June 2023

COLLECTING POWER CONSUMPTION METRICS FROM OPERATIONALLY INACCESSIBLE NETWORKS

Marcelo Yannuzzi

Carlos M. Pignataro

Roque Gagliano

Francisco Sedano Crippa

COLLECTING POWER CONSUMPTION METRICS FROM OPERATIONALLY INACCESSIBLE NETWORKS

AUTHORS:
Marcelo Yannuzzi
Carlos M. Pignataro
Roque Gagliano
Francisco Sedano Crippa

ABSTRACT

Operational traffic, such as management plane traffic carrying power consumption metrics from a device in the field, is often inaccessible to a device's vendor. Techniques are presented herein that support a novel mechanism for encoding power consumption metrics in standard communication protocols such as, for example, domain name system (DNS) requests. Aspects of the presented techniques ensure the atomicity of self-contained messages, as well as the confidentiality and integrity of the metrics that are sent to the corresponding vendors. Further aspects of the presented techniques support selectable levels of anonymity during the exporting of the above-described metrics. For example, selectable Terms and Conditions may not only allow administrators to choose among different levels of anonymity, but also facilitate frictionless operations and automatic configuration during the activation of a license.

DETAILED DESCRIPTION

Most information technology (IT) vendors are working to reduce the carbon footprint of their products, not only to increase their competitiveness and foster product renewal cycles but also to meet sustainability objectives. Those practices are leading to new levels of transparency from vendors, including the publication (through, for example, white papers, auditing reports, live data that is accessible from online sites, etc.) of the levels of greenhouse gas (GHG) and carbon dioxide equivalent (CO2E) emissions that are associated with a vendor's products.

Additionally, both IT vendors and users have incentives to share energy-related data such as power consumption metrics. More specifically, carbon footprint calculations entail a checks-and-balances process across organizations to avoid the duplication of

entries in carbon accounting. For instance, under category 11 of Scope 3 of The Greenhouse Gas Protocol: A Corporate Accounting and Reporting Standard (as published by the World Business Council for Sustainable Development (WBCSD) and the World Resources Institute (WRI)), vendors are required to include the emissions of sold products. The Scope 3 emissions for one organization (such as a vendor) are typically computed as the Scope 1 or 2 emissions of other organizations (such as users).

Hence the sharing of data between users and vendors in a controlled manner is mutually beneficial. On the one hand, users may not only obtain the calculations and emission reports directly from a vendor, but also may outsource the responsibility of such calculations and reporting. On the other hand, a vendor may now have access to data for their devices in the field and, consequently, be able to track and compare the results across different product releases, which may operate in different environments and under different hardware and software license configurations.

However, obtaining the above-described metrics from devices that are deployed in the field poses multiple challenges. Figure 1, below, presents elements of an exemplary arrangement 100 that helps to illustrate various of those challenges.

