February 2023

Defined-trust Limited Domains

Kathleen Nichols

Defined-trust Limited Domains

Kathleen Nichols
Pollere LLC
February 21, 2023

Abstract

Defined-trust Limited Domains are a type of Limited Domain [RFC8799] where the rules specifying the (networked) communication of application information are defined in a communications schema that governs the information communicated in a particular Limited Domain. The schema includes the required format of information and specifies identities and the attributes that are required to legally construct a particular format. Application-local trust management enforces the schema which allows members of the domain to be "definite in what they accept" [LANG] and hence secure the Domain without a physical perimeter. All communications must be signed by a verifiable member identity. The schema specifies the format of the communications as well as the format of identity certificates and all signing rules are specified as a chain of trust that terminates at the trust anchor of the Domain. Non-conformant communications are detected and discarded early in the arrival process, preventing external information from entering the Domain. Encryption can be specified in the schema for privacy of Domain information. A Defined-trust Limited Domain may also be referred to as a Trust Domain or just Domain where the context is clear.

There may be additional administrative communications in a Trust Domain, e.g. as part of the signing/crypto set up and maintenance or for reporting alerts, alarms, and faults. These may be specified in the communications schema. Example implementations with a Defined-trust Transport (DeftT) that handles communications and presents and API are available at [DCT].

Previous Work

The Defined-trust framework employs concepts from or is inspired by a body of previous work, including [DIFF,DLOG,DTM,DNMP,Graphene, LANG,NMUD,NDN,RFC2693,SDSI,SNC,SRM].

A Defined-trust Approach

A Defined-trust Limited Domain can be implemented as in [IOTK,DCT,DTLD] with an interface to *collections* of hierarchically named units of information (items or *publications*). A Trust Domain's collections are also hierarchically named where the prefix uniquely identifies a particular Trust Domain followed by the type of named items it contains. Collections are synchronized across all members of the Domain by a *sync* protocol. Sync sends and receives the items of its collection wrapped in its own protocol data units (PDUs) which are exchanged using system transports, e.g., UDP, TCP, IPv6, LoRA. PDUs are hierarchically named and prefixed with their collection name.

Synchronization is an inherently multi-party activity and can be efficiently implemented on broadcast media where internet protocols like IPv6 Link Local Multicast [RFC4291] are available; point-to-point links and protocols can also be used to carry sync PDUs. A sync operates on a single subnet thus its PDUs are not forwarded between subnets. *Syncps* [DCT] is a sync which uses [IBLT] for reconciling collection differences between Domain members on the

same subnet. A Trust Domain uses separate collections for application information and for such administrative information as certificates, keys, alerts and other types of data that implement and manage the Domain. This approach is detailed in [IOTK,TST] with reference applications [DCT].

Applications communicate by adding to and reading from collections containing information items identified by their own structured names beginning with topics or subtopics. Legal formats for these names are defined in the communications schema along with rules on the secured, validated identity a member must posses in order to form an item with a particular name structure. Member identities are specified as certificate signing chains that contain all the attributes required for a particular member to fulfill its role. An integrated trust management engine uses the communications schema to ensure that only valid information items and PDUs are built and accepted into a collection. Figure 1 illustrates this for a Trust Domain where a member adds the certificates of its identity to the Domain's **cert** collection and synchronizes application information items in **pubs**. Signing keys are always privately kept.

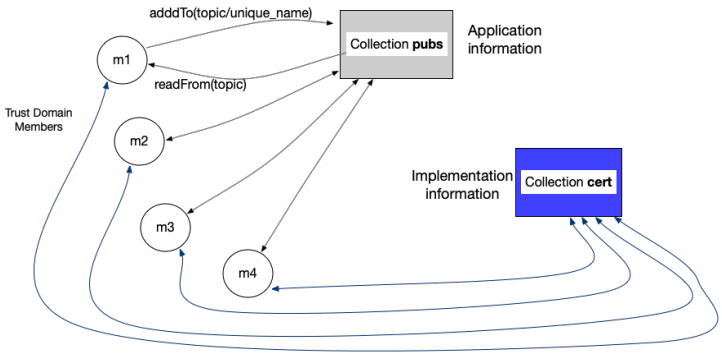

Figure 1. Defined-trust Domain Communications Model Showing Two Collections

Members communicate through the collection rather than with one another. Publish/subscribe (e.g. [MQTT]) is an example of this type of application interface. The Defined-trust model is distinguished from publish/subscribe application protocols by its use of communications schema, certificate chain identities, a per-member trust management engine that governs every item in a collection, and its sync that permits efficient operation on broadcast media.

