Message-locked Encryption with Deduplication Consistency

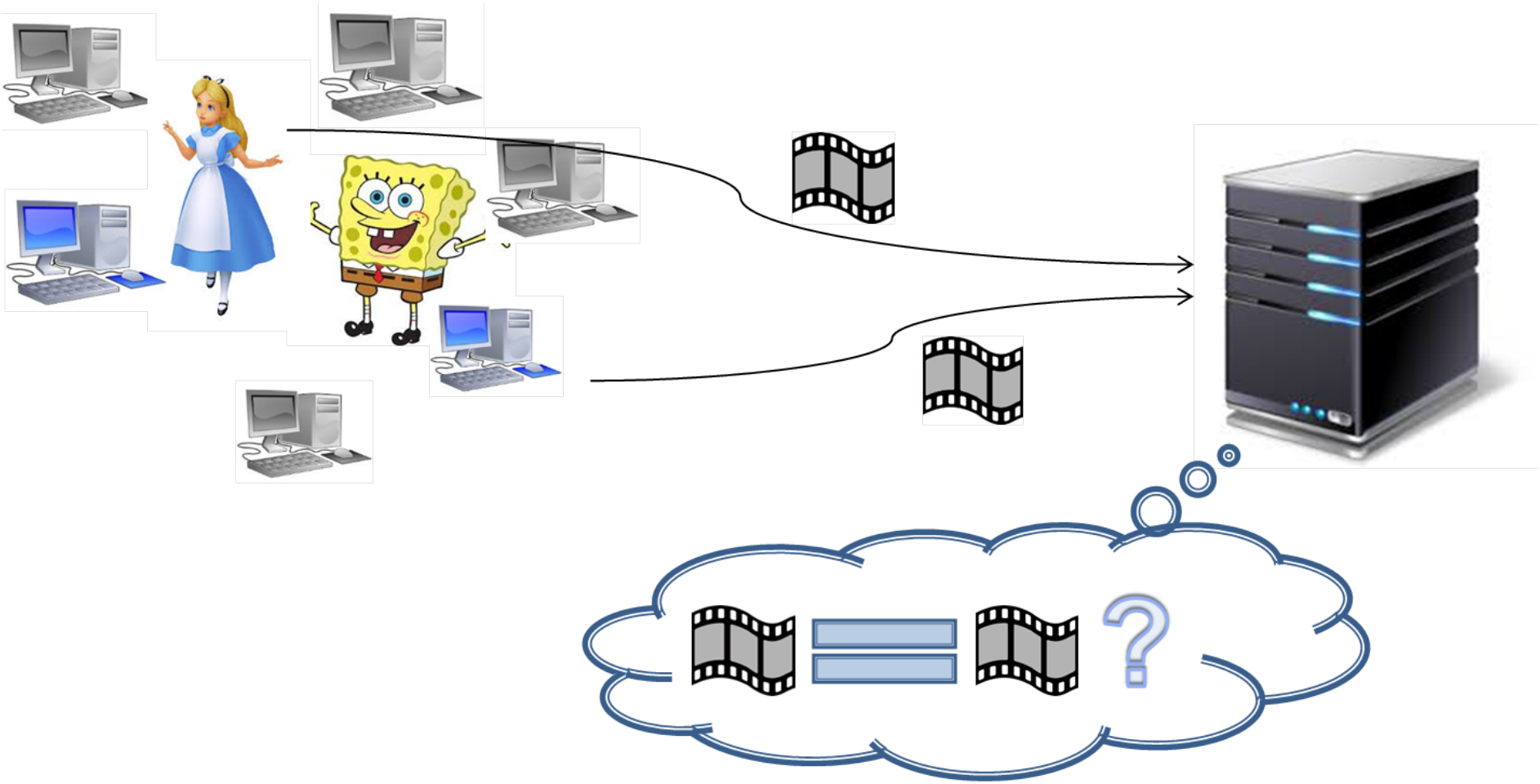
Sébastien Canard [1], Fabien Laguillaumie [2] and **Marie Paindavoine** [1,2]

[1]Orange Labs, Applied Crypto Group, France.

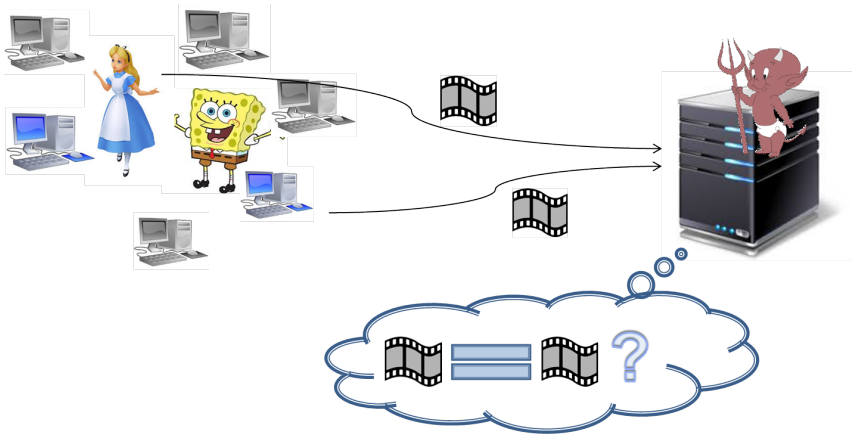
[2]Université Claude Bernard Lyon 1, LIP (CNRS/ENSL/INRIA/UCBL), France.

