



**HMAC**

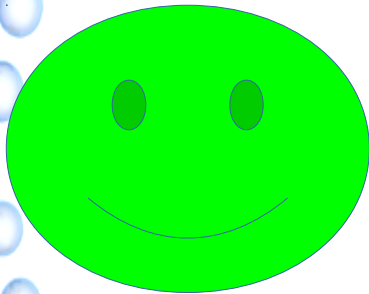
# Hashed Message authentication code

par Nicolas FRANCOIS et Joseph LEFEVRE

*décembre 2016*

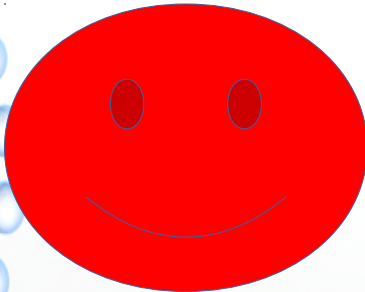
# Authentication

Bob



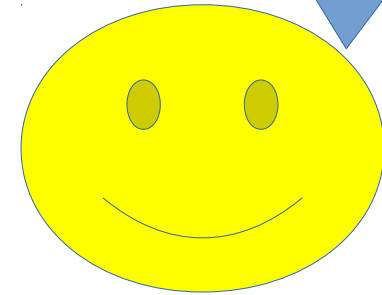
Bonjour,  
Je suis Bob et mon mail  
est bob@gmail.com

Jack



Bonjour,  
Je suis Bob et mon mail  
est bob@yahoo.fr

Répondre à  
bob@gmail.com  
ou  
bob@yahoo.fr ?



Alice

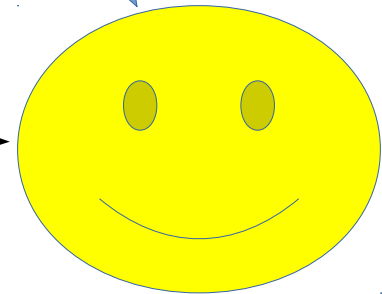
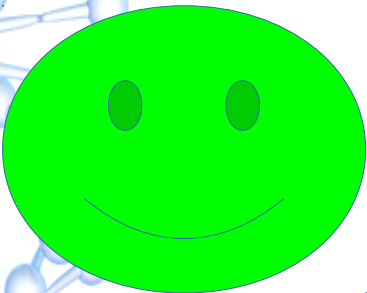
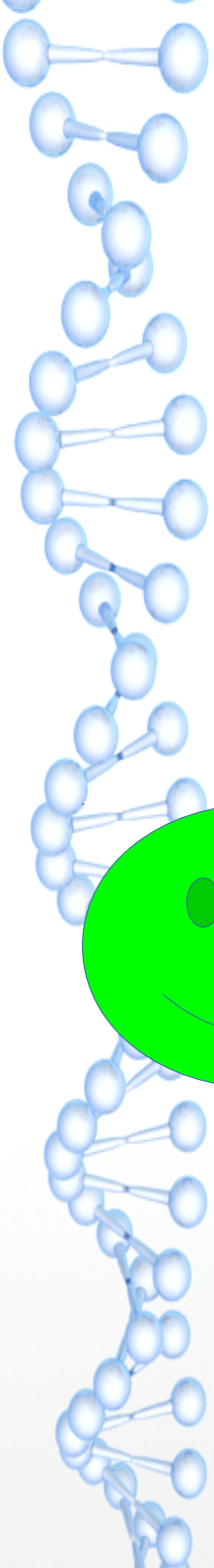
# Intégrité

Le mot de passe est  
A!idj\_5e

Le message  
a-t-il  
été déformé ?

