysis

3.1 The Conflation Problem

A persistent obstacle to rigorous analysis of revocation failures is the conflation of failure conditions with validation responses. Much of the existing literature frames revocation failures in terms of whether a client should accept or reject a certificate when revocation status cannot be determined. This framing treats the validation outcome as the primary object of analysis while leaving the underlying failure condition underspecified [12]. The consequence is that two incidents arising from entirely different root causes may be grouped together because they both produced soft-fail behavior, while two incidents with identical root causes may be treated as distinct because one was handled permissively and the other strictly.

This conflation is not merely a classification problem. It shapes how incidents are diagnosed and how architectural decisions are justified. When a revocation failure is characterized primarily by the validation outcome it produced, remediation efforts naturally target client behavior rather than the infrastructure, data, or policy conditions that caused the uncertainty in the first place [9]. Downgrade attacks provide an instructive analogy: a taxonomy of TLS downgrade attacks classifies attacks by the protocol element targeted, the vulnerability type, the attack method, and the resulting damage, rather than by how clients happened to respond to the degraded session [9]. The same analytical discipline is necessary for revocation failures.

3.2 Diversity of Operational Contexts

TLS operates across an exceptionally wide range of environments. General-purpose guidelines for TLS configuration acknowledge that while interoperability requirements suit most deployments, constrained implementations exist where security is required but broad interoperability is not, including embedded controllers and network infrastructure devices such as routers [1]. The 5G core network uses TLS to secure communication between Network Functions over REST/HTTP2 interfaces, and whether TLS is enforced at all depends on operator decisions rather than mandatory standardization [10]. In energy management systems, TLS has been implemented alongside protocols such as Modbus to secure communications between distributed energy resources and control systems, where computational constraints and intermittent connectivity create additional pressures on certificate validation infrastructure [15].

IoT deployments add another layer to this diversity. Devices in IoT ecosystems frequently operate with constrained computational resources, intermittent connectivity, and minimal human oversight [6]. Revocation checking mechanisms that impose acceptable latency and bandwidth costs in a data center context may be entirely infeasible on a battery-powered sensor. A failure that is operationally inconsequential in one environment can represent a critical trust breakdown in another. Classification based on observable properties of the failure itself, rather than on the assumptions of any particular deployment model, allows the taxonomy to remain applicable across this range [1].

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The emergence of post-quantum threats adds a further dimension to this diversity. Quantum computers can efficiently break the integer factorization and discrete logarithm problems on which current TLS key exchange and signature algorithms are based [14]. The future advancements pose a threat, as the enhanced capabilities could empower the defaulters with forging signatures and falsely authenticating themselves for users to access encrypted keys and read secret messages from past TLS sessions. Thus, there is a need for building certificate-level trust, which also includes revocation infrastructure that is capable of governing certificate validity, to ensure that false authentication can be avoided even if the cryptographic assumptions underpinning it evolve. A taxonomy grounded in structural failure properties rather than specific cryptographic mechanisms retains its applicability across this transition [12]. A systematic review of post-quantum migration efforts found that existing PKI and the infrastructure for managing the certificate lifecycle, including revocation mechanisms, represent one of the most operationally complex components to migrate. It is because changes to certificate formats and signature algorithms affect the interpretability of revocation artifacts by legacy clients during the transition period [19].

3.3 Foundation for Comparative Analysis

Beyond these isolated incidents, a unified taxonomy can be beneficial to reason about revocation failure at the scale of the Internet. With TLS 1.3 rapidly being adopted across the Internet (encrypted traffic alone constitutes over 94% of traffic as of 2024 on the most used Internet platforms [11]), the number of TLS connections requiring valid revocation checking has continued to grow. Analysis of 458,000 different sites showed that even minor improvements to configuration quality can have a material impact on the security of a population of end-entities, even at the cost of reachability [8]. To identify which classes of revocation failure are most frequent, most persistent, and most likely to produce divergent trust decisions across large populations of clients, one must have a framework that allows one to compare different events in such dimensions consistently [12].

Dimension	Core Question	Analytical Role
Failure Origin	Where does the breakdown occur?	Identifies cause and propagation path
Temporal Validity	Is the revocation signal current?	Captures freshness degradation and trust lag
Scope of Impact	How many parties are affected?	Measures reach and amplification risk
Trust Determinism	Are outcomes consistent across clients?	Characterizes predictability of trust decisions

Table 1: Summary of Taxonomy Dimensions and Their Analytical Roles [1, 9, 12]

4. Taxonomy Architecture and Design Principles

4.1 Dimensional Orthogonality

The four dimensions of this taxonomy are constructed to be orthogonal. Each captures a distinct aspect of revocation failure behavior, and none can be derived from the others. Failure origin is the causal locus of a failure. Temporal validity refers to whether or not these revocation signals are indicative of the trusted status of a given certificate at the time of validation. Scope is the size of the effect of the validation failure, and trust determinism is the congruence of validation outcomes for likewise situated clients with the same uncertainty regarding revocation. These dimensions interact in practice but remain analytically separable. A network-induced failure with systemic scope is a distinct classification from an infrastructure-induced failure with localized scope, even if both result in

