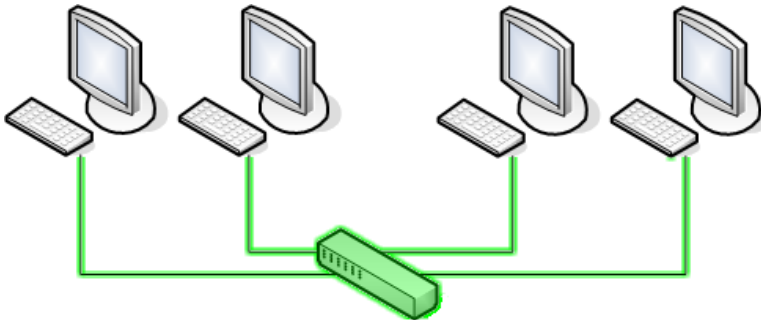


Security Features of CoCo Node 4.5

CoCo Communications is committed to supplying its customers with secure, reliable communication capabilities. CoCo Node, the packet-routing software at the heart of all our products, has been designed and built from the ground up to provide robust protection to the data that travels on a CoCo network. Our products make extensive use of state-of-the-art cryptographic technology and certificate management systems. These security features make it overwhelmingly difficult for potential intruders to overhear or disrupt our customers' mission-critical digital transmissions.

Every packet sent across a CoCo network is secured using two layers of encryption within CoCo Node, and optionally with additional layers of encryption by the underlying network transport and the by the client software application. These layers each shield against different kinds of attacks, and when combined together they present a solid wall of defense versus even highly sophisticated attack vectors.

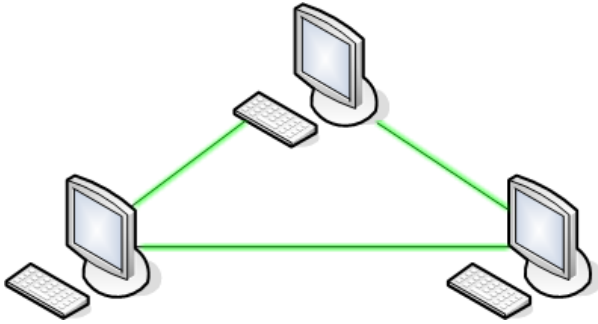


Transport Layer Security

Many digital communication systems have inherent, built-in encryption capabilities. This is particularly common for wireless networks, where connection to the shared medium is simply a matter of being within transmission range, and a potential attacker might gain access to the network simply by being in physical proximity to an access point. When two CoCo Nodes exchange packets across such an underlying medium, each packet gets wrapped in the existing security characteristics of that medium.

One example is WEP, the shared-key encryption system for 802.11, often used for home networks and small offices due to its ease of setup. A CoCo network can include, for example, a few machines whose wireless network interface cards (NICs) are configured to use the same network and WEP key. On each machine, CoCo Node will write Ethernet frames to the network interface. The NIC's drivers and firmware will take that data, encrypt it with the WEP key, and wrap it into the payload of an 802.11 frame before transmitting it out into the air. As such, a potential intruder would have to crack the WEP key before even being able to access the medium.

Nodes on a CoCo network form point-to-point connections, or “links”, with one another over Ethernet directly, or over IP using a VPN-like architecture. Therefore, CoCo Node will be able to take advantage of the protections offered by any data link emulation protocol that encrypts IP packets (such as a VPN tunnel) as well as any Ethernet-compatible transport layer (such as 802.11 or Type 1 Ethernet systems).



Link Layer Security

If attackers do manage to gain access to the medium, they will find that all traffic between any two adjacent nodes is encrypted with AES CFB using a 256-bit key that those two nodes negotiate using Diffie-Hellman key exchange. Every pair of nodes uses an AES key unique to that pair. Attackers might be able to compute the number of nodes on the local network segment, but they would be unable to even know how many nodes are on the CoCo network as a whole, much less send or receive traffic.

