Network Working Group
Internet-Draft
Intended status: Informational
Expires: 8 November 2026

R. Anderson
S. Berkson
Jolly Roger Telephone Company
P. Askew

7 May 2026

Caller-ID Vouching and Vetting (CIDVV)
draft-anderson-askew-cidvv-00

Abstract

Caller-ID spoofing remains a significant problem in telephony, particularly across inter-domain and international call paths where identity frameworks may not be consistently applied.

This document defines Caller-ID Vouching and Vetting (CIDVV), a lightweight verification mechanism that uses short-lived signaling exchanges encoded within the Calling Party Number to confirm that a calling party can receive calls at the Asserted Caller-ID.

CIDVV is designed to operate across heterogeneous SIP and SS7/TDM networks without requiring new protocol extensions or persistent identity infrastructure. It relies on existing call routing behavior and intentionally leverages failure responses as a signaling mechanism, using failed call attempts as evidence of number control rather than successful call completion.

The mechanism improves resistance to Caller-ID spoofing by requiring demonstrable control of the Asserted Caller-ID, while remaining incrementally deployable and tolerant of intermediate network modification.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://cidvv.org/draft-anderson-askew-cidvv.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-anderson-askew-cidvv/.

Source for this draft and an issue tracker can be found at https://github.com/Jolly-Roger-Telephone-Company/cidvv-spec.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 November 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Caller-ID spoofing is widely used in fraudulent and nuisance calling, particularly in environments where calls traverse multiple administrative domains and heterogeneous network technologies. Although mechanisms such as STIR/SHAKEN provide cryptographic attestation of caller identity, their effectiveness is limited by partial deployment and challenges in international interconnection.

This document defines Caller-ID Vouching and Vetting (CIDVV), a mechanism that verifies caller identity through network reachability rather than asserted identity. CIDVV requires that a party asserting a Caller-ID be able to receive a return call at that number within the Validity Window.

CIDVV operates by encoding signaling information within the Calling Party Number and leveraging existing call routing behavior to perform a challenge-response exchange. The protocol does not require new SIP headers, protocol extensions, or changes to SS7 signaling, and is designed to function across mixed SIP and TDM networks.

CIDVV is incrementally deployable and does not require universal adoption to provide benefit. It tolerates modification of signaling by intermediate networks and relies on signaling calls and distinct failure-response behaviors traversing the network path.

CIDVV provides strong, real-time evidence of Caller-ID validity by requiring that a party asserting a Caller-ID be able to receive and respond to a return call at that number within the Validity Window. While it does not provide absolute identity assurance, it offers a practical and robust signal of trust in the presented identity.

CIDVV leverages two key elements of the existing telephone ecosystem:

* Existing routing databases and numbering plans, which provide authoritative routing ownership for telephone numbers.

* Digit sequences chosen to minimize conflict with valid numbering plans (e.g., "100" and "101").

The mechanism operates entirely within standard PSTN routing behavior and requires no media exchange.

2. Terminology

In this document, the term "Caller-ID" refers to the identity presented to users, while "Calling Party Number" refers to the signaling field used to convey that identity.

* *Alice*: The calling party and verifier. In vouching flows Alice asserts a number; in vetting flows Alice verifies Bob’s number.

* *Bob*: The called party. In vetting flows Bob is the owner whose number is being vetted.

* *Mallory*: An attacker attempting to spoof a Caller-ID.

* *CIDVV Platform*: A system that implements the vouching and vetting procedures defined in this document.

* *CIDVV-aware Network Element*: An SBC or intermediary that recognizes CIDVV signaling prefixes and interprets associated responses, but does not implement the full CIDVV platform logic.

* *Vouch*: The act of a CIDVV platform asserting that it has verified control of a telephone number through the challenge-response mechanism described in this document, which may consist of one or more verification calls. A successful vouch provides strong evidence that the calling party controls the Asserted Caller-ID.

* *Vet* (or *Vetting*): The process by which a CIDVV platform confirms the relevant party controls the Asserted Caller-ID via the two-call challenge-response sequence. Vetting may be performed by the number owner directly or on behalf of third parties such as Caller-ID branding services, Google Business Profiles, trade organizations, or enterprise trust programs.

* *Vouching Call*: A short verification call used in the CIDVV protocol. CIDVV defines a primary vouching call ("100") and an optional secondary vouching call ("101").

* *Successful Vouch*: A verification result indicating that a matching cache entry was found.

* *Unsuccessful Vouch*: A verification result indicating that no matching cache entry was found.

* *Verification Not Performed*: A condition where verification could not be completed due to system or network conditions.

* *Validity Window*: A short time interval during which CIDVV signaling state is considered valid for correlation purposes, typically on the order of 10 seconds.

* *Asserted Caller-ID*: The Caller-ID value that is being vouched or vetted (i.e., the number whose control is being verified). This is the value used as the basis for the CIDVV Token payload and as the cache lookup key in the tuple (Asserted-Caller-ID, CIDVV-Token).

Once successfully vouched, an Asserted Caller-ID may be referred to informally as a "vouched number," but the formal term used in this document is "Asserted Caller-ID."

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 and RFC 8174 when, and only when, they appear in all capitals, as shown here.

2.1. Motivation and Advantages

The CIDVV vouching and vetting mechanism is designed to operate with minimal new infrastructure while providing strong protection against Caller-ID spoofing. Its primary advantages are:

* *Leverages existing PSTN infrastructure*: The mechanism uses existing numbering plans and routing databases to direct calls without requiring additional infrastructure.

* *Strong anti-spoofing protection*: A successful vouch provides strong evidence that the Asserted Caller-ID is controlled by the legitimate owner, because only the real owner can generate the correct challenge-response sequence. Spoofed calls are typically rejected early, often resulting in failure responses such as 404 Not Found.

