

SECURE OUT-OF-BAND SYMMETRIC ENCRYPTION KEY DELIVERY**FIELD**

[0001] The present embodiments relate to the fields of cryptographic communication systems and cryptographic communication in networks.

BACKGROUND

[0002] Encryption and decryption are commonly used for cryptographic communication over various networks. Typical encrypted communication uses symmetric keys to encrypt and decrypt data between 2 points. These keys are typically derived from Diffie-Hellman negotiation at the beginning of the data exchange utilizing the same link. With enough computing power, eavesdropping on the data link would allow the attacker to deduct the symmetric keys from the exchange. So far this danger was strictly theoretical, but advent of quantum computers makes this brute-force attack vulnerability very real.

[0003] QKD (Quantum Key Distribution) systems are currently being developed to provide symmetric keys to both parties with strong physical guarantee from eavesdropping. Practical usage of QKD systems is currently inhibited by limitations regarding distance and the fact of only being able to work as a pair of devices (Alice/Bob) with no capabilities to expand the network to more than pairs of Alice/Bob.

SUMMARY

[0004] A method is disclosed for secure out-of-band symmetric encryption key delivery. A first trusted node in a network receives a request. The request is from a first one of a pair of user devices. The request is to deliver symmetric encryption keys to the pair of user devices. The first trusted node delivers a second one of the symmetric encryption keys to a second one of the pair of user devices. The symmetric encryption key is delivered via further trusted nodes in the network. The first trusted node receives confirmation from one of the further trusted nodes in the network. The confirmation is of delivery of the second one of the symmetric encryption keys to the second one of the pair of user devices. The first trusted node delivers a first one of the symmetric encryption keys to the first one of the pair of user devices. This delivery is responsive to the confirmation of the delivery of the second one of the symmetric encryption keys to the second one of the pair of user devices.

[0005] A tangible, non-transitory, computer-readable media is disclosed. The media has instructions thereupon which, when executed by a processor, cause the processor to perform a method. A first trusted node in a network receives a request from a first user device. The

request is to deliver symmetric encryption keys to the first user device and a second user device. From the first trusted node, a second symmetric encryption key is delivered to the second user device. The key is delivered via trusted nodes in the network. At the first node, confirmation is received from one of the trusted nodes in the network. This is confirmation of delivery of the second symmetric encryption key to the second user device. From the first trusted node in the network, a first symmetric encryption key is delivered. The first and second symmetric encryption keys are symmetric to each other. The first symmetric encryption key is delivered to the first user device, responsive to the confirmation of the delivery of the second symmetric encryption key to the second user device.

[0006] An apparatus for secure out-of-band symmetric encryption key delivery is disclosed. The apparatus has a plurality of nodes of a network that are to form trusted nodes in a peer-to-peer or mesh network. A first trusted node, which is one of the trusted nodes, is to receive a request from a first user device to deliver symmetric encryption keys to the first user device and a second user device. The first trusted node is to deliver a second symmetric encryption key to the second user device. The key is to be delivered from the first trusted node, via the trusted nodes. The first trusted node is to receive confirmation of delivery of the second symmetric encryption key to the second user device. This confirmation is to be received from one of the trusted nodes. The first trusted node is to deliver a first symmetric key to the first user device. This first symmetric key is symmetric to the second symmetric encryption key. This delivery is responsive to the confirmation of the delivery of the second symmetric encryption key to the second user device.

[0007] Other aspects and advantages of the embodiments will become apparent from the following detailed description taken in conjunction with the accompanying drawings which illustrate, by way of example, the principles of the described embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The described embodiments and the advantages thereof may best be understood by reference to the following description taken in conjunction with the accompanying drawings. These drawings in no way limit any changes in form and detail that may be made to the described embodiments by one skilled in the art without departing from the spirit and scope of the described embodiments.

[0009] Fig. 1 depicts a system for out-of-band delivery of cryptographic keys in accordance with present embodiments. TX1 to TX7 represent Trusted Xchange Nodes (TX Nodes). U1 to U4 represent Users of the system. There is also a depiction of the system that can provide additional encryption keys, similar to, but not limited to, a QKD Alice/Bob pair.

[0010] Fig. 2 depicts a TX Node information collection loop (“chat with the neighbors”) from the perspective of Node TX1, suitable for use in embodiments of the TX Node network of Fig. 1.