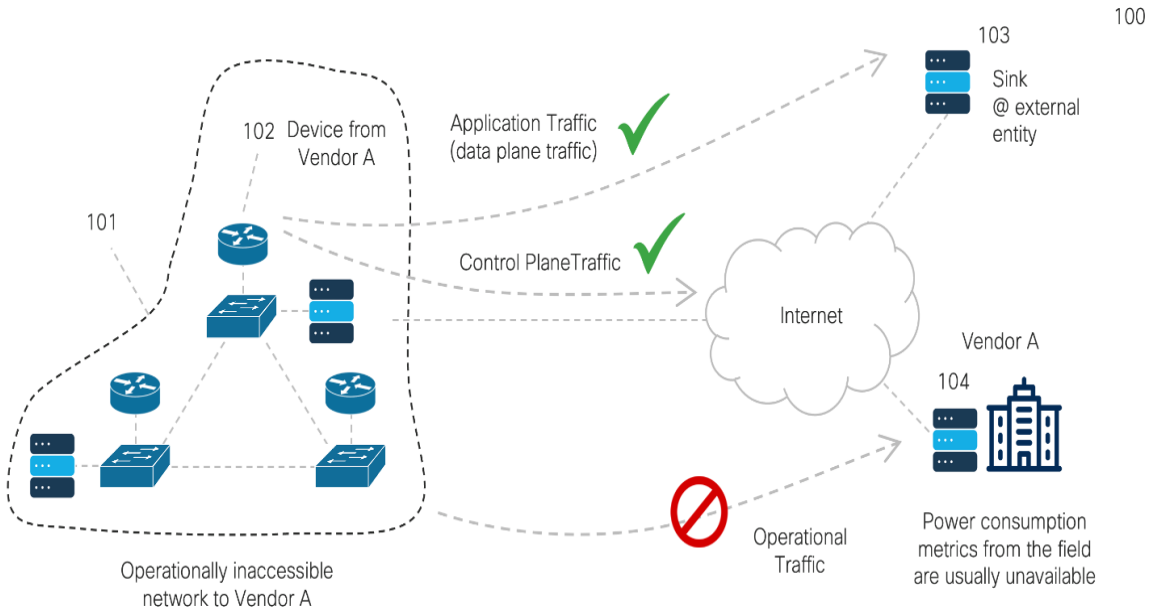

Figure 1: Exemplary Arrangement

As depicted in Figure 1, above, data plane and control plane traffic may be regularly sent to the Internet while operational traffic usually remains confined to just network 101.

In particular, using the arrangement that was presented in Figure 1, above, consider the issue that arises during the collection of power consumption metrics. As shown in the figure, network 101 may have deployed a device 102 that was manufactured by Vendor A. Device 102 may regularly forward data plane traffic from various applications to different endpoints across the Internet, including sink 103. Control plane traffic, such as DNS exchanges or routing traffic that is sourced from device 102, may also be forwarded to the greater Internet. However, operational traffic that is generated by device 102 (e.g., management plane traffic that is encoding power consumption metrics) is often inaccessible to the device's vendor. In other words, even though an endpoint 104 in Vendor A might be reachable from device 102, operational traffic that is produced by device 102, including power consumption metrics, is typically confined just to network 101.

Even if network device 102 is, by default, preconfigured to send data to its vendor, many zero-trust networks impose tight controls and may block outbound application and control plane communications that do not have a valid single-factor authentication, dual-factor authentication, or multi-factor authentication (MFA) facility. Further, network administrators are not used to explicitly configuring their IT devices to share power consumption metrics with the devices' corresponding vendors.

Today, there is no frictionless mechanism for sending power consumption data from different devices to their corresponding vendors. For example, such a conveyance may require the configuration of vendor-specific endpoints in the devices themselves. Additionally, it may require the collection of relevant metrics locally and then the pushing of the same to vendor-specific application programming interfaces (APIs) or other means. Some of the above-described approaches may also require authentication methods and/or the management of various API keys. As well, a conveyance as described above may also require the opening of ports, the reconfiguration of firewalls, etc., all of which translates into friction for network administrators. Further, network administrators may require that the data that is to be shared with various vendors comply with the data governance rules within their organization and/or comply with regulatory obligations including privacy-specific policies.

To address the challenge that was described above, techniques are presented herein that introduce a novel mechanism for encoding power consumption metrics in a standard communication protocol such as, for example, a DNS request. Aspects of the presented techniques ensure the atomicity of a self-contained message as well as the confidentiality and integrity of the metrics that are sent to a corresponding vendor. Further aspects of the presented techniques support selectable levels of anonymity when exporting such metrics to a vendor. For example, selectable Terms and Conditions may not only allow administrators to choose among different levels of anonymity, but also facilitate frictionless operations and automatic configuration during the activation of a license.

More specifically, the techniques presented herein may enable a first device (such as device 102 in Figure 1, above) to encode, encrypt, and carry power consumption metrics in a standard communication protocol (such as, for example, a DNS query) which may reach a second, external device (such as a root DNS server that is not shown in Figure 1), given that such communications are usually open and operationally available to the second device. The second device may resolve for the identity of a vendor (e.g., a vendor identifier (ID) may be encoded in a DNS query itself) and it may forward the encoded metrics to the corresponding vendor ID (as, for example, part of the DNS resolution process). A third device, managed by the vendor (such as, for example, a DNS server that is operated by the vendor, such as device 104 in Figure 1), may then receive the message, detect the ID of first device, decrypt the power consumption metrics, and acknowledge the receipt of such metrics to, or solicit other metrics from, the first device in a trustworthy manner. Importantly, the techniques presented herein require modifications neither to the protocols carrying the encoded metrics nor to their caching or communication methods.

Figure 2, below, illustrates an example system 200 that highlights the main elements of the techniques presented herein along with their potential use.

