A Framework for Pervasive Security in Infrastructure and Authority Independent Communications Networks

The mission of the Serval Project is to enable communications anywhere, any time, including in the complete absence of supporting infrastructure or authorities. This paper describes the security framework that is being implemented to support the Serval Project in the pursuit of this goal. In the context of this paper, security includes: (a) confidentiality and (b) authenticity, i.e., ensuring that a given piece of data is received untampered and unread by any third party, and that it was authored by whom it claims.

The remainder of this paper describes various features of the Serval Project's architecture and how they address these dual goals of confidentiality and authenticity in the absence of supporting infrastructure or authorities.

1 Public-Key Cryptography as Basis of Network Addresses

IPv4 and IPv6 networks use addresses that typically bear no intrinsic relationship to the device using any given address. That is, IP addresses are approximately fungible to the identities using them. As a result, IP addresses can be falsely claimed or spoofed without detection, and establishing secure communications between devices on the internet is problematic.

While IPSec provides some solutions to this problem, it is not without difficulty, and moreover does not solve the Birthday Paradox problem that faces IPv4 in global-scale mesh networking, nor the lack of availability of IPv6 on a substantial fraction of mobile telephone handsets, especially at the feature phone end of the spectrum. IPSec is also dependent on signing authorities, which cannot be assumed to be available in some use-cases that the Serval Project seeks to address. Thus IPSec is not a viable solution.

To address these difficulties, the Serval Project has adopted an overlay-network model where an underlying IPv4 network is used to transport datagrams for the Serval Overlay Mesh (SOM). The SOM uses as network addresses 256-bit public keys from the NaCl Curve25519 CryptoBox cryptography system.

CryptoBox is an Elliptic Curve Cryptography (ECC) system using the 2^255-19 curve developed by Dan Bernstein et al, and combined relatively high speed with a design that includes various measures to improve resistance against a variety of side-channel attacks.

Like most cryptography systems, CryptoBox does not have a mathematical proof to vouch for its security strength. Nonetheless, it is estimated that as of 2012 CryptoBox using 256-bit keys offers comparable security to that offered by the RSA cryptography system using 3072 bit keys. CryptoBox, as a member of the ECC-family is based on the discrete logarithm problem, which is believed to be much harder to solve than the number-factorisation problem that is the basis of the RSA cryptography system. There is reason to believe that the security strength of the ECC family will decay much more slowly than that of the RSA family as a result, and also as a result of the general focus of effort on the number-factorisation problem as against the discrete logarithm problem. Nonetheless, as with almost any cryptography system, there remains the risk that the security basis of the CryptoBox system will be invalidated, or some design or implementation flaw may undermine the security of the system at some point in time.

The use of public keys as network addresses immediately solves the problem of establishing secure and authenticated communications between a pair of nodes on the network. First, unidirectional secure communications can be achieved by using the authenticated encryption primitive of the CryptoBox system to send a message that is strongly resistant to unauthorised eavesdropping, modification or forging. Second, bidirectional communications can be established through the execution of the Diffie-Hellman key-exchange primitive of the CryptoBox system, to allow the secure generation of a shared-secret that can be used to seed a symmetric cipher to facilitate the creation of an encrypted session. Such an encrypted session can be protected against tampering by the use of appropriate message authentication codes.

Data overheads are mitigate to below IPv6 levels through the use of dynamic address abbreviation, that uses short prefixes of addresses whenever possible.

2 Authentication and Verification of Peers

Spoofing or fraudulent claiming of network addresses can be detected by use of a challenge-response protocol to verify that the party in question is in possession of the private key that corresponds to their claimed network address, i.e., public key.

Such mechanisms can be built into the higher levels of software to enable the operators of two devices to be confident that their devices have are communicating with one another. For example, the interface on two mobile telephones may display a series of words or an image derived from a shared secret obtained by methods previously described. If the words/image are identical on the two devices, then the operators can have confidence that their devices are communicating. If they are different, then a spoofing or man-in-the-middle attack is revealed.

Such a verification system can be used by parties to determine the correspondence between the public key on a device and its entitlement to claim a given telephone number, or its ownership by a given person. In this way users can through an initial interaction establish the authenticity of the other party. Such trust relationships are represented as a mapping between a telephone number or Direct Inward-Dial (DID) number and a public key/network address in the Serval framework. These public key network addresses are referred to as a Subscriber ID (SID) in the Serval framework. Thus the mapping is between a DID and SID. There may be more than one DID that a party authenticates as belonging to a given SID. Where more than one user/subscriber shares a given device, that device will have a corresponding number of SIDs. For example, an OpenBTS station running Serval Project software would have one SID per associated SIM card.

The presence of verified DID:SID mappings is leveraged during the dialling process. When a user dials a DID, if that DID has exactly one verified SID, then the call will be placed to that SID. In the absence of a verified SID, the Serval Distributed Numbering Architecture transmits an ARP like request asking for SIDs who claim to own the number. If one or more SIDs respond, then the caller will be informed that no verified subscribers could be found, but one or more unverified/insecure end points have been identified. The caller is presented with the opportunity to attempt verification of the remote party.

This verification process is performed by initiating a mesh telephone call to the aspiring recipient of the call. The Diffie-Hellman process is used to determine the shared secret corresponding to the caller and callee SIDs, and this is used to derive a random set of word which are displayed on both caller and callee's handsets. The caller is instructed to ask the callee to read the words back to them. If they match, then man-in-the-middle attack can be ruled out as highly improbable, and the communication channel considered secure for the remainder of the verification process. The caller can then seek to determine if the callee is who they claim to be. This wetware approach is much preferable to attempting to create a digital scheme for verifying identity, as such digital schemes typically require external infrastructure/authorities which may not be available for various Serval Project use cases. If the caller is satisfied that the callee is who they claim to be, they indicate this to the Serval Project software, which then records the DID:SID mapping, and allows future calls to be placed with confidence that the channel will be secure and addressed to the appropriate party.

3 Secured Datagram Transport: Serval Mesh Datagram Protocol (MDP)

Building on the security basis that the Serval Mesh Overlay Network provides, higher-level services such as carriage of voice will be implemented using a UDP-like overlay transport. This transport, the Serval Mesh Datagram Protocol (MDP), is structured around 256 bit public keys as network addresses and 32 bit port numbers for each end point of the socket.

To ease interoperability with conventional IP sockets, client programs obtain an MDP socket by connecting to a conventional loopback socket on the Serval DNA process, and performing a short handshake that informs the Serval DNA process about the MDP endpoints of the connection. Convenient wrapper functions will be provided for C and Java to make this process as simple as possible.

Transparent encryption and authentication of frames will be optional to allow for efficient cross-layer designs, especially for the carriage of voice where using advanced very low bit-rate voice codecs (such as David Rowe's Codec2 which operates at 1,400bps) where the provision of conventional encryption and authentication headers would dwarf the voice traffic. In addition to address abbreviation, MDP streams will be eligible for identification as Pseudo Virtual-Circuits (PVC), where between each pair of nodes in a multi-hop path the MDP stream is identified by an 8-bit PVC identifier. These measures are critical for efficiently channeling voice over low-bit-rate radio channels, where the total channel bandwidth may be less than 100kbits per second.