

Internet
Internet-Draft
Intended status: Informational
Expires: March 13, 2020

S. Rass
Universitaet Klagenfurt
Y. Qu
L. Han
Futurewei
September 10, 2019

Multipath Use Case and Requirement for Security
draft-rass-panrg-mpath-usecase-01

Abstract

This document describes a use case of multipath to achieve full CIA+ by using symmetric cryptography and point-to-point shared secrets.

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 March 13, 2020.

Copyright Notice

Copyright (c) 2019 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Internet-Draft

Mpath security use case

September 2019

Table of Contents

1.	Introduction	2
1.1.	Requirements Language	3
2.	Assumptions	3
3.	Multipath Routing	3
3.1.	Multi-path Service and User-Network Interface	4
3.2.	Path and Routing Reliability	4
3.3.	Cross Domain Path Reliability	5
3.4.	Cross Domain Network Connections	5
3.5.	Updates upon Changing Network Topologies	5
3.6.	Enforced Device Pairing and De-Pairing	6
4.	Summary	6
5.	Security Considerations	6
6.	Acknowledgements	6
7.	References	6
7.1.	Normative References	6
7.2.	Informative References	7
Appendix A.	Cryptographic and Graph-Theoretic Basics	9
A.1.	Secret Sharing	9
A.2.	Network Connectivity	9
Appendix B.	Multipath Transmission and Game-Theoretic Security	10
B.1.	End-to-end Confidentiality - Parallel Version	10
B.2.	End-to-end Confidentiality - Sequential-Parallel Version	10
B.3.	Randomized Routing to Maximize Security against Node (Failures)	11
B.4.	Availability	12
B.5.	End-to-End Authenticity	12
B.5.1.	Non-Repudiation	13
B.6.	Integrity	13
	Authors' Addresses	14

[1.](#) Introduction

Public-key cryptography is a convenient tool for end-to-end security, but in practice can be cumbersome or complicated for non-expert users to apply. Certificate- and key management rely on complex infrastructures and to a significant extent impose monetary cost and human effort.

This document describes a method of using symmetric cryptography and point-to-point shared secrets to establish full CIA+ (confidentiality, integrity, availability and authenticity) end-to-

end security. The respective schemes rely on multipath transmission and threshold cryptography, and are intended to work transparently for the users, i.e., entirely below the application layer. The only involvement of human action is for the key establishment, which is in

our setting equivalent to a pairing of devices, as is familiar from other contexts, such as Bluetooth.

[1.1.](#) Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

[2.](#) Assumptions

We assume a network of bidirectional links, represented as an undirected graph $G=(V,E)$. An edge $e=(v_1, v_2)$ in the set E represents a point-to-point connection between the nodes v_1 and v_2 in the network. We assume that every such pair (v_i, v_j) in E shares an individual secret k_{ij} , which is individually distinct for all edges (i.e., no two pairs have, other than by coincidence, the same secret). The secret exchange or establishment is left to arbitrary means, e.g., any device pairing scheme [[I-D.ietf-dnssd-pairing](#)] or cryptographic methods like Diffie-Hellman key exchange [[RFC2631](#)] would be admissible, up to quantum key distribution [BB84]. Indeed, end-to-end security in quantum networks is the most natural application area of multipath transmission as we discuss here.

We further assume that keys between adjacent nodes in the network have been exchanged in an authentic manner; say, by sufficient proximity during the device pairing (e.g., near field communication).

[3.](#) Multipath Routing

Multipath routing offers the remarking ability of establishing public-key like security without computational intractability. This means that periodic updates of keys or server certificates are no longer required in such systems; updates to keys for symmetric crypto are much easier by device re-pairing or refreshing keys from existing

key material, such as is done in quantum key distribution (QKD). Multipath transmission, requiring no quantum technology per se, offers nonetheless the same level of security QKD [BB84] and can resist attacks by quantum computers (like post-quantum cryptography [BD08]).

The key element to this end is using multiple paths to send a message, which in the simplest instance is just like humble symmetric encryption: consider two nodes A and B that have no direct connection between them (i.e., A and B are several hops apart). Let us assume that two paths connect A to B, where those paths intersect only at A and B (we call such paths node-disjoint). If so, then A can choose a

session key k that it sends to B over the first path, and deliver the encrypted payload over the second path. If the encryption is chosen properly (e.g., Vernam cipher [Ver55]) and the adversary does not intercept both paths, the connection remains secure.