Bob

Alice



# Fonction de hachage

## Vérifier le mot de passe

MonMotDePasse2016

Fonction de hachage MD5

49b1a1f31559e828  
a7d6e377e23094b5

## Vérifier l'intégrité d'un texte

[https://www.google.fr/  
search?client=ubuntu  
&channel=fs&q=google  
&ie=utf-8&oe=utf-8  
&gfe\\_rd=cr  
&ei=gsFGWIffIM3  
S8AfYvZj4Dg](https://www.google.fr/search?client=ubuntu&channel=fs&q=google&ie=utf-8&oe=utf-8&gfe_rd=cr&ei=gsFGWIffIM3S8AfYvZj4Dg)

Fonction de hachage MD5

813fa4b155a53522  
3547d120adc68ce1

# Fonction de hachage : Collision



MD5

253dd04e87492e4  
fc3471de5e776bc3d

MD5



# Fonction de hachage : Risque

Bob

MonMotDePasse2016

Fonction de hachage MD5

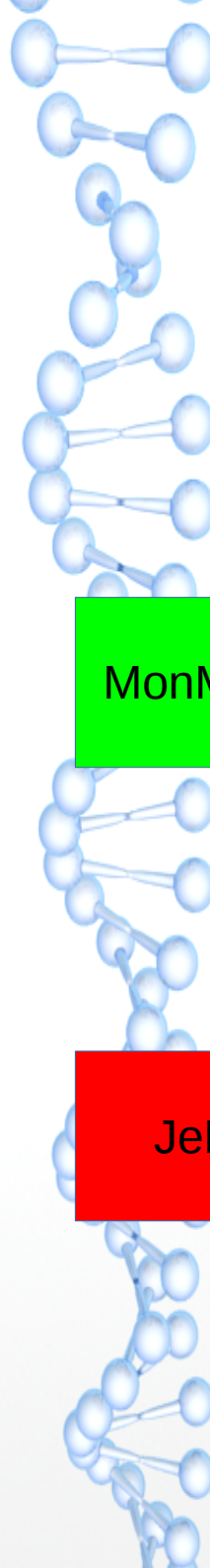
49b1a1f31559e828  
a7d6e377e23094b5

Jack

JeHackeTonMDP

?

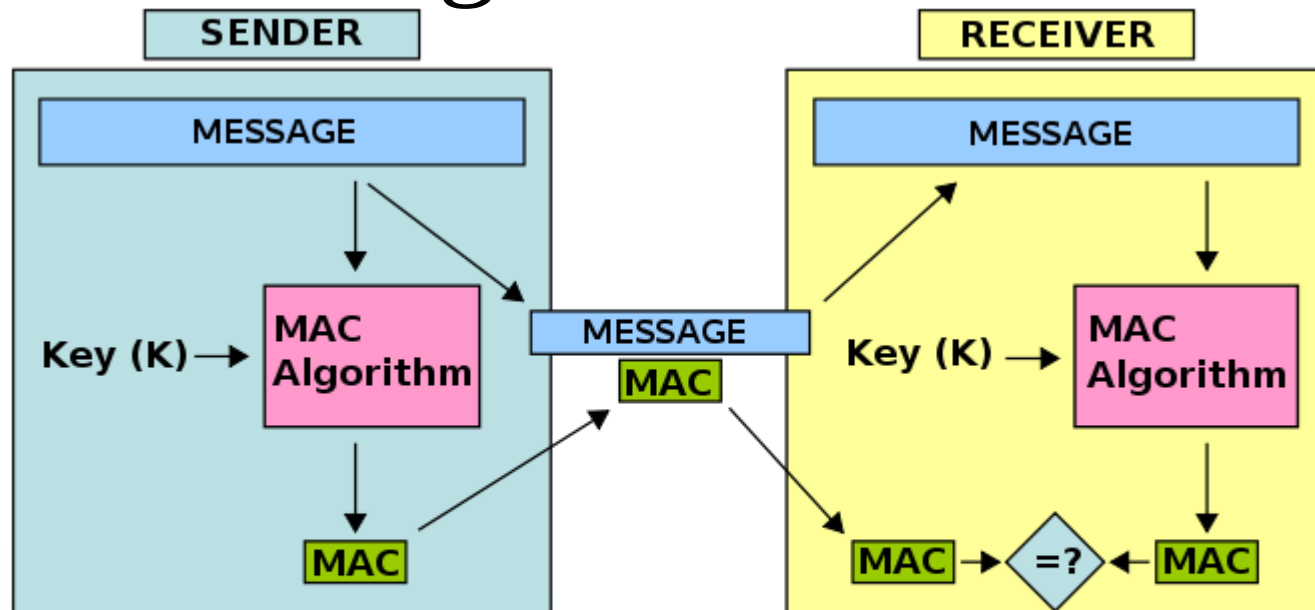
Si oui, alors Jack a réussi à hacker  
le compte de Bob  
(ex : Yahoo piraté)



