Asymetrický šifrovací algoritmus McEliece

Diplomová práce

Bc. Vojtěch Myslivec

vedoucí: prof. Ing. Róbert Lórencz, CSc.

Fakulta informačních technologií
České vysoké učení technické v Praze

11. června 2016

Obsah

Obsah

Postkvantová kryptografie

- Kvantové počítače – Shorův algoritmus 1994
    - *RSA*, *ECDSA*, ...

Postkvantová kryptografie

- Kvantové počítače – Shorův algoritmus 1994
  - *RSA*, *ECDSA*, ...
- Kandidáti pro postkvantovou kryptografii [3, 11]
  - Symetrická kryptografie – *AES*
  - *Lattice-based*
  - *Hash-based*
  - *Code-based*

Postkvantová kryptografie

- Kvantové počítače – Shorův algoritmus 1994
  - *RSA*, *ECDSA*, . . .
- Kandidáti pro postkvantovou kryptografii [3, 11]
  - Symetrická kryptografie – *AES*
  - *Lattice-based*
  - *Hash-based*
  - *Code-based* – McEliece

Obsah

Kryptosystém McEliece

- Asymetrický šifrovací algoritmus
    - *Robert McEliece* 1978 [1]

Kryptosystém McEliece

- Asymetrický šifrovací algoritmus
  - *Robert McEliece* 1978 [1]
- Využívá lineární kód pro opravu chyb
  - Náhodný chybový vektor jako součást šifry
  - Dekódovat neznámý lineární kód je NP-těžká úloha [2]

Kryptosystém McEliece

- Asymetrický šifrovací algoritmus
  - *Robert McEliece* 1978 [1]
- Využívá lineární kód pro opravu chyb
  - Náhodný chybový vektor jako součást šifry
  - Dekódovat neznámý lineární kód je NP-těžká úloha [2]
- Velké klíče (stovky kilobitů až jednotky megabitů)

Kryptosystém McEliece

- Asymetrický šifrovací algoritmus
  - *Robert McEliece* 1978 [1]
- Využívá lineární kód pro opravu chyb
  - Náhodný chybový vektor jako součást šifry
  - Dekódovat neznámý lineární kód je NP-těžká úloha [2]
- Velké klíče (stovky kilobitů až jednotky megabitů)
- Digitální podpis

Kryptosystém McEliece

- Asymetrický šifrovací algoritmus
  - *Robert McEliece* 1978 [1]
- Využívá lineární kód pro opravu chyb
  - Náhodný chybový vektor jako součást šifry
  - Dekódovat neznámý lineární kód je NP-těžká úloha [2]
- Velké klíče (stovky kilobitů až jednotky megabitů)
- Digitální podpis
  - Schéma publikováno až v roce 2001 [5]
  - Využívá příbuzný kryptosystém Niederreiter [8]

Generování klíčů

1. *Lineární kód* $\mathcal{K}$ (n, k) opravující t chyb, s $k \times n$ generující maticí G
2. Náhodná $k \times k$ regulární matice S
3. Náhodná $n \times n$ permutační matice P
4. Vypočítáme $k \times n$ matici $\hat{G} = SGP$

Kryptosystém McEliece

Generování klíčů

1. *Lineární kód* $\mathcal{K}$ (n, k) opravující t chyb, s $k \times n$ generující maticí G
2. Náhodná $k \times k$ regulární matice S
3. Náhodná $n \times n$ permutační matice P
4. Vypočítáme $k \times n$ matici $\hat{G} = SGP$

Vygenerované klíče

Veřejné parametry
: Čísla k, n, t

Veřejný klíč
: Matice $\hat{G}$ ($\hat{G} = SGP$)

Soukromý klíč
: Matice S, P a kód $\mathcal{K}$ generovaný G

Příklad

Kód Γ s parametry $(n, k, t) = (8, 2, 2)$:

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Příklad

Kód Γ s parametry $(n, k, t) = (8, 2, 2)$:

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Náhodné matice S a P:

$$S = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Příklad

Kód Γ s parametry $(n, k, t) = (8, 2, 2)$:

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Náhodné matice S a P:

$$S = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Veřejný klíč – matice $\hat{G}$:

$$\hat{G} = SGP = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Šifrování

Algoritmus E:

Máme zprávu m délky k, veřejný klíč $\hat{G}$ a parametr t

1. Vygenerujeme chybový vektor z délky n s *Hammingovou vahou* t
2. Šifrový text $c = m\hat{G} + z$

Kryptosystém McEliece

Šifrování

Algoritmus E:

Máme zprávu m délky k, veřejný klíč $\hat{G}$ a parametr t

1. Vygenerujeme chybový vektor z délky n s *Hammingovou vahou* t
2. Šifrový text $c = m\hat{G} + z$

Dešifrování

Algoritmus D:

1. Vypočítáme $\hat{c} = cP^{-1}$
2. Dekódujeme $\hat{m}$ z $\hat{c}$ pomocí použitého kódu
   $Dek(\hat{c}) = \hat{m}$
3. Vypočítáme původní zprávu $m = \hat{m}S^{-1}$

Kryptosystém McEliece

Příklad šifrování

Otevřený text $m = (\ 1\ 1\ )$, náhodný chybový vektor z *váhy* $t = 2$:

$$c = m\hat{G} + z = (\ 1\ 1\ ) \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} + (\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ )$$

$$c = (\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ )$$

Kryptosystém McEliece

Příklad dešifrování

Vynásobení c inverzí permutace:

$$\hat{c} = cP^{-1} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Dekódování kódem Γ – odstranění chyby:

$$\hat{m} = Dek_{\Gamma}(\hat{c}) = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

Vynásobení inverzí S:

$$m = mS^{-1} = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$$

Obsah

Binární Goppa kódy

- Základ pro *code-based* kryptografii

Binární Goppa kódy

- Základ pro *code-based* kryptografii
- Neexistují útoky na strukturu kódu

Binární Goppa kódy

Sestrojení binárního (ireducibilního) Goppa kódu

Kód Γ s parametry (n, k) opravující t chyb

Binární Goppa kódy

Sestrojení binárního (ireducibilního) Goppa kódu

Kód Γ s parametry (n, k) opravující t chyb

- **Goppův polynom** g
  (Ireducibilní) polynom stupně t z okruhu polynomů nad konečným tělesem $\mathbb{F} = GF(2^m)$

$$g \in \mathbb{F}[x] \quad deg(g) = t$$

Binární Goppa kódy

Sestrojení binárního (ireducibilního) Goppa kódu

Kód Γ s parametry (n, k) opravující t chyb

- **Goppův polynom** g
  (Ireducibilní) polynom stupně t z okruhu polynomů nad konečným tělesem $\mathbb{F} = GF(2^m)$

$$g \in \mathbb{F}[x] \quad deg(g) = t$$

$\Rightarrow$ *rozšířené* těleso $GF((2^m)^t)$

Binární Goppa kódy

Sestrojení binárního (ireducibilního) Goppa kódu

Kód Γ s parametry (n, k) opravující t chyb

- **Goppův polynom** g
  (Ireducibilní) polynom stupně t z okruhu polynomů nad konečným tělesem $\mathbb{F} = GF(2^m)$

  $$g \in \mathbb{F}[x] \quad deg(g) = t$$

  $\Rightarrow$ *rozšířené* těleso $GF((2^m)^t)$
- **Podpora** L
  Náhodná permutace všech prvků z tělesa $\mathbb{F}$

Binární Goppa kódy

Sestrojení binárního (ireducibilního) Goppa kódu

Kód Γ s parametry (n, k) opravující t chyb

- **Goppův polynom** g
  (Ireducibilní) polynom stupně t z okruhu polynomů nad konečným tělesem $\mathbb{F} = GF(2^m)$

  $$g \in \mathbb{F}[x] \quad deg(g) = t$$

  $\Rightarrow$ *rozšířené* těleso $GF((2^m)^t)$
- **Podpora** L
  Náhodná permutace všech prvků z tělesa $\mathbb{F}$
- **Kontrolní matice** H

  $$H = VD$$

Binární Goppa kódy

Příklad

Ireducibilní *Goppův* polynom $g(x) = (001)x^2 + (100)x + (001)$ nad tělesem $GF(2^3)$ s ireducibilním polynomem 1011.