The scheme straightforwardly generalizes to more than two paths, where the payload is (always) split into shares and transmitted over separate paths in parallel or sequentially. Security comes from the proper encoding/creation of the pieces so that an attacker needs to intercept a certain number of paths in order to breach confidentiality, insert a forged message, or cause a denial of service. The fundamental circumstance implying security here is the existence of k (≥ 2) node and link disjoint paths, so that an adversary needs to conquer at least k nodes in the network to breach security (by mounting a person-in-the-middle attack).

Security (up to QKD without trusted relays) thus hinges on the following network-related assumptions:

[3.1.](#) Multi-path Service and User-Network Interface

There are at least two disjoint paths between node A and node B, so A can send packets to node B via different paths efficiently and reliably.

New User-Network Interface (UNI) should be defined to exchange information between end device/application and network. The information may include but not limited to:

- o User expectation: such as number of paths, bandwidth required etc.
- o Path aware info: the network should dynamically provide end-device information such as number of paths available, each path's attributes: path reliability, routing quality, bandwidth, path elements etc.

[3.2.](#) Path and Routing Reliability

The sender A can deliberately choose any among the existing paths to its receiver B to transmit a message. The routing is reliable in the sense that there is at least a probabilistic guarantee for the packet to travel over exactly the chosen route with a likelihood p that A can quantify (not necessarily control). In other words, the chances for the path to be blocked, or for the packet to take a detour for any reason (e.g., load balancing, temporary congestions, or similar) is at most $1-p$, with the value of p being known to A. The ideal case $p = 1$ expresses that the chosen route has a perfect reliability (i.e., no deviations and guaranteed delivery).

It is admissible to express the path reliability in terms of several such probabilities, referring to different dimensions for the quality of service. That is, we may define a probability p_1 for the packet to stay on the chosen route, another probability p_2 for the packet to be delivered at all (i.e., not being blocked), or similar.

There are per se no stringent constraints regarding latencies or for several packets to arrive in the order of transmission, since the outer (cryptographic) transmission protocols can handle this. However, the aforementioned probabilities quantifying the quality of routing need to be accurately known to the sender A. Suitable protocols to handle path deviations (temporary detours) and to optimize quality-of-service tradeoffs based on such knowledge are found in the literature [[Ras13](#)], [RK12].

[3.3.](#) Cross Domain Path Reliability

If two distinct network domains are joined together, the topologies of both networks are reliably made known to the nodes in the respective other network. Chosen routes from one network into the other must remain quantifiably reliable in the sense of [section 3.2](#) above, i.e., a node A in one network must still be able to determine

a probability p for a packet to stay on its route and to arrive at the designated destination across all network domains that it traverses.

[3.4.](#) Cross Domain Network Connections

Whenever a node A has an outside connection to a node in another network domain N_2 , A should not have a second connection to another node in the same network domain N_2 . That is, if two network domains N_1 and N_2 are joined together via k links, those links should pairwise connect k distinct nodes in N_1 to another k distinct nodes in N_2 . This assures that the so-constructed larger network retains the necessary number of (at least) k node disjoint paths across the domains (by avoiding bottle-neck connections between networks N_1 and N_2).

[3.5.](#) Updates upon Changing Network Topologies

The information described under the preceding sections needs to remain up-to-date whenever A wishes to send a packet somewhere. Changing topologies such as in ad hoc networks call for a proper and reliable updating scheme to A 's local information about the network topology. This includes also changes in topologies of remote network domains (that the sender does not itself belong to).

[3.6.](#) Enforced Device Pairing and De-Pairing

Whenever a node X joins a network, it must establish shared secrets (for cryptography) with any neighbor with whom it has a direct point-to-point connection. Whenever a node X leaves the network, nodes losing the connection to X need to abandon their cryptographic key formerly assigned to the connection with X . The key exchange protocols can be arbitrary (cf. [section 2](#)), but the device pairing must in any case be authenticated.

[4.](#) Summary

The ability to route messages along chosen paths in a network, together with sufficient vertex connectivity and unique neighborhoods for each node opens up the possibility to achieve end-to-end

security:

- o without public-key cryptography.
- o using only light-weight symmetric cryptographic primitives (encryption and hashing).
- o and with the most trivial key-management consisting of only the exchange of keys between directly connected devices (along device pairing).

[5.](#) Security Considerations

TBD.

[6.](#) Acknowledgements

TBD.

[7.](#) References

[7.1.](#) Normative References

[I-D.ietf-dnssd-pairing]

Huitema, C. and D. Kaiser, "Device Pairing Using Short Authentication Strings", [draft-ietf-dnssd-pairing-05](#) (work in progress), October 2018.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", [RFC 2631](#), DOI 10.17487/RFC2631, June 1999, <<https://www.rfc-editor.org/info/rfc2631>>.