# Retour sur le HMAC

## *Principe d'un MAC*

(MAC : message authentication code)







HMAC = *hashed* MAC

Dépendance d'une fonction de hachage.

- SHA (SHA1, SHA2, SHA512)
- MD5

Critère pour le choix d'une fonction de hachage :

- Taille du MAC
- Temps de calcul





# Historique du HMAC

- 1996 : Définition du HMAC
- 1998 : OpenSSL
- 1999 : TLS (transport layer security)
- 2012 : Angular 1.0.0
- 2016 : Mysensors 2.0



# HMAC : De plus en plus populaire

C'est léger donc :

- Objets connectés
- Sécurisation de tous les flux Client/Serveur (HTTPS)

C'est de plus en plus demandé car :

- Avis des experts (comme Cisco)
- Sensibilisation des acteurs (effet Snowden)

# *La formule du HMAC :*

$$\text{HMAC}(K,M) = H[(K+ \text{ XOR opad}) \parallel H[(K+ \text{ XOR ipad}) \parallel M]].$$

H = fct hachage

M = message

L = number of blocks in

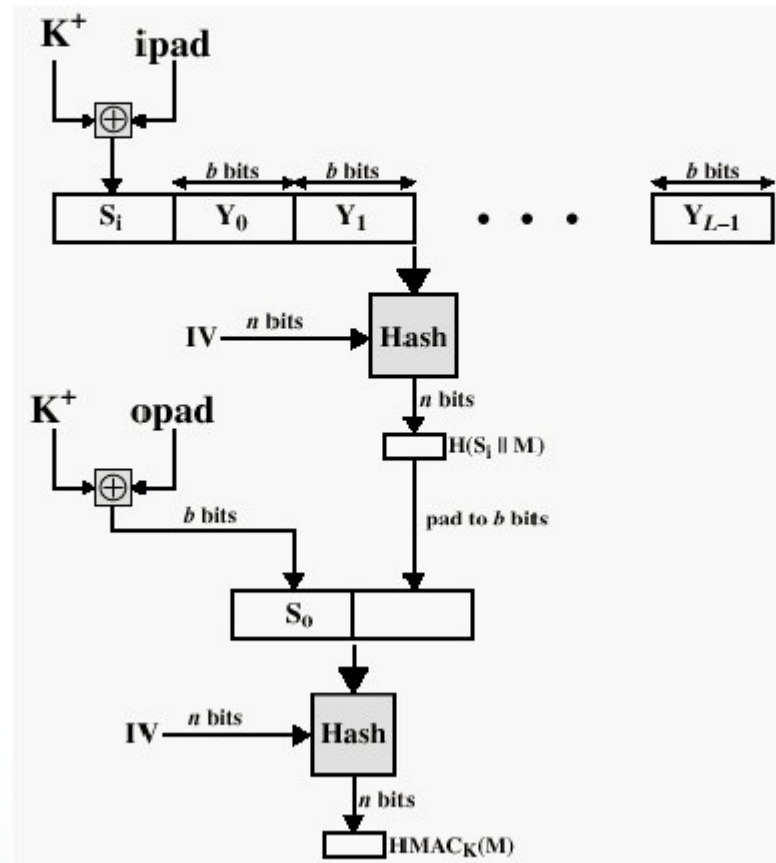
B = nb de bits d'un bloc

K = clé secrète

K+ = K avec des zéros à gauche  
(pour attendre taille de b bits)

