# Plan for Autokey Update

Dr. D. Sibold – PTB

IETF 84, Vancouver, Canada, July 29 – August 3, 2012

# Motivation

- **The current autokey specification has security issues as been presented at IETF 83 in Paris**
- **A more secure specification is needed, especially for cases where compliance requirements have to be fulfilled.**
- **As a consequence of IETF 83:**

  A project team has been setup

  - to develop a design paper for a new autokey specification.
  - The design paper shall be presented as I-D at the next IETF
  - Goal: the specification should be moved to RFC standard track
  - Coordinated effort between NTP developer and IETF community
  - IETF security group should be engaged
  - Implementation is intended as soon as the scope of the work is understood

# Requirements

**The new autokey specification shall provide:**

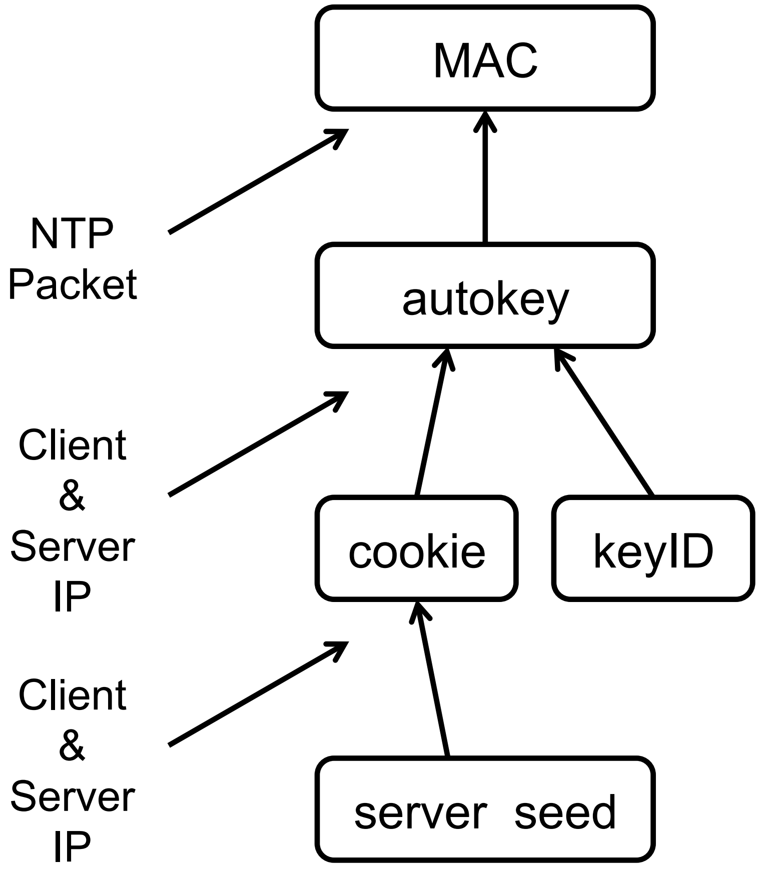
- **Authentication of the communication partners**
- **Integrity protection of the communication protocol**
- **Minimal impact on synchronization performance**
  - Therefore: no external security approach
  - Implementation at the application layer
- **Flexibility in the choice of cryptographic functions (Hash, …)**
- **Use of X.509 PKI infrastructure for authenticity verification**

