draft-urien-**hip-tag-02.txt**

HIP support for RFID

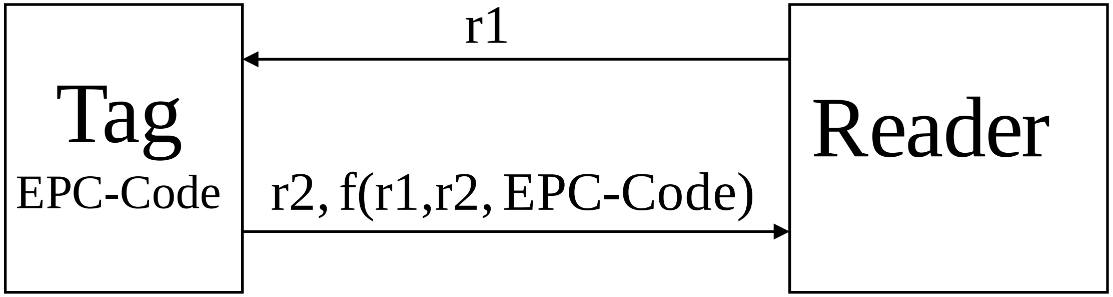
Pascal.Urien@telecom-paristech.fr

Summary

- This document describes an architecture based on the Host Identity Protocol (HIP) for active tags, i.e. RFIDs that include tamper resistant computing resources.
- HIP-Tags never expose their identity in clear text, but hide this value (typically an EPC-Code) by a particular equation (f) that can be only solved by a dedicated entity, referred as the portal.
- HIP exchanges occurred between HIP-Tags and PORTALs; they are shuttled by IP packets, through the Internet cloud.

Identity Protection

- **Privacy issues**
  - EPC-Code **MUST** be protected
  - EPC-Code is a solution of f(r1,r2,EPC-Code)



Tag

EPC-Code

r1

r2, f(r1,r2, EPC-Code)