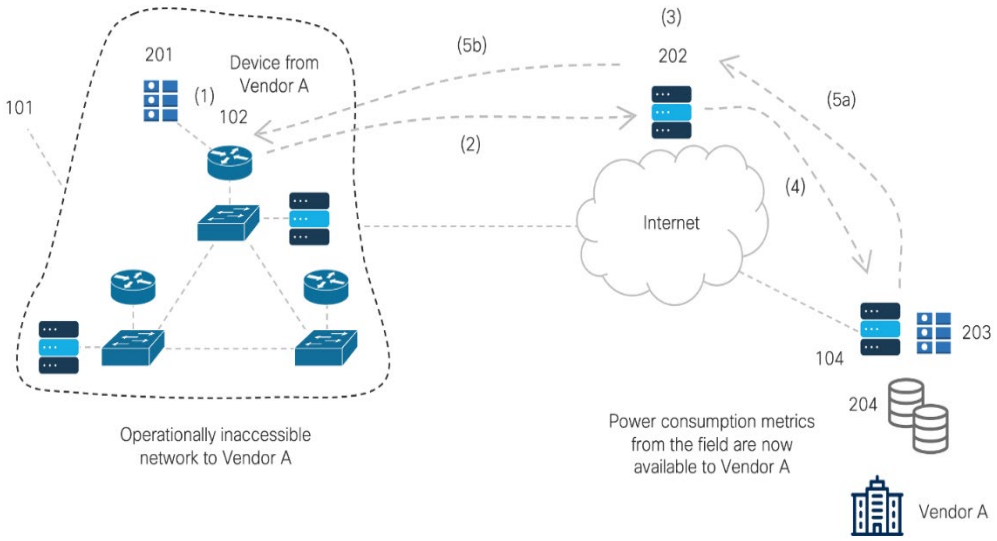

Figure 2: Illustrative Operation

As depicted in Figure 2, above, a network 101 may comprise a variety of IT devices that are developed by different vendors, including device 102 (such as, for example, a router) that is manufactured by vendor A (e.g., a network equipment vendor). Factory settings may preconfigure device 102 with a table or registry 201 that encodes and lists the categories of metrics that device 102 may send to vendor A. Table 1, below, shows how a number of exemplary categories of metrics may be encoded within registry 201.

Table 1: Exemplary Categories of Metrics Encoded in Registry

Category (3 octets, ASCII printable set)	Description
000	(Instantaneous Total Power, Null, Null)
001	(Instantaneous AC Power, Null, Null)
002	(Instantaneous DC Power, Null, Null)
003	(Instantaneous AC Power, Instantaneous DC Power, Null)
...	...
A00	(Instantaneous Total Power, Time up, C1)
A01	(Instantaneous Total Power, Time up, C2)
A02	(Instantaneous Total Power, Time up, C3)
...	...

As shown in Table 1, above, if vendor A receives a message that is indicated as category A02, vendor A may decode the received metrics as the triplet (Instantaneous Total Power, Time up, C3), where the three elements of the triplet are described below.

The first element in the triplet (i.e., Instantaneous Total Power) may represent the instantaneous total power that is consumed by the device (considering, for example, both alternating current (AC) and direct current (DC) metrics). The second element in the triplet (i.e., Time up) may represent the total amount of time since the device was last booted. The third element in the triplet (i.e., Cx) may represent the current Configuration code (or C code, for short) including both a specific hardware configuration and software license. Indeed, such a C code may represent a second level of encoding within the encoded information that is shown in Table 1, above, with the aim of differentiating among the various settings, encompassing different options in terms of hardware modules and software licenses, that a modular device might possess.

The three octets in the category field in Table 1, above, allow for 95^3 possible combinations, thereby leaving sufficient room to combine power consumption metrics and configurations for all of the different device families and models. Other combinations may also be used by decreasing or increasing the number of octets in the category field.

Referring again to Figure 2, above, device 102 may internally detect its C code at boot time or at run time (depending on, for example, a new configuration at run time such as a new module or line card that has been activated, a gigabit port that has been activated, etc.) and automatically filter the entries that currently apply in registry 201 depending upon its C code. Device 102 may also use a default metric triplet (such as Instantaneous Total Power, Null, Null) that is compatible with a computed C code.

Alternatively, the above-described C codes may be unavailable. For example, a device may not have the ability to compute such C codes. In that case, Table 1, above, may, according to aspects of the techniques presented herein, be simplified to capture only power consumption metrics.