Ipad = 36 (en hexa) répété b/8 fois

Opad = 5C (en hexa) répété b/8 fois

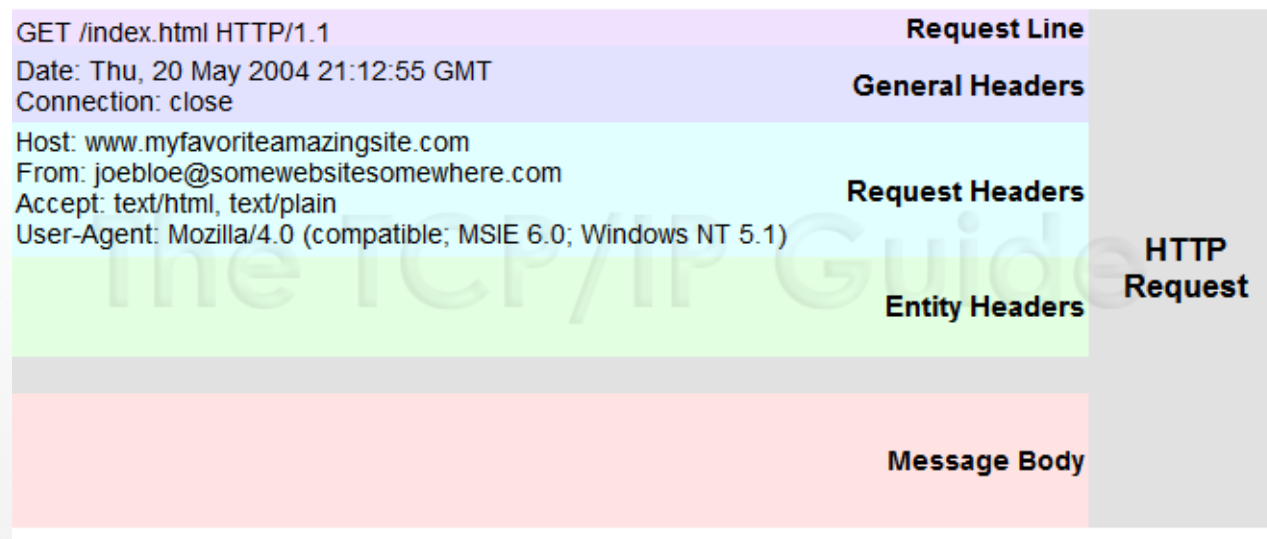


# Démonstration ! (1)

Communication client/serveur :

- 1) Jajoe a le droit de se connecter à NAS-Koisell.
- 2) Ils partagent la clé 'actemiumPower'.
- 3) NAS-Koisell n'accepte que les requêtes GET de Jajoe.

Format HTTP Request :





# Démonstration ! (1 suites)


requêtes reçus (voir paramètre id de l'URL):

- <https://www.nas-koisell.fr/images?id=4baef79af292cac4e1ccc7ca164c48733f82c13c>
- <https://www.nas-koisell.fr/images?id=4baef79af292cac4e1ccc7ca164c48733f82c13c>
- <https://www.nas-koisell.fr/images?id=5dea25dec1aab9cd7faededa51167aecccc551cad>

A quels requêtes répondre et renvoyer la page web ?  
Rappel : on ne renvoie la page web qu'à Jajoe.