[0011] Fig. 3 depicts an out-of-band encryption key(s) (“Parcel”) delivery in the Mesh of TX Nodes in accordance with present embodiments.

[0012] Fig. 4 depicts an embodiment of a path calculation process, suitable for use in embodiments of the out-of-band symmetric key delivery system of Fig. 1 and the Parcel delivery depicted in Fig. 3.

DETAILED DESCRIPTION

[0013] It is important to have additional pair(s) of keys delivered outside the channel used for the data exchange. Separating the data and key delivery channels makes the brute force quantum computer attack practically impossible. Combining keys delivered inline by traditional methods and out-of-band by using systems and methods described herein allows for unobtrusive deployment on existing networks while significantly increasing an encrypted channel’s resistance to attacks.

[0014] Various embodiments of a system disclosed herein remove limitations regarding distance and network expansion, work with or without the QKD system, and introduce a self-organizing mesh network of Trusted Xchange Nodes (i.e., Trusted Exchange Nodes or TX Nodes) with adaptive routing and fault tolerance, which also can benefit from additional in-transit key encryption from QKD Alice/Bob pairs where available.

[0015] A cryptographic communication system with key delivery, implemented on various networks including mesh networks and peer-to-peer networks, is described herein. TX Nodes in a network connect via TLS (Transport Layer Security) with strong encryption and mutual certificates, i.e. client validation of server and server validation of client, where client and server are defined per TLS communications.

[0016] TX Nodes communicate and cooperate to discover neighbors, generate keys per user requests, deliver key(s) to requested TX Node(s), confirm delivery of the key(s), find an optimal delivery route, find another route if one of the neighbor TX Nodes is inaccessible and repeat the procedure. In embodiments with features that can be separate, in variations, or in various combinations, the keys are encryption symmetric keys (or symmetric encryption keys), the network is a mesh network or a peer-to-peer network, and/or the keys may be additionally encrypted by external keys in each hop-to-hop transmission. External keys may be generated by connected QKD Alice/Bob pair(s), (e.g., each symmetric key being delivered is wrapped by an external key on that particular hop) in one or more, or all network hops. Some embodiments

perform out-of-band symmetric key delivery in combination with peer-to-peer communication. Some embodiments perform path optimization to various criteria, such as minimizing hops, or availability of external keys, or availability of external QKD keys. Neighboring nodes in a network can establish trust and become trusted nodes in a mesh network or peer-to-peer network through one or more mechanisms, including one-way or two-way exchange or verification of certificates or tokens. Further mechanisms or protocols for establishing trust, building a mesh network or peer-to-peer network, and pre-configuring nodes to trust each other are readily devised in keeping with the teachings herein.

[0017] Fig. 1 depicts a system for out-of-band delivery of cryptographic keys in accordance with present embodiments. TX1 to TX7 represent Trusted Xchange Nodes (i.e., Trusted Exchange or TX Nodes 106, 108, 110, 112, 114, 116, 118), U1 to U4 represent Users with corresponding User Devices 102, 120, 104, 122 of the system. Two user devices engage in in-band cryptographic (i.e., encrypted, decrypted) communication with each other, using cryptographic keys received through out-of-band delivery. There is also a depiction of the system that can provide additional encryption keys, similar but not limited to QKD Alice/Bob pair(s). A peer-to-peer or mesh network includes the TX Nodes 106, 108, 110, 112, 114, 116, 118 individually shown in Fig. 1, and these nodes are communicating in peer-to-peer mode with their neighbors. This communication is encrypted by TLS with both client-server and server-client certificate validation. TX Nodes only talk to neighbors which were configured by the system administrator and belong to the same Certificate Authority. Collectively, with more than two TX Nodes involved, they can create a mesh network. Some TX Nodes 108, 110, 112, 114 may not have user devices attached to them and only function as transition nodes, e.g., used for forwarding or routing. This is an important function that lets the system as a whole obtain fault tolerance and also allows for secondary encryption performed with external keys from limited-range QKD Alice/Bob pairs. The peer-to-peer or mesh network, and more specifically the TX Nodes 106, 108, 110, 112, 114, 116, 118 in the network, perform a process to deliver cryptographic keys between any given pair of user devices 102, 120, 104, 122 connected to the network. Processors executing software, firmware, hardware, and various combinations thereof can perform the various actions of this and other processes in various embodiments. Connections in the peer-to-peer or mesh network can be one-to-one (e.g., an end TX Node), many-to-one, or many-to-many, in various embodiments.