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

unavailable revocation signals. TLS downgrade attack taxonomies demonstrate the analytical value of this kind of multidimensional classification: by independently varying the element targeted, the vulnerability type, the method used, and the damage caused, the taxonomy captures relationships between attacks that would be invisible if only a single dimension were used [9]. The same principle applies to revocation failures. It also aligns with how TLS interception mechanisms are analyzed: by separating the motivations for interception from the technical methods and the security consequences, researchers can reason more precisely about the conditions under which interception becomes a risk [13].

4.2 Implementation Agnosticism

A defining design principle of this taxonomy is that classification attributes must remain stable across protocol versions and deployment models. TLS has evolved from SSL 2.0 through TLS 1.3, with each version introducing changes to handshake mechanics, cipher suite support, and extension handling [3, 11]. TLS 1.3 in particular made significant structural changes, including the removal of RSA key transport, the encryption of more of the handshake process, and the elimination of non-forward-secret cipher suites [13]. These changes affect traffic analysis and interception, but they do not alter the fundamental structure of certificate-based authentication or the mechanisms through which revocation is communicated. A taxonomy built around the observable properties of failure behavior remains stable across these protocol changes [5].

The post-quantum transition reinforces this requirement. Proposals for post-quantum TLS differ across security properties, hardness assumptions, application scenarios, TLS versions, and experimental setups [14]. A taxonomy of revocation failures anchored to the specific cryptographic assumptions of current TLS versions requires revision as post-quantum algorithms are standardized and deployed. By focusing on structural failure properties rather than mechanism internals, the taxonomy remains applicable regardless of which cryptographic building blocks underpin the certificates being validated.

4.3 Separation of Failure and Response

The taxonomy explicitly excludes validation behaviors such as soft-fail and hard-fail as primary classification attributes. This exclusion is important, as identical failure conditions will give different validation outcomes depending on client configuration, timing, and context. In opportunistic security deployments, systems configured to fail open will accept connections even when the TLS upgrade fails entirely, let alone when the revocation status is uncertain [9]. Enterprise TLS interception gateways introduce their own validation logic that may diverge substantially from the endpoint client's native behavior, further multiplying the range of possible responses to the same underlying failure condition [13]. Treating validation behavior as part of the failure definition would make classification contingent on observer-side variables rather than on the failure itself. Section 6 addresses validation behavior explicitly, positioning it as a secondary effect.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