Any two nodes on a CoCo network that can exchange packets over the same medium with one another (or across an IP network, if properly configured) will form a “link” with one another. A link refers to a relationship between two nodes that are adjacent to one another – that is, they can send and receive packets between one another directly, without needing to pass those packets through intermediary nodes.

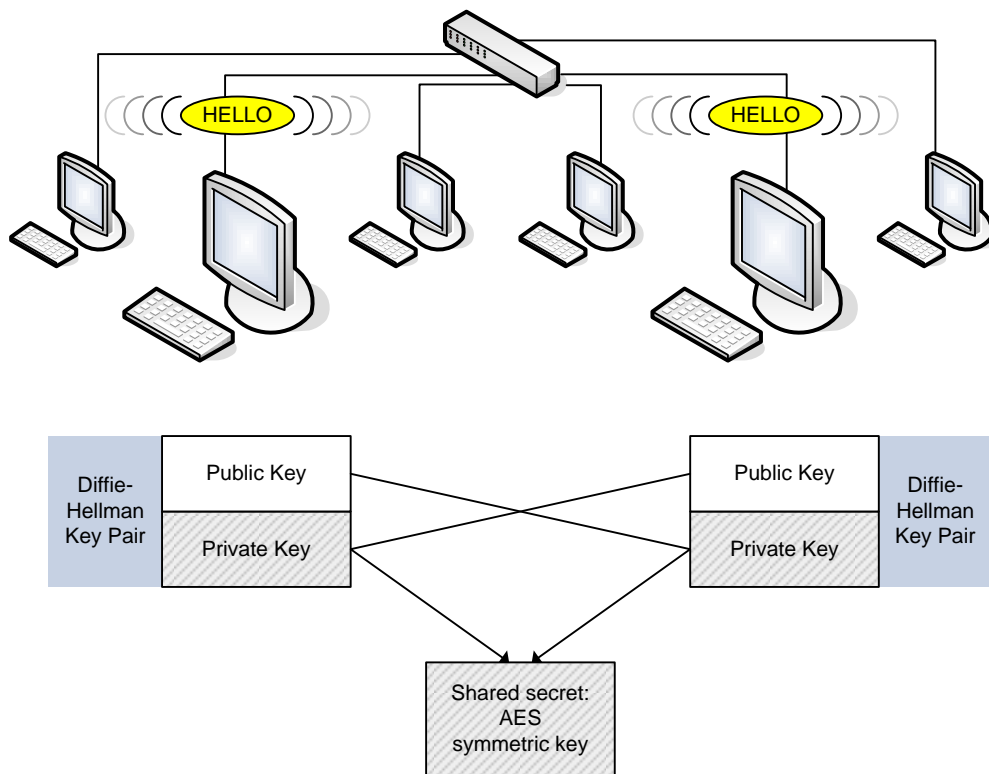
Each link represents a unique cryptographic context. When two nodes form a link, they perform a handshake to securely exchange cryptographic information. All data passed between those nodes is encrypted using this information.

When CoCo Node software is first installed on a machine, the installation process creates a Diffie-Hellman (DH) public/private key pair. This key pair is stored on the machine’s file system (in the registry on Windows-based machines), and is used for all link handshakes.

Every second, CoCo Node broadcasts a plaintext packet announcing its own existence to other nodes on the medium. This packet, called a “HELLO” packet, contains the node’s DH public key. Because it is broadcast in plaintext, all other nodes on the medium (within transmission range, in the case of wireless media) will receive this packet. In this manner, over the course of a second, all nodes on a medium learn one another’s DH public keys.