Under further aspects of the techniques presented herein, some of the metrics in Table 1, above, may encode values that could help provide additional signals for anti-counterfeiting measures. For instance, one of the metrics in a triplet might encompass the outcome of a set of internal checks, measurements, and/or the sampling of results to encode

data about the origin of the hardware and/or the software that is running on a device. Indeed, such information might be correlated with the power consumption metrics that are received by a vendor.

In order to send power consumption metrics from device 102 to vendor A, the various elements that are depicted in the example system 200 in Figure 2, above, may, under the techniques presented herein, proceed according to the following steps.

During a first step, device 102 may compute, encode, encrypt, and transmit power consumption metrics according to the procedure that is expressed in the following pseudo-code snippet:

```
#####################################################################
Pseudo-code snippet to be executed by device 102
#####################################################################
...
300 frequency = 86400
301 trigger = 10:00:00 CET
302 anonymity-level = 5 ## May allow to filter register 201 by anonymity level and C status
303
304 PAYLOAD = $metric & "." & timestamp & "." & $deviceID & "." & $category
305 EPM = encrypt ($PAYLOAD)
306 EPM32 = split-subdomain($EPM-$deviceID-$category | base32)
307 dig $EPM32.powermetrics.vendorA.com @dns-server
...
#####################################################################
```

For example, as shown in line 300 of the above code snippet, device 102 may report power consumption metrics to vendor A daily (i.e., with a frequency of 86,400 seconds) and such metrics might be reported every day at 10:00 AM CET (as indicated in line 301).

Additionally, the level of anonymity may be set to 5 (as indicted in line 302), which in this example may indicate that power consumption metrics may be sent to the vendor, but no specific details (such as the C status) may be shared. For example, the administrators of network 101 may allow for the sharing of a device ID and power metrics but not the specific hardware and software license configuration that is in use. A variety of anonymity

levels may be enabled as part of a selectable Terms and Conditions process, thereby allowing a network administrator to control how the data that is to be shared with various vendors complies with the data governance rules within an organization. Such parameters may also be reconfigured by an administrator once a device is in operation.

Line 304 of the above code snippet shows an example of how the payload that is to be sent by device 102 may comprise a metric (e.g., a power consumption metric), a timestamp, the device ID, and the category as defined by the three octets in the first column in Table 1, above. Such information may be appended (as indicted in line 304) and then encrypted as Encrypted Power Metrics (EPM) in line 305.

Line 306 shows how the resulting EPM may be appended to unencrypted information, such as the device ID and the category of metrics that are to be sent to vendor A, so that vendor A may identify the device, use the correct key to decrypt the payload, and use the category field to understand how to decode and interpret the data that is encoded in the received payload. Line 306 may also ensure that the concatenation of the EPM, the device ID, and the employed category are encoded in a base-32 numeral system (Base32) as defined by standardized DNS subdomain representations. Line 306 may further ensure that each subdomain comprises at most 63 characters, as defined by the DNS standard, through a use of the function split-subdomain.

Based on the above, device 102 may now issue a DNS request using, for example, a record type of text (TXT) as shown in line 307 of the above code snippet.

A concrete, step-by-step example of the above process is presented below:

vendor = Vendor A

anonymity-level = 5

default metric triplet = (Instantaneous Total Power, Null, Null)

instantaneous Total Power = 1196 W

triplet-encoding = 1196.000.000

timestamp = 1683031919884 (e.g., using a UNIX epoch - 64-bit value in ASCII format with millisecond granularity)

deviceID = AGD120920M (e.g., a router from Vendor A)

category = 000 (see Table 1)

domain-name = powermetrics.VendorA.com

record-type = TXT

and is summarized in Figure 3 which is presented later in the instant narrative.

Applying all of the above, line 304 of the previously presented code snippet yields PAYLOAD = 1196.000.000.1683031919884.AGD120920M.000.