* *Visibility into spoofing activity*: Telephone number owners gain direct insight into how often (and from where) their numbers are being spoofed worldwide through logged vetting attempts.

* *Low signaling overhead*: The two short vetting calls replace what would otherwise be a completed fraudulent call, resulting in lower overall network load.

* *Full TDM/SS7 compatibility*: The mechanism works natively across legacy SS7 and ISDN networks. SIP is not required.

* *International applicability*: The solution functions across international boundaries without relying on country-specific frameworks.

* *Independent of STIR/SHAKEN*: It provides an effective alternative (or complement) in environments where STIR/SHAKEN is unavailable, not deployed, or insufficient.

* *Enterprise and service-provider flexibility*:

  - Enterprises can deploy their own CIDVV platform using open-source tools such as Kamailio or Asterisk.

  - Service providers or third-party vendors (e.g., TransUnion, TNS, First Orion, Hiya, Numeracle, Numhub, or others) can operate cloud-based vouching and vetting services.

  - Customers can easily switch between providers, fostering real competition and driving down costs.

This design lowers the barrier to entry and encourages broad adoption while avoiding the single points of failure and high coordination costs associated with centralized solutions. These properties make the vouching mechanism particularly suitable for service providers, enterprises, and end users who need robust Caller-ID validation today, using only existing telephone infrastructure.

3. Design Principles

CIDVV is designed to operate under the assumption that intermediate networks may normalize, truncate, or otherwise modify signaling information. The protocol therefore encodes all required signaling in a numeric Calling Party Number that can survive traversal of mixed SIP and SS7/TDM networks.

CIDVV relies on the ability to distinguish between classes of call rejection behavior (e.g., "busy" vs. "not found"), rather than requiring specific numeric response codes to be preserved end-to-end.

4. Number Normalization

All telephone numbers used in CIDVV operations MUST be normalized to a digit string as follows:

1. Remove any leading "+" or other punctuation.

2. Use the full E.164 representation (country code + national significant number) as a plain digit string.

No padding is performed. Truncation (when required for the 15-digit limit) always removes leading digits of the telephone number, preserving the rightmost digits.

4.1. Protocol Overview

CIDVV uses special Caller-ID prefixes to signal protocol operations:

* "100" prefix - Primary Verification Call

* "101" prefix - Secondary Verification Call / Vetting Call

CIDVV Calling Party Numbers are numeric signaling values carried in the Calling Party Number field. They are not represented as E.164 numbers and are shown without a leading "+" in this document.

Ordinary subscriber telephone numbers (e.g., +12125550100) are shown in E.164 format for clarity, while CIDVV signaling values (e.g., 10019495550199) are shown as digit strings.

CIDVV signaling Calling Party Numbers MUST fit within the 15-digit Calling Party Number limit commonly encountered in SS7 and ISDN networks. For this reason, CIDVV uses a three-digit prefix followed by a payload derived from the Asserted Caller-ID:

CIDVV-CPN (CIDVV Calling Party Number) = Prefix || Payload

where CIDVV-CPN means CIDVV Calling Party Number, Prefix is "100" or "101", and the payload is derived from the Asserted Caller-ID (normalized per Section Section 4).

For vouching operations, the payload is derived from the called number associated with the verification. For vetting operations, the payload may be derived from computed token values.

In the common case where the Asserted Caller-ID has 12 or fewer digits, the Payload is used in full, so the CIDVV-CPN is simply the three-digit prefix directly concatenated with the full Asserted Caller-ID digits.

If the resulting CIDVV-CPN would exceed 15 digits (i.e., the asserted Caller-ID has more than 12 digits), the leading digits of the asserted Caller-ID are removed until the total length is exactly 15 digits, consistent with SS7 and ISDN Calling Party Number constraints. This truncation preserves the rightmost digits of the telephone number, which typically provide greater distinguishing information between individual subscribers than leading digits.

A CIDVV-aware element generating a CIDVV verification call MUST apply this construction. A CIDVV platform MAY cache and compare the complete 15-digit CIDVV Calling Party Number (including the prefix) rather than reconstructing it for comparison.

Because CIDVV payloads may be truncated to the rightmost 12 digits, distinct telephone numbers can, in rare cases, produce identical payload values. Correlation is therefore additionally scoped by the called number and the Validity Window.

In such cases, multiple call attempts may be indistinguishable to the CIDVV platform and treated as a single correlation event. As a result, a successful verification may apply to more than one call attempt within the Validity Window.

CIDVV verification consists of observing the behavior of one or more verification calls using distinct CIDVV prefixes.

A successful vouch requires that a verification call using the "100" prefix produce the expected response behavior. Additional verification calls (e.g., using the "101" prefix) MAY be used to achieve higher assurance.

The expected behavior is:

* Calls using the "100" prefix MUST result in SIP 486 Busy Here. Any other response, timeout, call progression, or successful call completion MUST be treated as an unsuccessful vouch.

* Calls using the "101" prefix are expected to result in SIP 404 Not Found. However, in the context of an active vetting procedure, a "101" call carrying a valid token MAY result in SIP 486 Busy Here.

A CIDVV-aware network element MUST NOT treat a single response as sufficient evidence of a successful vouch unless it corresponds to the expected behavior for the "100" prefix.

Additional verification calls (e.g., using the "101" prefix) MAY be used to increase assurance but are not required for a valid vouch.

The two verification calls MAY be sent in any order or in parallel. Implementations MUST NOT assume ordering.

If either expected response is missing, altered, delayed, replaced by call progression, or inconsistent with the expected pattern, the result MUST be treated as unsuccessful or indeterminate.

CIDVV exchanges occur using short signaling dialogs and do not require media establishment.

CIDVV signaling is encoded entirely within numeric Calling Party Number values to maximize survivability across heterogeneous SIP and SS7/TDM networks.