Reader

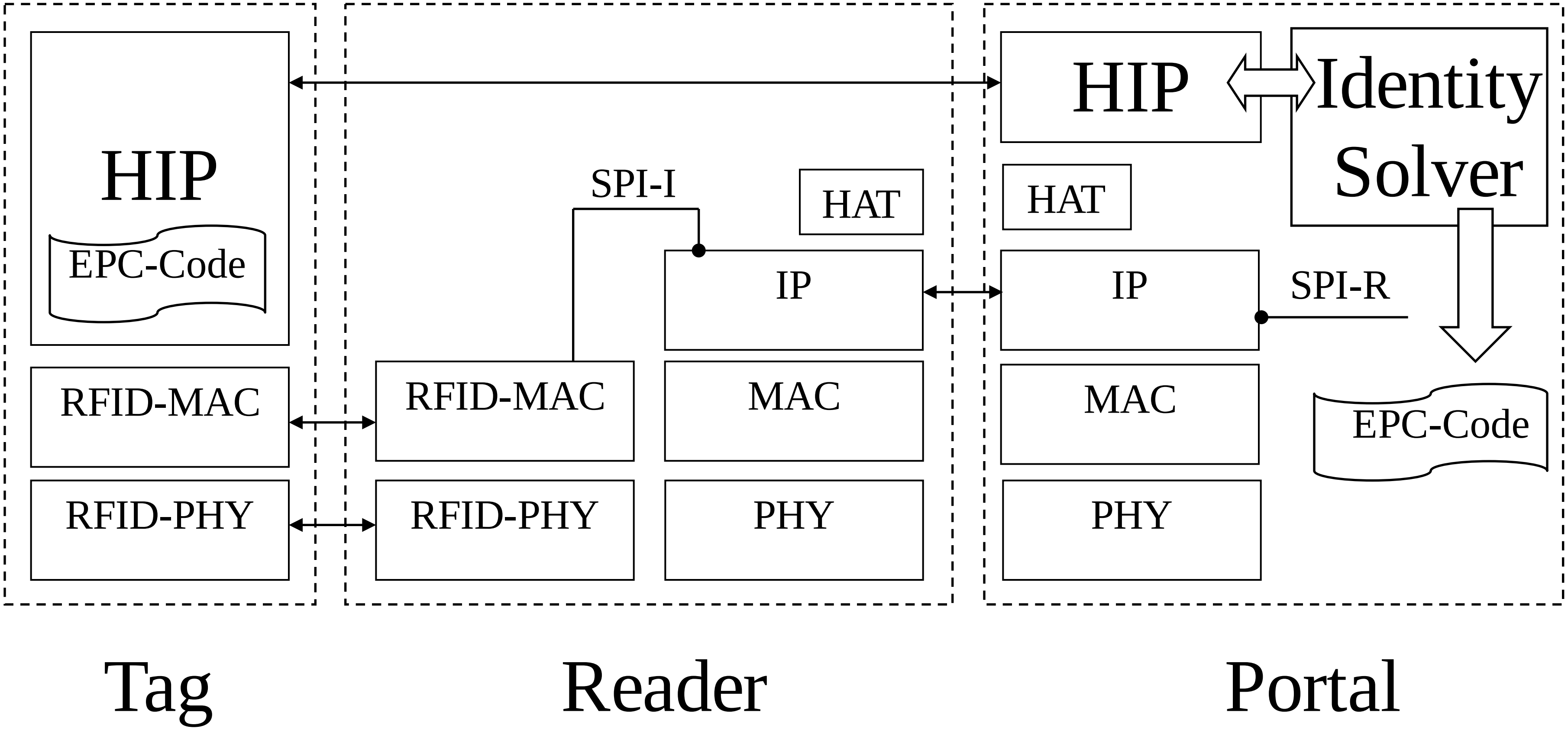
- **Example**
  - Many f proposal in the scientific literature
  - f(r1,r2, EPC-Code) = SHA1 (r1 | r2 | EPC-Code)

S. Weis, S. Sarma, R. Rivest and D. Engels. "Security and privacy aspects of low-cost radio frequency identification systems." In D. Hutter, G. Muller, W. Stephan and M. Ullman, editors, International Conference on Security in Pervasive Computing - SPC 2003, volume 2802 of Lecture Notes in computer Science, pages 454- 469. Springer-Verlag, 2003.