- [RFC5510] Lacan, J., Roca, V., Peltotalo, J., and S. Peltotalo, "Reed-Solomon Forward Error Correction (FEC) Schemes", [RFC 5510](#), DOI 10.17487/RFC5510, April 2009, <<https://www.rfc-editor.org/info/rfc5510>>.
- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", [RFC 6824](#), DOI 10.17487/RFC6824, January 2013, <<https://www.rfc-editor.org/info/rfc6824>>.

7.2. Informative References

- [FFGV07] Matthias Fitzi, Matthew K. Franklin, Juan Garay, and S. Harsha Vardhan, TCC, LNCS, vol. 4392, Springer, 2007, pp. 311--322., "Towards Optimal and Efficient Perfectly Secure Message Transmission".
- [MS81] R. J. McEliece and D. V. Sarwate, Commun. ACM 24 (1981), no. 9, 583--584., "On Sharing Secrets and Reed-Solomon Codes".
- [Ras13] Stefan Rass, Springer Journal of Network and Systems Management 21 (2013), no. 1, 47--64., "On Game-Theoretic Network Security Provisioning".
- [Ras14] Stefan Rass, International Journal of Advanced Computer Science and Applications 5 (2014), no. 2, 148--157., "Complexity of Network Design for Private Communication and the P-vs-NP question".
- [Ras18] Stefan Rass, CoRR abs/1810.05602 (2018)., "Perfectly secure communication, based on graph- topological addressing in unique-neighborhood networks".
- [RS10] Stefan Rass and Peter Schartner, Proceedings of the International Conference on Security and Management (SAM), vol. 1, CSREA Press, 2010, pp. 111--115., "Multipath Authentication without shared Secrets and with Applications in Quantum Networks".

- [Sha49] C. E. Shannon, Bell System Technical Journal 28 (1949),

656--715., "Communication Theory of Secrecy Systems".

[Sha79] Adi Shamir, ACM 22 (1979), no. 11, 612--613., "How to share a secret".

[Appendix A](#). Cryptographic and Graph-Theoretic Basics

[A.1](#). Secret Sharing

We assume a message m to come as a binary string of length L . A simple k -out-of- k secret sharing is by picking a set of $k-1$ random strings $s_1, s_2, \dots, s_{(k-1)}$ of the same length as m and computes $s_k := m \text{ XOR } s_1 \text{ XOR } s_2 \text{ XOR } \dots \text{ XOR } s_{(k-1)}$. Information-theoretically, one can prove [[Sha49](#)] that the recovery of m is impossible from any set of less than k of the strings s_1, \dots, s_k (since the missing string effectively acts as a one-time pad concealing m).

The sharing as just described is replaceable by more sophisticated schemes, such as Shamir's polynomial sharing [[Sha79](#)], which adds error correction capabilities [[MS81](#)] via using an isomorphism to Reed-Solomon encoding. We shall, however, hereafter not further relate to standardized versions of Reed-Solomon forward error correcting codes [LRPP09], but rather work with the above simple scheme instead.

Abstractly, we shall introduce a sharing function $\text{SPLIT}(m, k)$ that decomposes an input message m into a set of k shares according to any scheme of choice (for the description in this document, the above XOR-based scheme will suffice). The inverse of SPLIT will be the function $\text{COMBINE}(s_1, \dots, s_k)$, taking k (out of a potentially larger set) of shares to reconstruct the message m from it. Note that COMBINE internally may invoke error correction algorithms [LRPP09], which we do not further expand here.

[A.2](#). Network Connectivity

If a node A wants to transmit a message m to a node B , we assume that A can choose a path, or a set of paths through the network along a physical connection (over multiple hops) to the end-node B . Further, we assume that the network's node connectivity is such that more than one route from A to B exists, and that at least two routes exist that do not intersect other than at A and B (node-disjoint paths). It is known that the existence of k node-disjoint paths is equivalent to the graph admitting a k -vertex cut; equivalently, we call such a graph k -vertex-connected. The smallest graph with that property is the complete graph with $k+1$ nodes. Furthermore, if two k -vertex-connected graphs are given, we can combine them into one (big) k -vertex connected graph G as follows: we pick k distinct nodes u_1, \dots, u_k in G_1 and another k distinct nodes v_1, \dots, v_k in G_2 , and connect the two graphs by adding edges (u_i, v_i) for all $i=1,2,\dots,k$. The resulting graph contains all nodes and edges from

G_1 and G_2 , plus the connecting edges between the two graphs. It is provably a k -vertex-connected graph, admitting at least k node-

disjoint paths between any two nodes in either graph and from any node in G_1 to any node in G_2 and vice versa.

