SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND
INFORMATION TECHNOLOGY

Registration number: FEI-5382-5982

SECURE IMPLEMENTATION OF MCELICE CRYPTOSYSTEM

DIPLOMA THESIS

2015 **Martin Orem**

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Registration number: FEI-5382-5982

SECURE IMPLEMENTATION OF MCELICE CRYPTOSYSTEM

DIPLOMA THESIS

Study Programme:	Applied Informatics
Field Number:	2511
Study Field:	9.2.9 Applied Informatics
Training Workplace:	Institute of Computer Science and Mathematics
Supervisor:	doc. Ing. Pavol Zajac, PhD.
Consultant:	Ing. František Uhrecký

Bratislava 2015 **Martin Orem**

Slovenská technická univerzita v Bratislave
Ústav informatiky a matematiky

Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013
Evidenčné číslo: FEI-5382-5982

STU
FEI

ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Michal Ližičiar**
ID študenta: 5982
Študijný program: Aplikovaná informatika
Študijný odbor: 9.2.9 aplikovaná informatika
Vedúci práce: Ing. Matúš Jókay, PhD.

Názov práce: **Anonymizácia internetového prístupu**

Špecifikácia zadania:

Cieľom práce je vytvoriť zásuvný modul pre internetový prehliadač, ktorý bude schopný buď náhodne alebo selektívne meniť informácie používané na identifikáciu používateľa pri jeho prístupe na cieľový server.

Úlohy:
1. Analyzujte dostupnosť a funkčnosť podobných modulov.
2. Analyzujte informácie používané na identifikáciu používateľa pri prístupe na stránku.
3. Navrhnite, implementujte a otestujte anonymizačný modul pre zvolený internetový prehliadač.

Zoznam odbornej literatúry:

1. YARDLEY, G. Better Privacy. [online]. 2012. URL: http://nc.ddns.us/BetterPrivacy/BetterPrivacy.htm.
2. ECKERSLEY, P. A Primer on Information Theory and Privacy. [online]. 2010. URL: https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy.

Riešenie zadania práce od: 24. 09. 2012
Dátum odovzdania práce: 24. 05. 2013

L. S.

SLOVENSKÁ TECHNICKÁ UNIVERZITA
Fakulta elektrotechniky a informatiky
Ústav
informatiky
a matematiky
812 19 BRATISLAVA

Michal Ližičiar
študent

prof. RNDr. Otokar Grošek, PhD.
vedúci pracoviska

prof. RNDr. Gabriel Juhás, PhD.
garant študijného programu

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Aplikovaná informatika
Autor:	Martin Orem
Diplomová práca:	Bezpečná implementácia McEliece kryptosystému
Vedúci záverečnej práce:	doc. Ing. Pavol Zajac, PhD.
Konzultant:	Ing. František Uhrecký
Miesto a rok predloženia práce:	Bratislava 2015

Kryptografia poskytuje v súčastnosti bezpečne komunikovať nielen na internete. Dnešná kryptografia je prevažné založená na problémoch, ktoré môžu byž riešené pomocou Shorovho algoritmu s použitím kvantového počítača. Preto existencia kvantového počítača, ktorý by bol schopný spracovať niekoľko tisíc bitov naraz, by spôsobila skutočný problém. Táto diplomová práca sa venuje zlepšeniu bezpečnosti kryptografickej knižnice, ktorá by mala byť voči útoku kvantovým počítačom odolná. Začiatok práce je venovaný potrebným informáciám pre pochopenie McElice kryptosystému. V ďalšej sekcii je analyzovaná kryptografická knižnica spolu s CCA a KEM. V tretej sekcii je možné nájsť návrh nášho zlepšenia knižnice. Nasledujúce sekcie sa venujú samotnej implementácií s prisluchajúcimi testami. V závere popisujeme dosiahnuté výsledky.

