NTP Security Model

David L. Mills
University of Delaware
http://www.eecis.udel.edu/~mills
mailto:mills@udel.edu

Sir John Tenniel; *Alice's Adventures in Wonderland,*Lewis Carroll

NTP security model

- NTP operates in a mixed, multi-level security environment including symmetric key cryptography, public key cryptography and unsecured.
- NTP timestamps and related data are considered public values and never encrypted.
- Time synchronization is maintained on a master-slave basis where synchronization flows from trusted servers to dependent clients possibly via intermediate servers operating at successively higher stratum levels.
- A client is authentic if it can reliably verify the credentials of at least one server and that server messages have not been modified in transit.
- A client is proventic if by induction each server on at least one path to a trusted server is authentic.

Intruder attack scenarios

- An intruder can intercept and archive packets forever, as well as all the public values ever generated and transmitted over the net.
- An intruder can generate packets faster than the server, network or client can process them, especially if they require expensive cryptographic computations.
- In a wiretap attack the intruder can intercept, modify and replay a packet. However, it cannot permanently prevent onward transmission of the original packet; that is, it cannot break the wire, only tell lies and congest it. It is generally assumed that the modified packet cannot arrive at the victim before the original packet.
- In a middleman or masquerade attack the intruder is positioned between the server and client, so it can intercept, modify and replay a packet and prevent onward transmission of the original packet. It is generally assumed that the middleman does not have the server private keys or identity parameters.

Security requirements

- The running times for public key algorithms are relatively long and highly variable, so that the synchronization function itself must not require their use for every NTP packet.
- In some modes of operation it is not feasible for a server to retain state variables for every client. It is however feasible to regenerated them for a client upon arrival of a packet from that client.
- The lifetime of cryptographic values must be enforced, which requires a reliable system clock. However, the sources that synchronize the system clock must be cryptographically proventicated. This circular interdependence of the timekeeping and proventication functions requires special handling.

Security requirements (continued)

- All proventication functions must involve only public values transmitted over the net with the exception of encrypted signatures and cookies intended only to authenticate the source. Unencrypted private values must never be disclosed beyond the machine on which they were created.
- Public encryption keys and certificates must be retrievable directly from servers without requiring secured channels; however, the fundamental security of identification credentials and public values bound to those credentials must be a function of certificate authorities and/or webs of trust.
- Error checking must be at the enhanced paranoid level, as network terrorists may be able to craft errored packets that consume excessive cycles with needless result.

NTP subnet principles

- The NTP network is a forest of hosts operating as servers and clients
  - Primary (stratum 1) servers are the forest roots.
  - Secondary (stratum > 1) servers join the trunks and branches of the forest.
  - Clients are secondary servers at the leaves of the forest.
  - Secondary servers normally use multiple redundant servers and diverse network paths to the same or next lower stratum level toward the roots.
- An NTP subnet is a subset of the NTP network.
  - Usually, but not necessarily, the subnet is operated by a single management entity over local networks belonging to the entity.
  - The set of lowest-stratum hosts represent the roots of the subnet.
  - The remaining subnet hosts must have at least one path to at least one of the roots.
  - The NTP subnet is self contained if the roots are all primary (stratum 1) servers and derivative if not.
  - Subnets may include branches to other subnets for primary and backup service and to create hierarchical multi-subnet structures.

