Kryptografia post-kwantowa, algorytm New Hope
Poziom 5

Michał Artur Szlupowicz, Marek Brynda

Maj, 2020

1 Szyfry McEliece

Asymetryczny system szyfrowania wiadomości opierający się na trudności dekodowania kodów liniowych. Pozwala na odkodowanie wiadomości zawierającej t błędów.

1.1 Generowanie klucza

- Wybierz k x n macierz G dla t błędów.
- Losowo wybierz n x n macierz permutacji P oraz k x k odwracalną macierz S
- klucz prywatny = (S, G, P)
  klucz publiczny = (SGP, t)

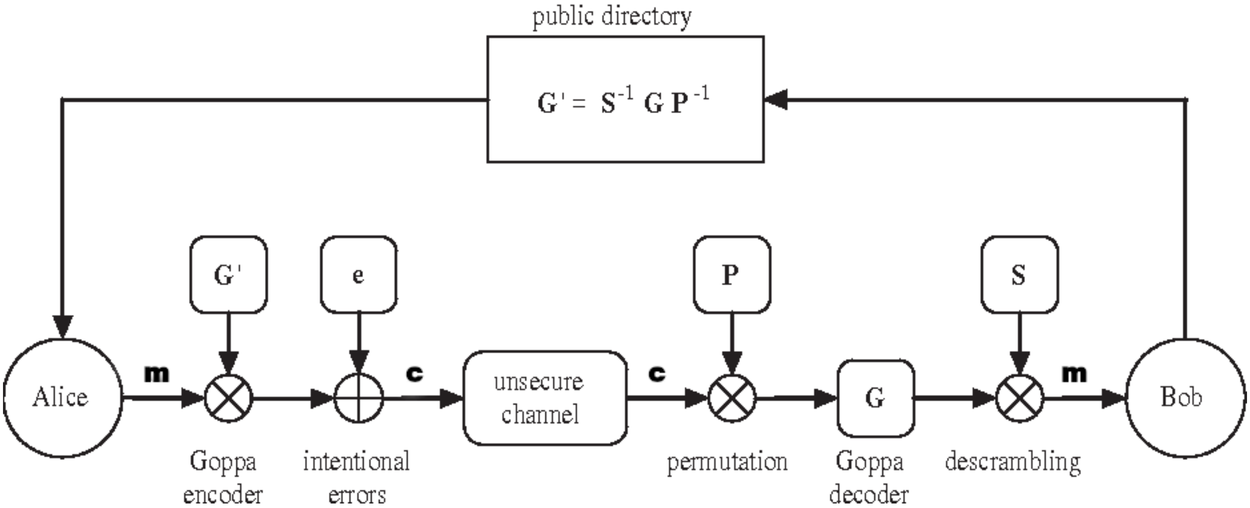
1.2 Szyfrowanie dla $m \in \mathbb{F}_q^n$

- Stwórz wiadomość m,
- Zakoduj wiadomość mnożąc ją przez losową macierz G_{pub} (SGP),
- Dodaj do wiadomości losowy błąd e (błąd ten może też wystąpić w trakcie przesyłu wiadomości)

$$c = mG_{pub} + e \tag{1}$$

1.3 Deszyfrowanie

- $z = cP^{-1}$ $\qquad z = mSG + eP^{-1},$
- $y = Decode_G(z)$ $\qquad y = mS,$
- $m' = yS^{-1}$ $\qquad m' = m.$



Rysunek 1: Schemat działania szyfrowania oraz deszyfrowania wiadomości (w opisie zastosowano odwrotne oznaczenia dla odwróconych P oraz S, tzn $P = P^{-1}$, a $P^{-1} = P$, tak samo dla S), źródło: https://www.semanticscholar.org/paper/LDPC-Codes-in-the-McEliece-Cryptosystem%3A-Attacks-Baldi/79f6c29884d52a870301ac7f683b75cdb135c32c

1.4 Tworzenie klucza szyfrującego

- Stwórz losową macierz generatora wybierając kod liniowy oraz wartości N,K,P,
- Pomnóż macierz G dla wybranego kodu przez macierz nieparzystą S a następnie przez macierz permutacji P.

1.5 Kody Liniowe

- N - rozmiar kodu
  K - rozmiar wiadomości
  T - maksymalna odporność na błędy

- $t \leqslant [\frac{n-k}{2}]$,
- Wielomian P(x) może zostać użyty do szyfrowania Systematic (error-correcting),
- P(x) - można zaprezentować jako kxn macierz G, taką że jej wiersze znajdują się w zakresie $\mathbb{F}_q^n$

2 Przykład

Odkodowana wersja w notebooku - napisana w pythonie.

- $\mathbb{F}_{23}$
- N = 6
- K = 3
- T = 1
- $m = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

2.1 Macierze

- $G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 4 & 9 & 16 & 25 \end{bmatrix}$
- $P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
- $S = \begin{bmatrix} 1 & 1 & 3 \\ 4 & 1 & 6 \\ 7 & 2 & 9 \end{bmatrix}$
- $SGP = \begin{bmatrix} 1 & 15 & 7 & 5 & 12 & 8 \\ 4 & 7 & 12 & 11 & 21 & 15 \\ 7 & 1 & 21 & 18 & 12 & 2 \end{bmatrix}$

2.2 Szyfrowanie wiadomość

- $c = SGP \cdot m$
- $c = \begin{bmatrix} 7 & 9 & 2 & 12 & 21 & 21 \end{bmatrix}$
- $c{+}e = \begin{bmatrix} 7 & 9 & 2 & 12 & 21 & 21 \end{bmatrix} + \begin{bmatrix} 0 & 6 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 7 & 15 & 2 & 12 & 21 & 21 \end{bmatrix}$

2.3 Odszyfrowywanie wiadomości

- $z = (c + e) \cdot P^{-1}$
- $z = \begin{bmatrix} 7 & 15 & 2 & 12 & 21 & 21 \end{bmatrix}$
- $y = Decode_G(z)$ (Algorytm Berlekamp-Welch)
- $y = \begin{bmatrix} 7 & 9 & 19 \end{bmatrix}$
- $m' = y \cdot S^{-1}$
- $m' = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
- $m' = m$

Literatura

[1] Suanne AuChristina Eubanks-TurnerJennifer Everson, The McEliece Cryptosystem
http://www.math.unl.edu/~s-jeverso2/McElieceProject.pdf

[2] Atif Khurshid, ESTR1004 - McEliece Cryptosystem
https://www.youtube.com/watch?v=GNgOJN9LD-I

[3] Atri Rudra, Lecture 27: Berlekamp-Welch Algorithm
https://cse.buffalo.edu/faculty/atri/courses/coding-theory/lectures/lect27.pdf

[4] James Cook, Error Correcting Codes
http://www-inst.eecs.berkeley.edu/~cs70/su14/notes/note_8.pdf