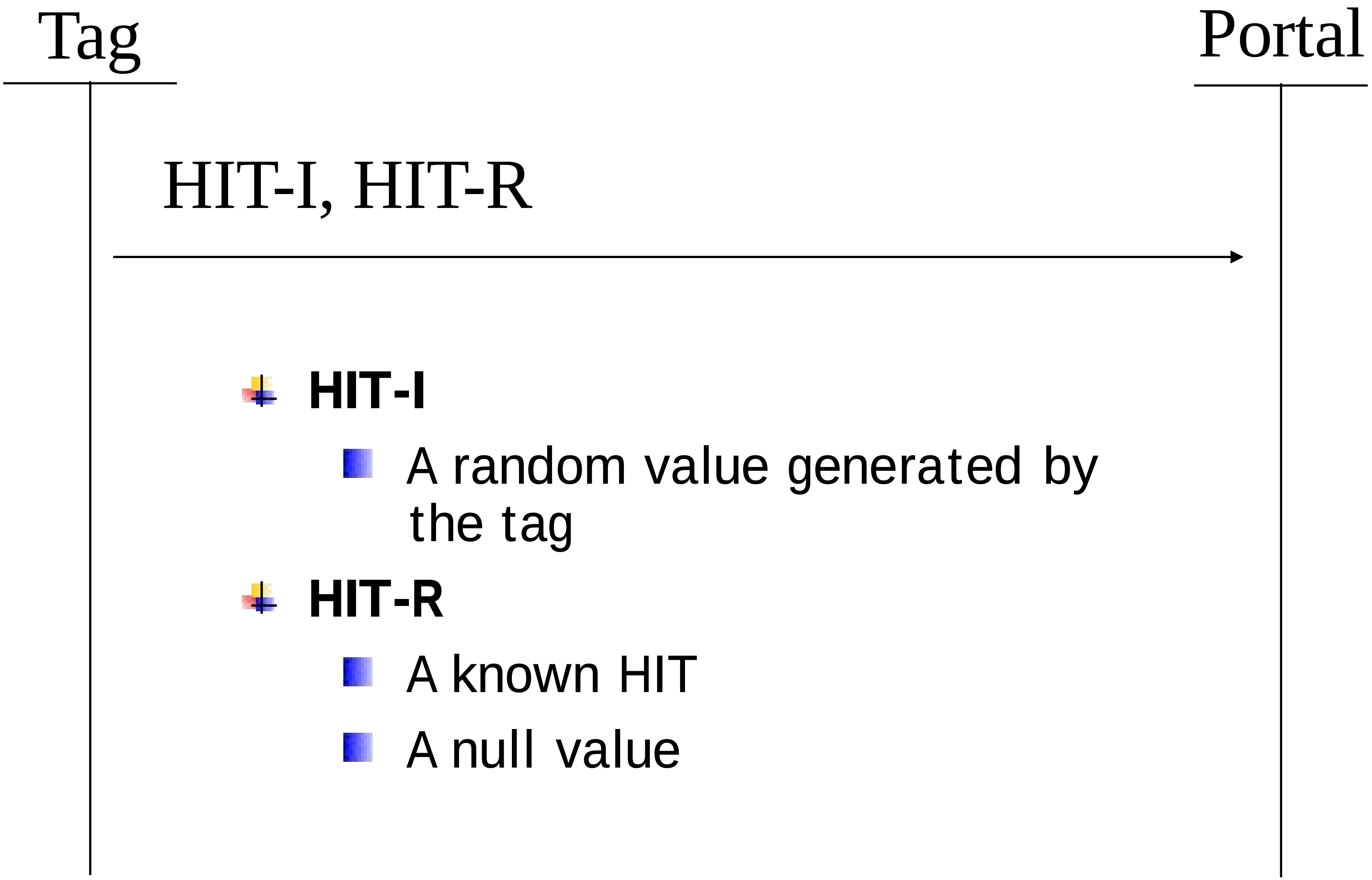
Main Ideas

- **The TAG runs a modified version of HIP**
  - HIP Only! – NO IP stack
  - HIT is a true 16 bytes random number generated by the TAG
- **The Reader is an IP node**
  - It acts as a docking host for HIP tag
- **The Reader is not able to solve the f equation**
  - The *identity solver* entity is located in a node called the PORTAL
- **HIP dialog between Tag and Portal**
  - HIP packets **MAY** be encapsulated by a HAT (*HIP Address Translation*) layer.