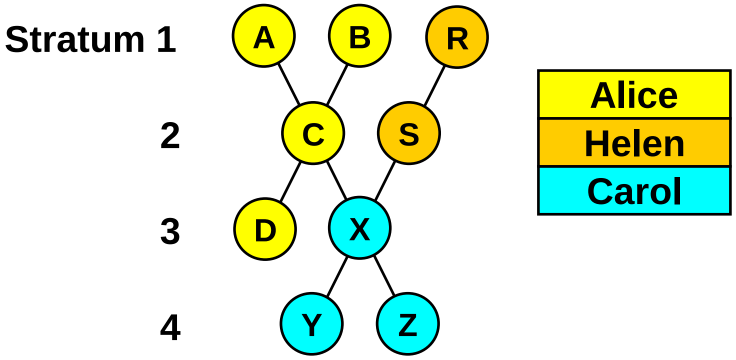
NTP secure group principles

- A NTP secure group is a subnet using a common security model, authentication protocol and identity scheme based on symmetric key or public key cryptography.
- Each group host has
  - Password-encrypted identity parameters and group key generated by a trusted agent.
  - For public key cryptography, a public/private host key pair and self-signed host certificate,
- Each group has one or more trusted hosts that
  - Provide cryptographic redundancy and diversity.
  - Operate at the lowest stratum of the group.
  - For public key cryptography, the host certificate must have a trusted extension field.
- A trusted agent acting for the group generates the current identity parameters and group key, which are distributed by secure means..

Hierarchical groups and trust inheritance

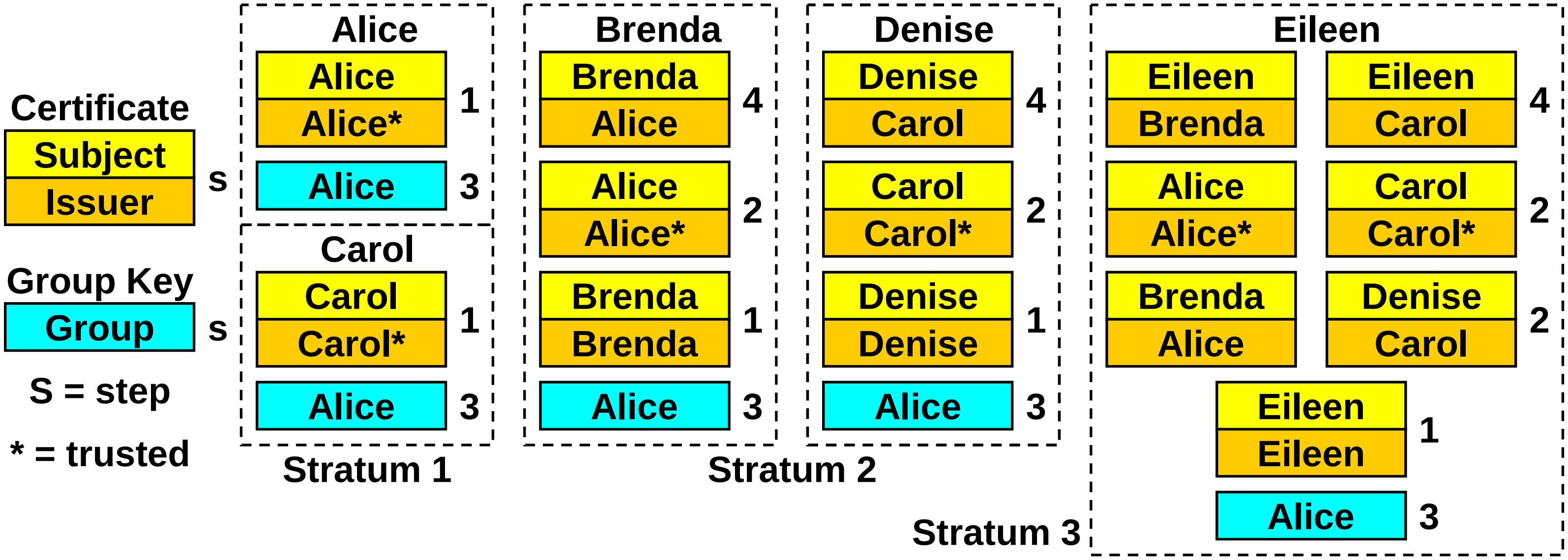
- A host authenticates neighbor hosts by credentials, including certificate, identity parameters, group key and identity scheme.
  - A certificate trail must exist from each host via intervening hosts having the same credentials to (one of) the trusted host(s) at the lowest stratum of the group. The name of each trusted host must be a pseudonym for the group.
  - The security protocol hikes the certificate trail to reveal the pseudonym which locates the credentials previously obtained from the trusted agent.
- This provides the framework for hierarchical group authentication.
  - The primary group includes multiple trusted primary (stratum 1) servers with primary group credentials.
  - A derivative group includes multiple trusted secondary servers at a higher stratum with both primary and secondary group credentials. These servers authenticate the primary group using certificate trails ending at the primary servers.
  - Dependent servers authenticate the derivative group using secondary group credentials and certificate trails ending at the secondary servers.
  - And so on to higher stratum groups.