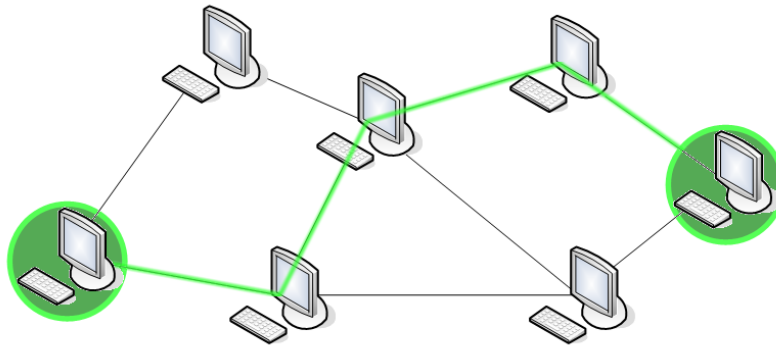
Diffie-Hellman allows any two nodes to securely derive the same shared secret. When two nodes have one another's DH public keys, each one can run its partner's public key and its own DH private key through a mathematical formula to produce a number. Its partner, performing the same operation with the opposite combination of keys, will produce the same number. This number is known only to these two nodes, and an attacker cannot compute it without knowing one of their private keys.

When two nodes hear one another's HELLO packets, each acquires the other's DH public key and computes this shared secret value. Each node then initializes a 256-bit AES cipher in CFB mode, using the shared secret as the AES key. They exchange unicast packets with one another to verify connectivity, and establish the link. From that point onward, all traffic from one node to the other is encrypted with AES.



Every node periodically sends a HELLO broadcast packet, which is received by every other node on the transmission medium. The HELLO packet contains the node's Diffie-Hellman (DH) public key. Whenever one node receives another's HELLO packet, it uses the other's DH public key, along with its own DH private key, to compute a shared secret. When it sends its own HELLO packet, the other node receives it, and performs the same operation with the opposite set of keys, computing the same shared secret. The two nodes use the shared secret as a symmetric key in a 256-bit AES cipher for all subsequent traffic between them.

Figure 1: Key negotiation during link establishment



Circuit Layer Security

More often than not, traffic on a network occurs between nodes that are on different media segments. This means that the two nodes that wish to exchange traffic might not have a direct link to each other, and so the traffic has to pass through a series of intermediary nodes. So even if the traffic is encrypted on a link-by-link basis, the two endpoint nodes would still want to protect their traffic from eavesdrop or insertion attacks by the nodes through which their packets pass.

CoCo offers this protection by integrating a certificate-based public key infrastructure (PKI) model into the network's name/address resolution system, and using these keys to encrypt all traffic between any two nodes engaged in a session-based multi-hop network connection.