# Current and new autokey specification

**Major differences between current and new autokey specification**

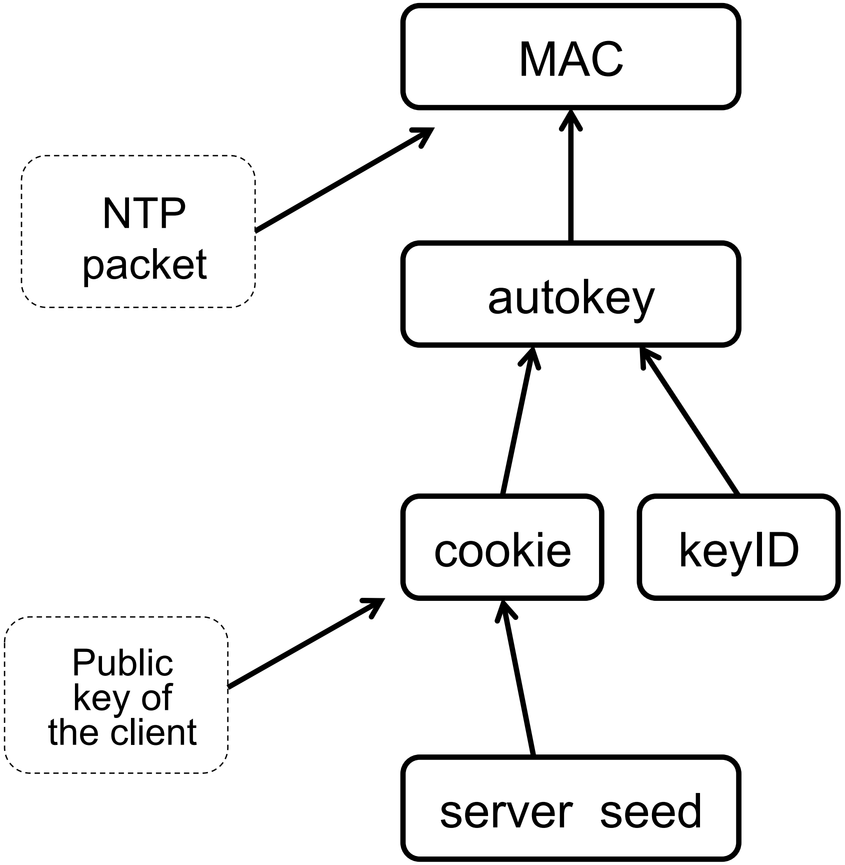
1. **Integrity protection of communication packets with Message Authentication Code (MAC)**
   - Short review of the vulnerabilities of the current autokey specification
   - Procedures to mitigate these vulnerabilities
2. **Verification of authenticity**
   - Shortcomings of autokey's identity schemes
   - Short discussion of hierarchical public key infrastructure

# MAC Calculation (current autokey)



1. Server seed is only 32 bits long
   → Client can request a cookie and brute force the seed
2. The cookie is only 32 bits long; it is the only secret in the generation of the autokey (in Client-Server Mode)
   → An adversary can capture a packet and brute force the cookie
3. Client Identity Check: authenticity verification of the client is based on the client's IP address
   → An adversary can masquerade as the client and obtain the client's cookie encrypted with his own public key.

# MAC Calculation (“new” autokey)



1. Server seed and cookie are 128 bits long.

2. The client’s public key is used for the calculation of the cookie.
   - **Note**: The server needs to recalculate the cookie at each sync request. Therefore the client has to attach its public key at each NTP packet!
   - Alternative: usage of a hash of the public key instead of the public key itself.

# Verification of authenticity

- **In the current autokey specification the verification of the authenticity of the server is done by means of challenge response schemes.**
- **These identity schemes are vulnerable against “man-in-the-middle” attacks.**
  - An adversary in able to send a faked response to a client challenge which the client will accept.
  - all identity schemes are affected
- **They shall be replaced by a hierarchical public key infrastructure based on X.509 certificates.**

# PKI Infrastructure

## Pros:

- **Widely accepted standard for authentication**
- **(Presumably) easy to implement**
- **Helpful in use cases with compliance requirements**

## Cons:

**In the beginning of the synchronization the client cannot verify the validity of the certificates**

**Feasible procedures:**

- TA’s certificate is trusted by default
- Certificates are checked against revocation lists (OCSP, (RFC 6277))
- Crosscheck with third party instance. E.g., utilization of TSP to get an initial certified time stamp from a TSA.

# Open Questions & Summary

## Open Questions

- Concept of proventication and how to implement it?
- Are alternatives to certificates useful: e.g. pre shared keys and Kerberos (like in TLS)?

## Summary

- A new autokey specification shall be formulated (NTP development team and IETF community)
- A first version of a new I-D is available (draft-ietf-ntp-autokey-v2-00)