



4th International Workshop on Storage Security and Survivability
StorageSS'08

Improving Secure Long-Term Archival of Digitally Signed Documents

Carmela Troncoso Danny De Cock
Bart Preneel

KULeuven COSIC/ESAT (Belgium)
31st October Washington

Why do we need long term archival?

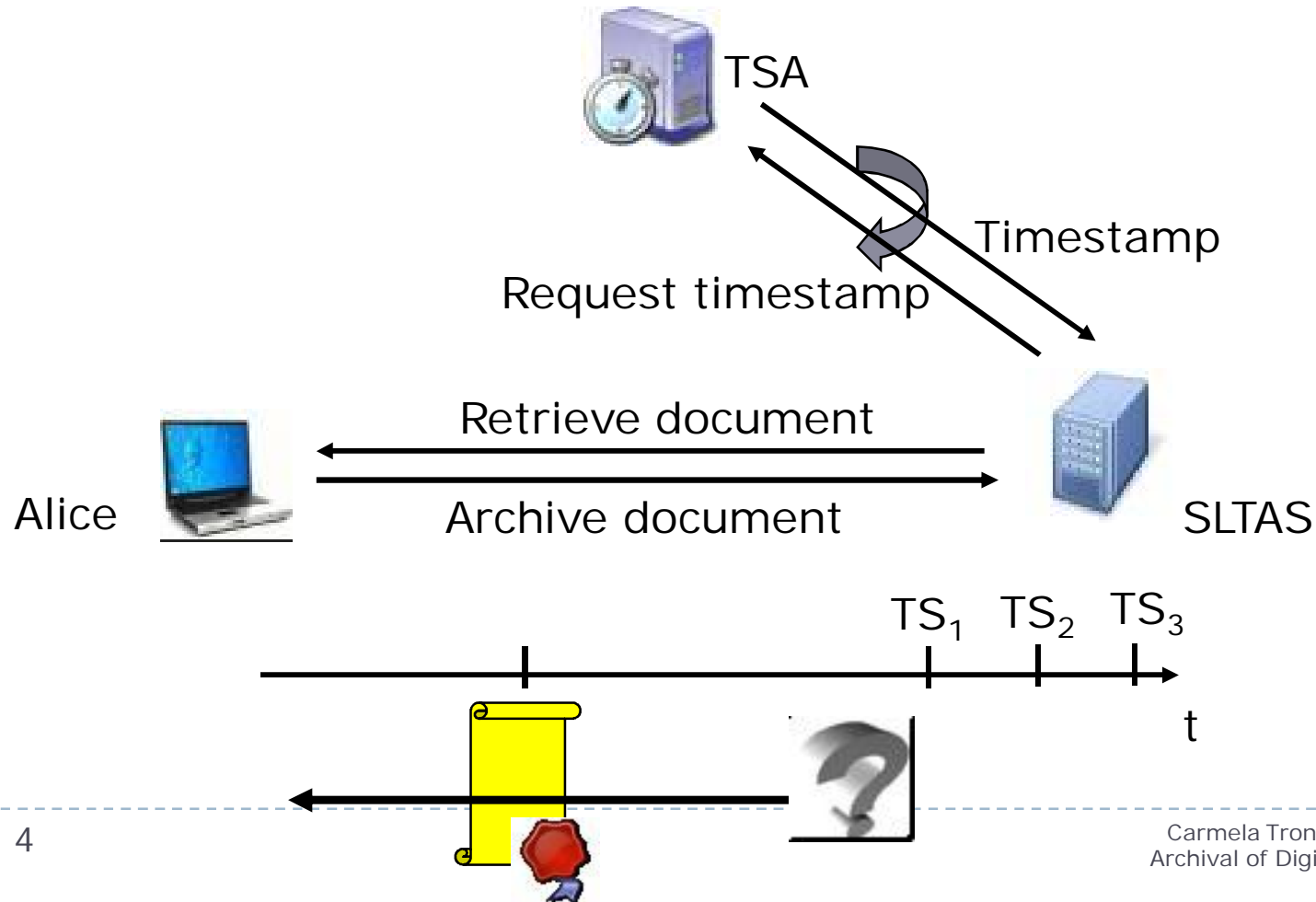
- ▶ **Digital archives:** documents and archives move to the electronic world
- ▶ **Secure Digital archives:** documents and archives move to the electronic world
 - ▶ Confidentiality
 - ▶ Integrity
 - ▶ Availability
- ▶ But... what happens in the long term?
 - ▶ Security properties degrade
 - ▶ Crypto primitives obsolescence: computing power and cryptanalysis
 - ▶ Invalidation of certificates,...