Příklad

Ireducibilní *Goppův* polynom $g(x) = (001)x^2 + (100)x + (001)$ nad tělesem $GF(2^3)$ s ireducibilním polynomem 1011.
Vygenerujeme podporu L:

$$L = (100, 001, 111, 011, 010, 000, 101, 110)$$

Binární Goppa kódy

Příklad

Ireducibilní *Goppův* polynom $g(x) = (001)x^2 + (100)x + (001)$ nad tělesem $GF(2^3)$ s ireducibilním polynomem 1011.
Vygenerujeme podporu L:

$$L = (100, 001, 111, 011, 010, 000, 101, 110)$$

Vandermondovu matice V a *diagonální* matice D:

$$V = \begin{pmatrix} 001 & 001 & 001 & \dots & 001 \\ 100 & 001 & 111 & \dots & 110 \end{pmatrix} \quad D = \begin{pmatrix} 001 & & & \\ & 111 & & \\ & & \ddots & \\ & & & 011 \end{pmatrix}$$

Binární Goppa kódy

Příklad

Ireducibilní *Goppův* polynom $g(x) = (001)x^2 + (100)x + (001)$ nad tělesem $GF(2^3)$ s ireducibilním polynomem 1011.
Vygenerujeme podporu L:

$$L = (100, 001, 111, 011, 010, 000, 101, 110)$$

Vandermondovu matice V a *diagonální* matice D:

$$V = \begin{pmatrix} 001 & 001 & 001 & \dots & 001 \\ 100 & 001 & 111 & \dots & 110 \end{pmatrix} \quad D = \begin{pmatrix} 001 & & & \\ & 111 & & \\ & & \ddots & \\ & & & 011 \end{pmatrix}$$

Vynásobením matic získáme *kontrolní* matici H (nad $GF(2^m)$):

$$H = VD = \begin{pmatrix} 001 & 111 & 110 & 110 & 011 & 001 & 111 & 011 \\ 100 & 111 & 100 & 001 & 110 & 000 & 110 & 001 \end{pmatrix}$$

Binární Goppa kódy

Dekódování

Binární Goppa kódy

Dekódování

Pattersonův algoritmus [10]

Binární Goppa kódy

Dekódování

Pattersonův algoritmus [10]

- Opravuje až t chyb
- Kritická místa:
  - Výpočet odmocniny
  - Hledání kořenů

Binární Goppa kódy

Dekódování

Pattersonův algoritmus [10]

- Opravuje až t chyb
- Kritická místa:
  - Výpočet odmocniny
  - Hledání kořenů
- Detaily v práci

Obsah

Kryptoanalýza systému McEliece

Známé útoky na McEliece

Kryptoanalýza systému McEliece

Známé útoky na McEliece

- Útoky na strukturu použitého kódu
    - *GRS* a další

Kryptoanalýza systému McEliece

Známé útoky na McEliece

- Útoky na strukturu použitého kódu
  - *GRS* a další
- Nalezení *kódového slova s nízkou vahou*
  - Canteaut a Chabaud [4]

Kryptoanalýza systému McEliece

Známé útoky na McEliece

- Útoky na strukturu použitého kódu
  - *GRS* a další
- Nalezení *kódového slova s nízkou vahou*
  - Canteaut a Chabaud [4]
- Útok s *informační množinou*, *Support Spliting Algorithm*, ...