HIP-Tags Architecture

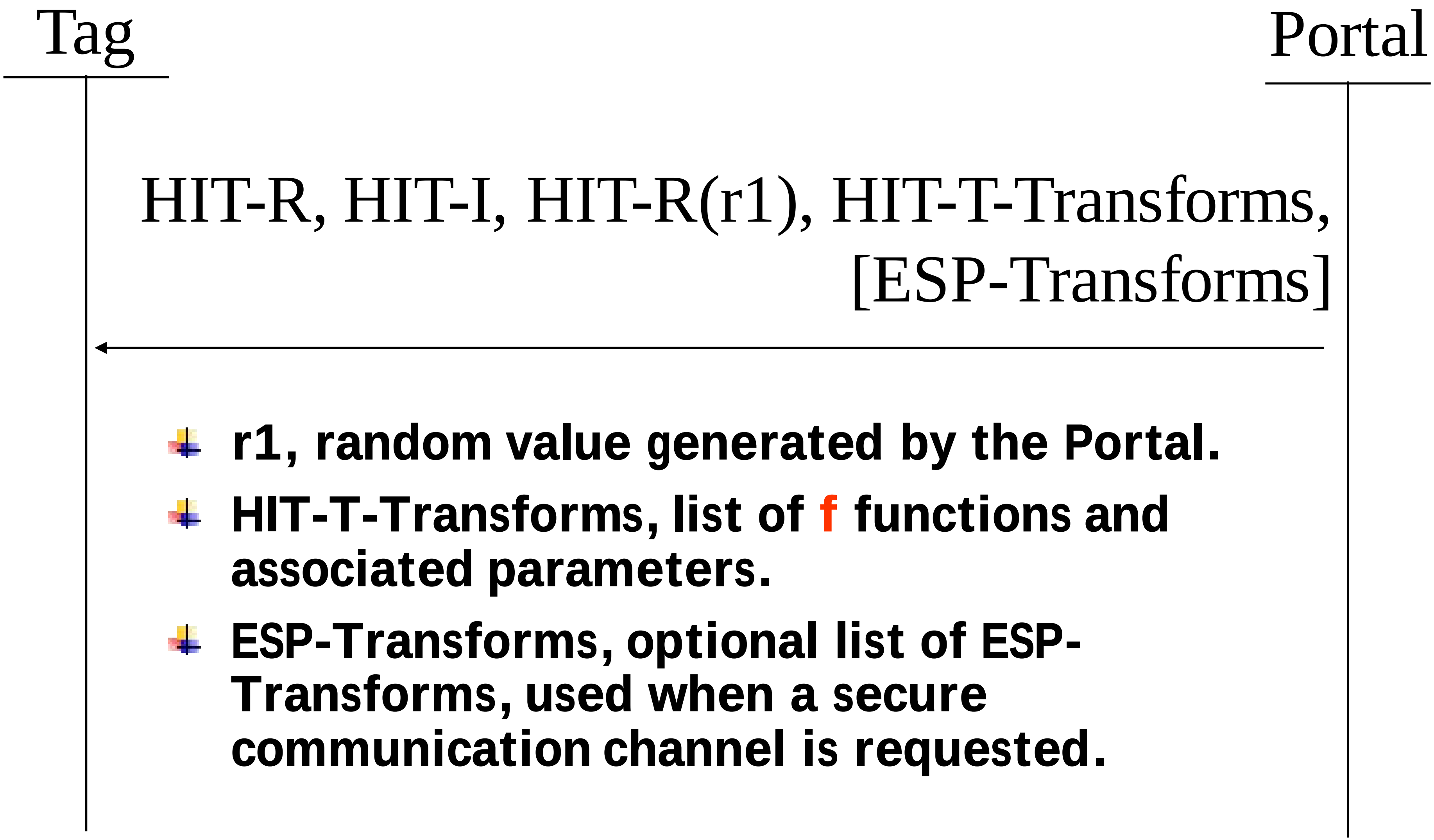


T-BEX Exchange, I1-T



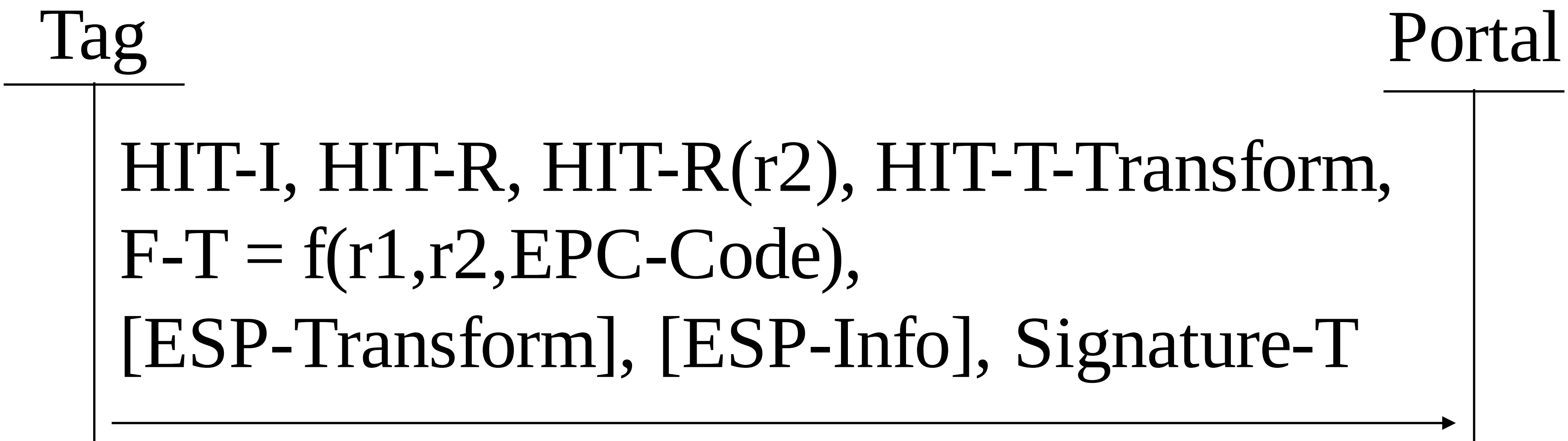
- **HIT-I**
  - A random value generated by the tag
- **HIT-R**
  - A known HIT
  - A null value

T-BEX Exchange, R1-T



- **r1, random value generated by the Portal.**
- **HIT-T-Transforms, list of f functions and associated parameters.**
- **ESP-Transforms, optional list of ESP-Transforms, used when a secure communication channel is requested.**