Secure Long-Term Archival System

- ▶ Focuses on **preserving integrity** and proof the **validity of signatures**
- ▶ Given a signed archived document, an SLTAS must be able to prove:
 - ▶ The signature was valid at the time of creation
 - ▶ The signing time (indisputable way)
 - ▶ The content has **not** changed
- ▶ even if...
 - ▶ the cryptography of its digital signature becomes obsolete
 - ▶ the certificates are not longer available

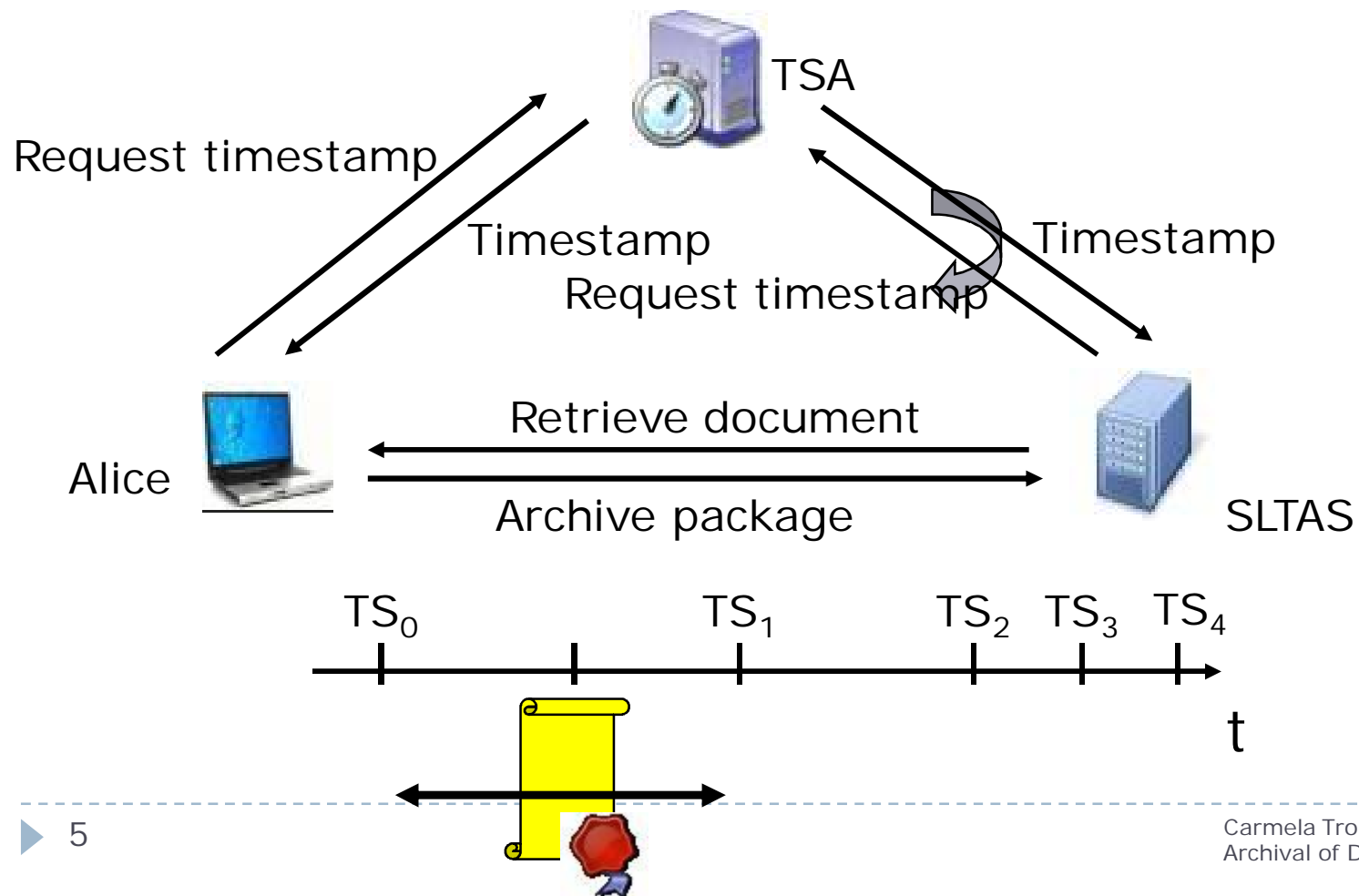
Architecture & Components (I)

