

Protokol s koncovým šifrováním pro IEEE 802.15.4

Author: JAROMÍR BAČA

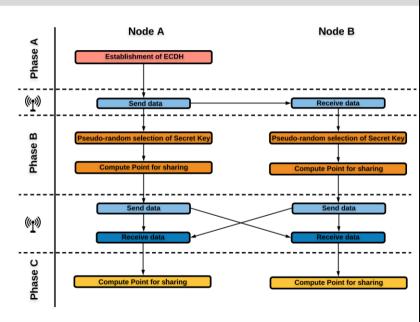
Advisor: Ing. ONDŘEJ KRAJSA, Ph.D.

Brno, 23.6.2020

Metodika výměny klíčů

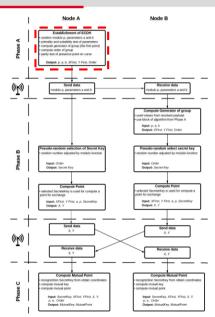


- Odesílatel, příjemce
- Inpirováno IKEv2
- Asym. krytptografie
- Eliptické křivky
- Tři fáze



Metodika výměny klíčů – Fáze A





Fáze A

Establishment of ECDH

- random modulo p, parameters a and b
- primality and suitability test of parameters
- compute generator of group (the first point)
- compute order of group
- parity test of presence point on curve

Output: p, a, b, XFirst, Y First, Order

Fáze A – sestavení parametrů výměny



- Pseudonáhodný výběr hodnoty p (modulo), a a b (parametry křivky)
- Kontrola správnosti vstupních hodnot
- Výpočet generátoru grupy (první bod) a řádu grupy
- Kontrola parity a důkaz, že poslední bod leží na křivce

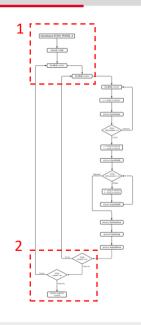
```
ECDH_PHASE_A(&mod, &a_parameter, &b_parameter, &X_first, &Y_first, &G_Order);
```

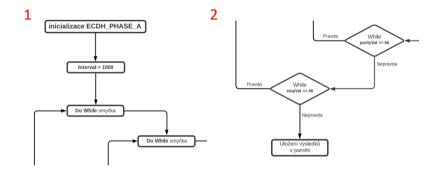
Vstup: žádný, aplikace si samostatně volí parametry

Výstup: modulus, a parameter, b parameter, XFirst, YFirst, GOrder

Fáze A – vývojový diagram



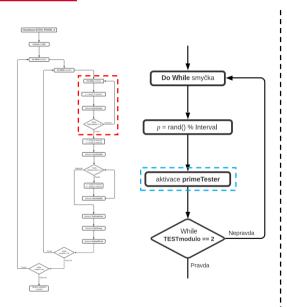




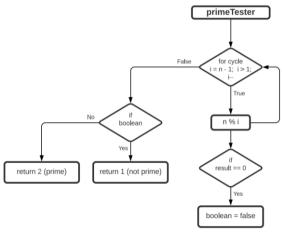
- Nastavení intervalu je součástí algoritmu
- Spuštění dvou testovacích smyček
- Test parity a test přítomnosti bodu na křivce

Fáze A – vývojový diagram (Výběr prvočísla)



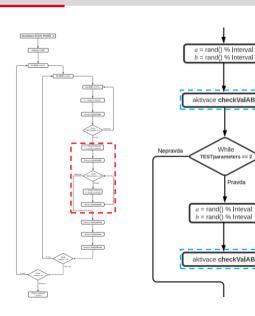


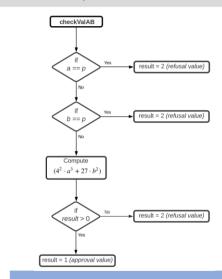
primeTester



Fáze A – vývojový diagram (Výběr parametrů)



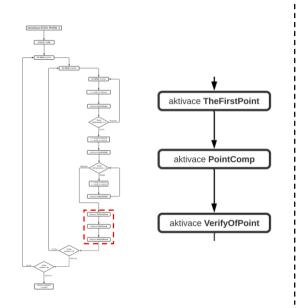




 $(4 \cdot a^3 + 27 \cdot b^2) \bmod p$

Fáze A – vývojový diagram





- TheFirstPoint
 - Výpočet generátoru grupy
 - Výpočet řádu grupy (počet bodů)
- PointComp
 - Výpočet všech bodů
 - Test parity
- VerifyOfPoint
 - Test přítomnosti bodu na křivce