T-BEX Exchange, I2-T



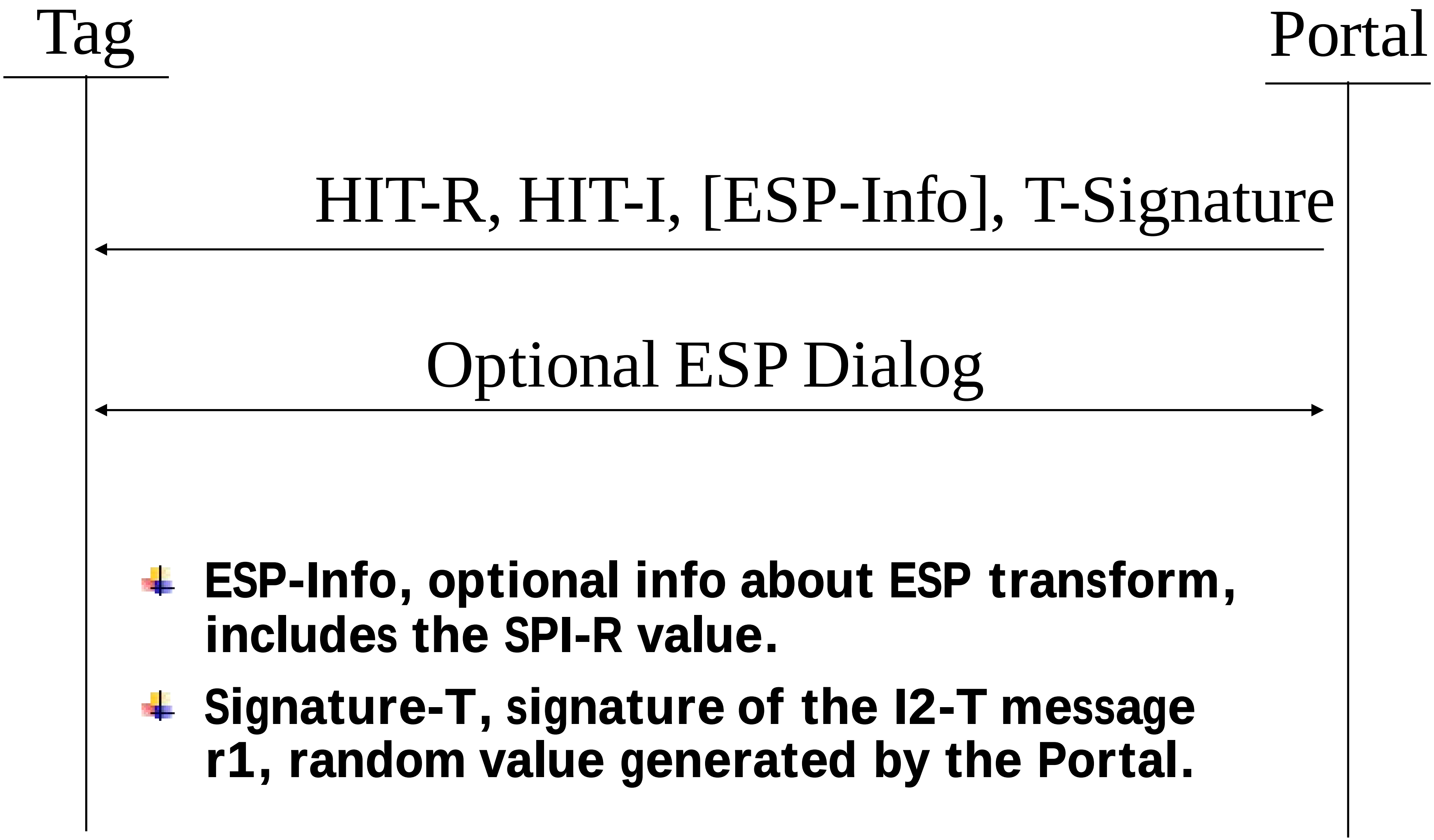
Tag

Portal

HIT-I, HIT-R, HIT-R(r2), HIT-T-Transform, F-T = f(r1,r2,EPC-Code), [ESP-Transform], [ESP-Info], Signature-T

- **r2, random value generated by the Tag.**
- **HIT-T-Transform, selected f function.**
- **F-T, equation to solve**
  - f(r1,r2, EPC-Code)
- **ESP-Transform, optional selected ESP-Transform**
- **ESP-Info, optional info about ESP transform, includes the SPI-I value.**
- **Signature-T, signature of the I2-T message**
  - KI-Auth-key = g(r1, r2, EPC-Code)

T-BEX Exchange, R2-T (Optional)