While it may not be too optimistic to hope for a large k in the existing internet topology, matters of resilience against failure of single nodes in the network call for a least $k=2$, so that the network remains connected if one node (and hence the adjacent edge) fails.

Let A 's available routes to B be enumerated as R_1, \dots, R_k , which A picks with likelihoods p_1, \dots, p_k , say, $p_i := 1/i$ for an equiprobable choice of a single route. Moreover, we let each transmission use their point-to-point shared secrets to encrypt a message along the network edge (v_i, v_j) under the key k_{ij} (e.g., by means of the Advanced Encryption Standard or others).

[Appendix B](#). Multipath Transmission and Game-Theoretic Security

[B.1](#). End-to-end Confidentiality - Parallel Version

To confidentially transmit the message m , A proceeds as follows:

1. Decompose m into shares $\{s_1, \dots, s_k\} := \text{SPLIT}(m)$
2. Send each share s_i over the route R_i (for $i = 1, \dots, k$) in parallel to B .
3. B , upon receiving all shares recovers the message as $m := \text{COMBINE}(r_1, \dots, r_k)$.

By construction, the attacker needs to gather all k shares to recover m , so that if the attacker can intercept only less than k paths, the message m remains perfectly concealed (by the aforementioned arguments). A picture of the scheme is found at

<https://www.syssec.at/user/themes/syssec-theme/images/publikationen/MPTrans.png>

[B.2](#). End-to-end Confidentiality - Sequential-Parallel Version

The above scheme can be further strengthened by a two-stage sharing

as follows: as before, let m the message that A wishes to send to B in perfect privacy. It proceeds as follows:

1. Decompose m into n shares $\{s_1, \dots, s_n\} := \text{SPLIT}(m)$
2. For $i = 1, 2, \dots, n$: send each share s_i by the parallel scheme described above; resulting in the transmission of shares r_{i1}, \dots, r_{ik} for the share s_i

3. The receiver B then needs to (i) reconstruct every share $s_i := \text{COMBINE}(r_{i1}, \dots, r_{ik})$ (as in the parallel version above), and (ii) reconstruct the overall message as $m := \text{COMBINE}(s_1, \dots, s_n)$.

An attacker needs to intercept the entirety of shares for each individual transmission, as well as for all the sequential transmissions. Unless the attacker can mount a full person-in-the-middle attack, the message m remains perfectly concealed. Even if the attacker has a positive probability $q < 1$ to catch all shares for a single transmission in step 2, the probability to catch the entirety of n sequential shares (created in step 1) equals $(1-q)^n$ (the path choices are made stochastically independent). In choosing n large enough, A can make the adversary's success chances exponentially small.

[B.3.](#) Randomized Routing to Maximize Security against Node (Failures)

Suppose that the attacker can intercept a fixed maximum number $t < k$ of nodes, where k is the network's vertex connectivity. If the network is such that certain routes are more or less reliable than others (e.g., some routes may be easier to intercept for the adversary or temporarily be unavailable), there is no obligation in the above scheme to use the full set of paths per parallel transmission. Instead, to transmit a share (whether in the parallel or sequential-parallel version of the transmission), the sender may randomly pick the route R_i with likelihood p_i , and transmit the share over the chosen route.

Knowing the choice rules p_1, \dots, p_k for the k routes that A can choose from, the attacker may seek to compute an optimal strategy for intercepting, resulting in probabilities $q_1, \dots, q_{|V|}$ for nodes to

attack (excluding the nodes for A and B here, since our security goal is confidentiality, disregarding impersonation attacks for the moment).

The optimal computation of probabilities to choose routes, and individual likelihoods to intercept nodes amounts to a simple two-person matrix game [[Ras13](#)], whose saddle-point value (computable by means of linear optimization) systematically quantifies (bounds) the likelihood for the attacker to succeed. For a simplified example, assuming that all nodes are equally "easy" for the attacker to conquer, yet with a bound to no more than 1 node to be under the adversary's control at a time, the optimal choice for the sender A would be an equiprobable pick among the routes, i.e., $p_i := 1/k$ for all i , and an equiprobable choice of victim nodes for the attacker (here, we assumed that the sender uses only a single path at a time).