Kľúčové slová: kryptografia, KEM, McElice kryptosystém

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Author:	Martin Orem
Diploma Thesis:	Secure implementation of Mcelice Cryptosystem
Supervisor:	doc. Ing. Pavol Zajac, PhD.
Consultant:	Ing. František Uhrecký
Place and year of submission:	Bratislava 2015

Crystography provides a secure way of communication in today's world. Current cryptography is mainly based on problems, which could be solved by Shor's algorithm by using a quantum computer. Therefore, the existence of Quantum Computers in the range of 1000 bits would be a real world threat. This thesis is dedicated to improvement of cryptographic library, that should be resistant against such threats. The thesis contains an overview of McElice cryptosystem and it's security against CCA attacks. The first section discuss necessary preliminaries. In the further section, current state of the library is analysed. The section also contains an analysis of missing features like secure CCA2 conversion, KEM. The other sections of the thesis are dedicated to our contribution and implementation with corresponding tests. In the end of the thesis, accomplished results are discussed.

Keywords: crypto, McElice, KEM, CCA

Declaration

I declare that I developed this diploma thesis myself with listed references.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, doc. Ing. Pavol Zajac PhD., for his support and guidance.

Contents

List of Figures and Tables

List of Abbreviations and Symbols

WWW - World Wide Web

List of Algorithm

Introduction

TODO: v skratke, co chceme dosiahnut...

In today's world, cryptography has found its place in many areas of ordinary living and it is important to understand its limits. We have learnt the meaning of the word privacy, especially on the internet. We distribute private information on web sites, we save confidential information on our computers, we pay through internet banking believing and expecting that all of this is adequately defended. Quantum computer may be a breakthrough in current cryptography, regarding its enormous power.

On of the best candidates for post-quantum cryptography comes from times back to the beginning of modern cryptography. McElice has proposed cryptosystem based on error code problem, which was proven to be NP hard. In recent years, dedicated to implementation of cryptographic library called Bitpunch. The project was a part of programme called: "Secure implementation of post-quantum cryptography", NATO Science for Peace and Security Programme Project Number: 984520.

In the first section, we discuss needed preliminaries, that are necessary to understand the McEliece cryptosystem. The further section is dedicated to basic analysis of Bitpunch and its CCA security. We also provide an overview of missing key encapsulation management. The third section presents our contribution to this project. The fifth section shows basic conducted test of the implementation. In the end of this thesis, we conclude accomplished results.

1 Preliminaries

In this section, we introduce some important preliminary results, which are required throughout the thesis. We present notation, a short overview of McElice PKC, CCA security and the introduction to key encapsulation.

1.1 Notation

If x is a string, then $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If S is a set then $s \Leftarrow S$ denotes the operation of picking an element s of S uniformly at random. If $k \in N$ then 1^k denotes the string of k ones. If P is a matrix, then P^{-1} denotes an inverse matrix to the matrix P.

1.2 McElice cryptosystem

McElice cryptosystem was proposed by R. J.McEliece in 1978. [1] The cryptosystem belongs to a public key cryptography family of protocols, it basically means that two different keys are used for encryption and decryption. McElice PKC provides fast way of encryption and decryption process of messages as was proven by ... [referencia] The original construction of MECS is based on Goppa codes, which seems to be secure with the right choice of parameters. Goppa codes are well suited for cryptographic application due to its generator matrix, which is hardly to be distinguished from a random binary matrix and also for their high error-correcting capabilities. Many other variants of the cryptosystem using various linear codes have been proposed over the years, but the most of them were subsequently proven to be insecure by researchers. [referencia] MECS is considered to be one of the best candidate for post-quantum cryptography, which causes a growing interest in implementation and further cryptoanalysis. Security of MECS relies on a decoding problem, which is known to be NP-hard. [referencia] The most important part of secret key is the description of the structured linear code, created by an irreducible polynom in key generation process. An efficient decoding algorithm for the chosen linear-code is required for successful decryption of messages. It is clear, that the knowing the structure of the underlying linear code provides a way for fast decryption. McElice cryptosystem takes an advantage from randomness. A public key is "permuted" and "varied" form of chosen-linear code (secret key), which should be hardly distinguished from a completely random linear code. We can formally define McElice cryptosystem by describing algorithms $\Pi = (Gen, Enc, Dec)$ as follows.