SEC2, Lorient, July 5.

Deduplication : Saving Space Storage.



The Secure Deduplication Problem

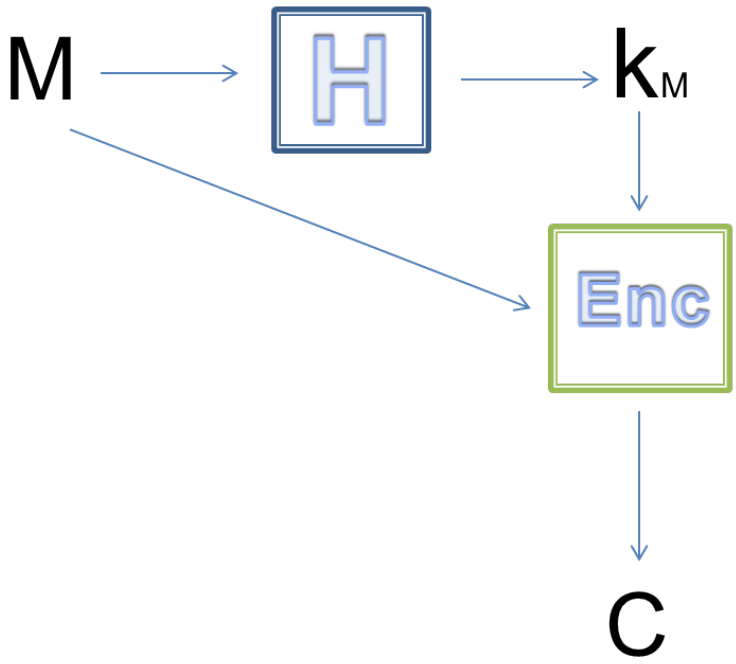


- What if the cloud server is distrusted?
- Alice and Bob could use encryption
- How can the server perform deduplication?

Two main challenges

- The server should be able to check that two ciphertexts encrypt the same message.
- Bob should be able to decrypt Alice's ciphertexts.

The Convergent Encryption Solution [DABS02]



- H is a deterministic hash function.
- Enc is a deterministic encryption scheme.
- Two encryptions of M (even by different persons) yield the same C.
- Server can test if two ciphertexts are equal.

The MLE Model [BKR13,ABMRS13]

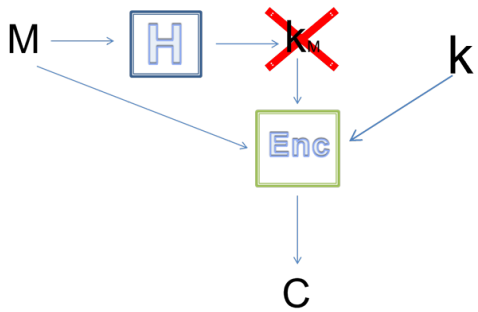
- Formalization and generalization of convergent encryption.
- Definition of a formal security model.
- Give a solution in a non-deterministic setting.
- It suffices to have a tag and an equality test procedure.

Main Security Requirements

- Privacy for unpredictable data only.
- Tag-consistency.
  - $T_1 = T_2$ implies that underlying messages are equal.
- Privacy holds when messages are correlated?
- Privacy holds when messages are dependent from public parameters?
- Construction fulfilling all of those requirements are (very) inefficient [ABMRS13,BK15]

Deduplication Consistency?

- New security requirement.
- If the messages are equal, then the equality test on ciphertexts returns 1.
- Not achieved in convergent encryption and (most of) MLE.



- Adds verifiability to MLE.
- Useful for the right-to-be-forgotten.

Our Scheme

KeyGen: algebraic hash function.
M is divided into (small) blocks M_i

$$k_M = \prod a_i^{M_i}$$

Enc ElGamal encryptions of each M_i

$$T_{1,i} = g_i^{r_i}, \quad T_{2,i} = h^{M_i} g_i^{k_M r_i}.$$

Tags : $(t_1^u, t_2^{k_M u})$.

Use of a bilinear map e for the equality testing.
With 2 tags : $(t_1^{u_1}, t_2^{k_{M_1} u_1})$.
and $(t_1^{u_2}, t_2^{k_{M_2} u_2})$.
Test if

$$e(t_1^{u_1}, t_2^{k_{M_2} u_2}) = e(t_1^{u_2}, t_2^{k_{M_1} u_1}).$$

Ensuring Deduplication Consistency

KeyGen: algebraic hash function.
M is divided into (small) blocks M_i

$$k_M = \prod a_i^{M_i}$$

Enc ElGamal encryptions of each M_i

$$T_{1,i} = g_i^{r_i}, \quad T_{2,i} = h^{M_i} g_i^{k_M r_i}.$$

Tags : $(t_1^u, t_2^{k_M u})$.

Goal: proving all those values were consistently computed.
We use zero-knowledge proofs.
The user can prove every value is consistently derived from the secret message M without revealing it.
Algebraic hash function ensures that the efficiency is linear in the size of the message.

Conclusion and Perspectives

- A probabilistic scheme with new security features.
- (Sort of) efficient.
- Can we have all security features and still be efficient?
- "Fuzzy" deduplication?

Thank you!
Any question?