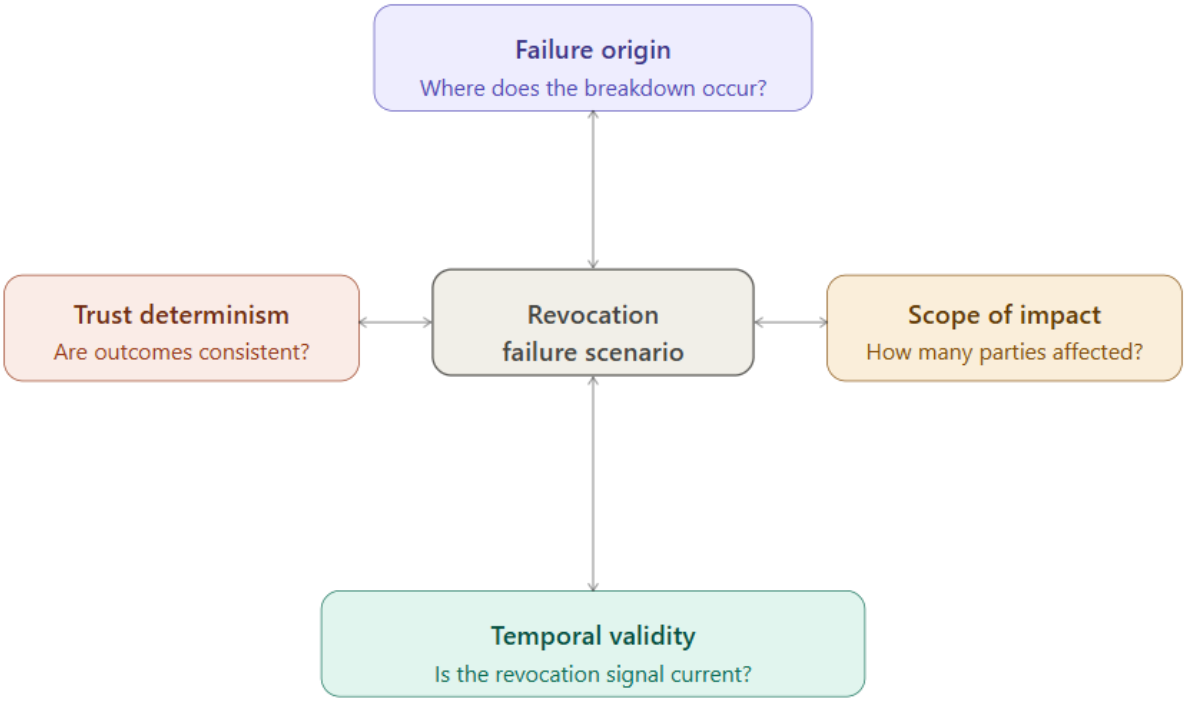

Figure 1: Taxonomy Dimensional Model (Revocation Failure Scenario) [1, 9, 12]

5. Formal Taxonomy of TLS Revocation Failure Scenarios

5.1 Failure Origin

Failure origin characterizes where in the revocation signaling path the breakdown occurs. This dimension focuses on causal structure rather than observable symptoms. Failures with similar surface presentations can arise from fundamentally different root conditions and therefore exhibit different persistence and propagation behavior [12]. Network-induced failures arise when connectivity constraints prevent relying parties from accessing revocation information at validation time. These include routing failures, network partitions, filtering by intermediary devices, and egress access control restrictions that block communication with CRL distribution points or OCSP responders [1, 5]. The revocation mechanisms remain operationally intact; the failure is one of reachability. TLS servers are required to be configured with certificates issued by CAs that publish revocation information through CRL or OCSP, but this requirement presupposes that clients can reach those publication endpoints [1]. In environments with restricted egress, this presupposition frequently does not hold. Network-induced failures are often transient but can recur unpredictably, particularly in segmented enterprise networks or constrained deployment contexts such as IoT ecosystems where connectivity is intermittent by design [6].

Infrastructure-related failures arise from disruptions within the revocation distribution ecosystem. For instance, outages of OCSP responder services, misconfigured distribution endpoints, synchronization failures between CAs and publication systems, and overload conditions are some examples of revocation distribution ecosystem disruptions [12]. A distinguishing property of infrastructure-induced failures is that they may persist even when network connectivity is otherwise healthy. Their effects can also scale rapidly when revocation infrastructure is shared across many relying parties. A performance evaluation of TLS in the 5G core found that TLS introduces measurable overhead in control-plane communications, and infrastructure components under high load present additional latency and failure risks [10]. When OCSP responders experience similar overload; the resulting failures affect every client that depends on them for revocation verification [12].

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Data-induced failures occur when revocation artifacts are present but cannot be reliably interpreted. Malformed CRLs, improperly encoded OCSP responses, missing metadata, and signature verification failures are representative examples [2, 5]. These failures occupy an analytically distinct position because they make it difficult to tell availability and correctness apart. Revocation information exists and is technically accessible, yet its trustworthiness remains ambiguous. Longitudinal measurement studies have found that a substantial proportion of TLS certificates in active deployment contain incorrect or inconsistent identity information, indicating that data quality issues in PKI artifacts are not hypothetical edge cases but observed operational conditions [12]. Data-induced failures tend to manifest sporadically and are difficult to diagnose without detailed inspection of the revocation artifact itself.

Policy-induced failures arise from operational, regulatory, or environmental constraints that limit revocation reachability or interpretation by design. These are not technical malfunctions but deliberate configurations. Constrained TLS implementations in embedded controllers and network infrastructure devices are explicitly acknowledged as contexts where a subset of full PKI capabilities may be acceptable [1]. In 5G core deployments, the decision of whether to enforce TLS at all, let alone enforce revocation checking, is left to the system operator rather than mandated by standardization [10]. In energy management systems using Modbus/TLS, device capabilities may constrain the computational overhead of full certificate validation, including revocation checking [15]. Policy-induced failures are frequently persistent by intent and can interact with other failure origins in compounding ways, particularly when network or infrastructure failures occur within an environment already operating under constrained revocation access [6, 12].

Origin Type	**Primary Cause**	**Persistence**	**Diagnostic Visibility**
Network-Induced	Connectivity disruption	Typically transient	Moderate
Infrastructure-Induced	Revocation system failure	Variable, potentially extended	High
Data-Induced	Artifact integrity or encoding failure	Sporadic	Low
Policy-Induced	Operational or regulatory constraint	Persistent by design	Context-dependent

Table 2: Failure Origin Categories with Characteristic Properties [1, 2, 6, 10]

Implementation Scenario - Enterprise and IoT: In enterprise network environments, policy-induced failures frequently arise from egress-filtering firewalls that block outbound connections to CRL distribution points or OCSP responders operated by external CAs. When an enterprise gateway intercepts and re-terminates TLS sessions, it introduces a secondary revocation dependency: the revocation status of the intermediary's certificate must itself be verifiable, yet that status may be subject to the same egress restrictions affecting end clients. In IoT deployments, policy-induced and network-induced failures commonly co-occur. A sensor operating on a low-power wide-area network (LPWAN) may have no persistent IP-layer connectivity to external PKI infrastructure, making any form of online revocation checking operationally infeasible. A systematic review of IoT authentication schemes confirms that certificate-based validation mechanisms are frequently simplified or disabled in resource-constrained device classes precisely because the operational overhead of maintaining live revocation access exceeds device capability constraints [20]. In both scenarios, the failure origin

classification remains analytically stable regardless of how individual clients respond to the resulting revocation uncertainty [1, 6, 10].