Key generation algorithm In order to generate keys, we have to define a Goppa code, which is created over irreducible polynomial. The output of this algorithm is:

- Private key - S, G, P (random singular matrix, generation matrix, permutation matrix).
- Public key - $\widehat{G}, t$ (masked generation matrix and number of error, which is capable to correct up to t errors).

Algorithm 1 Key generation

```
1. Pick a random irreducible polynom $g$ over $GF(2^m)$ of degree $t$,
2. Compute a $k \times n$ generation matrix $G$, of goppa code $\Gamma = (\alpha_1, ..., \alpha_n, g)$, with dimension $k = n - td$,
3. Generate a random $k \times k$ singular matrix $S$,
4. Generate a random $n \times n$ permutation matrix $P$,
5. Compute a $k \times n$ matrix $\widehat{G} = SGP$,
6. Public key is pair of $(\widehat{G}, t)$, where $t$ is maximum of corrected errors, private key is consisted of $S, G, P$.
```

Algorithm 2 Encryption

```
1. Let $m$ be a $k$-bit message, and let $e$ be an random $n$-bit vector with $wH(e) \leq t$. Then $c = m \cdot \widehat{G} + e$ is a ciphertext.
```

Algorithm 3 Decryption

```
1. Calculate $S^{-1}$ and $P^{-1}$,
2. Calculate $\mathbf{c'} = \mathbf{c}P^{-1}$,
3. Apply proper decoding algorithm $Dec$ of the code $\Gamma$ to decode $\mathbf{c'}$ to $\widehat{\mathbf{m}}$,
4. Obtain $m$ by $\mathbf{m} = \widehat{\mathbf{m}}S^{-1}$.
```

Note that a linear code, which is part of public key G is permutation-equivalent to the chosen secret key. As was mentioned above, the original construction in [25] uses irreducible binary Goppa codes, for which an efficient decoding algorithm was presented by Patterson [30]. In order to apply Patterson's algorithm, the polynomial generating the Goppa code must be known. Considering this fact, the chosen polynomial is considered to be a part of secret key, due to the fact that describes a structure of underlying linear code.

1.3 CCA security

In order to define IND-CCA, we need to consider following experiment with a private-key encryption scheme Π = (Gen, Enc, Dec), adversary A, and value n for the security parameter.

Algorithm 4 The CCA experiment

1. A random key k is generated by running $Gen(1^n)$.
2. The adversary A is given input 1^n and oracle access to encryption and decryption. It outputs a pair of messages m_0, m_1 s.t $|m_0| = |m_1|$.
3. A random bit $b \leftarrow \{0, 1\}$ is chosen, and then a cipher text is computed and subsequently given to A. We call c the challenge.
4. Adversary continues to have an access to oracle encryption and decryption, however it is not allowed to query later challenge itself.
5. The output of the experiment if 1 if $b' = b$, otherwise 0.

A private-key encryption scheme Π is CCA-secure if for all probabilistic polynomial-time adversaries A there exists a negligible function *negl* such that:

$$Pr[PrivK_{A,\Pi}(n) = 1] \leq \frac{1}{2} + negl(n)$$

where the probability is taken over all random coins used in the experiment.
The formal definition of CCA security is discussed in more detail at [modern crypt].

1.4 Key encapsulation

A key-encapsulation mechanism $KEM = (Gen, Enc, Dec)$ consists of three polynomial-time algorithms.