Kryptoanalýza systému McEliece

Bezpečné parametry

Kryptosystém	Parametry	Míra bezpečnosti	Velikost klíče	Složitost šifr.	Složitost dešifr.
RSA	1024b modul	~ 80 b	1 kb	2^{30}	2^{30}
	2048b modul	~ 112 b	2 kb	2^{33}	2^{33}
	4096b modul	~ 145 b	4 kb	2^{36}	2^{36}
McEliece	$(2048, 1608, 40)$	~ 98 b	691 kb	2^{20}	2^{23}
	$(2048, 1278, 70)$	~ 110 b	961 kb	2^{20}	2^{24}
	$(4096, 2056, 170)$	~ 184 b	4096 kb	2^{22}	2^{26}

Tabulka: Porovnání *McEliece* a *RSA* dle [6, 9]

Obsah

Implementace

- Software *Wolfram Mathematica*
- Rozděleno do samostatných balíků

Implementace

- Software *Wolfram Mathematica*
- Rozděleno do samostatných balíků
- Implementováno:
  1. (Rozšířená) konečná tělesa
  2. Goppa kódy
  3. McEliece

Implementace

Příklad

Rozšířený Euklidův algoritmus pro výpočet *inverze* polynomu $(101)x^3 + (010)x^2 + (110)x + (111)$ *modulo* $(001)x^4 + (011)x^3 + (011)x^2 + (001)x + (011)$ (nad tělesem $GF(2^3)$ s ireducibilním polynomem 1101):

Implementace

Příklad

Rozšířený Euklidův algoritmus pro výpočet *inverze* polynomu $(101)x^3 + (010)x^2 + (110)x + (111)$ *modulo* $(001)x^4 + (011)x^3 + (011)x^2 + (001)x + (011)$ (nad tělesem $GF(2^3)$ s ireducibilním polynomem 1101):

Podíl	Zbytek	Koeficient
	(001)(011)(011)(001)(011)	(000)
	(101)(010)(110)(111)	(001)
(111)(000)	(110)(011)(011)	(111)(000)
(111)(001)	(001)(100)	(010)(111)(001)
(110)(001)	(111)	(001)(111)(110)(001)

Příklad

Rozšířený Euklidův algoritmus pro výpočet *inverze* polynomu $(101)x^3 + (010)x^2 + (110)x + (111)$ *modulo* $(001)x^4 + (011)x^3 + (011)x^2 + (001)x + (011)$ (nad tělesem $GF(2^3)$ s ireducibilním polynomem 1101):

Podíl	Zbytek	Koeficient
	(001)(011)(011)(001)(011)	(000)
	(101)(010)(110)(111)	(001)
(111)(000)	(110)(011)(011)	(111)(000)
(111)(001)	(001)(100)	(010)(111)(001)
(110)(001)	(111)	(001)(111)(110)(001)

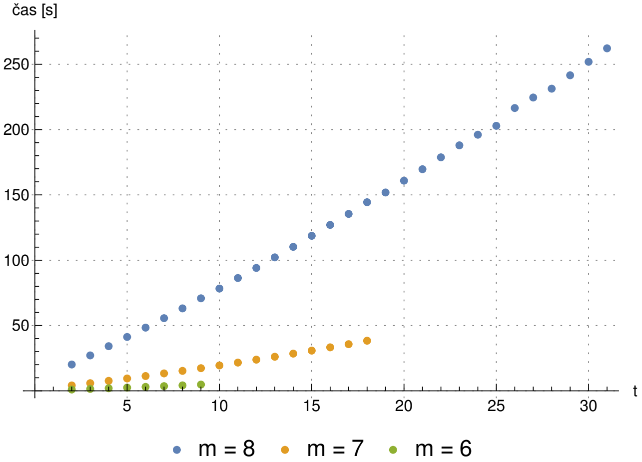
$\Rightarrow ((101)(010)(110)(111))^{-1} = (101)(001)(100)(101)$