The encryption technique that is employed in the instant example may use the following asymmetric keys. The private key:

MIIBVgIBADANBgkqhkiG9w0BAQEFAASCAUAwggE8AgEAAkEAma2eGQHwOc6kQ0C5Lpyl7VkPu/QuopYg3guKhFVvflordQCnWDooFXttW/6pW8z3iS50HwmQxQ9pF4RIuApt1wIDAQABAkA7irkml0msHBQounsFRExFmawu9nzbIa+6WF5ix3dbTJqvzNo/F8avryWC63u6aFv18iTedjrrk81AIKn3zyWBAiEAyhYZzXn0aX5Ca6hJ/jDA+1AL+Uu3CRPKj66Cz0xBYTUCIQDCrWJQT6dHd96ezDcFcVZJBi38MMVJ5GrUXrPxgxVgWwIhAKtEK73+m2tEfBotV/g7bXIPlvZCeu8QfN1kcqwo3kf5AiEAuD6yIkIQljIbbnzlwe/u8rO3h8t8nJ9if9WVRgiCjasCIQDDBeWW2IzPM952XluvOuonoWfRwIeDHYQmyykG6h1u0A==

may be used for encrypting the queries that are generated by device 102. This may help ensure important properties of the payloads that are received by vendor device 104 including non-forgeability, confidentiality, and integrity. This key may also be used by device 102 for decrypting a response that is received from device 104. The public key:

MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAJmtnhkB8DnOpENAuS6cpe1ZD7v0LqKWIN4LioRVb35aK3UAp1g6KBV7bVv+qVvM94kudB8JkMUPaReESLgKbdcCAwEAAQ==

may be used by device 104 for decrypting the queries that are sent by device 102. This key may also be used by device 104 for encrypting the responses that are sent to device 102.

Employing the above, in line 305 of the previously presented code snippet the expression EPM = encrypt ($PAYLOAD) yields:

l8A3JhCDF0n3EcGxYo6LsfyWLmNE04BPnVP1WiuCa+4fDpkQw772pMGW+LXpkPw/T029POpYJ7vxY8330I0cTw==

and the expression ($EPM-$deviceID-$category | base32) in line 306 yields:

NQ4ECM2KNBBUIRRQNYZUKY2HPBMW6NSMONTHSV2MNVHEKMBUIJIG4VSQGFLWS5KDM
EVTIZSEOBVVC5ZXG4ZHATKHK4VUYWDQNNIHOL2UGAZDSUCPOBMUUN3WPBMTQMZTGBETAY2U
O46T2LKBI5CDCMRQHEZDATJNGAYDA===

Consequently, in line 306 the expression EPM32 = split-subdomain($EPM-$deviceID-$category | base32) yields three subdomain groups, split by the dot character (“.”), each of which is limited to a maximum of 63 characters:

NQ4ECM2KNBBUIRRQNYZUKY2HPBMW6NSMONTHSV2MNVHEKMBUIJIG4VSQGFLWS5K.DM
EVTIZSEOBVVC5ZXG4ZHATKHK4VUYWDQNNIHOL2UGAZDSUCPOBMUUN3WPBMTQM.ZTGBETAY2U
O46T2LKBI5CDCMRQHEZDATJNGAYDA===

In line 307 of the previously presented code snippet, device 102 may now issue a DNS request to its dns-server (e.g., 8.8.8.8) for the following domain name:

NQ4ECM2KNBBUIRRQNYZUKY2HPBMW6NSMONTHSV2MNVHEKMBUIJIG4VSQGFLWS5K.DM
EVTIZSEOBVVC5ZXG4ZHATKHK4VUYWDQNNIHOL2UGAZDSUCPOBMUUN3WPBMTQM.ZTGBETAY2U
O46T2LKBI5CDCMRQHEZDATJNGAYDA===.powermetrics.VendorA.com.

As described and illustrated in the above example, device 102 is able to successfully encode the power metrics in an atomic and self-contained manner, without needing to fragment and sequence the metrics across various payloads. More specifically, in the above example device 102 may generate a DNS query comprising 193 characters, which honors the DNS standard, since the total size of the records is limited in practice to 253 octets, with each subdomain name filled up to a maximum of 63 characters.

During a second step, as shown in Figure 2, above, a generated DNS query may reach an external device 202, such as a root DNS server. In an alternate arrangement, device 202 might be part of a DNS layer security solution. Indeed, while Steps 3 to 5 below are described as part of the standard operation of a DNS query-response flow, an alternative approach may leverage a DNS layer security solution to detect DNS queries that match a specific domain name (such as powermetrics.VendorA.com). In such cases, the security solution may capture the query and redirect it directly to device 104. The DNS response

may be treated and routed similarly, thus avoiding an increase in traffic in the standard DNS system due to the collection of sustainability metrics.

During a third step, external device 202 may now resolve for the vendor ID that is encoded in the query itself. For example, device 104 in Figure 2, above, may represent an authoritative DNS server for the domain name powermetrics.VendorA.com.