NTP secure group configuration example



- There are three groups, primary Alice and Helen and derivative Carol.
  - Each member has the credentials for its group generated by a trusted authority. Alice trusts AB, Helen trusts R and Carol trusts X.
  - C authenticates using Alice credentials and either A or B certificate.
  - D authenticates using Alice credentials and certificate trails via C.
  - S authenticates using Helen credentials and R certificate.
  - Y and Z authenticate using Carol credentials and X certificate.
  - X authenticates either with Alice credentials and trails via C and/or Helen credentials and trails via S. Which credentials to use are determined by the security protocol and trusted host at the end of the trail.
  - Each trusted host must have credentials for all next downstratum trusted hosts.

Identity verification - outline



Certificate

Subject

Issuer

s

Group Key

Group

s

S = step

* = trusted

Alice

Alice / Alice* 1

Alice 3

Carol

Carol / Carol* 1

Alice 3

Stratum 1

Brenda

Brenda / Alice 4

Alice / Alice* 2

Brenda / Brenda 1

Alice 3

Denise

Denise / Carol 4

Carol / Carol* 2

Denise / Denise 1

Alice 3

Stratum 2

Eileen

Eileen / Brenda

Eileen / Carol 4

Alice / Alice*

Carol / Carol* 2

Brenda / Alice

Denise / Carol 2

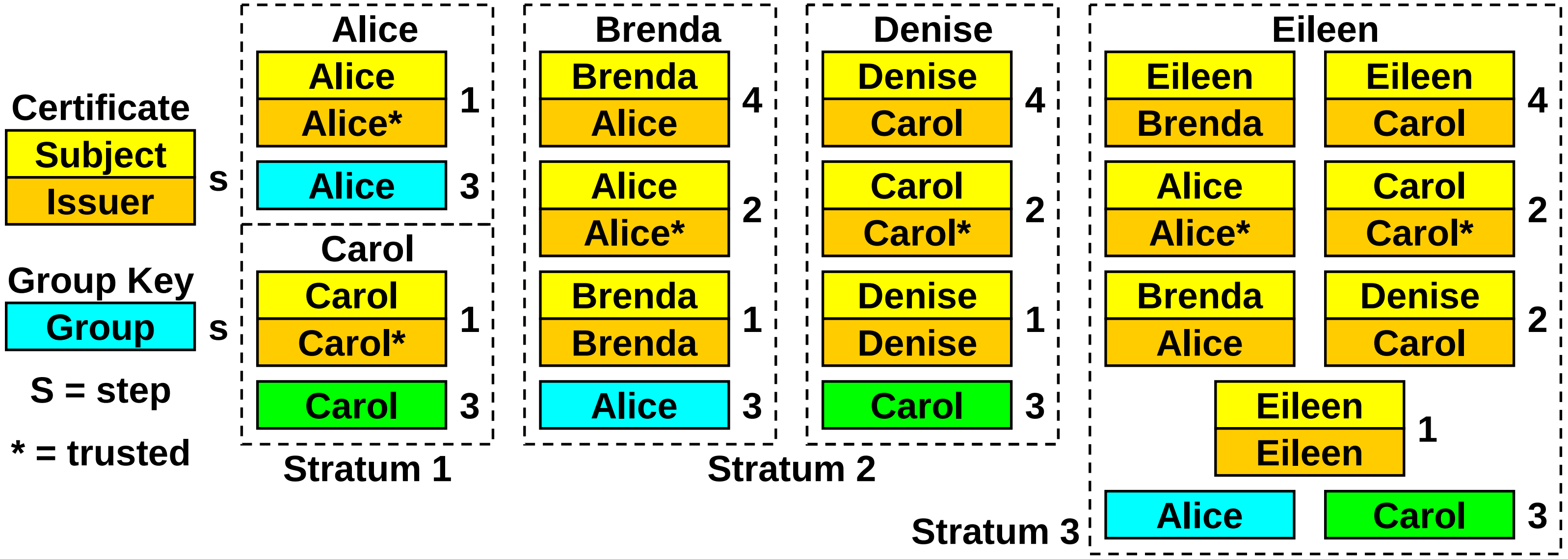
Eileen / Eileen 1

Alice 3

Stratum 3

- Eileen (stratum 3) chimes both Brenda and Denise, Brenda (2) chimes Alice (1) and Denise (2) chimes Carol (1). Alice and Carol have trusted certificates; Alice trusted group keys have been securely deployed.
  - Step 1: Host loads self-signed subject certificate at startup.
  - Step 2: Autokey loads server certificate signed by next lower stratum issuer. The trail continues until a trusted certificate is found.
  - Step 3: Autokey loads group key and verifies server identity.
  - Step 4: Autokey presents self-signed certificate to server for signature.

Multiple groups



Certificate
Subject
Issuer
s

Group Key
Group
s

S = step

* = trusted

Alice
Alice / Alice* 1
Alice 3

Carol
Carol / Carol* 1
Carol 3

Stratum 1

Brenda
Brenda / Alice 4
Alice / Alice* 2
Brenda / Brenda 1
Alice 3

Denise
Denise / Carol 4
Carol / Carol* 2