Obsah

Výsledky měření

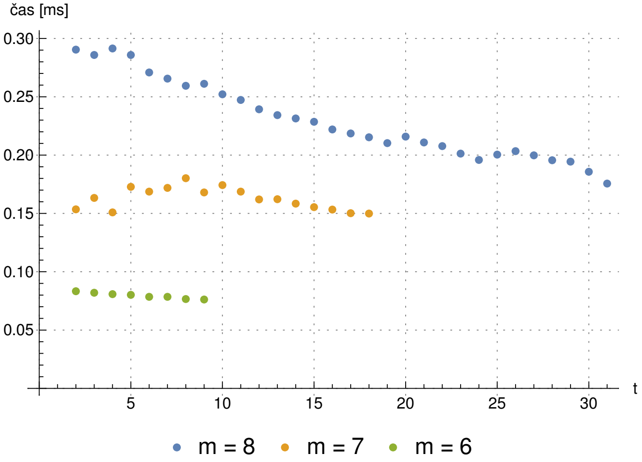
- Měření prováděno v *GPU laboratoři* (T9:350)
  - čtyřjádrový procesor Intel i5-6500, *3.2 GHz*
  - *16 GB RAM* DDR3

Výsledky měření



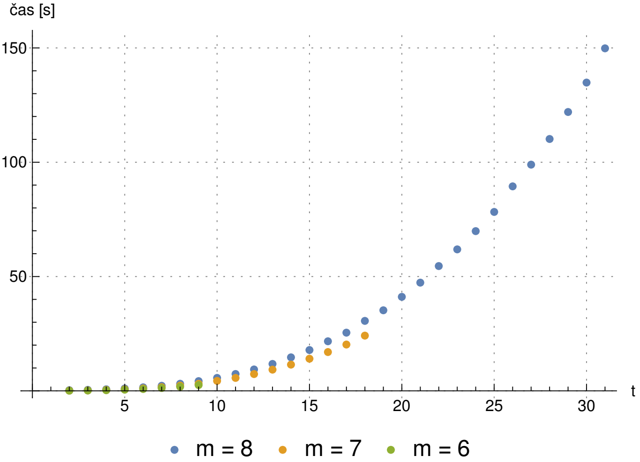
Obrázek: Závislost doby generování klíčů na parametru t

Výsledky měření



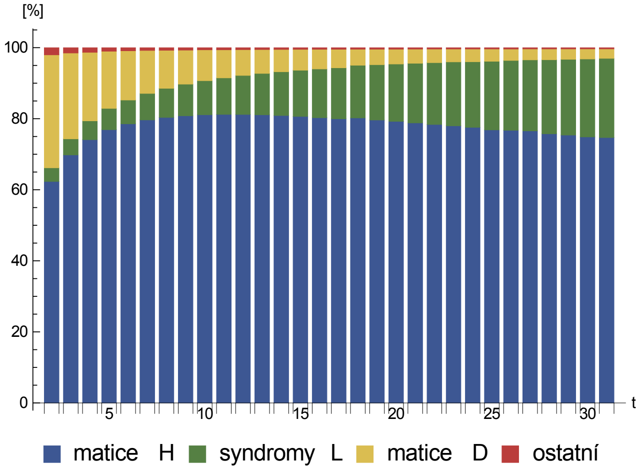
Obrázek: Závislost doby šifrování zprávy na parametru t

Výsledky měření



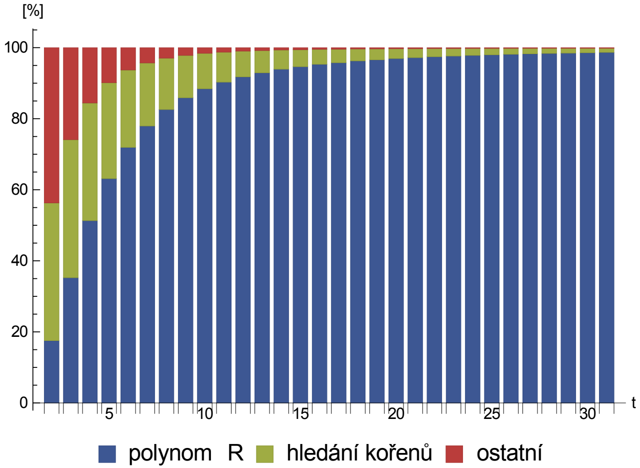
Obrázek: Závislost doby dešifrování zprávy na parametru t