During a fourth step, the DNS query may be forwarded to device 104 which may now proceed to decode it. To this end, device 104 may first need to remove the dot characters (".") that were used to split the various subdomains into a maximum of 63 characters of length. It may subsequently decode the message using Base32. It may then strip the tail "-AGD120920M-000" from the decoded subdomain fields and identify both the device ID (as AGD120920M) and the category of the payload received (000 in this example). As mentioned above, this may allow the vendor not only to identify the device, and the corresponding key to decrypt the payload, but also its type and how to decode it. More specifically, the vendor may also have access to a catalog of registries 203 and to the device IDs, their corresponding families, and models through data store 204. Using the public key, device 104 may now proceed to decrypt the message and obtain the artifact payload = 1196.000.000.1683031919884.AGD120920M.000.

The tail of the payload (i.e., AGD120920M.000) that was extracted after the decryption process must match with the device ID and the category that had been encoded in Base32 and sent in an unencrypted form. This may enable device 104 to determine that the payload that was received was generated by the correct device (e.g., device 102 with an ID of AGD120920M).

Device 104 may now identify the category as 000 and conclude that the encoded metrics denote the triplet (Instantaneous Total Power, Null, Null), and, from the triplet encoding of 1196.000.000, obtain the Instantaneous Total Power, which in this example is 1196 watts (W).

During a fifth step, device 104 may now respond to the received DNS query. In the example that was presented above, the query that was issued by device 102 was for a record-type of TXT, hence device 104 may include text in the response. Various possible messages may be encoded, encrypted (using the public key in the above example), and then sent back in the response from device 104. Different fields may be encoded in the response,

including a request to rotate the category of metrics that are sent each day. For example, device 104 may request a rotation, such as in the next update metrics should be sent according to category 001 (as indicated in Table 1, above). Device 102 may also support a calendar feature, so that it may schedule the delivery of different message categories on Mondays, Tuesdays, etc., leading to, for example, a weekly rotation. Additionally, a response may include a timestamp, or be salted, so as to avoid potential replay attacks. Such a response may also include the original timestamp, which was sent by device 102, which could opportunistically be used as a nonce (to, for example, match the response to a pending query).

Device 102 may now receive the TXT response and decrypt it. It may then update the parameters that are used by the scheduler as requested by device 104. Device 102 may also use the reply as an acknowledgement by device 104 of the receipt of the corresponding power consumption metrics.

Figure 3, below, summarizes the main elements of the different steps that were described above.