Denise / Denise 1
Carol 3

Stratum 2

Eileen
Eileen / Brenda — Eileen / Carol 4
Alice / Alice* — Carol / Carol* 2
Brenda / Alice — Denise / Carol 2
Eileen / Eileen 1
Alice — Carol 3

Stratum 3

- Alice and Carol are trusted agents in different groups.
  - Alice group key previously deployed to Brenda and Eileen.
  - Carol group key previously deployed to Denise and Eileen.
  - Eileen hikes trail via Brenda to Alice and verifies identity with Brenda using Alice key.
  - Eileen hikes trail via Denise to Alice and verifies identity with Denise using Carol key.
  - Basic rule: each server must have all group keys for all possible hikes.

Authentication scheme A (Diffie-Hellman)

- Scheme is based on Diffie-Hellman key agreement and conventional symmetric cryptosystem.
  - Certificated public values for server are provided by X.509 infrastructure.
  - Private session keys are distributed out-of-band in advance or derived using certificated Diffie-Hellman agreement (Station-Station protocol)
  - The message digest is computed and verified using the session key
- Advantages
  - Requires no protocol modifications.
  - Conforms to current IPSEC security models (Photuris, etc.).
  - Can be adapted to multicasting in small groups.
- Disadvantages
  - Server requires separate state variables for each client.
  - Does not scale to large subnets with many clients and few servers.
  - Not practical for multicasting in large groups.

Authentication scheme B (Kent)

- Scheme is based on RSA public key signature, Diffie-Hellman key agreement and MD5 one-way hash function.
  - Certificated public values for server are provided by X.509 infrastructure.
  - Server computes session key as MD5 hash of source and destination addresses, key identifier and cookie as hash of private value.
  - On request, server encrypts cookie using provided client public key. Server sends this and RSA signature to client. Client verifies and stores for later.
  - The message digest is computed and verified using the session key.
- Advantages
  - Requires no protocol modifications.
  - Server needs no persistent state variables for clients .
- Disadvantages
  - Not practical for multicasting.