This communications model can be implemented through use of a defined-trust transport at each member containing the API logic, the trust management engine, the collection management and synchronization protocol, item and PDU building, signing and validation logic, certificate storage, and distribution of certificates, keys, and other transport management functions (see fig. 2). Application-visible information item collections (**pubs**) are supported by the transport's own infrastructure of collections, e.g., identity chain certificates (**cert**) and group keys (**keys**). Each of these has its own sync, exchanging PDUs via an a system transport like UDP (multicast or unicast), TCP, IPv6. The trust management engine and crypto signing and validation functions are used or both PDUs and the pubs.

A defined-trust transport can be deployed with trusted software enclaves like TPMs and Trust Execution Environments (TEEs) [TPM,ATZ] to secure the critical code, communications schema, identity cert chain, and/or the private signing keys. In fig. 2, the shaded areas can be

located in a TEE with gates where the information items enter and leave. Defined-trust Limited Domains secure information only from the time it is received from the API until the time it is passed to an API. If code security is required, Trusted Execution Environments (TEEs) [ATZ,HSE] should be used.

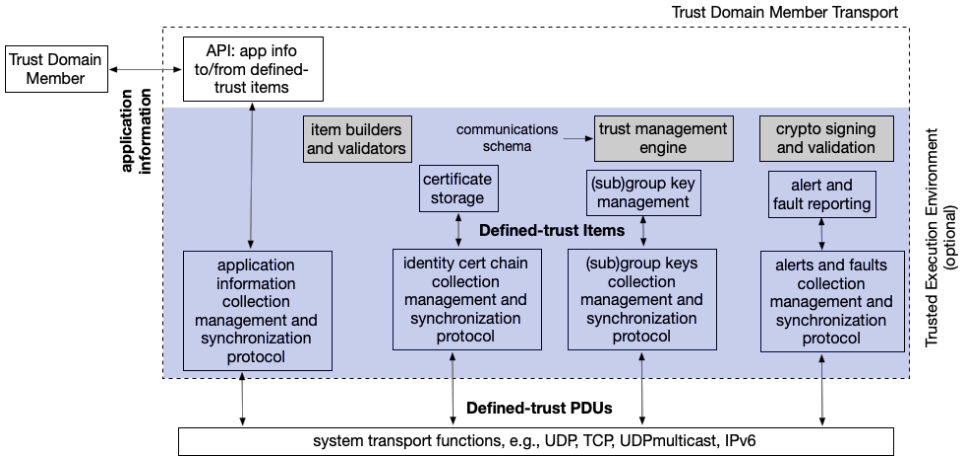

Figure 2. Transport model for a Trust Domain member

A Trust Domain is uniquely identified by its communications schema and particular trust anchor that signs the schema and all identities within the TD. In some cases, it is convenient to define subdomains governed by a communications schema that contains a subset of the overall definitions, creating a subdomain that can be uniquely distinguished by these particular schema. [DCT] uses an efficient binary representation of communications schema that is distributed as a certificate signed by the Domain's trust anchor. A hash of the schema certificate is the prefix of all collection PDUs so can be used to identify which PDUs are part of the domain or subdomain. Trust domains and subdomains may use the same physical subnet; trust boundaries are enforced by member trust management not by physical separation or perimeters.

Member identities are distributed as a chain of trust, public certificates that have each been signed by the signing key associated with the next certificate in the chain and, at the root, signed by the trust anchor of the Trust Domain (see fig. 3). A member is enrolled in a Trust Domain through secure configuration with the trust anchor (public cert), the trust schema (in a cert) and its own identity cert chain with private signing key. These provide the member everything it needs to participate in the Trust Domain. There are many existing approaches to perform the enrollment [COMIS,RFC8995]; custom variants are possible.