Tag

Portal

HIT-R, HIT-I, [ESP-Info], T-Signature

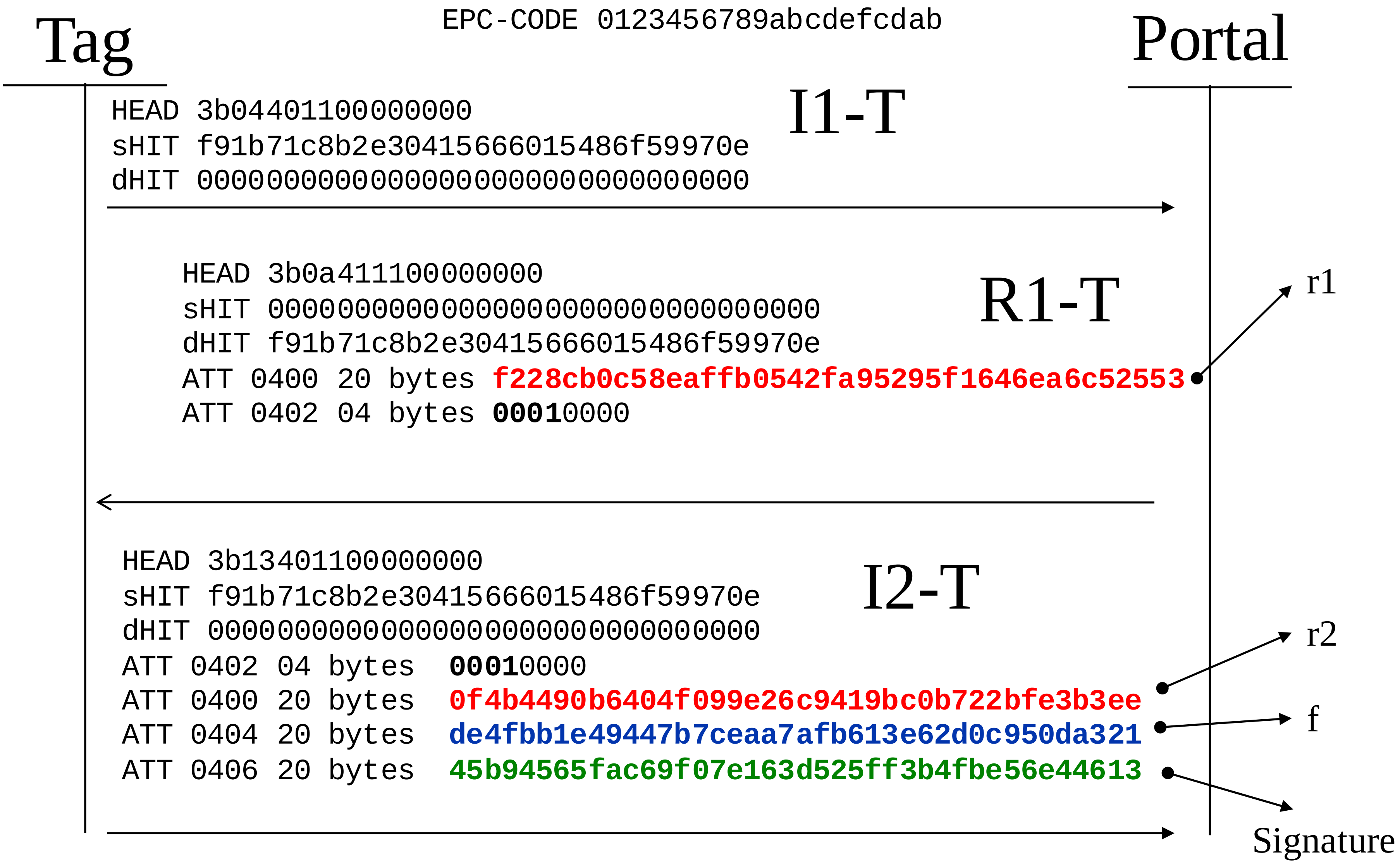
Optional ESP Dialog

- **ESP-Info, optional info about ESP transform, includes the SPI-R value.**
- **Signature-T, signature of the I2-T message r1, random value generated by the Portal.**

T-Transform 0001 - HMAC

- **K = HMAC-SHA1(r1 | r2, EPC-Code)**
- **F-T = HMAC-SHA1(K, CT1 | "Type 0001 key ")**
  - CT1 = 0x00000001 (32 bits)
- **K-AUTH-KEY = HMAC-SHA1(K, CT2 | "Type 0001 key")**
  - CT2 = 0x00000010 (32 bits)

Example, with T-Transform = 0x0001



EPC-CODE 0123456789abcdefcdab

Tag

Portal

I1-T

HEAD 3b04401100000000
sHIT f91b71c8b2e30415666015486f59970e
dHIT 00000000000000000000000000000000