Výsledky měření



Obrázek: Poměr významných částí výpočtu při generování klíčů v závislosti na parametru t (při $m = 8$)

Výsledky měření



Obrázek: Poměr významných částí výpočtu při dešifrování zprávy v závislosti na parametru t (při $m = 8$)

Obsah

Závěr

- Zadání splněno

Závěr

- Zadání splněno
- Hluboká rešerše kryptosystému nad rámec této prezentace
    - Kryptoanalýzy systému
    - Metody zkrácení klíčů
    - Slabiny a moderní varianty

Závěr

- Zadání splněno
- Hluboká rešerše kryptosystému nad rámec této prezentace
  - Kryptoanalýzy systému
  - Metody zkrácení klíčů
  - Slabiny a moderní varianty
- Ukázková implementace

Závěr

- Zadání splněno
- Hluboká rešerše kryptosystému nad rámec této prezentace
  - Kryptoanalýzy systému
  - Metody zkrácení klíčů
  - Slabiny a moderní varianty
- Ukázková implementace
- Experimentálně ověřené složitosti
  - Kritické části výpočtu

Obsah

Otázky oponenta

Otázky oponenta

1. Má vůbec smysl hledat úsporu prostoru u soukromého klíče, i s ohledem na vaše tvrzení, že kapacity disků jsou téměř neomezené a limit je primárně v přenosu veřejného klíče?

Otázky oponenta

1. Má vůbec smysl hledat úsporu prostoru u soukromého klíče, i s ohledem na vaše tvrzení, že kapacity disků jsou téměř neomezené a limit je primárně v přenosu veřejného klíče?
2. Zkoušel jste změřit dobu generování klíčů, šifrování a dešifrování pro bezpečné parametry, tedy např. $m = 12$, $t = 41$?