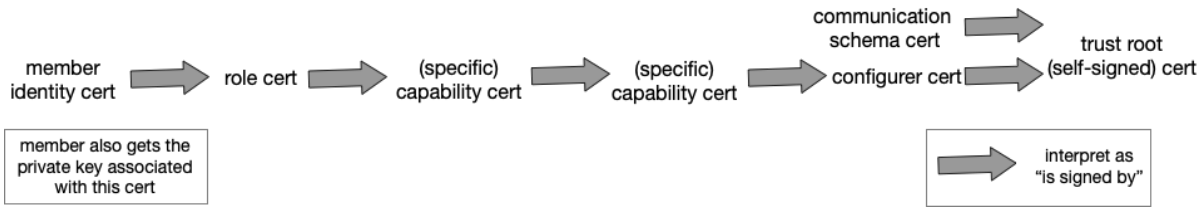

Figure 3. Member's identity chain of public certificates

Relays

PDUs are confined to a single subnet while the *publications* (information items) of a Domain-wide collection may move between subnets of the same trust domain or between compatible subdomains. Defined-trust communications employs **relays** to move publications between physically separate subnets and/or subdomains governed by different (but compatible in at least one type of publication) communications schema that may or may not use different subnets. These simple entities can be deployed to create a wide range of low or no configuration secure networks.

A relay has two or more separate defined-trust transports (as in fig 2), each of which is part of the same overall Trust Domain but may be in different subnets and/or subdomains, i.e., each transport may be governed by a different communications schema but must have the same trust anchor and there must be some compatibility or overlap of publication and identity definitions for there to be anything to relay (though there may be some disjoint transports in a multi-interface relay). Different transports of a relay may be configured with different identities or may all use the same identity. Relays pass publications from the collection at one of its transports to all its other transports; a relay doesn't *originate* any items for the **pubs** collection. Filtering of publications may be carried out but isn't required since relays ensure that publications are only added to a collection if all the criteria of receiving communications schema are met. The trust management engine of each transport uses its schema to determine whether to add publications to its local collections and will discard non-conforming items locally, i.e., before they are published, so filtering can be managed via schema.

Relays handle all collections of the Trust Domain, but publications, certificates, and keys each require different handling. Further, the collection of information about keys for encrypting PDUs is handled differently from keys for encrypting publications. Relays must be able to encrypt and decrypt PDUs on their subnets but should not be able to decrypt the publications they carry, passing them on with content intact.

Relays can be used to extend a trust domain geographically, to isolate communications to those subnets of a trust domain where they are relevant, and to create self-managing meshes. The sync's set reconciliation means that information propagates to all collections throughout the relay-connected Domain, schema permitting, transiting links or connected areas only once. Fig. 4 illustrates some relay roles: connecting broadcast subnets (that may use different media types) and extending a Domain geographically. On the right hand side of fig. 4, a TD is extended geographically by using a unicast connection (e.g., over a cell line or tunnel over the Internet) between two Relays which also interface to local broadcast subnets. Everything on each local subnet shows up on the other. A subDomain could be used here to limit the types of publications sent on the remote link, e.g., logs or alerts. Using this approach, local communications for subnet 1 can be kept local while subnet 2 might send commands and/or collect log files from subnet 1.

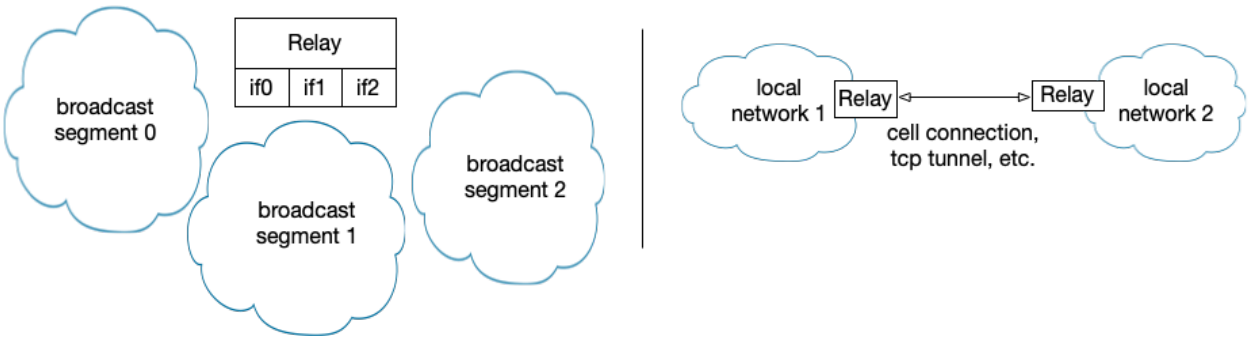

Figure 4. Relays extend trust domains and can separate subdomains

Another possible use of relays is to contain communications of a Domain in their local subDomain, only passing schema-permitted publications. This might be deployed to connect environmental (or other) monitoring information of a large company with offices distributed across the United States while keeping local information (e.g. instantaneous temperature measurements) within a subDomain while permitting some control information and summary measurements to pass (see fig. 5).