R1-T

HEAD 3b0a411100000000
sHIT 00000000000000000000000000000000
dHIT f91b71c8b2e30415666015486f59970e
ATT 0400 20 bytes **f228cb0c58eaffb0542fa95295f1646ea6c52553**
ATT 0402 04 bytes **0001**0000

r1

I2-T

HEAD 3b13401100000000
sHIT f91b71c8b2e30415666015486f59970e
dHIT 00000000000000000000000000000000
ATT 0402 04 bytes **0001**0000
ATT 0400 20 bytes **0f4b4490b6404f099e26c9419bc0b722bfe3b3ee**
ATT 0404 20 bytes **de4fbb1e49447b7ceaa7afb613e62d0c950da321**
ATT 0406 20 bytes **45b94565fac69f07e163d525ff3b4fbe56e44613**

r2

f

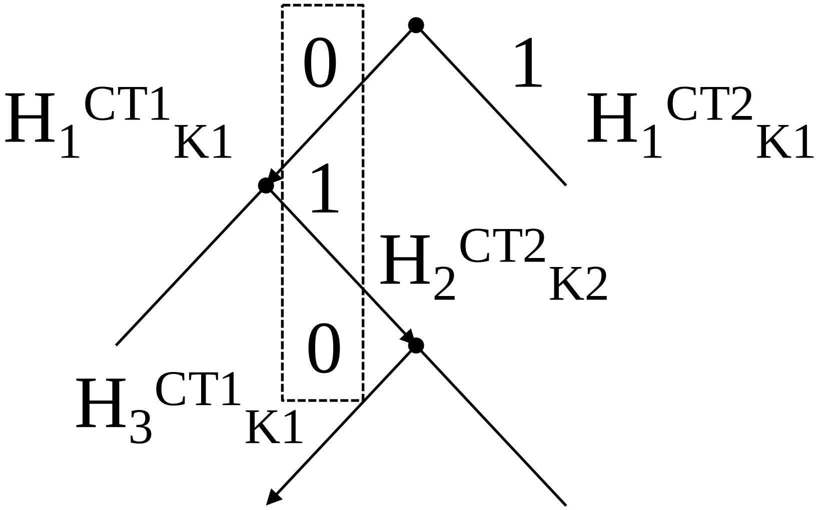
Signature

T-Transform 0002 – Tree (early proposal)

- **F-T = H1 | H2 | Hi | Hn**
  - Hi = HMAC-SHA1(r1 | r2, Ki | CT1),or
  - Hi = HMAC-SHA1(r1 | r2, Ki | CT2)
    - CT1 = 0x00000001, CT2 = 0x00000010
  - Notation: $H_{i}^{CTk}{}_{Ki}$ k=1,2 i=1…n
- **K-AUTH-KEY = HMAC-SHA1(K, CT1 | "Type 0002 key")**
  - K = HMAC-SHA1(r1 | r2, EPC-Code)
    - CT1 = 0x00000001 (32 bits)

EPC-Code = 010….

F-T = $H_1^{CT1}{}_{K1}\ H_2^{CT2}{}_{K2}\ H_3^{CT1}{}_{K1}$



Open Resources

- **http://perso.telecom-paristech.fr/~urien/hiptag**
  - Java code for portal.
  - Java card code for tags.
    - ISO 14443 tags work at 13,56 MHz.
    - Java card are widely deployed, about 1 billion devices per year.
    - Thanks to the NFC technology, HIP-TAG could be supported by billions of mobile phones.
- **http://gforge.cnam.fr/gf/project/t2tit**
  - Code source of the T2TIT project, funded by the French National Research Agency (ANR).
- **Papers: HIP-tags, a new paradigm for the Internet Of Things**
  - Urien, P.; Elrharbi, S.; Nyamy, D.; Chabanne, H.; Icart, T.; Pepin, C.; Bouet, M.; Cunha, D.; Guyot, V.; Krzanik, P.; Susini, J.-F.; Wireless Days, 2008. WD '08. 1st IFIP, 24-27 Nov. 2008 Page(s):1 – 5. Available at IEEE Explorer.

Conclusion

- **Is Internet Of Thing a working item for the IRTF?**
- **Is HIP a good candidate for the IoT ?**
- **Is privacy a main request for the IoT ?**
- **Is it acceptable to have fix identifier for the IoT?**