Vetting procedures MAY use full telephone numbers or truncated forms as input to cryptographic operations, independent of the CIDVV Calling Party Number encoding.

CIDVV operations rely on state within the Validity Window.

4.2. Vouching vs. Vetting Response Patterns

CIDVV uses the same signaling prefixes for both operations but with distinct expected behaviors. The table below summarizes the differences:

Prefix	Vouching (live call)	Vetting (pre-shared secret)	Notes
100	Expect 486 Busy Here	Not used	Primary vouch signal
101	Expect 404 Not Found	First call: 404 (deposit), Second call: 486	Secondary / vetting

Table 1

Implementations distinguish context (vouching vs. vetting) primarily by the presence of a pre-agreed vetting Caller-ID and shared secret for the Asserted Caller-ID. Because vetting uses a specific Caller-ID designated for the procedure, overlap with ordinary vouching calls on the same number is expected to be rare. A CIDVV platform MUST treat calls using a known vetting Caller-ID according to the vetting response pattern (even if a live vouch cache entry exists) and MUST NOT treat a 101->404 response as a successful vouch when an active vetting procedure is in progress for that number.

4.3. Response Semantics

Because intermediate SIP and SS7/TDM networks may translate, modify, or replace response codes, implementations MUST interpret responses based on behavioral class (e.g., "Busy"-class vs. "Not Found"-class) rather than exact numeric values.

Implementations SHOULD use SIP 486 (Busy Here) and 404 (Not Found) as the canonical representations of these behaviors where possible.

CIDVV requires that these two rejection behaviors remain distinguishable across the signaling path. Environments that cannot preserve this distinction may not support enhanced verification.

4.3.1. "100" Prefix (Primary Verification)

A call using the "100" prefix is considered successful only if it results in an immediate rejection consistent with a "Busy"-class response (e.g., SIP 486 Busy Here).

A CIDVV implementation MUST treat any response other than this expected behavior - including ringing, call completion, timeout, or alternative error responses - as an unsuccessful verification.

A successful "100" verification provides a baseline level of confidence that the Asserted Caller-ID is routable and under the control of the originating party.

4.3.2. "101" Prefix (Secondary Verification)

A call using the "101" prefix is used as an optional secondary validation signal.

The expected behavior is an immediate rejection consistent with a "Not Found"-class response (e.g., SIP 404 Not Found).

In the context of an active vetting procedure, a "101" call carrying a valid vetting token MAY instead result in a "Busy"-class response (e.g., SIP 486 Busy Here).

Because such responses are relatively common in the PSTN, a "101" verification alone MUST NOT be treated as evidence of a successful vouch.

4.3.3. Combined Verification

Implementations MAY perform both "100" and "101" verification calls to achieve a higher level of assurance.

In this case, a stronger validation result is obtained when:

* The "100" verification produces the expected "Busy"-class behavior, AND

* The "101" verification produces the expected "Not Found"-class behavior

within the Validity Window.

Implementations MUST NOT require the two verification calls to occur in any specific order.

4.3.4. Failure Handling

If the expected behavior for a given prefix is not observed, the verification for that prefix MUST be treated as unsuccessful.

If only the "100" verification succeeds, the result MAY be treated as a valid but lower-assurance vouch.

If both "100" and "101" verifications succeed (i.e., "100" -> 486 and "101" -> 404), the result MAY be treated as a higher-assurance vouch.

If neither verification succeeds, or if results are inconsistent or ambiguous, the vouch MUST be treated as unsuccessful or indeterminate.

5. Protocol Operation

5.1. Vouching Procedure

Alice’s CIDVV platform receives an attempted call from Alice to Bob. It MUST construct a CIDVV token as defined in Section Section 4.1 by prefixing "100" to the dialed number.

The CIDVV platform MUST cache the call attempt using the tuple:

(Called Number, CIDVV Token)

for the Validity Window.

The CIDVV platform then rejects the call with SIP response 486 (Busy Here).

Alice’s SBC receives the 486 and advances the original call through the PSTN toward Bob using the original Caller-ID.

When Bob’s system receives the call, a CIDVV-aware network element (e.g., SBC) initiates a verification call toward Alice.

5.1.1. Primary Verification ("100")

Bob’s CIDVV-aware element constructs the CIDVV token using the same method (prefix "100" plus the rightmost 12 digits of the dialed number) and initiates a verification call toward Alice using that value as the Calling Party Number.

When Alice’s SBC receives a call with a Calling Party Number beginning with "100", it MUST route the call to the CIDVV platform.

Upon receiving the verification call, Alice’s CIDVV platform MUST look up the tuple:

(Called Number, CIDVV Token)

cached for the Validity Window.

If a matching cache entry exists, the CIDVV platform MUST reject the verification call with SIP response 486 (Busy Here).

If no matching cache entry exists, the CIDVV platform MUST reject the verification call with SIP response 404 (Not Found).

A successful "100" verification (i.e., receipt of 486) indicates that the originating party can receive calls at the Asserted Caller-ID and constitutes a valid baseline vouch.

5.1.2. Secondary Verification ("101")

Bob’s CIDVV-aware element MAY initiate a second verification call using a CIDVV token constructed by prefixing "101" to the same 12-digit payload.

When Alice’s SBC receives a call with a Calling Party Number beginning with "101", it MUST route the call to the CIDVV platform.

Upon receiving such a call, the CIDVV platform MUST reject the call with SIP response 404 (Not Found), unless the call corresponds to an active vetting procedure (see Section Section 5.4).

A "101" verification call does not require cache lookup for vouching purposes and MUST NOT be used as a standalone indicator of a successful vouch.

5.1.3. Combined Verification Behavior

Implementations MAY perform only the primary ("100") verification or MAY perform both primary and secondary verification.

A successful "100" verification alone provides a valid vouch with baseline assurance.