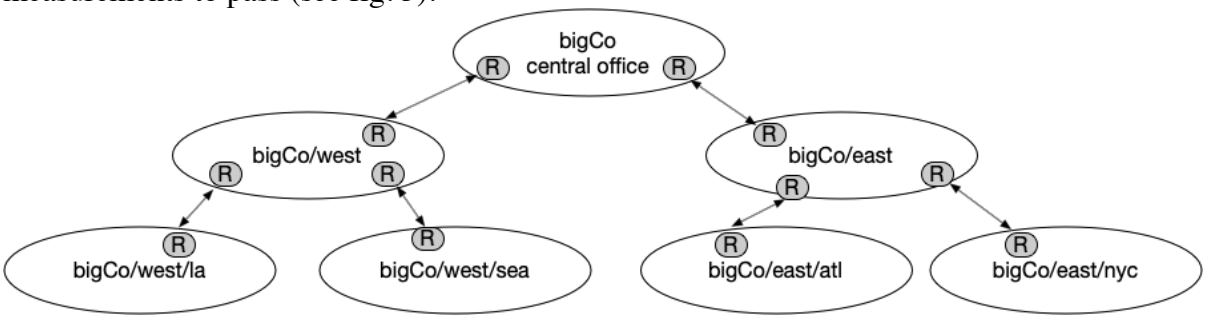

Figure 5. Relays extending a trust domains hierarchically

More generally, Relays can form a mesh of broadcast subnets with no additional configuration (i.e., Relays on a broadcast network do not need to be configured with others' identities). The mesh is efficient: publications are only added to an individual DeftT's collection once regardless of how it is received. Figure 6 illustrates relays (“R”) deployed where they may be connected via radio or wired media. It’s not necessary for every relay to be in direct contact with every other relay since their sync will ensure each has all publications (as long as there are not some relays that are completely disconnected). Relays communicate with members (“M”) on a different transport subdomain or subnet. The different member shapes are to indicate that they may be using different media to communicate or different subdomains (subschemas). The mesh of relays will ensure that all information gets where it needs to go and the trust management engines ensure information does not go where it should not. This requires no additional configuration of the relays beyond the schema and identity for each transport. Full connectivity of entities is not required. Note also that a relay entity can be colocated on a device with a member entity.

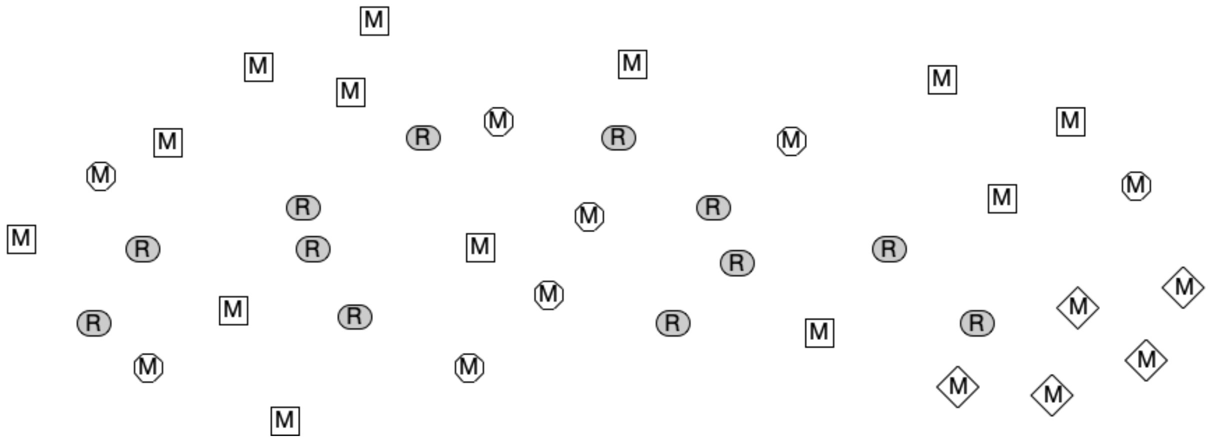

Figure 6. Relays provide meshed connectivity with no additional configuration

Subdomains are identified by the communications schema they are using and a relay can, in turn, use a subschema to limit the publications and certificates that are passed to it to those that are in use in that subdomain. For efficient communications, a subnet consisting of only relays may negotiate the subcollections and/or subschemas they will use on the subnet.

References

[ATZ] Ngabonziza, B., Martin, D., Bailey, A., Cho, H., and S. Martin, "TrustZone Explained: Architectural Features and Use Cases", IEEE 2nd International Conference on Collaboration and Internet Computing (CIC) November 1-3, 2016, https://doi.org/10.1109/CIC.2016.065.