[0018] TX Nodes 106, 108, 110, 112, 114, 116, 118 periodically communicate with their pre-configured neighbors, pulling the information about neighbors of the neighbors and all the connected User Devices. Thus, in case of a Mesh setup, each TX Node 106, 108, 110, 112, 114, 116, 118 maintains a near-real-time map of all nodes in the Mesh and all active User Devices

while only requesting the information from the allowed neighbors. This Map automatically changes if new TX Nodes / User Devices are added/removed by the System Admin or if some nodes become inaccessible.

[0019] In one scenario, one of the User Devices 102, for example U1, requests key(s). U1 also specifies another User Device 104 which should receive the same key(s), for example U3. The request for keys is sent to the TX Node 106 this User Device 102 is connected to, in this example TX1. The key request is performed in accordance to mutually agreed specifications and may include TLS encryption and certificate validation and may be performed, e.g., via REST API (Representational State Transfer Application Programming Interface). As another variant, it can be performed via serial communication in a format both User Device and TX Node agree upon.

[0020] Upon receiving the request, a TX Node 106 (TX1 in this example) generates the requested key(s) and prepares to deliver them to the U3 first before returning the keys to the requestor user device 102 (U1). TX1 looks up the name of the TX Node 116 that U3 is connected to, in this example TX6. TX1, being only allowed to talk to its pre-configured neighbors, in this case TX Nodes 108, 112 TX2 and TX4, calculates the next hop based on its current nodes Map. In this example TX Node 106 TX1 will see several equal-length paths -- i.e., both TX2 and TX4 would have equal weight. In other examples, paths could have different weighting. In this example TX1 will perform a load-balancing action and send the generated key to TX4 if the previous key delivery transaction was performed via TX2 and vice-versa. The internal data block is called a Parcel and has a destination TX Node label attached to it. Upon receiving this Parcel from TX1, TX4 will perform a similar calculation of next hop and find out that TX4 has only one alternative -- namely, TX Node 114 TX5. The operation of optimal path calculation and potential load-balancing is performed on every transitional TX Node (hop) until the Parcel reaches its destination TX Node 116 (in this example TX6). Each TX Node in the chain keeps the connection from the previous peer open in this transaction so any potential error is immediately returned within the context of this transaction. Upon successful delivery of the Parcel with the keys to TX6, TX1 returns a success code and closes the connection. Each previous TX Node does the same until the success code reaches the originator (in this example TX1). Upon receiving the success code, TX1 finally returns the requested key(s) to the U1 User Device 102. With key(s) successfully delivered to TX6, U3 can now request the key(s) after which the key(s) are erased from TX6. There is no storage of the delivered keys in transitional TX Nodes beyond the time of the Parcel delivery transaction.

[0021] The system can be configured such that TX Nodes supplied with additional encryption keys receive the path priority, for example by according greater weight to such paths.

In case of additional keys, for example from a QKD Alice/Bob pair, TX Nodes additionally encrypt/decrypt the Parcel using those keys on both sides of that particular hop. Such quantum key encryption and decryption can occur on one hop, two or more hops, or each and every hop of the Parcel along a path, in various embodiments.

[0022] The TX Node that generates the keys on request for delivery could get the Key Material from a pre-configured RNG (Random Number Generator) system, including but not limited to a QKD system, or a specialized QRNG (Quantum Random Number Generator), from an internal hardware RNG or from a standard software entropy-based RNG, or generate keys in various embodiments.