Fáze A – test aplikace

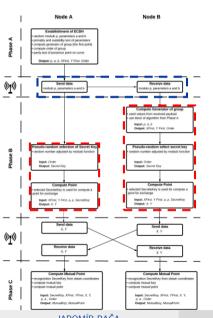


```
Modulo value is 5
Selected parameters are 1 and 6
Body isou [
             0, 1] [0, 4]
Body jsou [ 2, 1] [2, 4]
Body jsou [ 3, 1] [3, 4]
Body jsou [ 4,
                  21 [ 4, 31
The first point is [0, 1] and order of group is 9
Bod: [4, 2]
Bod: [2, 1]
Bod: [3, 4]
Bod: [3, 1]
Bod: [2, 4]
Bod: [4, 3]
Bod: [0, 4]
 Value X of the last point in group: 0
 Value Y of the last point in group: 4
 Result of parity test: 1
 Result of non-zero test: 1
Selected imput values are: mod = 5, parametry jsou 1 a 6
The first point has these coordinates [0, 1], Order is 9
```

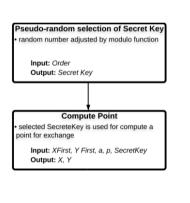
- Testováno s deaktivovaným pseudonáhodným výběrem
- Aplikace (na pozadí) vypočítá celou grupu
- Pokud jsou výsledky obou testů rovny 1, jsou hodnoty podstoupeny k dalším fázím procesu

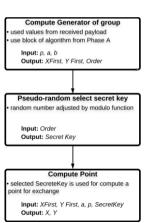
Metodika výměny klíčů – Fáze B





Fáze B





Fáze B – úvod



- Rozdělena na dvě subverze
- Příjemce si ze znalosti p, a, b si sám vypočítá generátor a řád grupy
- Pseudonáhodný výběr hodnoty tajného klíče
- Výpočet bodu pro sdílení

```
ECDH_PHASE_BA(&Order, &Xf, &Yf, &mod, &a, container);
```

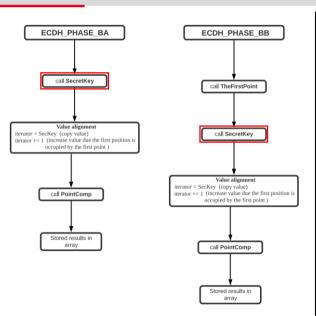
```
ECDH_PHASE_BB(&mod, &a, &b, container);
```

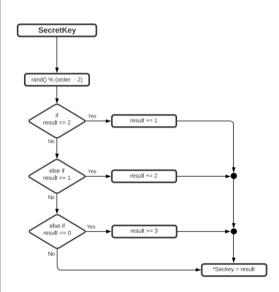
Vstup (subverze BA): řád, generátor grupy [X, Y], modulus, parametetry a, b Vstup (subverze BB): modulus, parametetry a, b

Výstup: pole obsahující souřadnice sdíleného bodu a tajný klíč

Fáze B – vývojový diagram a blok SecretKey

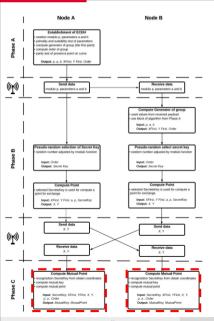






Metodika výměny klíčů – Fáze C





Fáze C

Compute Mutual Point

- recognizition SecretKey from obtain coordinates
- compute mutual key
- compute mutual point

Input: SecretKey, XFirst, YFirst, X, Y,

p, a, Order

Output: MutualKey, MutualPoint

Fáze C – úvod



- Ze znalosti vlastního tajného klíče a obdrženého bodu vypočítá společný klíč
- Aplikace z obdrženého bodu rozpozná tajný klíč protistrany
- Ze získaného společného klíče se spočítá společný bod

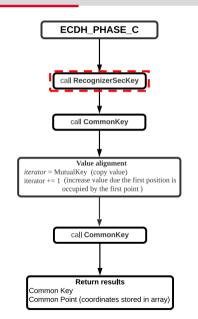
```
void ECDH_PHASE_C(&MSKey, &XF, &YF, &XO, &YO, &mod, &a, &order, &MutKEY, container);
```

