Gestion des clés en Bluetooth Low Energy

Tristan Claverie, José Lopes Esteves

Agence nationale de la sécurité des systèmes d'information

17 mai 2020

A propos

ANSSI, Laboratoire Sécurité des Technologies Sans-Fil :

- Sécurité électromagnétique (TEMPEST, AGREMI)
- Protocoles de radiocommunication
- Traitement du signal
- Simulations, mesures, électromagnétisme
- Systèmes embarqués

A propos

ANSSI, Laboratoire Sécurité des Technologies Sans-Fil :

- Sécurité électromagnétique (TEMPEST, AGREMI)
- Protocoles de radiocommunication
- Traitement du signal
- Simulations, mesures, électromagnétisme
- Systèmes embarqués

Tristan Claverie

- Sécurité des protocoles de radiocommunication
- IoT
- TNT, Bluetooth LE, Classic, LoRaWAN
- Radio logicielle

Programme

- Contexte
- Sécurité du Bluetooth Low Energy
- Cas 1 : Diffie-Hellman sur courbes elliptiques
- Cas 2 : Génération de clés
- Résultats des tests
- Conclusion

Contexte

- BLE standardisé en 2010
- Protocole de communication
- De plus en plus déployé
- Le standard définit des mesures de sécurité
- Que valent-elles ? Comment vérifier ?

$\Rightarrow$ Besoin de vérifier ce qui est fait

La sécurité selon le standard

Propriétés de sécurité

- Vie privée
- Confidentialité [des communications]
- Intégrité [des communications]
- Authenticité [des communications]

Modèle d'attaquant

- Passif : peut écouter des messages
- Actif : peut écouter, intercepter, forger des messages

Mécanismes de sécurité

Plusieurs mécanismes sont définis pour apporter ces propriétés. Ils se reposent sur un certain nombre de clés : STK, LTK, CSRK, IRK.

Mécanismes de sécurité

Plusieurs mécanismes sont définis pour apporter ces propriétés. Ils se reposent sur un certain nombre de clés : STK, LTK, CSRK, IRK.

- *Pairing* : génération de clé et authentification
- *Link Encryption* : chiffrement des communications, nécessite STK ou LTK
- *Bonding* : génération de clé, nécessite un lien chiffré
- *Data Signing* : protection de certains messages en intégrité, nécessite CSRK
- *Private Address Resolution* : rotation d'addresse d'un équipement, nécessite IRK
- *LE Privacy* : rotation d'addresse d'un équipement

Mécanismes de sécurité

Plusieurs mécanismes sont définis pour apporter ces propriétés. Ils se reposent sur un certain nombre de clés : STK, LTK, CSRK, IRK.

- *Pairing* : génération de clé et authentification
- *Link Encryption* : chiffrement des communications, nécessite STK ou LTK
- *Bonding* : génération de clé, nécessite un lien chiffré
- *Data Signing* : protection de certains messages en intégrité, nécessite CSRK
- *Private Address Resolution* : rotation d'addresse d'un équipement, nécessite IRK
- *LE Privacy* : rotation d'addresse d'un équipement

Quelques changements ont eu lieu entre sur le mécanisme *Pairing* entre le BLE 4.0 (Legacy Pairing) et 4.2 (LE Secure Pairing).