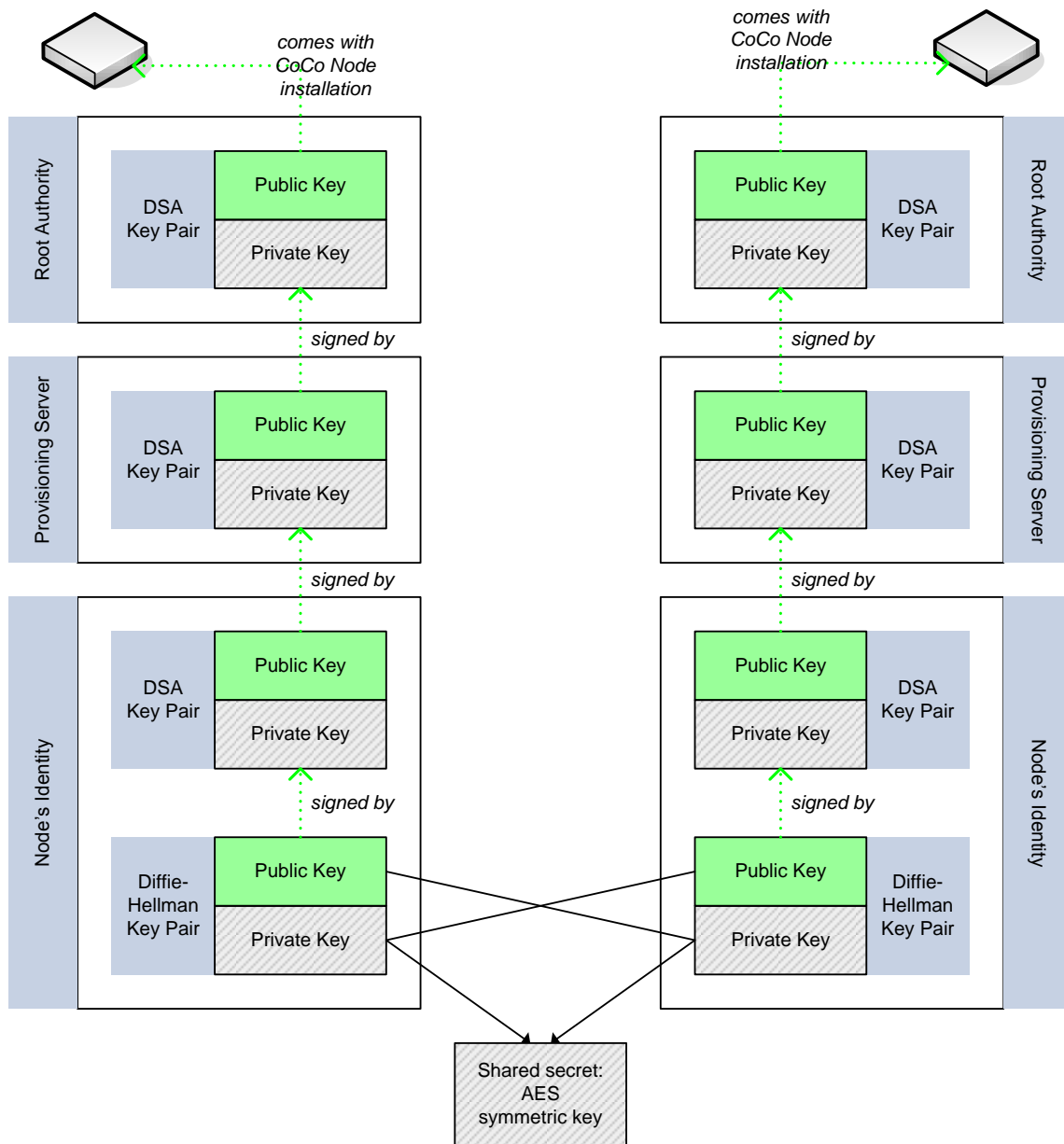
Every CoCo network has one Provisioning Server, a certificate authority with a user interface that allows a human system administrator to permit or deny nodes access to the network. When installed, the Provisioning Server creates a DSA key pair. The Provisioning Server puts its DSA public key into a certificate signature request and submits it to a root certificate authority hosted by CoCo Communications. The root certificate authority itself has a DSA public/private key pair, and its public key comes installed automatically in each copy of CoCo Node. As such, every node can always verify any certificate chain that extends to the root certificate authority.

In order to send or receive traffic on a CoCo network, a node must register with the network's Provisioning Server. A node creates an "identity", a human-readable name that other nodes will use in order to find this node in the network. In conjunction with this identity, it creates two public/private key pairs for use with that identity: one with DH keys and one with DSA. The node signs its DH public key with its DSA private key, and then submits its name and DSA public key to the Provisioning Server in the form of a certificate signature request. The Provisioning Server, pending the administrator's approval, signs the node's certificate request using its own DSA private key. This creates a chain of certificates from the root authority's DSA public key, to the Provisioning Server's DSA public key, to the node's identity's DSA public key, to the node's identity's DH public key.

When one node wishes to send packets to another, it opens a persistent end-to-end connection, or “circuit”, to its intended recipient. In order to do so, the sender must first determine its intended recipient’s location in the network based on its human-readable name. This process is conceptually analogous to performing a DNS lookup on a conventional IP network – in order to open a connection to, say, www.cococorp.com, one must first resolve its IP address (to 69.17.116.124) so that the underlying routing system will know where to send the packets.