- ▶ Based on refreshing the validity of the signatures using timestamps (Time Stamping Authority)

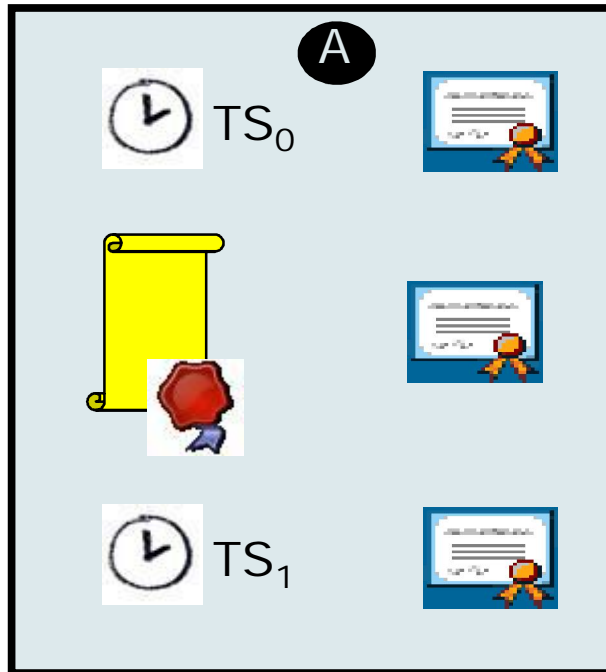
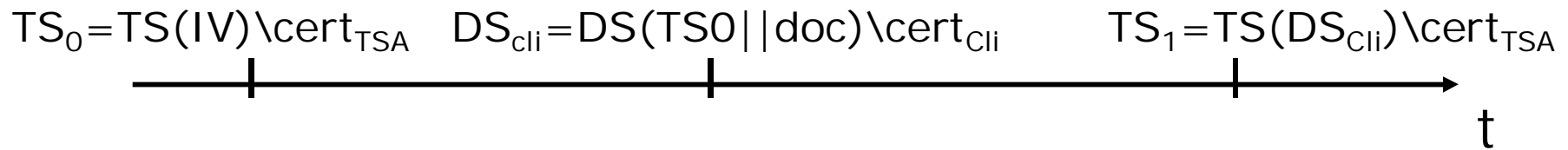


Architecture & Components (II)

- ▶ **Solution:** Timestamping in the client side!