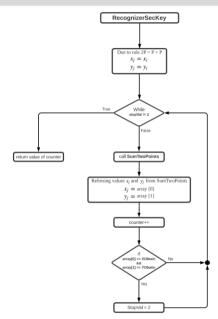
```
Vstup (subverze BA):
Vlastní tajný klíč, generátor grupy [X, Y], získaný bod [X, Y], modulus,
parametr a
```

Výstup: pole obsahující společný klíč, resp. společný bod

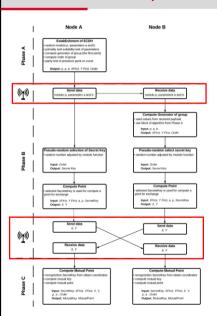
Fáze C – vývojový diagram a blok RecognizerSecKey





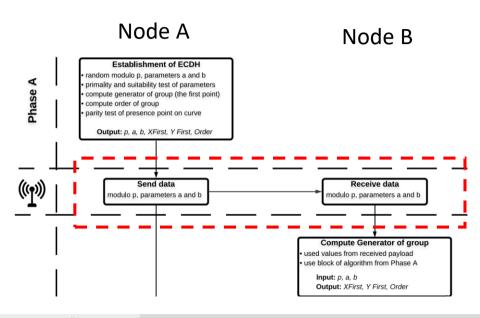




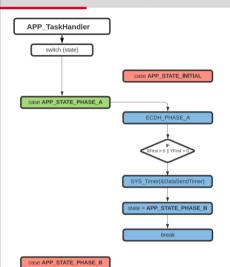


- Řešení bezdrátového přeposílání dat je řešeno na základě předchystaného LWM stacku
- V rámci implementace řešeno přepínání jednotlivých fází a spouštění vysílacích relací





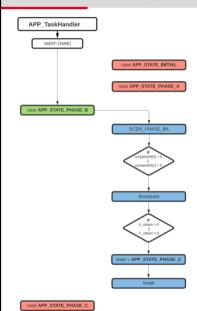




- Aktivace Fáze A, kde proběhne výběr vstupních parametrů a výpočtu generátoru grupy
- Pokud je splněna podmínka if, tedy že X nebo Y jsou větší než nula, znamená to, že první fáze úspěšně proběhla
- Pomocí SYS_Timer, který ovládá interval odesílání dat, jsou odeslány data protistraně
- V posledním příkazu je řešeno přepnutí programu na další fázi

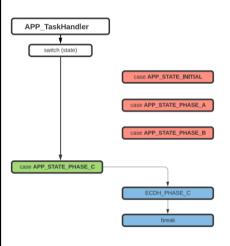
case APP_STATE_PHASE_C





- Aktivace Fáze BA, kde proběhne volba tajného klíče a následně výpočet bodu pro sdílení
- Pokud je splněna podmínka if, tedy že X nebo Y jsou větší než nula, znamená to, že fáze BA úspěšně proběhla
- Poté jsou souřadnice bodu X a Y odeslány protistraně
- Pokud je splněna další podmínka if, znamená to, že byly obdrženy data od protistrany a algoritmus se může přepnout na třetí fázy





Aktivace Fáze BA, kde výpočet společného klíče

- Do této fáze je implementován blok, který má za úkol zvýšit bitovou velikost vypočítaného společného klíče
- V této fázi je klíč předán do bloku AES, který jej použije k šifrování další komunikace



Děkuji za pozornost!

Otázky oponenta 1/2



- Proč ve svém algoritmu pro výměnu klíčů nevyužíváte bezpečné standardizované eliptické křivky?
 - Snahou bylo navrhnout jednoduché řešení, které bude optimalizované pro mikrokontrolery AVR.
- Jakým způsobem se bráníte útoku Man-in-the-middle?
 - Možným řešením by bylo zařazení předsdíleného klíče, který by šifroval odeslání veřejně dostupných parametrů pro výpočet eliptických křivek a hašovací funkci (např. SHA-1), který by správnost přenosu potvrdil.
 - Kryptografie na eliptických křivkách je asymentrického typu. Zachycení veřejných parametrů (p, a, b) útočníkem nepředstavuje riziko.

Otázky oponenta 2/2



Jaká je časová náročnost Vámi navrženého algoritmu pro testování prvočísel? Časová náročnost se odvíjí od velikosti čísla. Prvočíslo je testováno hrubou silou (hledání dělitele).

Jak dlouho trvá otestování bitového řetězce o délce 512 b pomocí Vámi navrženého algoritmu?

25 vteřin