5.2 Temporal Validity of Revocation Signals

Temporal validity captures whether revocation information accurately reflects the current trust state of a certificate at the moment validation occurs. Revocation is inherently time-sensitive, and trust decisions may rely on information whose relevance has degraded since it was produced [12]. CRLs are published on a scheduled basis, meaning that a revocation event occurring between publication cycles will not appear in the current CRL until the next update [2]. OCSP responses carry explicit validity windows, and responses cached beyond those windows represent stale data even if they are technically accessible [5].

Unavailable revocation signals represent a complete absence of revocation information at validation time. Whether caused by an unreachable endpoint, a missing distribution path, or a blocked access channel, unavailability introduces immediate trust uncertainty. The relying party cannot distinguish between the absence of a revocation record and the absence of the data that would contain such a record. OCSP is designed to provide per-certificate status on demand, but this assumption only holds while the responder remains reachable [5]. In environments where outages of security infrastructure have been documented, we consider unavailability a practical risk rather than a theoretical one [8, 12].

Stale revocation signals are present at validation time but exceed acceptable freshness bounds. Staleness is particularly problematic because validation may complete without any explicit error condition. Systems accept the stale signal as valid, which means they are using information that is no longer correct about the revocation state. Private key compromises, such as those enabled by the Heartbleed vulnerability, may result in certificates that were valid at the time of last CRL publication but have since been revoked. A client relying on a cached pre-revocation CRL would accept such a certificate without warning [12]. The failure is invisible to the relying party and produces no observable anomaly in the protocol exchange.

Delayed revocation signals represent cases where a revocation event has occurred but has not yet propagated to all relying parties. This reflects an inherent lag in revocation publication and distribution processes. During the propagation window, some clients may operate with accurate current-state information while others continue to rely on pre-revocation data. The duration of this window depends on CRL publication frequency and OCSP cache lifetimes, both of which involve deliberate trade-offs between freshness and infrastructure load [2, 5]. Indeterminate temporal states arise when revocation metadata is insufficient to assess freshness conclusively. Missing timestamps, ambiguous validity intervals, and inconsistent update indicators prevent relying parties from determining whether revocation information is current, delayed, or stale [12].

Implementation Scenario - Enterprise and IoT devices: In enterprise deployments, stale revocation signals may be encountered if OCSP responses are aggressively cached to reduce the OCSP responder load. An example is where a gateway used by endpoint clients in an enterprise environment may cache OCSP responses for an unreasonably long period of time, so a response with a valid status may be encountered when the certificate was subsequently revoked. The indeterminate state is common in IoT. Devices that seldom validate certificates directly or only synchronize PKI state with the network during maintenance windows will experience long-lived trust gaps with out-of-date revocation data that cannot be avoided or worked around without communicating with the network. This persistent trust gap is invisible at the device level and difficult to detect at the network operations level without dedicated observability infrastructure [2, 12, 17].