Client Side

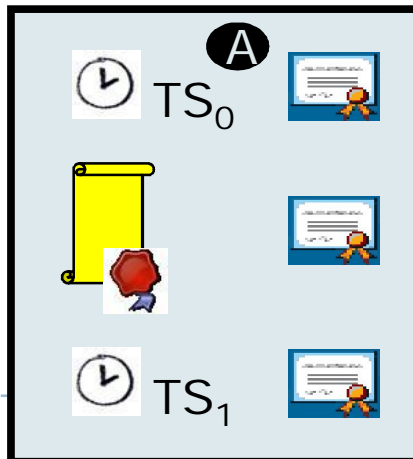
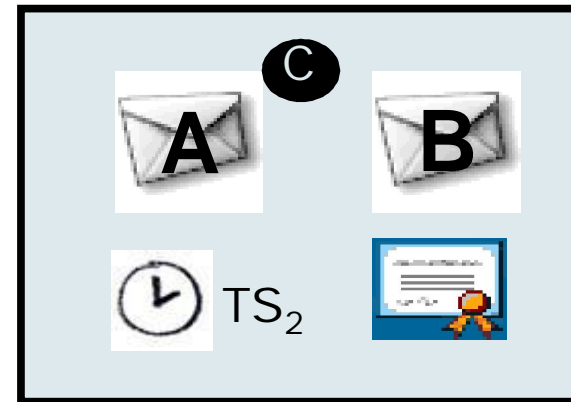
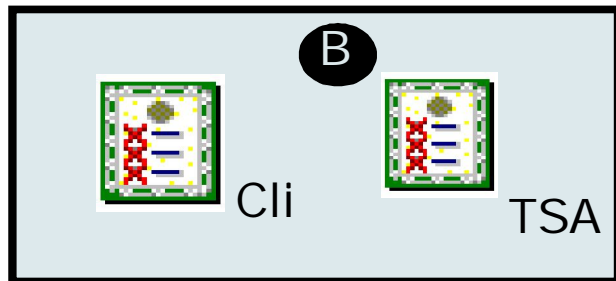


Server Side

Check
signatures
&
Collect
evidence

$$TS_2 = TS(A || B) \setminus cert_{TSA}$$

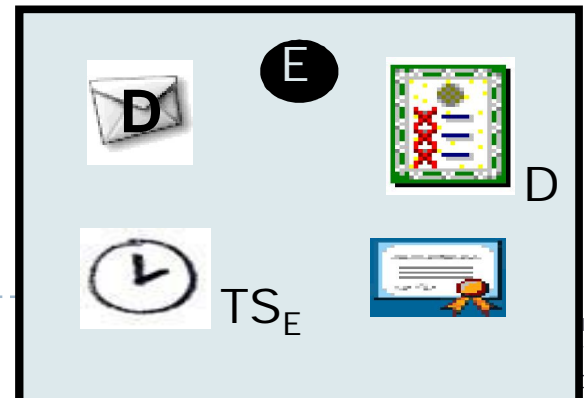
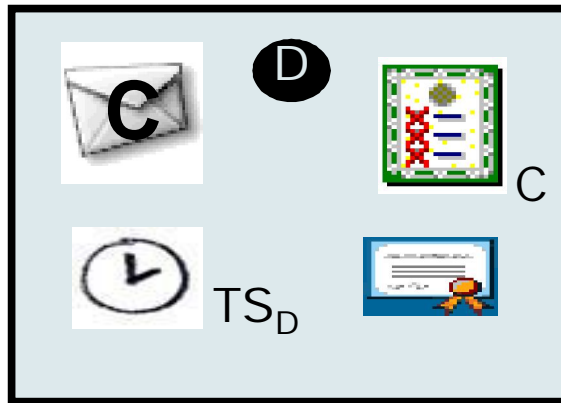
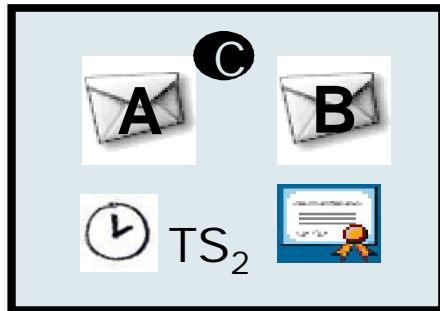
t