If both verification calls are performed, a higher-assurance vouch is obtained when:

"100" -> 486, and "101" -> 404

are both observed within the Validity Window.

The two verification calls MAY occur in any order or in parallel. Implementations MUST NOT assume ordering.

If the "100" verification fails, the vouch MUST be treated as unsuccessful regardless of any "101" result.

5.2. Correlation Model

CIDVV vouching correlates calls using the Asserted Caller-ID, the called number, and a Validity Window. It does not attempt to identify individual call legs across the PSTN.

If multiple calls with the same Asserted Caller-ID and called number occur within the cache interval, implementations MAY treat them as a single aggregate vouching state or MAY maintain a count of pending attempts.

A successful vouch indicates that at least one matching call attempt occurred during the Validity Window, rather than proving a one-to-one correspondence between specific call legs.

5.3. Hash Function for Vetting and State Storage

The same deterministic algorithm MUST be used for: 1. Vetting token computation. 2. Any short-term cache (e.g., Redis) that stores vouching or vetting state.

Algorithm (normative)

1. Normalize both numbers: E.164 digit string, no leading "+", no punctuation (see Section Section 4).

2. Concatenate as UTF-8 bytes: normalized-calling-number || "|" || normalized-called-number || "|" || shared-secret

3. Compute SHA-256 digest of the concatenated bytes.

4. Take the first 8 hexadecimal characters of the digest.

5. Convert that 8-hex string to a decimal integer.

6. Left-pad with zeros to 10 digits if needed, then prepend ’1’ to produce an 11-digit token.

Example (for illustration only): - calling = 12125550100, called = 19495550199, secret = "hamburger" - Concatenated: "12125550100|19495550199|hamburger" - SHA-256 first 8 hex -> decimal -> padded/prepended token = 12953388433 (or similar)

Implementations MUST use the identical normalization and concatenation order for both vetting calls and any Redis (or equivalent) cache lookups. The token is valid only inside the Validity Window.

5.3.1. Multi-Tenant Considerations

CIDVV platforms that perform vouching on behalf of multiple independent customers MUST ensure that correlation state is scoped per customer. This prevents unintended interaction between unrelated vouching operations that may produce identical CIDVV payload values.

Implementations MAY use separate storage, partitioning, or customer-specific identifiers to achieve this isolation.

This requirement does not apply to vetting operations, which are already scoped by the shared secret.

5.4. Vetting Procedure

Vetting a remote number requires two separate calls (distinct SIP dialogs) using a pre-agreed shared key. The process confirms that the *called party (Bob)* controls the target telephone number and possesses the correct shared secret. In the examples below, Alice is the verifier who initiates the two calls to Bob’s number in order to vet Bob’s number.

Before vetting begins, Alice and Bob agree on a shared secret, Bob’s vetting Caller-ID, and a Validity Window.

Alice places a vetting call to Bob using a Caller-ID beginning with the digits "101".

When Bob’s CIDVV platform receives the first vetting call, it removes the "101" prefix and verifies that the resulting Caller-ID is expected for the current vetting attempt.

Bob’s platform MUST compute the vetting token using the algorithm defined in Section Section 5.3 and store the resulting token for the Validity Window. It then rejects the call with SIP response 404 (Not Found).

Alice performs the same SHA-256 calculation and places a second vetting call to Bob. This second call uses a Caller-ID beginning with the Vetting Token Check prefix of "101" followed by the computed numeric code.

When Bob’s CIDVV platform receives the Vetting Token Check call, it removes the "101" prefix and compares the remaining numeric code to the recently cached value.

If the numeric code matches, Bob’s CIDVV platform MUST reject the call with SIP response 486 (Busy Here). Alice’s platform treats this response as a successful vet.

Any other response, timeout, code mismatch, expired cache entry, or unexpected Caller-ID MUST be treated as an unsuccessful vet.

6. Examples

6.1. Successful Vouch Call Flow (Baseline)

The following diagram shows a baseline successful vouch using only the primary "100" verification call. This provides a valid vouch with baseline assurance.

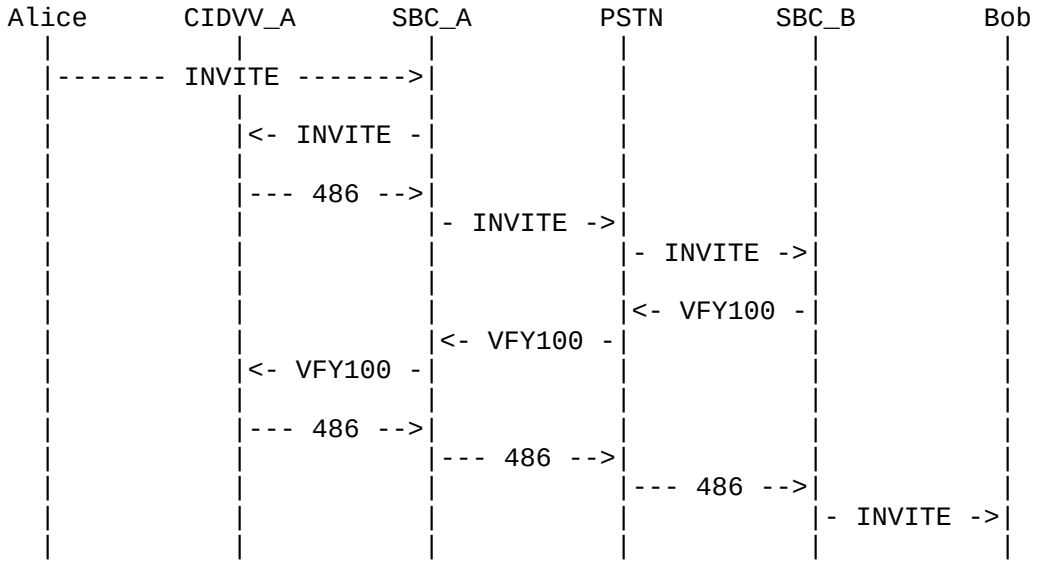


Figure 1: Example Successful Vouch (Baseline)

In the diagram, "VFY100" represents a verification call whose Calling Party Number is the CIDVV token formed as "100" followed by the rightmost 12 digits of the dialed number.