Mécanismes de sécurité en image

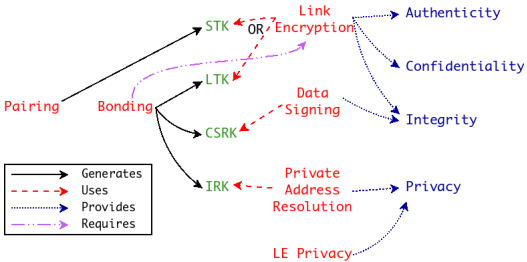


Figure – LE Legacy Pairing

Mécanismes de sécurité en image

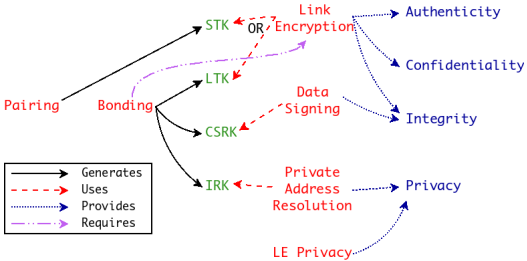


Figure – LE Legacy Pairing

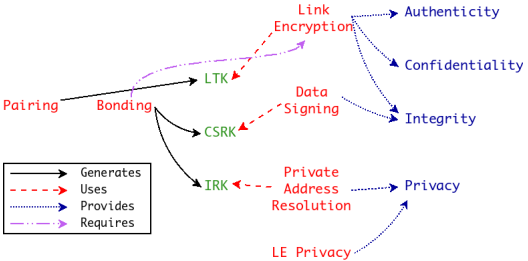


Figure – LE Secure Pairing

Contexte de l'étude

Focus

On s'intéresse à la Confidentialité, l'Intégrité et l'Authenticité des communications.
Les mécanismes impliqués sont *Data Signing* et *Link Encryption*.

Tout se base sur du chiffrement symétrique :
$\Rightarrow$ La sécurité du mécanisme dépend de la sécurité de la clé
$\Rightarrow$ Les analyses se sont concentrées sur la phase de *Pairing*, qui initie la génération des clés

Différents protocoles

- BLE 4.0 : famille *LE Legacy Pairing*
  - JustWorks
  - Passkey Entry
  - Out of Band
- BLE 4.2 : famille *LE Secure Pairing*
  - JustWorks
  - Passkey Entry
  - Numeric Comparison
  - Out of Band

- Le nom réfère à l'interaction utilisateur nécessaire
- Pour *Numeric Comparison*, il faut vérifier une donnée
- Pour tous les autres cas, il s'agit d'échanger une donnée entre deux équipements

Legacy et Secure Pairing : fonctionnement

Legacy Pairing :

- Echange d'une donnée secrète (TK) entre deux équipements
- La donnée est utilisée pour dériver STK

Secure Pairing :

- Echange de clés ECDH entre deux équipements : partage de DHKey
- Echange d'une donnée d'authentification entre les équipements
- La donnée est utilisée pour authentifier l'échange ECDH
- La clé partagée DHKey est utilisée pour dériver LTK

Bonding

Pré-requis : utilisation de STK ou LTK pour chiffrer les communications.

Clés générées :

- Legacy Pairing : LTK, CSRK, IRK
- Secure Pairing : CSRK, IRK

Mode de génération non imposé, deux approches proposées :

1. Génération aléatoire
2. *Key Hierarchy*

$\Rightarrow$ La spécification mentionne que le second cas ne permet qu'une entropie limitée.

Etat de l'art

Auteur	Résultat
Ryan [8]	Deux procédures du LE Legacy Pairing vulnérables
Lindell [7]	Passkey Entry en LE Secure Pairing vulnérable si code prédictible
Antonioli [2]	LE Secure Pairing vulnérable à une réduction de taille de clé
Biham [4]	Mauvaise validation des clés publiques dans les échanges ECDH
Cremers [5]	Usurpation d'identité si un équipement ne valide pas la clé publique

Cas 1 : Echange Diffie-Hellman sur courbes elliptiques

ECDH : Résultats

- Une clé publique sur une courbe elliptique est un point (x,y) de la courbe
- Lors de l'échange ECDH, les clés publiques sont envoyées en entier
- Dans les protocoles d'appairage, seules les abscisses sont authentifiées
- Certaines implémentations ne vérifient pas les clés publiques (CVE 2018-5383)