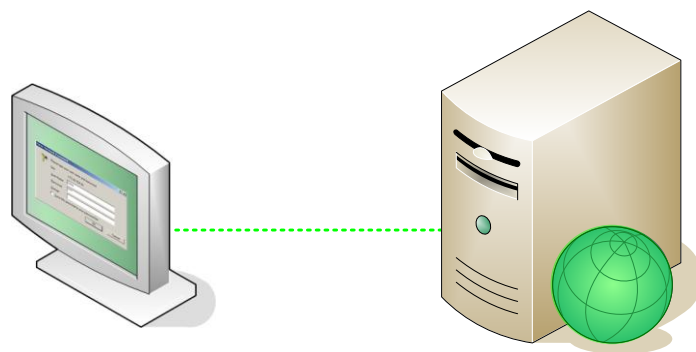
With CoCo, the process of name/address resolution includes transmitting the sender’s certificate chain, as well as the sender’s network location, to the intended recipient. The recipient gets this information before the sender has a chance to learn about the recipient’s network location, so the recipient can verify the sender’s identity long before communication begins. If the sender’s certificate chain is valid all the way up from the DH key to the root key (which the recipient can verify because the root public key comes installed with every copy of CoCo Node), the recipient knows that the sender has permission to open connections on the CoCo network. The recipient then transmits its own certificate chain and network location to the sender. The sender likewise verifies the recipient’s identity by virtue of its certificate chain.

At that point, the sender and recipient both have one another’s DH public keys – and thanks to the valid certificate chains, each node knows for certain that the DH public key of the other node really does belong to the intended partner. Armed with each other’s DH public keys, the two nodes compute a shared secret, and use the shared secret as a symmetric key in a 256-bit AES cipher, just like in the Link Layer. All traffic between these two nodes (or, more pedantically, between the respective *identities* of these two nodes) is encrypted using this AES cipher, and even if the traffic passes through many intermediary nodes, only the two nodes at either end of the circuit are able to read it.



When one node opens an end-to-end connection to another, the nodes exchange certificate chains during the name/address resolution phase. Each node verifies the other's certificate chain, confirming that each certificate in the chain is correctly signed by the public key in the certificate above it. At the top of the chain is the root authority's public key, which comes pre-loaded on every installation of CoCo Node. At the bottom of the chain are the DH public keys of the respective identities of the two nodes forming a connection. With a valid certificate chain, each node can confirm that the other's DH key really does belong to it. Given this guarantee, the nodes use each other's DH public keys to compute a shared secret, which they then use as a symmetric key for a 256-bit AES cipher for all subsequent traffic between these two nodes.

Figure 2: Certificate chain validation for secure circuits



Application Layer Security

Every network application has its own unique security concerns. The Internet offers no network-level security of its own, and years of experience have taught software developers that hackers can easily take down any network application without additional safeguards. Whether they use common protocols such as SSL or custom-built cryptographic handshakes, today's developers take for granted that IP is inherently unsafe and build their applications to remain robust and hacker-proof regardless of the underlying network's lack of security.

CoCo technology honors these application-layer security measures by remaining fully compatible with all IP-based application features. CoCo Node uses IP tunneling technology to appear as a network interface, complete with an IP address binding. When a user installs CoCo Node on a computer, that machine will behave as though it has a new network interface, called a "tunnel device", alongside its existing network cards. Applications will send IP packets to this "imaginary" network interface, just like they would send packets to the computer's Ethernet interface or WiFi card. Instead of traveling out to the network right away, though, these packets written to the tunnel will get processed by CoCo Node and routed in accordance with CoCo's mesh network technology.

Under the hood, the network may be a highly complex, mobile, dynamic structure, but to third-party applications running on machines with CoCo Node, it all simply looks like IP. They can continue to function as normal, using the security mechanisms their developers built in order to remain robust against IP network attacks – while at the same time benefiting from the additional protection offered by CoCo's link-layer and circuit-layer security features.