6.1.1. Successful Vouch (Baseline) Step-by-step description

The diagram above shows the high-level message flow. The following numbered steps provide the detailed behavior, including Caller-ID manipulation performed by CIDVV platforms and CIDVV-aware elements.

1. The originating user (Alice, Caller-ID +12125550100) initiates a call to the destination user (Bob, dialed number +19495550199).

2. The call is routed from Alice’s User Agent to her SBC, which forwards it to the originating CIDVV platform (CIDVV_A).

3. *CIDVV_A*:

   * Constructs a CIDVV token by prefixing "100" to the rightmost 12 digits of the dialed number. In this case, the payload is 19495550199, resulting in the token 10019495550199.

   * Caches the call attempt using the tuple: (Called: +12125550100, Token: 10019495550199) for the Validity Window.

   * Rejects the call with *486 Busy Here*.

4. Alice’s SBC receives the 486 and advances the original call toward the PSTN using the original Caller-ID.

5. The call reaches Bob’s SBC via the PSTN.

6. *Bob’s SBC (CIDVV-aware element)*:

   * Constructs the same CIDVV token by prefixing "100" to the rightmost 12 digits of the dialed number (+19495550199), resulting in 10019495550199.

   * Initiates a verification call toward Alice’s number (+12125550100) using the Caller-ID 10019495550199.

7. The verification call arrives at Alice’s SBC via the PSTN.

8. *Alice’s SBC*:

   * Detects the leading "100" prefix on the Caller-ID.

   * Routes the call to CIDVV_A for vouch verification.

9. *CIDVV_A*:

   * Receives the call with: To: +12125550100 From: 10019495550199

* Looks up the tuple: (Called: +12125550100, Token: 10019495550199) cached for the Validity Window.

* Finds a matching entry from step 3.

* Considers this a successful primary verification and returns *486 Busy Here*.

10. Bob’s SBC receives the 486 via the PSTN, recognizes it as a successful primary verification, and advances the original call to Bob’s User Agent.

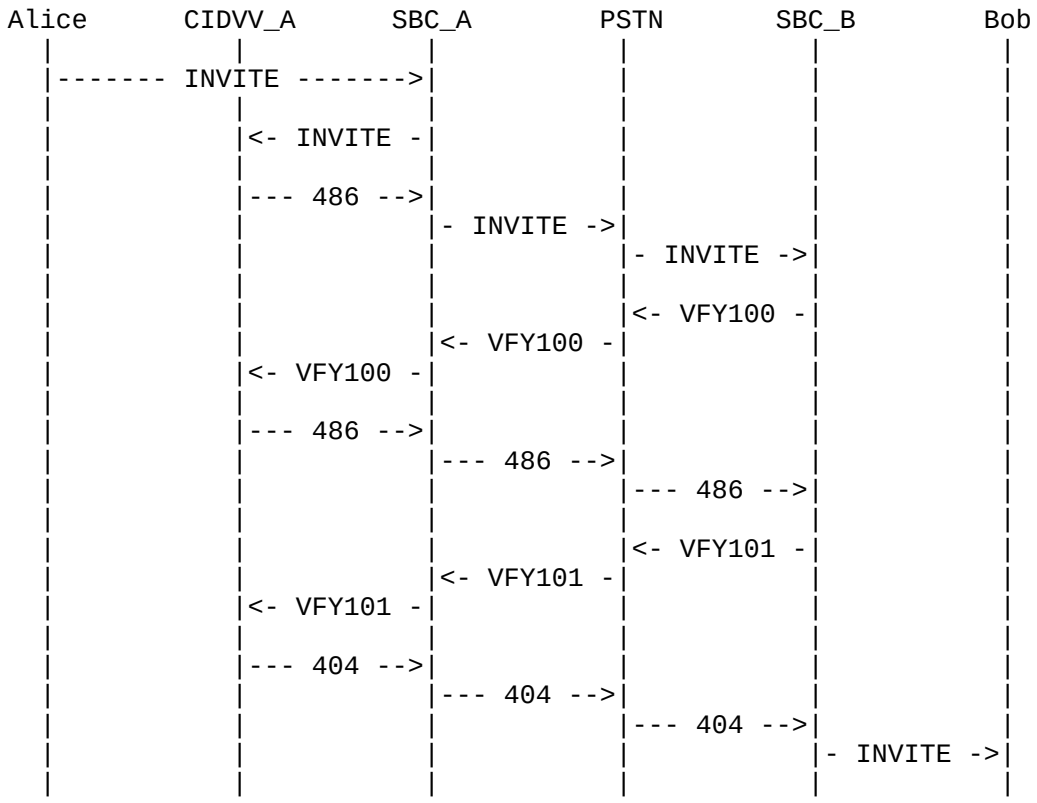
11. Bob’s telephone rings.

This mechanism allows the originating CIDVV platform to confirm that the Asserted Caller-ID is valid without completing the initial call.

6.2. Successful Vouch Call Flow (Enhanced)

The following diagram shows a successful vouch (Enhanced) using both the primary "100" verification call and an optional secondary "101" verification call. The "101" call provides additional assurance by producing a distinct response pattern when combined with a successful "100" verification. The "100" verification alone is sufficient for a baseline successful vouch.

For clarity, the verification calls are shown sequentially. In practice, the two verification calls MAY occur in any order or in parallel.