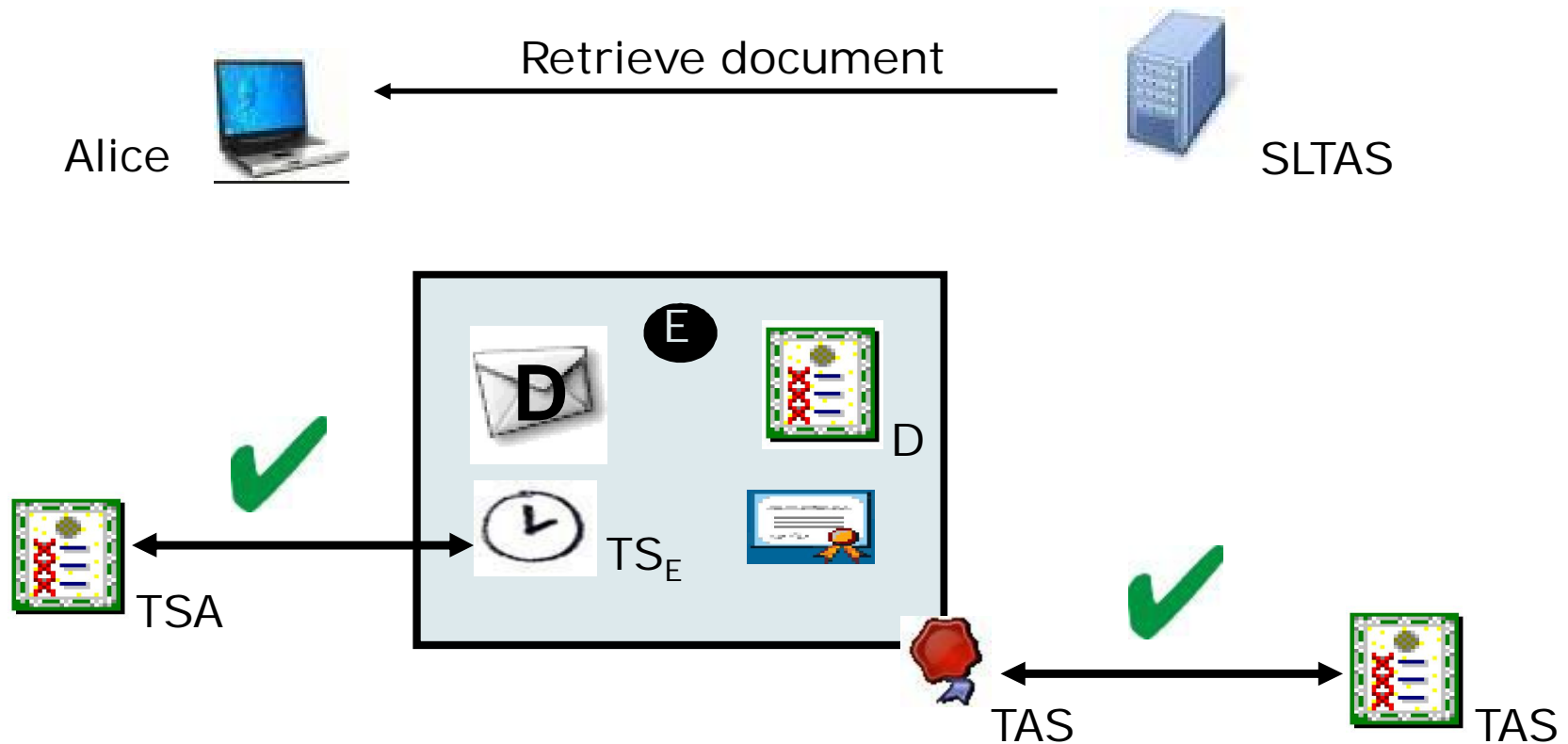
Re-timestamping

$$TS_D = TS(C || CRL_C) \setminus cert_{TSA}$$

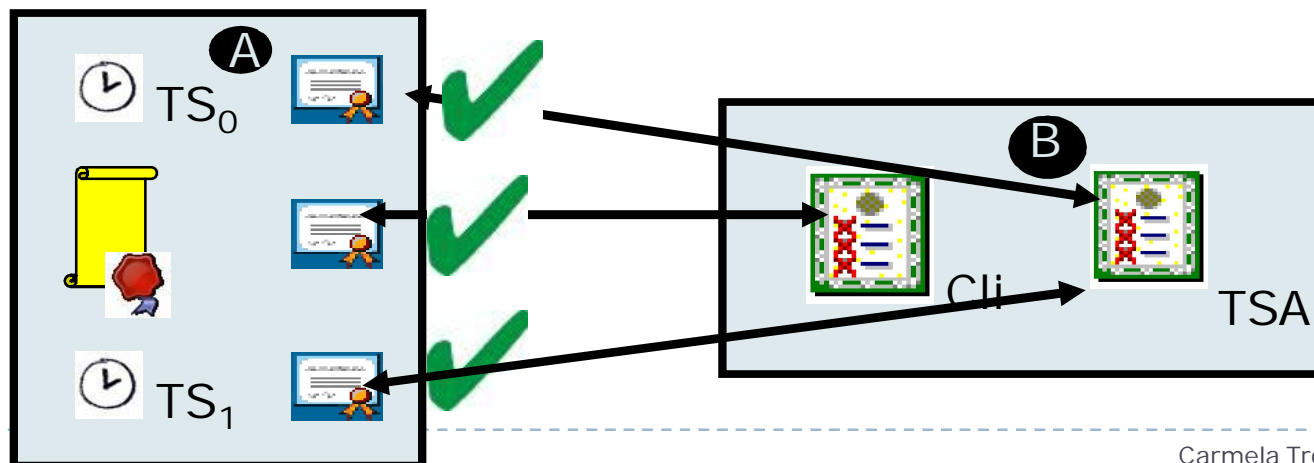
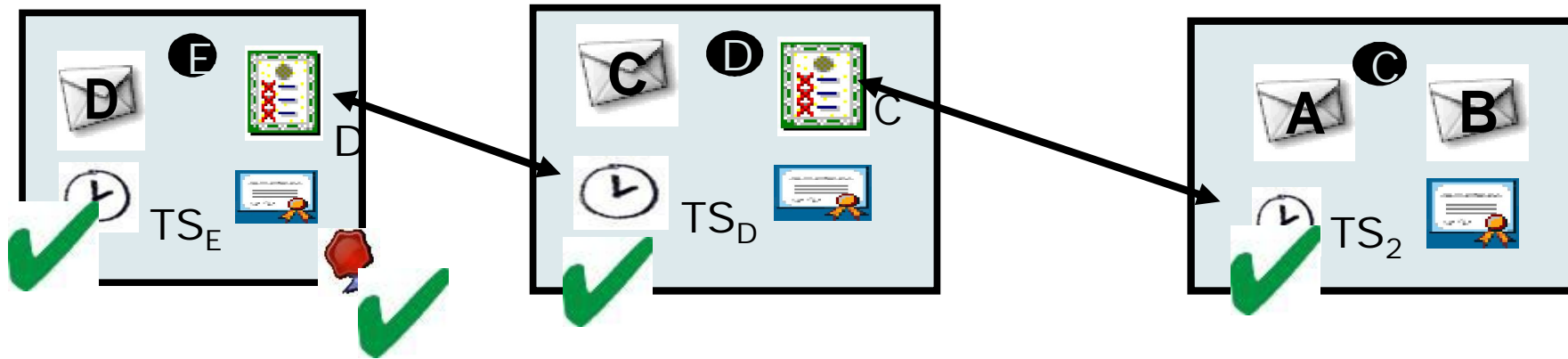
$$TS_E = TS(D || CRL_D) \setminus cert_{TSA}$$



Retrieval - One-signature validation



Retrieval - Complete validation



Discussion

- ▶ Cryptographic algorithms take time to break
- ▶ No modification operation (against goal)
 - ▶ Store modification (Haber and Kamat, 2006)
 - ▶ Could solve migration
- ▶ Potential “infinite” storage space needed
 - ▶ Kryder’s Law: ~Moore’s law disk storage cost (half cost per year)
 - ▶ Not that large...

Discussion

- ▶ Confidentiality
 - ▶ SLTAS will be able to read after long time
 - ▶ What is the SLTAS archiving?
 - ▶ Key management?

- ▶ Availability
 - ▶ Replication and backups
 - ▶ PASIS, SafeStore,...