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

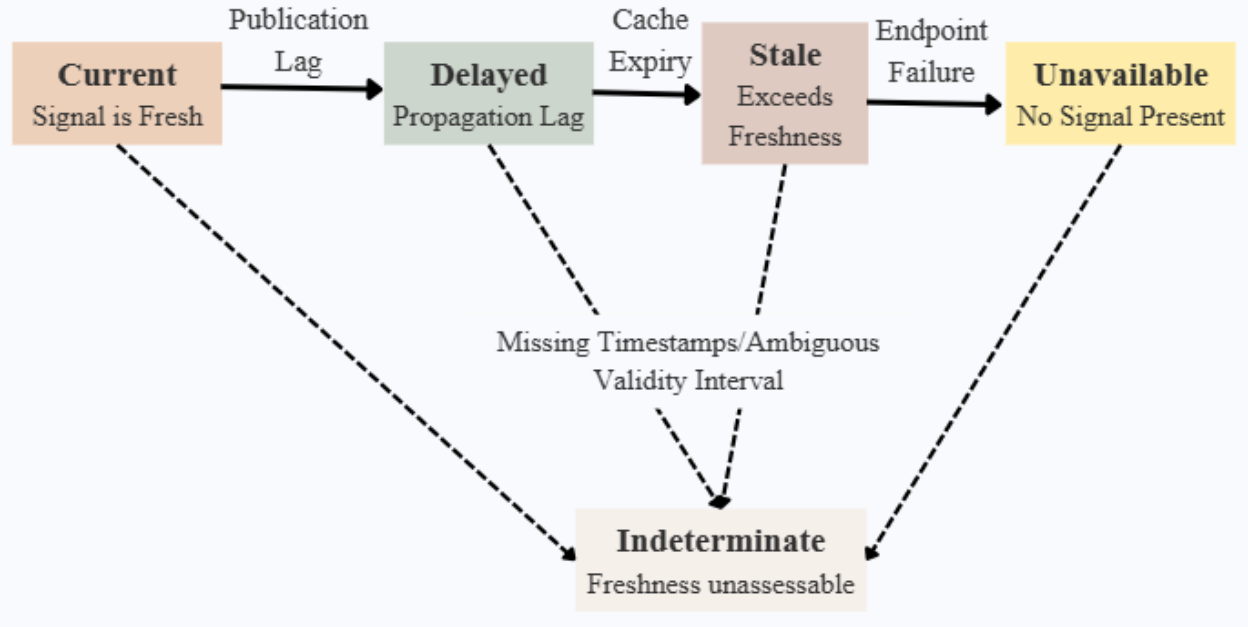

Figure 2: Temporal Validity State Transitions [2, 5, 12]

5.3 Scope of Impact

The scope of impact is the number of relying parties and trust consequences that a particular failed revocation will affect in the TLS ecosystem. Two identical failures occurring at the same time and with the same attributes can have vastly different impacts due to infrastructure dependencies [12]. Localized failures may affect only a small number of clients' services or network paths. They can include the failure of a single isolated network link, a particular client configuration, or limited infrastructure failures and are generally localized, remaining hidden from the outside world. Minimal TLS server implementations for embedded controllers and network appliances, for example, are a population where local failures are expected and considered a normal operating condition [1]. In energy management systems, a single endpoint failure in device revocation over Modbus/TLS may not impact other devices on the same network segment [15].

Segmented failures occur for a well-defined group of entities that share a common property, such as geographic region, platform type, common network egress path, or common trust domain. In this case, the certificate is partially inconsistent, as it may be validated in one segment but not in others. This feature makes it challenging to apply some kinds of analysis to the incident since there is no single global point of failure. In every IoT deployment, devices deployed in a given geographical area or under the policy of a given network operator may face different constraints on device revocation [6, 10]. Failures are systemic when they impact many customers and services, often due to dependencies or shared infrastructure. For example, if a widely used OCSP responder, a commonly trusted intermediate CA, or a common validation code path is impacted, the failure is systemic. The only thing common to all of these is rapid amplification. Each of them allows one manipulation to lead to trust implications for Internet-scale populations before any compensation can be made. The 460,000 domains measured have shown the size of configuration-level effects on reachability and security in large endpoint populations [8]. In addition, that scale effect is exacerbated by CDNs: when a single operator controls certificates for many content sources, a failure to propagate revocation into that operator's certificate-management infrastructure can multiply the impact across all domains served by that CDN [12, 13].Internet-wide TLS scanning has further demonstrated that the distribution of revocation mechanism support is highly uneven across certificate populations, with some CA hierarchies accounting for disproportionately large shares of revocation infrastructure load, meaning

that a failure concentrated in one CA's distribution infrastructure can produce systemic effects across a much broader certificate population than its nominal footprint would suggest [16].

Implementation Scenario - Enterprise and IoT: Because this is an enterprise scenario, what would otherwise be tolerable as an infrastructure outage in a single zone can lead to an enterprise-wide degradation of the business in the absence of an OCSP responder for a shared intermediate CA. In particular, this is true for all internal services, remote access clients, and partner-facing APIs that rely on a CA's revocation infrastructure. Many IoT devices use the same provisioning pipeline with trust roots signed by firmware. This means that trust anchor mismatches can quickly move from being a mismatch in a single piece of data to a mismatch in an entire family of devices deployed in the field, potentially thousands in number [1, 6, 12].

5.4 Trust Determinism

Trust determinism characterizes the consistency and predictability of trust outcomes produced by a revocation failure across relying parties. This dimension addresses whether similarly situated clients facing the same revocation uncertainty arrive at the same trust decision [12]. Deterministic trust failures produce consistent validation results across clients regardless of client-specific variables. The failure condition reliably produces either acceptance or rejection. Even when the outcome is unfavorable, determinism supports reasoning about the system-wide trust posture because the outcome is anticipatable. Hard-fail enforcement policies, where implemented consistently, produce deterministic rejection under defined failure conditions. This predictability lets operators think clearly about how those conditions affect security [12].