In the diagram, "VFY100" and "VFY101" represent verification calls whose Calling Party Numbers are formed using the CIDVV token construction defined in Protocol Overview.

6.2.1. Enhanced Successful Vouch Step-by-step description

The diagram above shows an enhanced vouch using both the primary "100" verification call and an optional secondary "101" verification call. The following steps describe the detailed behavior.

1. The originating user (Alice, Caller-ID +12125550100) initiates a call to the destination user (Bob, dialed number +19495550199).

2. The call is routed from Alice’s User Agent to her SBC, which forwards it to the originating CIDVV platform (CIDVV_A).

3. *CIDVV_A*:

   * Constructs a CIDVV token by prefixing "100" to the rightmost 12 digits of the dialed number (19495550199), resulting in 10019495550199.

   * Caches the call attempt using the tuple: (Called: +12125550100, Token: 10019495550199) for the Validity Window.

   * Rejects the call with *486 Busy Here*.

4. Alice’s SBC receives the 486 and advances the original call toward the PSTN using the original Caller-ID.

5. The call reaches Bob’s SBC via the PSTN.

6. *Bob’s SBC (CIDVV-aware element)*:

   * Constructs the CIDVV token 10019495550199.

   * Initiates a primary verification call toward Alice’s number (+12125550100) using the Caller-ID 10019495550199.

7. The primary verification call arrives at Alice’s SBC via the PSTN.

8. *Alice’s SBC*:

   * Detects the leading "100" prefix.

   * Routes the call to CIDVV_A for verification.

9. *CIDVV_A*:

   * Looks up (Called: +12125550100, Token: 10019495550199) cached for the Validity Window

   * Finds a matching entry.

   * Rejects the call with SIP response *486 Busy Here*.

10. Bob’s SBC receives the 486 and recognizes a successful primary verification.

11. *Bob’s SBC (optional secondary verification)*: - Constructs a second CIDVV token by prefixing "101" to the same 12-digit payload, resulting in 10119495550199. - Initiates a secondary verification call toward Alice’s number using the Caller-ID 10119495550199.

12. The secondary verification call arrives at Alice’s SBC via the PSTN.

13. *Alice’s SBC*: - Detects the leading "101" prefix. - Routes the call to CIDVV_A for processing.

14. *CIDVV_A*: - Receives the call with: To: +12125550100 From: 10119495550199 - Performs no matching cache lookup for this prefix in the vouching context. - Rejects the call with *404 Not Found*.

15. Bob’s SBC receives the 404 and recognizes the expected secondary verification behavior.

16. Having observed:

    * "100" -> 486, and

    * "101" -> 404 within the Validity Window,

    Bob’s SBC treats this as a higher-assurance successful vouch.

17. Bob’s SBC advances the original call to Bob’s User Agent.

18. Bob’s telephone rings.

6.3. Unsuccessful Vouch

A vouch attempt is considered unsuccessful or indeterminate if the expected verification behavior is not observed.

Specifically:

* If a verification call using the "100" prefix does not result in SIP 486 (Busy Here), the vouch MUST be treated as unsuccessful.

* If both verification calls are performed, and the expected pattern of:

  - "100" -> 486, and

  - "101" -> 404 is not observed within the Validity Window, the vouch MUST be treated as unsuccessful or indeterminate.

* A "101" verification result alone MUST NOT be treated as evidence of a successful vouch.

Implementations MUST fail closed. Any ambiguity, unexpected response, timeout, or call progression MUST result in an unsuccessful or indeterminate outcome.

6.4. Vetting a Caller-ID Number

Vetting uses two independent verification calls that form a challenge-response sequence. For clarity, the calls are shown separately, but together they constitute a single vetting operation.

6.4.1. First Vetting Call

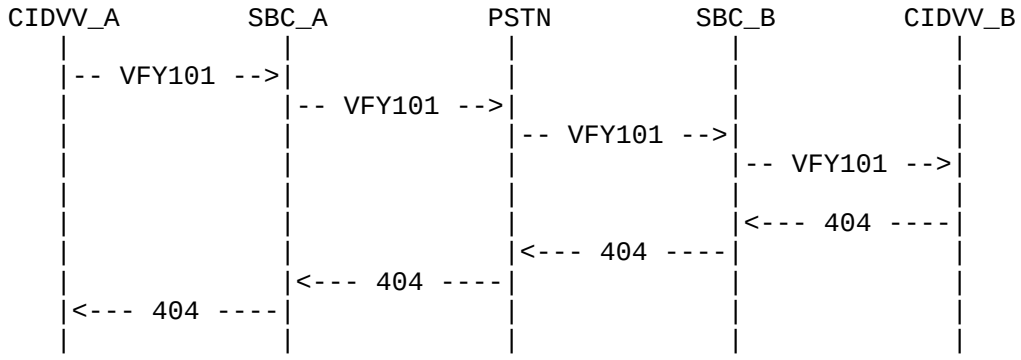


Figure 2: First vetting call with 101 - creates cache entry and responds with 404

6.4.2. Second Vetting Call

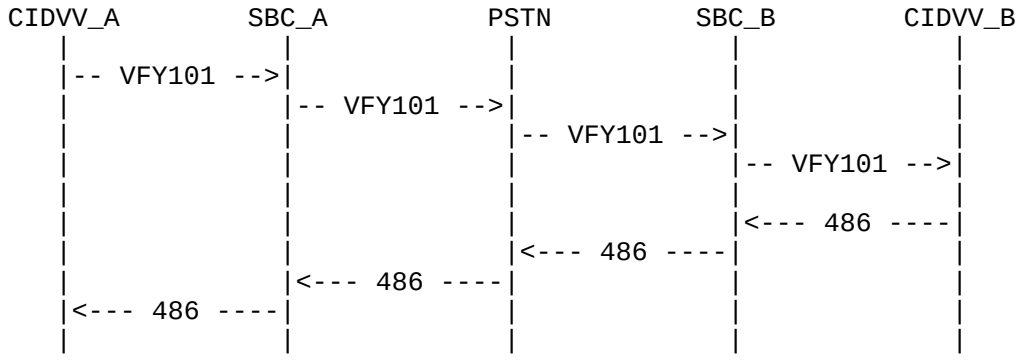


Figure 3: Vetting token check call with 101 - confirms vouch with 486 Busy Here

6.4.3. Successful Caller-ID Vetting Flow

Vetting a remote number requires two separate calls (distinct SIP dialogs) using a pre-agreed shared key. The process confirms that the called party (Bob) controls the target telephone number and possesses the correct shared secret.

