JOHANNES KEPLER
UNIVERSITY LINZ | JKU

Timestamps

Michael Sonntag

Institute for Information processing and microprocessor technology (FIM)

Johannes Kepler University Linz, Austria

sonntag@fim.uni-linz.ac.at

Why timestamping?

- Validity of electronic signatures:
  - Revocation of certificates: Everything signed before is valid, so we need to know exactly, when the certificate was revoked (or that it was not at a certain point of time)
  - Time a document was signed: To prevent backdating
- The time some fact was established (=some data existed)
  - IPR: Inventions, copyrighted works, … were completed/existed
  - Measurements: They were made then and not before/later
- Note: Timestamp is not necessarily the time of the event, just when it was "notarized" that the event occurred before!
  - Often irrelevant, but an important distinction
- Not event the owner of the "document" can change it later on

How to create a timestamp: Principle

- Create a digital representation of the data you want to timestamp
- Calculate a hash value from this representation
- Send the hash value to a third party
- The third party does:
  - Add the current time to the data
  - Sign the whole package (+ optionally archive it/some parts)
  - Send it back
- The recipient checks, that the timestamp is valid
- Storing it for later references
  - Potentially: Re-signing/re-timestamping later on

Third party receives only the hash, NOT the data itself!

When is a timestamp valid?

- Result:
  - Data existed before the timestamping took place
  - Someone was in possession of the original data at that time
    - Based on impossibility of creating a document matching an arbitrary hash value
      - Note: Broken hash algorithms → This does not apply any more (but the frist one is then false as well!)
- Validation procedure:
  - Calculate hash of original data and append the time stated in the timestamp
  - Validate the digital signature against this data (mathematics, certificate)
  - The timestamp must be within the validity period of the certificate
  - If the certificate was revoked, this must have been after the timestamp

What if the CA key is compromised?

- CA key = Private key of the third party signing the hash values (=timestamper)
- Then you are out of luck!
  - The timestamp may become invalid, as the revocation must be dated back to the earliest point in time the disclosure of the private key may have taken place
  - There is no possibility to “save” the timestamps between “earliest disclosure date” and “date the disclosure was detected”
- Possible help: Use several timestamps simultaneously

Important specifications

- RFC 3161 (from 08/2001)
  - Specification for format and protocol between client and timestamping server
  - Uses ASN.1 notation
  - Time always in UTC
    - Second precision obligatory; more allowed
      - Representation of upper limit of deviation optional; maximum 999 seconds
  - Specifies four transports: E-Mail, File, HTTP and Socket (=TCP; port 318; uses polling to check for the response)

Important specifications

- ETSI TS 101 861 (v1.3.1 from 01/2006)
  - European standardization institution
  - Defines what a timestamping server/client must support
    - Hash algorithms: SHA1, MD5, RIPEMD-160
    - Signatures: RSA+SHA1 (1024 Bit must, 2048 Bit should); nothing else specified
    - One online and one store&forward protocol must be supported; HTTP should be
  - Otherwise references RFC 3161

Time sources

- "Primary" sources are typically very good
  - GPS satellites: Accuracy to ≈ 100 ns
  - Atomic clocks: E.g. Austrian official time has a deviation smaller than $5*10^{-14}$ sec.
  - Radio stations (DCF-77): Precision ≈ 1-2 ms (radio itself: 100 µs)
- They are much too good for timestamping!
  - Transportation time to the service is much bigger than their resolution
  - Normally there is no need for timestamps with more than second precision
    - Often even less; the important part is, that it was an unrelated third person providing the stamp, not which absolute exact time it was
  - Actual source for timestamping: Good NTP synchronization
    - Typical precision: 1-20 ms

Authenticode timestamps

- The signtool can communicate with timestamping services for creating timestamps for signed code
- Method:
  - HTTP 1.1 POST message (=HTTP timestamping request)
  - Body: Base64-encoded timestamping request
    - The request itself is a DER-encoded ASN.1 structure
  - Response: Similar
    - Format: PKCS#7 signed message
  - Content info from the response is copied as a countersignature into the code signature
    - Including the certificate chain of the timestamp
- Attention: This differs from RFC 3161!
  - Same methods used, but different request/response structures!

Conclusions

- Its not that complicated, but you can't do it yourself
  - The basic idea is, that you need a third party!
- Timestamping services are uncommon and mostly require payment
  - Free (and no registration required) service available for Authenticode
- Practical use seems to be scarce
  - There is no need for a timestamp **now** …
  - Every digital signature should have a timestamp: Validation is much better/easier

Thank you for your attention!

Michael Sonntag

Institute for Information processing and microprocessor technology (FIM)

Johannes Kepler University Linz, Austria

sonntag@fim.uni-linz.ac.at

Literature

- Microsoft: Time Stamping Authenticode Signatures

  http://msdn.microsoft.com/en-us/library/bb931395%28v=vs.85%29.aspx