[0023] Fig. 2 depicts an embodiment of the information collection loop (“chat with the peers”) from the perspective of TX Node 106 TX1. Each Node 106, 108, 110 in the network must have a unique name. In this example node TX1 is pre-configured by a System Administrator so that TX1 is allowed to talk to Nodes TX2 and TX3. The exchange is implemented via REST API over TLS with both client – server and server – client certificate validation. TX1 in this example runs an independent “chat” thread per each peer Node that TX1 is configured to talk to. TX1 initiates a REST ‘GET /chat’ request with pre-set interval T to TX2 and TX3. Each TX Node generates a one-time UUID (Universally Unique Identifier) that lasts while a TX Node process is running and is replaced when the process restarts. There is no long-term storage for this UUID. This UUID is supplied alongside of the TX Node name in each response to a “chat” request. In this example, TX1 receives information from each of its peers that is structured as an array of ([TX Name, TX UUID] -> array of (peer names), array of (User Device names)), although other ways of organizing and recording this information are readily devised in keeping with the teachings herein. TX1 then creates memory structures for lookups for where User Devices are located and which TX Node knows which peers. TX1 also checks for changes of UUID associated with TX Names and replaces appropriate lookup slots when a UUID has changed. TX1 also marks TX name/UUID pair as “stale” if TX1 did not see that particular UUID in responses in $T \times 2$ time. Absence of that UUID in more than $T \times 2$ time indicates that the TX Node in question is inaccessible and must be avoided during the path calculations. In the example in Fig. 2, all other Nodes perform similar REST “chat” requests to each other, thus building a collective picture of the TX Node network while only talking to allowed peers. Each TX Node 106, 108, 110 keeps its own copy of this collective memory as the TX node is relative to it. If the UUID/TX name pair was not seen within $T \times 11$ timeframe – the TX Node in question is removed from the picture completely. Otherwise it has a chance to be marked as an operational node again. Various further time frames or tests for removal and/or restoration of a node to the network are readily devised in keeping with the teachings herein.

[0024] The run-time UUID refresh method was chosen for various embodiments because it does not rely on the notion of global time that otherwise needs to be maintained across the TX Node network. Each TX Node 106, 108, 110 only relies on its own clock to calculate the “stale” timeouts and each TX Node clock can be skewed significantly without affecting inter-operability and collective memory. There are no timestamps exchanged in “chat” request/response, in some embodiments.

[0025] The GET method of REST API was chosen as most secure so no “impostor” TX Node can inject the incorrect information into the collective memory by just pushing the incorrect information, even if somehow the “impostor” TX Node passes the certificate validation.

[0026] Path finding thus embodies a TX Node receiving the Parcel with the keys for delivery and finding the next optimal peer TX Node to give this parcel to, based on the current TX Node network picture the TX node created in memory.

[0027] There is no permanent storage of the network memory on any TX Node, in some versions.

[0028] Approximate time of propagation of the newly configured TX Node across the network is bracketed between $T \times L$ and $T \times L \times 2$ where L is the length of the chain from TX1 (as in the Fig 2. example) to the TX Node in question. Thus, propagation time in the network is linear relative to the length of the chain of TX nodes.

[0029] Time to mark the Node (i.e., TX Node) “stale”, on the other hand, is of less significance because next hop calculation happens immediately on the TX Node that currently holds the Parcel with the keys and needs to find the next hop, independently of what other Nodes know about the current state of the Network. Relative to this Node this time is constant and bracketed between T and $T \times 2$.

[0030] Fig. 3 depicts an out-of-band encryption key(s) (“Parcel”) delivery in the Mesh of TX Nodes 106, 108, 110, 112, 114, 116 in accordance with present embodiments. This example also covers the rare situation when TX Node 116 TX6 is experiencing problems (or became unavailable) very recently so the information about this failure did not yet propagate through the Mesh “Chat” as described above.

[0031] Per example in Fig. 3, the TX Nodes Mesh is configured as follows: TX1 talks to TX4; TX4 talks to TX1 and TX5; TX5 talks to TX2 and TX6; TX2 talks to TX5 and TX3; TX3 talks to TX2 and TX6; TX6 talks to TX5 and TX3.

[0032] User Device 102 U1 requests the encryption keys from the TX Node 106 TX1 and also specifies its counterpart User Device 120 U2 which should receive the same key(s). U1 then waits for response. TX1 generates the requested key(s) and generates the Parcel. TX1 finds the

TX Node 110 that U2 is connected to from the current node map of TX1. This TX Node 110 happens to be the TX3. Thus, the Parcel is formed with the final destination specified as TX3. TX1 then sends the Parcel to its only peer, TX4 as Step 1 and waits for response. TX4 sends the parcel to TX5 in exactly same manner in Step 2 and waits for response.

[0033] TX4 and TX5 also happen to receive the additional symmetric encryption keys from external devices -- for example, quantum keys from a QKD Alice/Bob pair. Thus, in addition to a TLS layer for the communication, the Parcel itself is encrypted using those quantum keys.