Conditionally deterministic failures yield predictable outcomes only under specific conditions. Validation behavior may be consistent within a defined freshness window but diverge once temporal thresholds are crossed. A downgrade attack taxonomy identifies analogous conditional behavior: some attacks succeed only when the communicating parties are in a particular negotiation state, and the attack becomes inert once that condition no longer holds [9]. TLS interception mechanisms introduce a related form of conditional determinism: validation behavior at the intercepting gateway may be consistent, while the behavior of the endpoint client behind the gateway diverges depending on how the re-created session is constructed and whether revocation checking is performed independently at each layer [13]. Empirical study of revocation and replacement practices in production environments has found that this propagation delay is frequently non-trivial, with operators taking hours to days to complete the revocation and re-issuance cycle following a confirmed compromise event, and with some replacement certificates observed to carry residual metadata inconsistencies that further extend the effective delay window [18]. These findings establish that delayed revocation is a systemic operational pattern rather than an exceptional occurrence.

Indeterminate trust failures produce inconsistent outcomes across otherwise comparable relying parties. Some clients accept a certificate; others reject it; others classify its status as unknown. Indeterminacy emerges from the interaction of multiple factors: temporal validity states, caching behavior, client-specific interpretation logic, and timing relative to revocation publication events. Empirical studies of certificate validation behavior have found substantial variation across client implementations in how revocation failures are handled, with some implementations silently accepting certificates when OCSP is unreachable while others enforce rejection [12]. Post-quantum migration further complicates this picture: as new signature algorithms are introduced; clients that have not yet adopted the relevant certificate validation logic may interpret signatures differently, adding another source of outcome divergence [14].

Implementation Scenario - Enterprise and IoT: Trust determinism has operational impacts on security monitoring and incident response in enterprise and IoT environments. In enterprise environments, multiple TLS intermediaries (i.e., proxies, load balancers, and API gateways) may be in use that enforce certificate revocation independently of one another, which can lead to an

indeterminate outcome for the same certificate when traversed through different paths. Security operations teams may monitor the validation outcome and act upon it without comprehending the path dependency, which causes indeterminate failures to be misinterpreted as deterministic failures. In the Internet of Things, firmware-embedded trust anchor configurations (which may differ across alternate generations of devices or across geographic regions) convert what would otherwise be a conditionally deterministic failure across a homogeneous fleet into an indeterminate failure across a heterogeneous fleet, with inconsistent trust decisions for the same certificate and revocation condition from different device cohorts [9, 12, 13].

Determinism Class	Outcome Consistency	Trigger Stability	Analytical Complexity
Deterministic	Consistent across clients	Stable across conditions	Low
Conditionally Deterministic	Consistent within conditions	Unstable at thresholds	Moderate
Indeterminate	Inconsistent across clients	Inherently unpredictable	High

Table 3: Trust Determinism Categories and Characteristics [9, 12, 14]

6. Validation Outcomes as Secondary Effects

6.1 The Misclassification Problem

Validation outcomes are commonly the primary focus of revocation failure discussions, yet treating them as defining properties of failures rather than as responses to them leads to analytical confusion. Soft-fail behavior permits certificate acceptance when revocation status cannot be confirmed. Hard-fail behavior requires confirmed non-revocation before proceeding. A significant proportion of SMTP servers configured to support TLS operate in opportunistic security mode, meaning they fail open and revert to unauthenticated plaintext when the TLS upgrade fails for any reason [9]. This behavior reflects a policy preference for availability over security, not a distinct category of revocation failure. Classifying the underlying failure by this policy preference would conflate the failure condition with the response to it [12].

The same failure condition can produce different validation outcomes across clients even within a single incident. A network-induced failure rendering an OCSP responder unreachable for a brief interval may result in acceptance at one client due to a permissive caching policy, rejection at another due to hard-fail enforcement, and cached validity at a third due to a recently refreshed response still within its validity window [5, 12]. These three outcomes do not represent three different failure scenarios. They represent three interpretations of the same condition shaped by local implementation choices. Classifying them as distinct failures would fragment the analysis of a single event into incommensurable categories. TLS interception intermediaries add a further layer to this complexity: an enterprise gateway may perform revocation checking independently of the endpoint client, meaning that the same certificate may be evaluated twice by parties with different enforcement configurations [8, 13].

6.2 Causal Relationship Between Taxonomy Dimensions and Outcomes

The relationship between taxonomy dimensions and validation outcomes is causal rather than definitional. Failure origin shapes which aspect of the revocation signaling path is disrupted. Temporal validity determines how much the revocation signal has degraded at the time of validation. Scope defines how many relying parties are simultaneously exposed to the same failure condition.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Trust determinism captures whether those parties converge on consistent outcomes. Validation behavior emerges from this combination rather than preceding it [9,23].