1. Alice and Bob agree on a shared secret (e.g., hamburger) and Alice’s vetting Caller-ID (+12125550100, or its rightmost 12 digits for matching purposes).

2. Both parties enter the shared secret, Alice’s vetting Caller-ID, and an optional Validity Window (e.g., one week) into their respective CIDVV platforms.

3. Alice’s CIDVV platform (CIDVV_A) initiates the first vetting call with Caller-ID 10112125550100 toward Bob’s number (+19495550199). The call traverses the PSTN.

4. Bob’s SBC recognizes the leading 101 prefix on the incoming Caller-ID and forwards the call to Bob’s CIDVV platform (CIDVV_B).

5. *CIDVV_B*:

   * Strips the leading 101, recovering Alice’s Caller-ID.

   * For matching purposes, CIDVV_B MAY use only the rightmost 12 digits of the Caller-ID, consistent with CIDVV payload constraints.

   * Recognizes this as a pre-agreed Vetting Caller-ID

   * Computes the SHA-256 digest over the UTF-8 string formed by concatenating normalized-calling-number, normalized-called-number, and shared-secret, where telephone numbers are represented as digit strings without separators or leading "+".

   * Takes the first 8 hexadecimal characters (b0092191), converts to decimal (2953388433), pads to 10 digits, and prepends 1, yielding 12953388433.

   * Caches this value for the Validity Window.

   * Rejects the call with *404 Not Found*.

6. CIDVV_A receives the 404 and performs the identical hash calculation to derive 12953388433.

7. CIDVV_A immediately places a second vetting call to +19495550199 using the Vetting Token Check Caller-ID 10112953388433.

8. Bob’s SBC recognizes the 101 prefix on the Caller-ID and forwards the call to CIDVV_B.

9. *CIDVV_B*:

   * Strips the leading 101.

   * Observes that the remaining Caller-ID (12953388433) matches a recently cached vetting token.

   * Responds with *486 Busy Here* to signal a successful vet.

10. CIDVV_A receives the 486 Busy Here and reports a successful vet to Alice.

Use of the rightmost 12 digits is sufficient because collisions within the Validity Window are expected to be rare.

6.4.3.1. Vetting Failure Cases

A vetting attempt may fail for the following reasons:

* Bob does not have a participating CIDVV platform - the first call will not return 404, or the second call will not return 486.

* The shared secret, Alice’s vetting Caller-ID, or time window does not match - the two calls will not produce the expected 404 + 486 sequence.

* Network or policy restrictions prevent one or both calls from reaching the remote CIDVV platform.

In all such cases, the vetting attempt MUST be treated as unsuccessful.

This two-call challenge-response mechanism provides strong confirmation that the remote number is both reachable via the PSTN and controlled by an entity that knows the shared secret.

7. Deployment Considerations

7.1. Behavior of Non-CIDVV Systems

Systems that do not implement CIDVV are not expected to recognize CIDVV signaling prefixes. Such systems will typically process CIDVV calls as ordinary calls and may return a wide range of responses.

CIDVV implementations MUST treat any response that does not match the expected protocol behavior as indicating a non-participating system (see Section Section 4.1 for response patterns).

7.2. Handling of CIDVV Signaling Calls

Networks that recognize CIDVV SHOULD NOT present calls with Calling Party Numbers beginning with "100" or "101" to end users.

Such calls SHOULD be intercepted by network elements or CIDVV platforms and SHOULD result in a non-success response (e.g., 4xx, 5xx, or 6xx class response codes). Implementations commonly use responses such as 404 (Not Found), 486 (Busy Here), or 603 (Decline).

Call analytics, labeling, and fraud detection systems SHOULD recognize CIDVV signaling prefixes ("100" and "101") and treat such calls as protocol signaling rather than ordinary subscriber calls.

CIDVV-aware elements SHOULD recognize and internally route CIDVV signaling calls using the Asserted Caller-ID without user presentation.

CIDVV signaling calls are not intended to complete. Implementations SHOULD minimize call duration and signaling load and SHOULD avoid any media establishment.

7.3. Response Variability

Implementations SHOULD interpret responses based on behavioral class (e.g., success vs. immediate rejection) rather than relying solely on exact numeric values, as intermediate networks may translate or modify response codes.

7.4. Short-Term State Management

CIDVV relies on short-lived state for the (Asserted Caller-ID, CIDVV-Token) tuple, valid only for the Validity Window. Implementations MUST expire this state automatically and MUST fail closed: on restart or state loss, treat all verification requests as unsuccessful until fresh state has been deposited. The same hash algorithm defined in Section Section 5.3 MUST be used for any vetting-related state.

8. Operational Considerations

8.1. Protocol Operation - Vouching

If a vouching call results in a provisional response (e.g., 180 Ringing) or a successful response (200 OK), the originating system SHOULD immediately cancel the call and treat the remote system as not implementing CIDVV.

8.2. Failure and Restart Behavior

CIDVV platforms rely on short-lived state. Upon restart or loss of state, implementations SHOULD continue accepting new call deposits but MUST treat all verification requests as unsuccessful until sufficient state has been rebuilt.