[B.4.](#) Availability

The XOR sharing used in [Section 2.1](#) is vulnerable against packet loss (whether this happens by coincidence or due the attacker's actions; DoS attacks). Making the scheme resilient against packet loss or damage calls for error correction capabilities within the COMBINE function, e.g., using the methods described in [LRPP09]. A full-fledged scheme using Reed-Solomon error correction towards optimized availability and confidentiality is described by [[FFGV07](#)].

[B.5.](#) End-to-End Authenticity

Using a similar idea [[RS10](#)], authenticity of messages is accomplishable by message authentication codes. Since the sender A shares secrets only with her/his direct neighbors, it can only use their secrets to attach a message authentication code. The receiver B, being several hops away from A, does not know the secrets to verify the MAC, but, thanks to its ability of chosen path routing, can ask A's neighbors to verify the MACs on B's behalf.

Putting this to practice, A authenticates a message m for B as follows, using the keys $\{k_1, \dots, k_n\}$ that A shares with its direct neighbors in the network. We write $\text{MAC}(m, k)$ to denote a message authentication code (MAC) for a message m computed under the (secret) key k . Moreover, let H be a cryptographically strong hash function

(e.g., SHA-3 or likewise).

1. A computes hash-MACs, e.g., using the HMAC scheme in [RFC2104], and attaches the MACs $\{a_i := \text{MAC}(H(m), k_i) \mid i=1,2,\dots,n\}$ to the message.
2. B receives the message m' (say, over a multipath transmission scheme with chosen routes as described above). To verify that m' is authentic, B computes the hash $h' = H(m')$ and asks A's neighbors to verify the respective MACs. To this end, B contacts the i -th neighbor of A on a chosen route, and sends the data $\{h', a_i'\}$ to A's neighbor with whom A shares the secret k_i . Here, the value a_i' is the MAC that B received (which could equally well have been corrupted).
3. A's neighbor no. i uses its secret k_i to verify if $\text{MAC}(h', k_i) =?= a_i'$. It replies the result ("yes" or "no") back over the same route as how the query came in. This process happens concurrently at all of A's neighbors (for $i = 1, \dots, n$).
4. B collects all replies and takes either a majority decision or (in the most stringent setting) rejects if any of the replies comes back negative.

A picture of the scheme is found at

<https://www.syssec.at/user/themes/syssec-theme/images/publikationen/MPAuth.png>

The condition upon which B accepts A's message as authentic may depend on how resilient one needs to be about an adversary potentially manipulating the verification query to B. If B rejects upon a single negative verification, then even an attacker that can conquer only a single node on any of the chosen paths can mount a denial-of-service. On the contrary, if B accepts the majority vote, then the attacker needs to intercept (and manipulate) more than half of the routes chosen.

The security of this scheme follows by similar arguments as in the case for confidentiality: the scheme is secure as long as the adversary cannot mount a full person-in-the-middle attack, conditional on the attacker's inability to find hash-collisions (in that sense, the scheme is, unlike the multipath transmission for

confidentiality), only computationally secure.

Note that confidentiality of the message against the verifying neighbors is not directly addressed here beyond the point of sending a hash of m for verification instead of the full message.

Heuristically, the message thus remains concealed to the extent of the neighbor's inability to find a meaningful pre-image for the received value h' . We assumed the neighbors to be honest, unless being under the attacker's control, so that a denial-of-service or intentionally incorrect response is in any case possible, and cannot be ruled out by this protocol.

[B.5.1.](#) Non-Repudiation

Under proper graph topological properties, the above authentication scheme, though based on symmetric cryptography only, shares the non-repudiation feature of public-key digital signatures. In fact, if the set of secrets shared between a node and its direct neighbors (or a subset thereof) is unique, i.e., distinct, for each node, then no other node than A can create the MAC-set attached to the message m . Networks with that property are easy to recognize based on their adjacency matrix [[Ras18](#)]; moreover, the "unique-neighborhood property" is preserved upon the same network merging operations as described above for k -vertex-connectivity.

[B.6.](#) Integrity

From the construction of [Section 3.4](#), integrity is directly implied by the use of hashes that additionally act as checksums. That is, any distortion on the transmission line will with overwhelming

probability invalidate the MAC or inner hash, thus causing the protocol to indicate this error. Conversely, if B accepts A 's message as authentic, integrity verification is accomplished in the same blow, unless the attacker managed to forge the message as a whole (in which case, integrity is also unhedgeable).

Authors' Addresses

Stafan Rass
Universitaet Klagenfurt

EMail: stefan.rass@aau.at

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: yingzhen.qu@futurewei.com

Lin Han
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: lin.han@futurewei.com