Authentication scheme C (RSA)

- Scheme is based on RSA public key signature
  - Certificated public values are provided by X.509 infrastructure.
  - Server computes MD5 message digest and encrypts with RSA private key. This value is included in the message authentication code (MAC).
  - Clients decrypt MAC and compare with computed message digest.
  - Servers either
    - Estimate encryption delay and compensate timestamp or
    - Include timestamp in following message.
- Advantages
  - Best among all schemes for multicast security with man-in-middle attacks.
  - Requires no client-specific state at server.
- Disadvantages
  - Requires protocol changes; not backwards compatible.
  - Requires significant processing time for each message.
  - Unpredictable running time degrades timestamp accuracy.

Authentication scheme D (S-Key)

- Scheme is based on public key (RSA) encryption and S-Key scheme
  - Certificated public values are provided by X.509 infrastructure.
  - Server generates session key list, where each key is a one-way hash of the previous key, then computes the RSA signature of the final session key
  - Server uses keys in reverse order and generates a new list when the current one is exhausted; clients verify the hash of the current key equals the previous key
  - On request, a server returns the final session key; clients use this if many messages are lost
  - The message digest is computed and verified using the current key
- Advantages
  - Requires few protocol changes; backwards compatible
  - Requires only one additional hash
- Disadvantages
  - Vulnerable to certain man-in-the-middle attacks
  - Lost packets require clients to perform repeated hashes

NTP symmetric key cryptography

- NTP symmetric key cryptography is based on keyed MD5 message digests.
  - A message authentication code (MAC) is computed as the MD5 digest of the message concatenated with the group key.
  - The computed MAC follows the message in the transmitted packet.
  - The receiver computes the MAC in the same way and verifies it matches the MAC in the packet.
- The group key consists of a 32-bit key ID and a 128-bit MD5 key.
  - Each group has a different key distinguished by the key ID included in the MAC.
  - Keys are created by the group trusted host and distributed by secure means.
  - Keys have indefinite lifetime, but can be activated and deactivated by configuration or remotely.

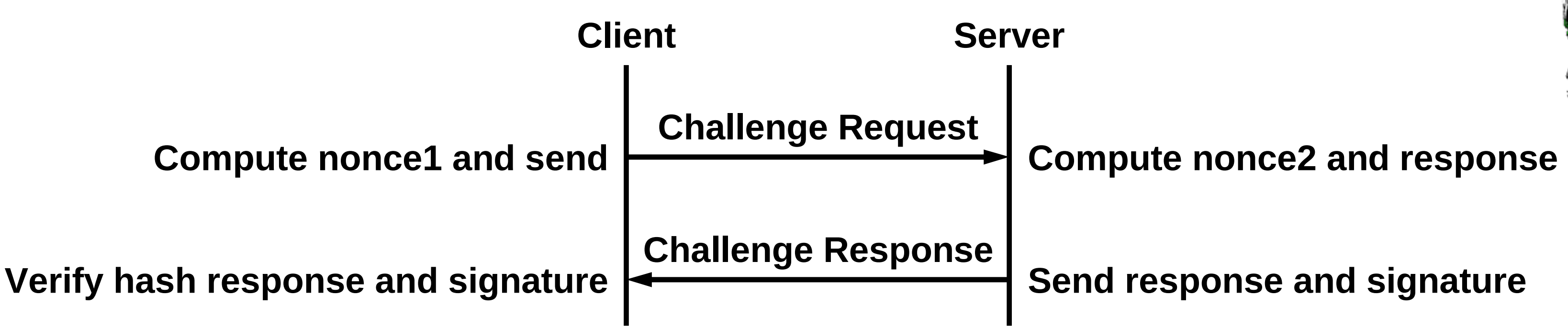
NTP public key cryptography

- NTP and security protocol work independently for each client, with tentative outcomes confirmed only after both succeed.
- Public keys and certificates are obtained and verified relatively infrequently using X.509 certificates and certificate trails.
- Session keys are derived from public keys using fast algorithms.
- Each NTP message is individually authenticated using session key and message digest (keyed MD5).
- A proventic trail is a sequence of NTP servers each synchronized and cryptographically verified to the next lower stratum server and ending on one or more trusted primary time servers.
- Proventic trails are constructed by induction from the primary servers to secondary servers at increasing stratum levels.
- When server time and at least one proventic trail are verified, the peer is admitted to the population used to synchronize the system clock.