A failure characterized by infrastructure-induced origin, stale temporal validity, systemic scope, and indeterminate trust determinism is precisely the kind of condition that produces divergent soft-fail and hard-fail responses across a large client population. Certificate revocation failures in the TLS ecosystem have been shown to vary substantially in their behavioral consequences across different client implementations, reflecting exactly this kind of interaction between failure conditions and implementation-specific response logic [12,21]. The process of negotiating a cipher suite in the TLS handshake provides an analogy: the outcome of negotiation depends on the interaction of client and server capabilities, not on either party's preferences in isolation [15]. Revocation validation outcomes follow the same logic.

6.3 Normative Neutrality and Policy Applicability

This analytical separation preserves the normative neutrality of the taxonomy. Debates over soft-fail versus hard-fail enforcement involve genuine trade-offs between availability and security assurance that vary across deployment contexts. A regulated financial application handling sensitive transactions faces different risk tolerance than a content delivery server [1, 7]. The tension between security and accessibility was illustrated during the standardization of TLS 1.3, when some industry participants pushed for means to facilitate traffic inspection while privacy advocates raised concerns about weaknesses introduced by intermediaries [13,22]. A taxonomy that embeds a preference for one enforcement approach would compromise its applicability across environments with legitimately different operational requirements. By positioning validation behavior as a downstream effect, the taxonomy supports rigorous failure analysis without taking a position on how those conditions should be handled in any particular context.

6.4 Analytical Benefits of Decoupling

When an analyst uses this taxonomy to analyze an incident, the classification of the failure remains unchanged regardless of how different clients respond to it. This makes comparative analysis of incidents tractable [25]. Two incidents that produced predominantly soft-fail outcomes but differed in scope from localized to systemic are recognized as analytically distinct despite their similar surface presentations. Conversely, two incidents that produced mixed soft-fail and hard-fail responses may be recognized as arising from identical failure conditions, differing only in the distribution of client enforcement configurations. Neither comparison is straightforward when validation behavior serves as the primary classification attribute [9, 12].

The decoupling also supports forward-looking analysis as the TLS ecosystem evolves. The transition toward post-quantum cryptography will require changes to certificate formats, signature algorithms, and key exchange mechanisms [14]. These changes may introduce new classes of data-induced and policy-induced revocation failures as legacy clients encounter certificates they cannot fully interpret [24]. The taxonomy, being anchored in structural failure properties rather than specific cryptographic mechanisms, provides a stable framework for classifying these emerging failure classes as they arise, without requiring revision of its foundational dimensions [12, 14].

Conclusion

Certificate revocation failures are not uniform events. They arise from distinct causal conditions, manifest across different temporal and infrastructural dimensions, and produce trust outcomes ranging from entirely predictable to deeply inconsistent. The taxonomy presented in this article addresses this complexity by classifying revocation failures along four orthogonal dimensions: failure origin, temporal validity of revocation signals, scope of impact, and trust determinism. Together, these dimensions provide a structured vocabulary for characterizing revocation failures as phenomena in

their own right, independent of the validation responses they produce in specific environments. Decoupling failure classification from validation behavior preserves analytical neutrality and remains applicable across the full diversity of TLS deployments, ranging from constrained embedded systems and IoT infrastructure to CDN-mediated architectures and post-quantum-capable environments. Localized, transient failures are analytically distinguishable from systemic, persistent ones. Deterministic failures are distinguishable from indeterminate ones that silently undermine trust consistency at scale. This precision matters because revocation is not a solved problem. As encrypted communications continue to expand across critical sectors and TLS underpins an ever-widening range of services, the structural reliability of revocation infrastructure carries greater consequence. The taxonomy offered here is intended as a foundational analytical tool that complements resilience-oriented models, supports clearer incident analysis, and provides a stable basis for future architectural and policy-oriented work in PKI and certificate lifecycle management.

References

[1] Tim Polk, Kerry McKay, and Santosh Chokhani, "Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations," NIST Special Publication 800-52, 2014. Available: https://cdn.atraining.ru/docs/NIST.SP.800-52r1.pdf

[2] D. Cooper et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," Internet Engineering Task Force, RFC 5280, May 2008. Available: http://www.ietf.org/rfc/rfc5280.txt

[3] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," Internet Engineering Task Force, RFC 2246, January 1999. Available: http://www.ietf.org/rfc/rfc2246.txt

[4] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," Internet Engineering Task Force, RFC 4346, April 2006. Available: http://www.ietf.org/rfc/rfc4346.txt

[5] S. Santesson et al., "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," Internet Engineering Task Force, RFC 6960, June 2013. Available: http://www.ietf.org/rfc/rfc6960.txt

[6] Yashvant Dev and Kismat Chhillar, "A Layered Security Perspective on the Internet of Things Ecosystem: Threat Taxonomy, Vulnerabilities, and Mitigation Strategies," 2025. Available: https://www.ijcsejournal.org/wp-content/uploads/2025/12/A-Layered-Security-Perspective-on-the-Internet-of-Things-Ecosystem-Threat-Taxonomy-Vulnerabilities-and-Mitigation-Strategies.pdf