Implementations SHOULD return a non-success response (e.g., 4xx, 5xx, or 6xx). A 603 (Decline) response is commonly used to indicate that verification could not be performed.

Implementations SHOULD fail closed (treating requests as unverified) rather than risk false-positive validation.

8.3. Prefix Preservation

SIP intermediaries and SBCs that support CIDVV SHOULD preserve the Calling Party Number digits used for CIDVV signaling, including the leading "100" or "101" prefix, across trusted interfaces unless local policy explicitly rejects the call.

CIDVV does not rely on Type-of-Number (TON) preservation and assumes that intermediate networks may normalize or reinterpret numbering format.

8.4. Interaction with Call Analytics and Fraud Detection

CIDVV signaling calls use Calling Party Number values that may appear anomalous to call analytics, labeling, and fraud detection systems.

Systems that support such analytics SHOULD recognize CIDVV signaling prefixes (e.g., "100" and "101") and treat such calls as protocol signaling rather than ordinary subscriber traffic.

CIDVV signaling calls are not intended to be presented to end users and SHOULD NOT be labeled or blocked as malicious traffic when processed within cooperating networks.

Failure to recognize CIDVV signaling may result in increased false positives or suppression of verification attempts.

9. Security Considerations

CIDVV verification is probabilistic and based on reachability. It does not provide cryptographic identity guarantees and is intended to complement, not replace, mechanisms such as STIR/SHAKEN.

Its security properties derive from the inability of an attacker to receive calls at the Asserted Caller-ID (the number being vouched).

CIDVV does not provide per-call correlation and instead validates reachability within the Validity Window. This may result in multiple calls being validated by a single successful vouch.

The use of distinct response patterns across multiple verification calls (e.g., "100" -> 486 and "101" -> 404) increases resistance to false-positive validation arising from common network behaviors.

9.1. Trust Model

CIDVV assumes that: - The PSTN routes calls to the correct terminating service provider for a given telephone number. - The terminating service provider has authoritative control over the number and can originate return calls. - Intermediate networks may modify signaling but will generally preserve sufficient information to allow correlation of requests and responses.

CIDVV does not assume that Caller-ID values are trustworthy; instead, it verifies control through network reachability.

9.2. Replay Attacks

CIDVV uses short-lived state (typically on the order of seconds) to correlate signaling exchanges. This limits the effectiveness of replay attacks.

Implementations MUST: - Expire cached state quickly (e.g., within ~10 seconds) - Reject verification attempts that do not match recent state

Replay within the Validity Window remains theoretically possible but requires precise timing and routing alignment (see Section Section 5.3 for vetting tokens).

9.3. Spoofing Resistance

CIDVV prevents spoofing by requiring the party asserting a Caller-ID to successfully receive and respond to a return call routed via the PSTN. An attacker (Mallory) who does not control the corresponding number cannot receive the verification call and therefore cannot complete the vouching process.

9.4. Denial of Service

CIDVV introduces additional signaling traffic, which may be abused for denial-of-service (DoS) purposes.

Implementations MUST: - Rate-limit CIDVV signaling requests - Detect and suppress repeated unsuccessful attempts - Bound resource usage for temporary state

Implementations SHOULD: - Apply per-source and per-destination limits - Monitor for anomalous traffic patterns

9.5. Amplification and Reflection

CIDVV generates return calls as part of its operation. Care MUST be taken to ensure that this behavior cannot be exploited for amplification or reflection attacks.

Implementations SHOULD: - Only initiate return calls in response to valid inbound attempts - Limit the rate of outbound verification calls - Avoid generating multiple responses for a single triggering event

9.6. Response Code Manipulation

CIDVV does not require specific SIP response codes to be preserved end-to-end, but it does require that distinct rejection behaviors (e.g., "busy" vs. "not found") remain distinguishable.

Implementations MUST interpret responses based on behavioral class (e.g., "Busy"-class vs. "Not Found"-class) rather than exact numeric values.

9.7. Data Privacy

CIDVV exchanges inherently expose calling and called numbers within signaling messages.

Implementations SHOULD: - Avoid storing telephone numbers in plaintext where possible - Use derived values (e.g., cryptographic hashes) for temporary state - Limit retention of any identifying data

Temporary state MUST be short lived and automatically expired.

9.8. Hash-Based Token Security (Vetting)

Vetting operations use shared secrets and derived tokens.

Implementations MUST: - Use cryptographically secure hash functions (e.g., SHA-256) - Protect shared secrets from disclosure - Ensure tokens are valid only within the Validity Window

Implementations SHOULD: - Include sufficient entropy in derived tokens - Avoid predictable or reusable values

9.9. Failure Modes

CIDVV implementations MUST fail closed. If verification cannot be completed due to: - network errors - state loss - unexpected responses

the result MUST be treated as unverified.

9.10. Interoperability Risks

CIDVV operates across heterogeneous networks, including SIP and SS7/TDM environments. Intermediate systems may: - modify Calling Party Number values - truncate digits - alter signaling behavior

These behaviors may cause verification to fail but MUST NOT result in false-positive validation.

9.11. Residual Risk

CIDVV improves resistance to Caller-ID spoofing but does not provide absolute identity assurance. It reduces the effectiveness of spoofing attacks rather than eliminating them and relies on probabilistic verification based on reachability and response behavior, not cryptographic identity binding.

10. IANA Considerations

This document has no IANA actions.

Appendix A. Acknowledgments

The authors thank contributors to telephony security research and PSTN infrastructure development.

Authors' Addresses

Roger Anderson
Jolly Roger Telephone Company
United States of America
Email: roger@jollyrogertelephone.com

Steven Berkson
Jolly Roger Telephone Company
United States of America
Email: steveb@jollyrogertelephone.com

Phillip Askew
United States of America
Email: phillip.askew@theaskewcrew.com