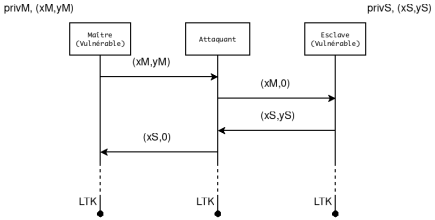


Figure – Attaque de Biham

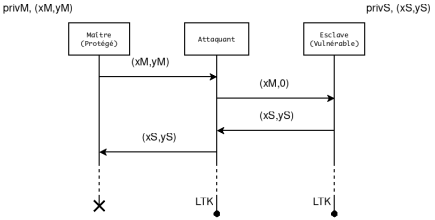


Figure – Attaque de Cremers

ECDH : Récupération de clé

Problème connu

Si un serveur qui garde la même clé privée, ne vérifie pas les clés publiques et qu'il est possible de récupérer la clé qu'il a dérivée, ça devient un oracle et il est possible de retrouver sa clé privée. [3, 6]

$\Rightarrow$ La spécification oblige à vérifier les clés publiques et à changer sa clé privée régulièrement.

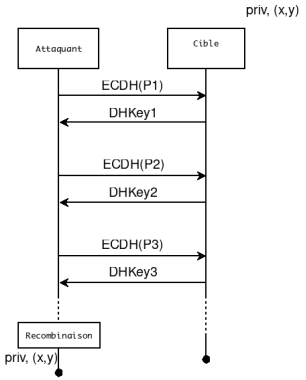


Figure – Principe de la récupération de clé

ECDH : Récupération de clé en BLE

- On sait que certains équipements ne vérifient pas les clés publiques
- On sait que certains équipements gardent la même clé privée

Principe de la récupération de clé en BLE :

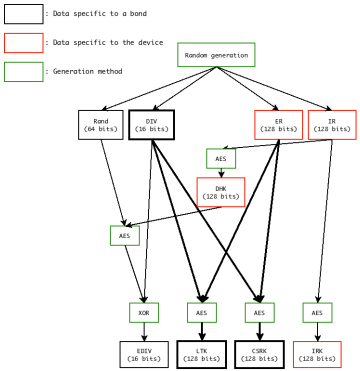
- Génération de points invalides
- Faire des appairages successifs
- Retrouver DHKey générée par la cible
  - Plus simple d'attaquer un équipement ayant le rôle de maître.

ECDH : Comment tester

1. L'équipement change sa clé privée (facile)
   - Plusieurs tentatives d'appairage, la clé publique doit changer.
2. L'équipement valide les clés publiques reçues (difficile)
   - Principe : envoi d'une clé publique invalide, l'appairage doit échouer.
   - Problème : pas de message d'erreur spécifique en recevant une clé publique invalide.
     - Echec dès réception d'une clé publique invalide
     - Echec dès utilisation d'une clé publique invalide
   - Besoin de réitérer le test pour être sûr que l'équipement valide correctement les clés.

Cas 2 : Génération de clé par Key Hierarchy

Génération de clé : Key Hierarchy



$\Rightarrow$ C'est un générateur aléatoire qui génère 2^{16} clés possibles.

$\Rightarrow$ La spécification mentionne qu'il faut lui préférer le tirage parfaitement aléatoire si on a besoin de sécurité (Vol 3, Partie H, Appendice B., paragraphe 2.2)

Problèmes avec Key Hierarchy

- Possibilité d'énumération des clés possibles

- Legacy Pairing : les clés CSRK et LTK sont impactées
- Secure Pairing : la clé CSRK est impactée

$\Rightarrow$ En Legacy Pairing, énumérer toutes les LTK possibles permet de déchiffrer les communications d'un équipement **sans avoir assisté à l'appairage**

Key Hierarchy : Comment tester

En considérant que l'une des deux méthodes de génération de clé décrite est implémentée, on doit savoir discriminer deux cas :

- Un générateur aléatoire à 2^{16} sorties possibles (équipement vulnérable)
- Un générateur aléatoire à 2^{128} sorties possibles (équipement non vulnérable)

Principe du test en boîte noire

On réalise beaucoup de *Pairing* + *Bonding*, on vérifie s'il y a des collisions entre les clés générées.