Otázky oponenta

1. Velikosti klíče

Otázky oponenta

1. Velikosti klíče

1. Veřejný klíč
   - Přenos klíče

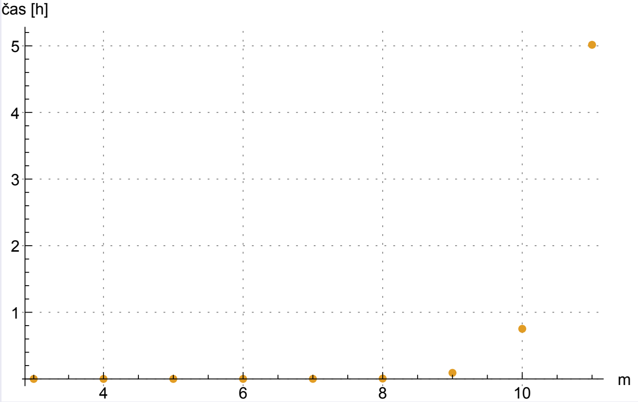
Otázky oponenta

1. Velikosti klíče

1. Veřejný klíč
   - Přenos klíče
2. Soukromý klíč
   - Vestavěné systémy

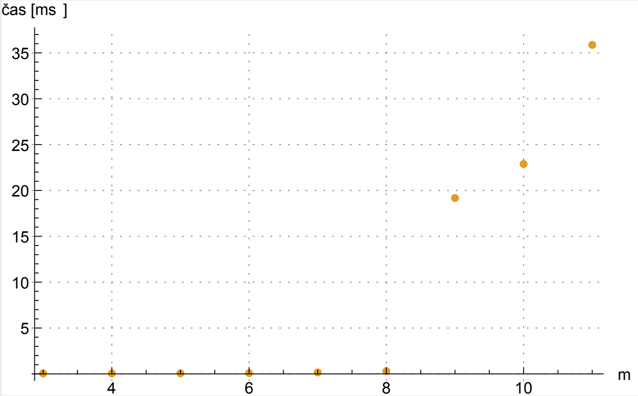
Otázky oponenta

2. Rozumné parametry



Obrázek: Závislost doby generování klíčů na parametru m

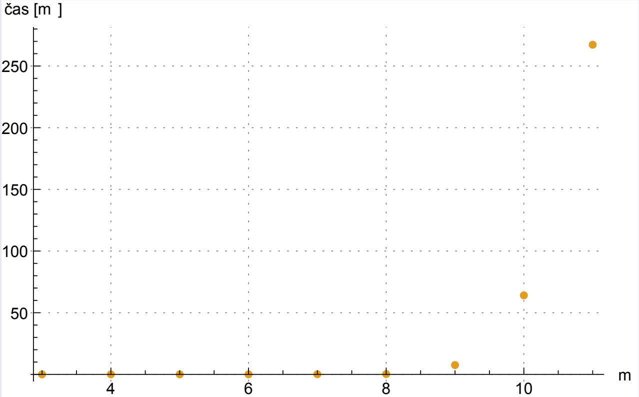
2. Rozumné parametry



Obrázek: Závislost doby šifrování zprávy na parametru m

Otázky oponenta

2. Rozumné parametry



Obrázek: Závislost doby dešifrování zprávy na parametru m

Reference I

[1] Robert J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory v *JPL Deep Space Network Progress Report*, strany 114-116. 1978. Dostupné online
http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF

[2] Elwyn R. Berlekamp, Robert J. McEliece, Henk C. A. van Tilborg. On the Inherent Intractibility v *IEEE Transactions of Information Theory*, vol. IT-24, No. 3, strany 384-386. IEEE, květen 1978.

[3] Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen. *Post-Quantum Cryptography*. ISBN 978-3-540-88701-0. Springer Berlin Heidelberg, 2009.

Reference II

[4] Anne Canteaut, Florent Chabaud. Improvements of the Attacks on Cryptosystems Based on Error-Correcting Codes v *Research Report LIENS-95-21*. École Normale Supérieure, 1995 Dostupné online http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.1645

[5] Nicolas T. Courtois, Matthieu Finiasz, Nicolas Sendrier. How to Achieve a McEliece-Based Digital Signature Scheme v *Advances in Cryptology – ASIACRYPT 2001*, strany 157-174. Springer Berlin Heidelberg, 2001. Dostupné online http://link.springer.com/chapter/10.1007%2F3-540-45682-1_10

[6] Daniela Engelbert, Raphael Overbeck, Arthur Schmidt. A Summary of McEliece-Type Cryptosystems and their Security v *Journal of Mathematical Cryptology*. IACR 2006. Dostupné online http://eprint.iacr.org/2006/162

[7] Valery D. GOPPA. A New Class of Linear Correcting Codes v *Problemy Peredachi Informatsii*, vol. 6, strany 24-30. 1970.

[8] Harald NIEDERREITER. Knapsack-type cryptosystems and algebraic coding theory v *Problems of Control and Information Theory 15*, strany 19-34. 1986

[9] Christof PAAR, Jan PELZL. *Understanding Cryptography*: A Textbook for Students and Practitioners. Springer-Verlag Berlin Heidelberg, 2010. Dostupné online: `https://www.springer.com/us/book/9783642041006`

[10] Nicholas J. PATTERSON, The algebraic decoding of Goppa codes v *IEEE Transactions on Information Theory*, vol. 21, strany 203-207. IEEE 1975. Dostupné online `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1055350`

Reference IV

[11] J. M. Schanck, W. Whyte, Z. Zhang. Criteria for selection of public-key cryptographic algorithms for quantum-safe hybrid cryptography (Internet-draft). IETF, 2016. Dostupné online https://datatracker.ietf.org/doc/draft-whyte-select-pkc-qsh/