[7] Hiren Parmar and Atul Gosai, "Analysis and Study of Network Security at Transport Layer," International Journal of Computer Applications, vol. 121, no. 13, 2015. Available: https://www.researchgate.net/publication/281978834

[8] Mat Phillips, "TLS Filter: An Application-Level Firewall for Transport Layer Security," Final Report, June 2014. Available: https://kevincurran.org/papers/Thesis%20-%20Application%20Level%20Firewall.pdf

[9] Eman Salem Alashwali and Kasper Rasmussen, "What's in a Downgrade? A Taxonomy of Downgrade Attacks in the TLS Protocol and Application Protocols Using TLS," in Proc. International Conference on Security and Privacy in Communication Systems, Springer, 2018, pp. 468–487. Available: https://arxiv.org/pdf/1809.05681

[10] Oliver Zeidler et al., "Performance Evaluation of Transport Layer Security in the 5G Core Control Plane," in Proc. 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2024, pp. 78–88. Available: https://dl.acm.org/doi/pdf/10.1145/3643833.3656140

[11] Jiuxing Zhou et al., "Challenges and Advances in Analyzing TLS 1.3-Encrypted Traffic: A Comprehensive Survey," Electronics, vol. 13, no. 20, 2024. Available: https://www.mdpi.com/2079-9292/13/20/4000

[12] Hyunsoo Kwon et al., "Certificate Revocation in the TLS Ecosystem: A Survey," ACM Computing Surveys, vol. 58, no. 7, 2026, pp. 1–36. Available: https://dl.acm.org/doi/pdf/10.1145/3785653

[13] Xavier de Carné de Carnavalet and Paul C. van Oorschot, "A Survey and Analysis of TLS Interception Mechanisms and Motivations: Exploring how end-to-end TLS is made 'end-to-me' for web traffic," ACM Computing Surveys, vol. 55, no. 13s, 2023, pp. 1–40. Available: https://dl.acm.org/doi/pdf/10.1145/3580522

[14] Nouri Alnahawi et al., "A Comprehensive Survey on Post-Quantum TLS," IACR Communications in Cryptology, 2024. Available: https://inria.hal.science/hal-04845617/file/1-2-6.pdf

[15] Matheus K. Ferst et al., "Implementation and Analysis of a Secure Communication with SunSpec ModBus and Transport Layer Security Protocols for Short-Term Energy Management Systems," IEEE Access, 2025. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=11028069

[16] Markus Sosnowski et al., "An internet-wide view on https certificate revocations: observing the revival of CRLs via active TLS scans," In 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2024. Available: https://www.net.in.tum.de/fileadmin/bibtex/publications/papers/sosnowski2024certificates.pdf

[17] Aparna Tiwari and Dinesh Kumar, "A dynamic fuzzy-based blockchain transport layer security algorithm for attack detection in distributed dynamic honeypot systems," Proceedings of the Indian National Science Academy, 2025. Available: https://link.springer.com/article/10.1007/s43538-025-00576-y

[18] David Cerenius et al., "Trust issue(s): Certificate revocation and replacement practices in the wild," In International Conference on Passive and Active Network Measurement, Cham: Springer Nature Switzerland, 2024. Available: https://www.diva-portal.org/smash/get/diva2:1846836/FULLTEXT01.pdf

[19] Christian Näther et al., "Migrating software systems toward post-quantum cryptography: a systematic literature review," IEEE access, 2024. Available: https://ieeexplore.ieee.org/iel8/6287639/6514899/10648683.pdf

[20] Jameel Shehu Yalli et al., "Authentication schemes for Internet of Things (IoT) networks: A systematic review and security assessment," Internet of Things, 2025. Available: https://www.sciencedirect.com/science/article/pii/S2542660524004104

[21] D. Puthiya, "Adaptive growth models in the era of enterprise AI transformation," Journal of Computational Analysis and Applications, vol. 31, no. 4, pp. 2796–2812, 2023.

[22] G. A. Ascanio, "Material performance and longevity in luxury kitchens: Architectural approaches to durability and use," Journal of International Crisis and Risk Communication Research, vol. 7, no. S9, pp. 3575–3584, 2024.

[23] F. K. Darteh, "Challenges in revenue and expenditure reporting: Implications for budget management," Sarcouncil Journal of Entrepreneurship and Business Management, vol. 2, no. 11, pp. 1–8, 2023.

[24] S. Surana, "The efficacy of internal controls and audit committees in mitigating financial risk: Perspectives from Indian corporate governance," Journal of International Crisis and Risk Communication Research, vol. 8, no. S10, pp. 377–386, 2025.

[25] A. Belhassen, "Machine learning for predictive maintenance: Fusing vibration sensor data and thermal imaging to forecast bearing failure," Sarcouncil Journal of Engineering and Computer Sciences, vol. 1, no. 3, pp. 9–18, 2022.