$\Rightarrow$ Il y a un compromis à faire entre la durée du test et sa précision.

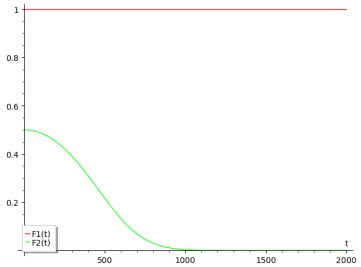
Interprétation de résultats

- Il y a une collision au bout de X essais, quelle est la probabilité que l'équipement testé soit vulnérable ?
- Il n'y a pas de collision au bout de X essais, quelle est la probabilité que l'équipement testé soit vulnérable ?
- Combien d'essais sont nécessaires pour être sûr à 99% de détecter un équipement vulnérable ?

$\Rightarrow$ Utilisation de quelques probabilités pour tenter de répondre à ces questions.

$\Rightarrow$ Dépend d'une inconnue : la proportion d'équipements vulnérables.

Interprétation de résultats



- t : Nombre d'essais
- F1 (rouge) : $\sim$Taux de vrais positifs (doit être proche de 1)
- F2 (vert) : $\sim$Taux de faux négatifs (doit être proche de 0)

Détails complets dans les actes, pour étudier le compromis temps/précision

Résultats des tests

- Mirage [1] pour l'implémentation des tests.
- Détecte des équipements vulnérables à la CVE-2018-5383. Pas vu d'équipement vulnérable à la récupération de clés.
- Pour tester l'utilisation de *Key Hierarchy*, besoin d'instrumenter les équipements.
- Estimation du temps pour faire le test de la génération de clé : $\sim$8.5 secondes par clé générée sur un équipement instrumenté au niveau BLE, $\sim$25 secondes sur un équipement instrumenté au niveau utilisateur.
- Exemple : 1 collision au bout de 389 essais. En supposant une proportion d'1% d'équipements vulnérables, probabilité de $1.11e^{-16}$ que l'équipement sous test ne soit PAS vulnérable.

Conclusion

- Les implémentations ne respectent pas nécessairement les bonnes pratiques
- Il faut pouvoir tester des équipements, y compris en boîte noire
- Les procédures proposées fonctionnent
- Certains équipements utilisent le mode de génération *Key Hierarchy* (Divulgation en cours)
- En respectant la spécification à jour, pas de problème sur ces points-là
- Au delà de l'appairage, la génération de clé est importante pour la sécurité

Questions

- Merci de votre attention
- tristan.claverie@ssi.gouv.fr

Bibliographie

[1] Mirage documentation — Mirage 1.1 documentation.

[2] Antonioli, D., Tippenhauer, N. O., and Rasmussen, K.
Low Entropy Key Negotiation Attacks on Bluetooth and Bluetooth Low Energy.
13.

[3] Biehl, I., Meyer, B., and Muller, V.
Differential Fault Attacks on Elliptic Curve Cryptosystems (Extended Abstract).
16.

[4] Biham, E., and Neumann, L.
Breaking the Bluetooth Pairing – Fixed Coordinate Invalid Curve Attack.
26.

[5] Cremers, C., and Jackson, D.
Prime, order please! revisiting small subgroup and invalid curve attacks on protocols using diffie-hellman.
Cryptology ePrint Archive, Report 2019/526, 2019.
https://eprint.iacr.org/2019/526.

Bibliographie (cont.)

[6] Jager, T., Schwenk, J., and Somorovsky, J.
Practical Invalid Curve Attacks on TLS-ECDH.
19.

[7] Lindell, A. Y.
Attacks on the Pairing Protocol of Bluetooth v2.1.
10.

[8] Ryan, M.
Bluetooth : With Low Energy comes Low Security.
7.