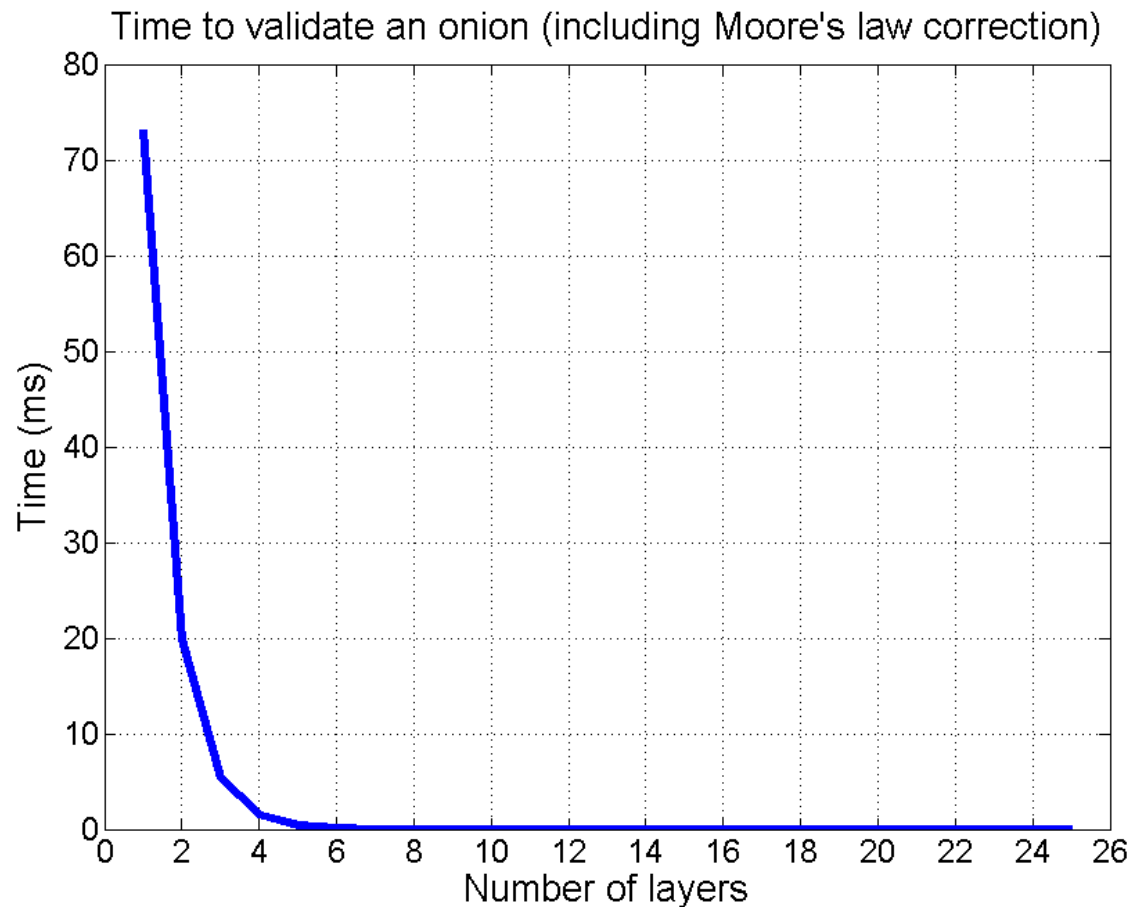
- ▶ Retrieval token storage and integrity preservation

Evaluation

- ▶ Java non optimized implementation
 - ▶ RSASSA-PSS signatures (Client: 1024 bits, SLTAS and TSA: 2048 bits)
 - ▶ X.509 certificates
- ▶ 1st step: archiving a document
 - ▶ Client create the first packet (A): overhead 9.7Kb, <1s
 - ▶ SLTAS reception and verification: <350ms
- ▶ 2^o step: retimestamp
 - ▶ each 3 years
 - ▶ +256bits per retimestamping iteration
 - ▶ <700ms (Moore's Law)