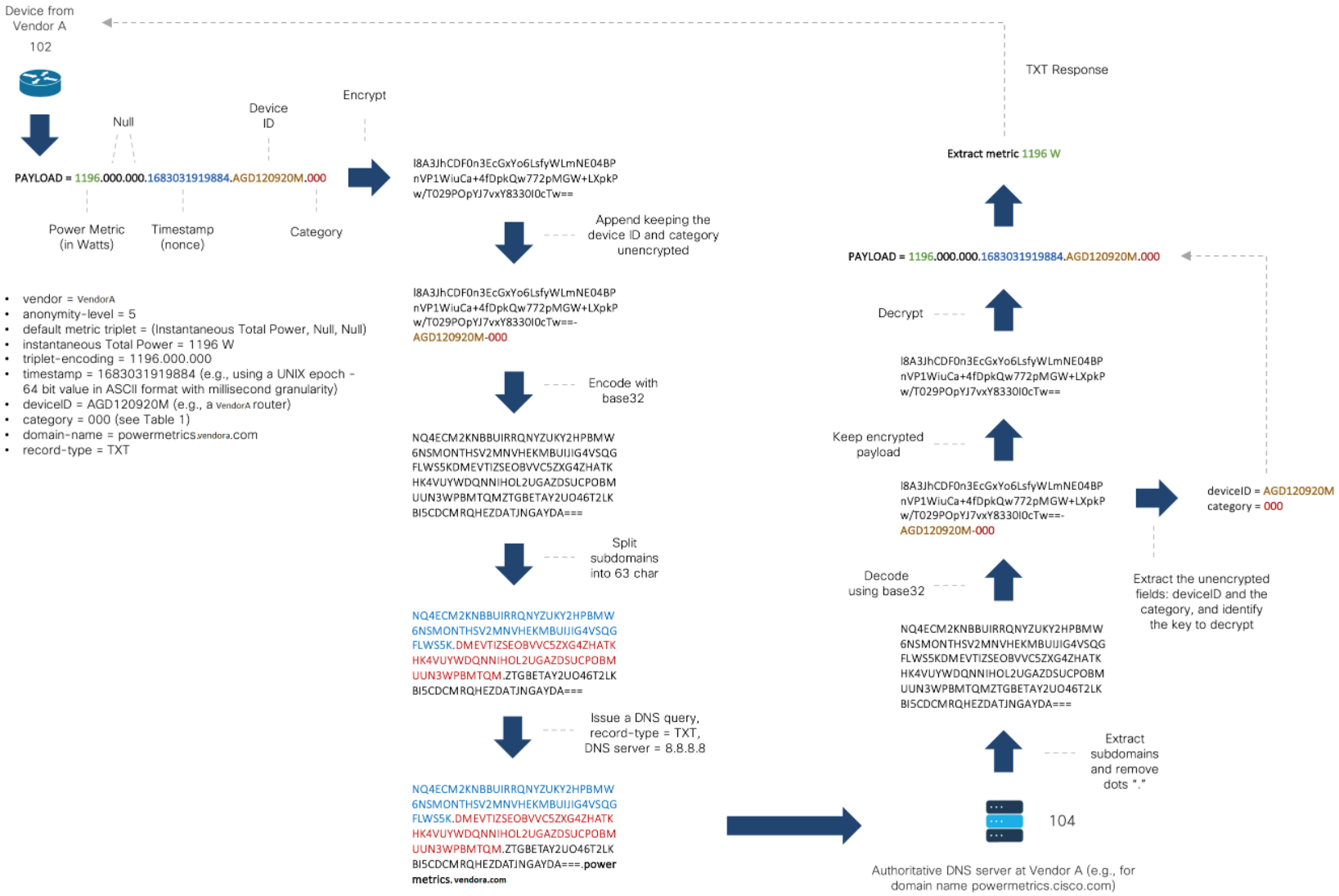

Figure 3: Overall Process Flow

Under aspects of the techniques presented herein, devices 102 and 104 of Figure 2, above, may, as an alternative, employ symmetric keys instead of the asymmetric keys that were described above.

It is also important to note that the dispersion in the subdomain names, such as in the following example:

NQ4ECM2KNBBUIRRQNYZUKY2HPBMW6NSMONTHSV2MNVHEKMBUIJIG4VSQGFLWS5K.DMEVTIZSEOBVVC5ZXG4ZHATKHK4VUYWDQNNIHOL2UGAZDSUCPOBMUUN3WPBMTQM.ZTGBETAY2UO46T2LKBI5CDCMRQHEZDATJNGAYDA===

may help avoid the caching mechanisms that many devices typically implement as part of the DNS system.

Under further aspects of the techniques presented herein, some of the devices involved in the above-described flows may flag support for Extension Mechanisms for DNS (EDNS). Under such a case, it may be possible to expand the size of several parameters and avoid the size limitations in legacy DNS systems.

Under still further aspects of the techniques presented herein and referring again to Figure 2, above, device 102 may piggyback power consumption metrics into other standard communication protocols beyond, for example, DNS queries. Such an approach may be implemented using optional and transitive attributes in type-length-value (TLV) triplets, which may be detected by external devices. More specifically, an external device 202 may be able to receive messages from device 102 and detect the presence of those optional and transitive attributes. External device 202 may then extract and parse such attributes. The process in charge of that activity may run within a router, a virtual router (vrouter), a controller leveraging a Layer 3 (L3) device API, or another device represented as 202 in Figure 2, above. Such a process may parse the received piggybacked information and obtain the ID of the IDP (such as, for example, vendor A) and perform a lookup process for the IDP to determine the vendor's endpoint (e.g., device 104) to which the information must be sent. External device 202 may then forward the information to vendor A's endpoint 104.

In summary, techniques have been presented herein that support a novel mechanism for encoding power consumption metrics in standard communication protocols, such as

DNS requests. Such metrics may be periodically, such as once a day, etc. Broadly, techniques herein facilitate access to such information across different administrative domains (e.g., vendor – user). Aspects of the presented techniques ensure the atomicity of self-contained messages, as well as the confidentiality and integrity of the metrics that are sent to the corresponding vendors. Further aspects of the presented techniques support selectable levels of anonymity during the exporting the above-described metrics. For example, selectable Terms and Conditions may not only allow administrators to choose among different levels of anonymity, but also facilitate frictionless operations and automatic configuration during the activation of a license.