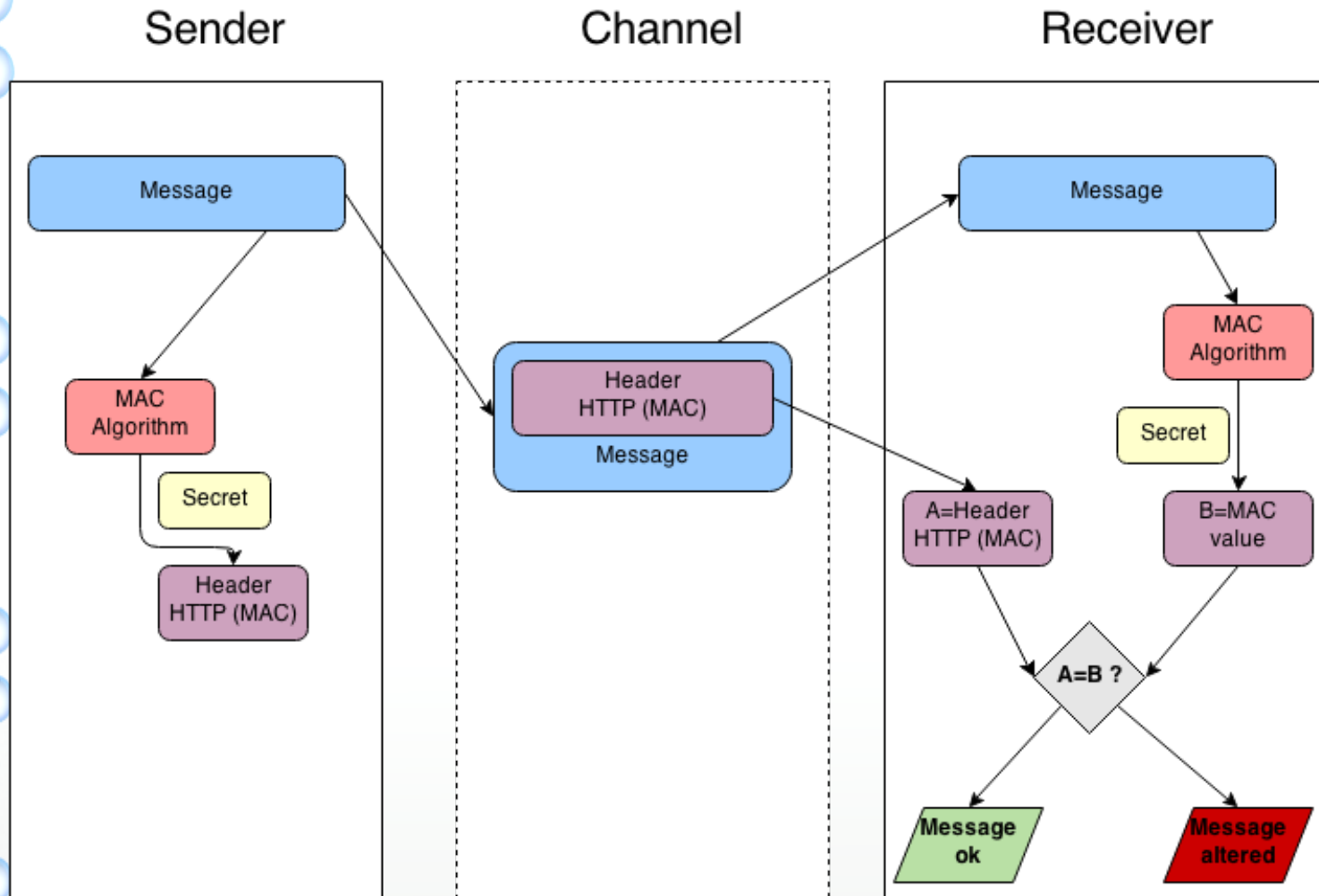
Calcul du HMAC-SHA1 de la requête  
avec notre GUI !

# Application 1 : Mysensors

**IOT + DIY =**   
**MySensors**

# Application 2 : AngularJS

Intérêt : Eviter le «Man in the middle»







# Avantages/inconvénients

- Rapide
  - Pas de contraintes propriétaires
  - Supporté par de nombreuses plate-formes
  - Pas de failles de sécurité algorithmique (NGE et QR selon Cisco)
- 
- N'est pas aussi répandu que d'autres standards du Web.
  - Echange de clés = faille.



# Faibles de sécurité

- Attaques par force brute.
- Interception de la clé privée (plus répandu)



# Conclusion

Une solution qui a de l'avenir ?