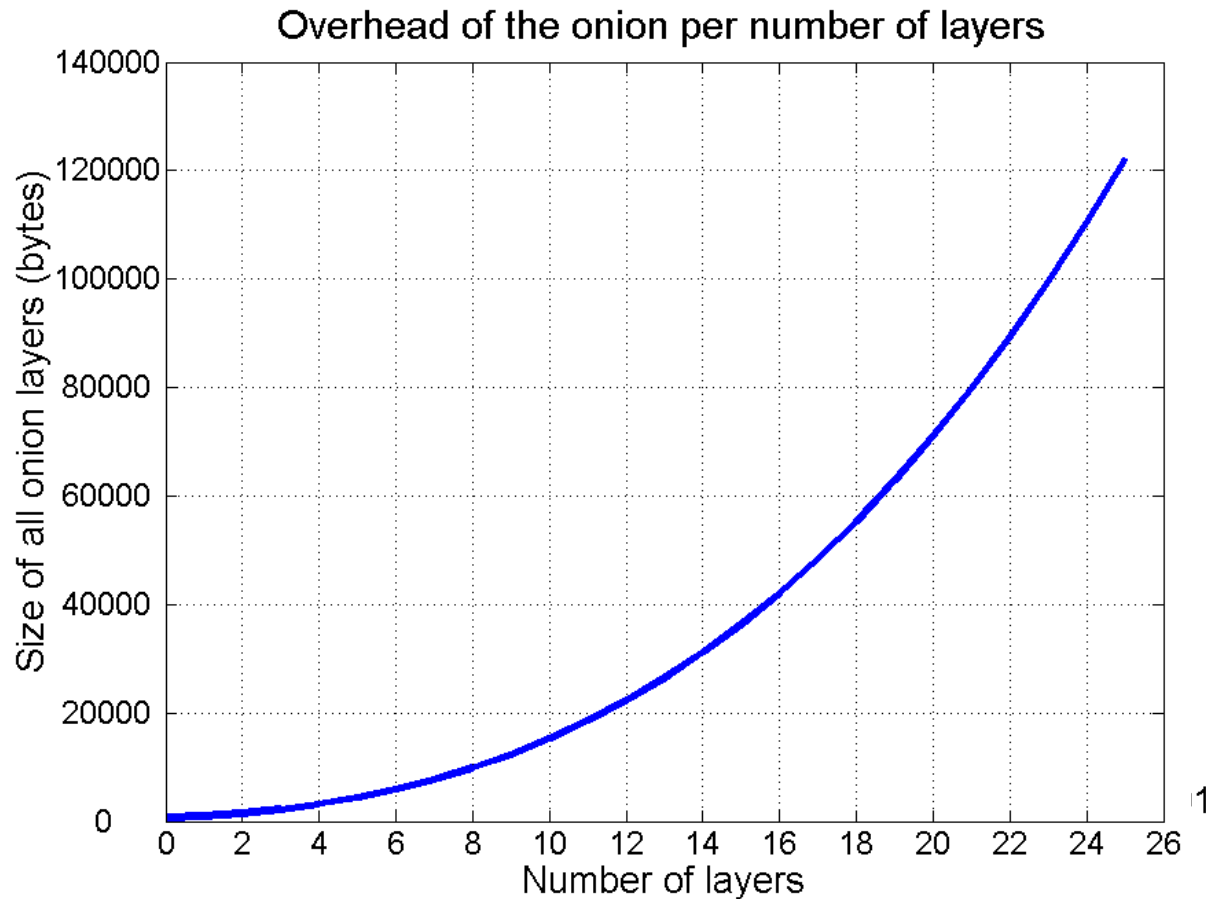
Evaluation

- ▶ 3° step: retrieving a document (full-validation)



Evaluation

- ▶ Storage overhead (all but document)



Conclusion

- ▶ Step forward in the design of SLTAS:
 - ▶ Integrity over time
 - ▶ Validity of signatures
 - ▶ Even if certificates revoked \emptyset unavailable
 - ▶ **Bounded** time of signing
- ▶ Space and time efficient
- ▶ Future work
 - ▶ Privacy friendly?
 - ▶ Other schemes: e.g., data checking



COSIC

Car mela.Troncoso@esat.kuleuven.be



**Thanks for your attention!!
Questions?**