[COMIS] Lydersen, L., "Commissioning Methods for IoT", February 2019, https://www.silabs.com/documents/public/presentations/ew-2019-iot-security-commissioning-methods-for-iot.pdf.

[DCT] Pollere LLC, "Defined-trust Communications Toolkit", 2022, https://github.com/pollere/DCT.

[DIFF] Eppstein, D., Goodrich, M. T., Uyeda, F., and G. Varghese, "What's the difference?: efficient set reconciliation without prior context", ACM SIGCOMM Computer Communication Review, August, 2011.

[DLOG] Li, N., Grosof, B., and J. Feigenbaum, "Delegation logic", ACM Transactions on Information and System Security, February 2003, https://doi.org/10.1145/605434.605438.

[DNMP] Nichols, K., "Lessons Learned Building a Secure Network Measurement Framework Using Basic NDN", Proceedings of ACM ICN '19, September 24-26, 2019, Macao, China.

[DTLD] Nichols, K., Jacobson, V., King, R., "Defined-trust Transport for Limited Domains", talk July 29, 2022, https://datatracker.ietf.org/meeting/114/materials/slides-114-iotops-defined-trust-transport-for-limited-domains-00

[DTM] Blaze, M., Feigenbaum, J., and J. Lacy, "Decentralized Trust Management", Proceedings IEEE Symposium on Security and Privacy, June 1996, https://doi.org/10.1109/SECPRI.1996.502679.

[Graphene] Ozisik, A. P., Andresen, G., Bissias, G., Houmansadr, A., and B. N. Levine, "Graphene: A New Protocol for Block Propagation Using Set Reconciliation", 2017, https://doi.org/10.1007/978-3-319-67816-0_24.

[HSE] Kapersky, "Secure Element", 2022, https://encyclopedia.kaspersky.com/glossary/secure-element/.

[IBLT] Goodrich, M. T. and M. Mitzenmacher, "Invertible bloom lookup tables", Forty-Ninth Annual Allerton Conference, UIUC, Illinois, 2011, https://doi.org/10.1109/Allerton.2011.6120248.

[IOTK] Nichols, K., "Trust schemas and ICN: key to secure home IoT", Proceedings of ACM ICN '21, September 22-24, Paris, France, https://doi.org/10.1145/3460417.3482972.

[LANG] LANGSEC: Language-theoretic Security "The View from the Tower of Babel"", 2021, http://langsec.org](http://langsec.org/.

[MQTT] OASIS, "MQTT: The Standard for IoT Messaging", 2022, mqtt.org

[NDN] "Named Data Networking Packet Format Specification 0.3", 2022, https://named-data.net/doc/NDN-packet-spec/current/.

[NMUD] D. Dodson et al, "Securing Small-Business and Home Internet of Things (IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD)", NIST Special Publication 1800-15, May 2021, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1800-15.pdf.

[RFC2693] Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and T. Ylonen, "SPKI Certificate Theory", RFC 2693, DOI 10.17487/RFC2693, September 1999, https://www.rfc-editor.org/info/rfc2693.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, https://www.rfc-editor.org/info/rfc4291.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, https://www.rfc-editor.org/info/rfc8995.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, https://www.rfc-editor.org/info/rfc8799.

[SDSI] Rivest, R. L. and B. W. Lampson, "SDSI - A Simple Distributed Security Infrastructure", April 1996.https://people.csail.mit.edu/rivest/sdsi11.html

[SNC] Smetters, D. K. and V. Jacobson, "Securing Network Content", October 2009, https://named-data.net/wp-content/uploads/securing-network-content-tr.pdf.

[SRM] Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L., **A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing**, IEEE/ACM Transactions on Networking, December 1997, Volume 5, Number 6, pp. 784-803.

[TPM] Griffiths, P., "TPM 2.0 and Certificate-Based IoT Device Authentication", September 2020, https://www.globalsign.com/en/resources/white-papers-ebooks/white-paper-tpm-20-and-certificate-based-iot-device-authentication.

[TST] K. Nichols, "Versec/DCT to create and use trust schemas", ACM ICN 2021 Tutorial, https://conferences2.sigcomm.org/acm-icn/2021/assets/tutorial-trust-schema/2-versec-dct-e1c6ddae2a10c47df55846b58ab84b93aeaff30d1eb00fa4fe9383afb3abca59.pdf