NTP Autokey

- NTP public key cryptography is based on the Internet security infrastructure and Public Key Infrastructure (PKI) principles.
  - Each group host generates a RSA or DSA public/private key pair and self-signed X509v3 certificate.
  - The trusted group host certificate is explicitly designated as trusted using a X509v3 extension field.
  - A certificate trail is established dynamically where a client convinces the next lower stratum server to sign its certificate, which is then available to its own dependent clients.
  - A special purpose security protocol called Autokey verifies and instantiates cryptographic values as required.
  - At initialization Autokey recursively obtains certificates until terminating with the trusted certificate which authenticates the path.
- In order to protect against middleman attacks, an optional cryptographic identity scheme can be used.

Identification exchange



Client

Server

Compute nonce1 and send

Challenge Request

Compute nonce2 and response

Challenge Response

Verify hash response and signature

Send response and signature

- This is a challenge-response scheme
  - Client Alice and server Bob share a common set of parameters and a private group key *b*.
  - Alice rolls random nonce *r* and sends to Bob.
  - Bob rolls random nonce *k*, computes a one-way function *f*(*r*, *k*, *b*) and sends to Alice.
  - Alice computes some function *g*(*f*, *b*) to verify that Bob knows *b*.
- The signature prevents message modification and binds the response to Bob’s private key.
- An interceptor can see the challenge and response, but cannot determine *k* or *b* or how to construct a response acceptable to Alice.

Identity schemes

- Private certificate (PC) scheme
  - Trusted agent generates a certificate marked private and transmits it by secure means to all group members. The certificate is never divulged outside the group and never presented for signature.
- Trusted certificate (TC) scheme (default)
  - The certificate trail is validated to a self signed certificate marked trusted. The identity exchange is not used. This scheme is vulnerable to a middleman masquerade.
- Schnorr (IFF) scheme
  - Trusted agent generates the IFF parameters and transmits them by secure means to all group members. The IFF identity exchange is used to verify group credentials.
- Guillou-Quisquater (GQ) scheme
  - Trusted agent generates the GQ parameters and transmits them by secure means to all group members. Each member generates a GQ private/public key pair and certificates with the public key in an extension field. The GQ identity exchange is used to verify group credentials.

Identity schemes (continued)

- Mu-Varadharajan (MV) scheme
  - This scheme is intended for servers with untrusted dependent clients and where the ultimate trust rests with a trusted agent. The trusted agent generates parameters and private encryption keys for the server group and private decryption keys for the client group. The MV identity exchange is used to verify server credentials.

Future plans

- Deploy, test and evaluate NTP Version 4 daemon in testbeds, then at friendly sites in the US, Europe and Asia
- Revise the NTP formal specification and launch on standards track
- Participate in deployment strategies with NIST, USNO, others
- Prosecute standards agendae in IETF, ANSI, ITU, POSIX
- Develop scenarios for other applications such as web caching, DNS servers and other multicast services

Further information

- Network Time Protocol (NTP): http://www.ntp.org/
  - Current NTP Version 3 and 4 software and documentation
  - FAQ and links to other sources and interesting places
- David L. Mills: http://www.eecis.udel.edu/~mills
  - Papers, reports and memoranda in PostScript and PDF formats
  - Briefings in HTML, PostScript, PowerPoint and PDF formats
  - Collaboration resources hardware, software and documentation
  - Songs, photo galleries and after-dinner speech scripts
- FTP server ftp.udel.edu (`pub/ntp` directory)
  - Current NTP Version 3 and 4 software and documentation repository
  - Collaboration resources repository
- Related project descriptions and briefings
  - See “Current Research Project Descriptions and Briefings” at http://www.eecis.udel.edu/~mills/status.htm