[0034] TX5 would normally be aware that its peer TX6 has problems by performing the Chat with time interval T, but in this very rare case TX6 is experiencing difficulties or became unavailable in a time frame bracketed between 0 and T. So TX5 still thinks that its peer TX6 is operational and has equal-path load-balancing opportunity between TX2 and TX6. If TX6 is finally chosen as a next hop, TX5 sends the Parcel to TX6 in Step 3. The Error response 302 (or timeout) is raised (response arrow) and TX5 then sends the Parcel to its other peer TX2 in Step 4 and waits for response. TX2 sends the Parcel to TX3 in Step 5 and waits for response.

[0035] TX3 sees that the Parcel destination matches its name and perform actions in accordance with the method e.g., through an Application Programming Interface (API) for the User Device key(s) delivery. The parcel is deleted immediately after the key(s) are retrieved by the User Device 120 U2.

[0036] TX3 then returns a Success response 304 (response arrow) to TX2 and closes the transaction. TX2 returns a Success response 306 to TX5 and closes the transaction. TX5 returns a Success response 308 to TX4 and closes the transaction. TX4 returns a Success feedback 310 to TX1 and closes the transaction. TX1 gives the key(s) to waiting User Device 102 U1 in accordance with the method (e.g., through an API) for the User Device key(s) delivery and closes the transaction.

[0037] Typical delivery would involve TX5 learning about TX6 problems from the Chat and marking TX6 as Stale. This knowledge also propagates across all the nodes in the Mesh via Chat as described above. In this typical case the Parcel delivery path would consist of Steps 1, 2, 4 and 5 – avoiding Step 3 altogether.

[0038] The Parcel delivery is implemented using REST method “POST /parcel”. The Parcel is never stored on HDD (hard disk drive) or other storage memory while in transit to a User Device and is immediately deleted from memory after retrieval by the User Device.

[0039] It is typical that a User Device and its associated TX Node perform both client/server and server/client certificate validation and/or are connected via an otherwise highly-secure link, which may be implemented via serial interface for example.

[0040] Fig. 4 depicts an embodiment of a path calculation process, suitable for use in embodiments of the out-of-band symmetric key delivery system of Fig. 1 and the Parcel delivery depicted in Fig. 3. In the Fig. 4 example, TX Node 106 TX1 is the initiator of the Parcel delivery to the TX Node 116 TX6. Tx Node 106 TX1 keeps the current map of all the nodes in the Mesh and their peers as obtained via Chat as described above.

[0041] TX1 uses Step 1 to create a just-in-time map consisting of all the possible destinations, peers of TX1 through which those destinations are accessible, and number of hops to take to deliver to a particular destination via a particular peer. Stale TX Nodes are excluded from mapping.

[0042] TX1 then uses Step 2 to filter out all irrelevant records so only TX6 remains as a destination. In Fig. 4 example it yields 2 records with an equal number of hops. So TX1 switches to load-balancing mode and sends Parcels in round-robin fashion alternatively to TX2 and TX4.

[0043] As the original Map obtained from Chat changes, so does the Just-in-Time hop map which is being produced fresh on every Parcel delivery. As an example, when TX2 receives the Parcel destined to TX6, TX2 performs identical calculation taking its copy of the current map of all the nodes and their peers as obtained via Chat and, converting the map to an all possible destinations map with hops count, only now relative to the TX Node 108 TX2 (itself), and then filtering out by destination TX6. This will yield TX6->TX3->2 and TX6->TX5->2 alternatives which trigger again the load-balancing round-robin between TX3 and TX5. One will be chosen and the next identical step will result in delivery of the Parcel to the TX Node 116 TX6.

[0044] Path calculation happens on each TX Node 106, 108, 110, 112, 114, 116, 118 independently, on every Parcel delivery based on the freshest information obtained from the Chat as described above.

[0045] The above-described embodiments, examples and scenarios are generalizable to further embodiments in which a request to deliver symmetric encryption keys to user devices is received at a TX Node that is not directly connected to a user device, but which communicates with the user device(s) through further TX Node(s). The TX node that receives the request for symmetric encryption keys from a user device delivers one of the symmetric encryption keys to another specified user device, receives confirmation of delivery, then delivers the other symmetric encryption key to the user device that requested the keys, all through TX Nodes.

[0046] The foregoing specification references specific exemplary embodiments. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope intended. The specification and drawings are accordingly to be regarded in an illustrative rather than a restrictive sense.