- Randomized algorithm Gen for generation of key pair for security parameter $k \in N$ produces $(priv, pub) \Leftarrow Gen(1^k)$,
- A pair $(K, C) \Leftarrow Enc(pub)$ is provided by randomized encapsulation algorithm, where $K \in K(k)$ is uniformly distributed symmetric key and C is a ciphertext,
- Decryption algorithm provides a way to extract original key $K \leftarrow Dec(priv, C)$

Obviously, we estimate the decryption to be deterministic in such a way that $\forall k \in N$ and $\forall (K, C) \Leftarrow Enc(pub)$ we have probability $Pr[Dec(priv, C) = K] = 1$. One of the most common requirement of KEM is CCA indistinguishability, which is discussed more in detail at[referenca na clanok 418]. TODO: dostudovat

2 Analysis

2.1 Bitpunch library

In this section, we present current status of Bitpunch library. In other years, the project called Bitpunch implemented very lightweight crypto library in C language. Following list shows already implemented features of the library, which are points of our interests:

- modular architecture,
- variation of Pointcheval conversion,
- simple testing environment

2.1.1 Modular architecture

Modular architecture of the library provides easy extendibility. The modules are interconnected via relationships, which are depicted on Figure 1. The simplicity of the library itself allows us to implement another CCA conversion.

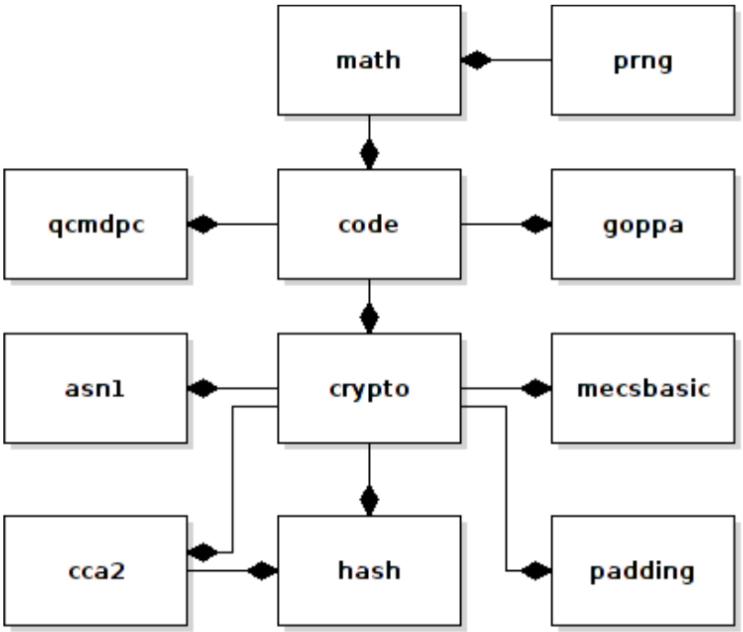

Figure 1: Modular architecture of Bitpunch.

- asn1 - provides import/export of keys using libtasn1, code - provides implementation of used codes (Goppa, QC-MDPC [reff]),

- crypto:
    - cca2 - implementation of CCA2 conversions,
    - hash - implemented hash from PolarSSL
    - mecsbasic - implementation of basic McElice PKC,
    - padding - implementation of padding,
- math - provides arithmetic functions over $Gf(2^m)$, etc.
- prng - API for a pseudo random generator

As we consider the architecture to be ....TODO: treba popisat, ze libka je dobre rozdelena na moduly a nebudeme to menit

2.1.2 CCA conversions

In this section, we discuss CCA conversions, which should provide indistinguishability under adaptive chosen-ciphertext attacks.

The original version of MECS is not secure in theoretical way. In practice, a PKC can be considered secure to use if it provides IND-CCA security. The current version of Bitpunch includes a version of Pointcheval conversion. An input of the conversion is a plaintext.

Algorithm 5 The Pointcheval CCA2 conversion

1. An input of MECS is $\widehat{m} = r_1||hash(m||r_2)$, r_1 is a random bit vector with lenght $k - l$ and output of hash function is indistinguishable from random l-bit vector,
2. A ciphertext consists of $(c_1, c_2, c_3) = (\widehat{m}\widehat{G} + e, hash(r_1) + m, hash(e) + r_2)$.

Variant of Pointcheval CCA2 conversion This version of Pointecheval conversion is susceptible to the following attack. TODO: treba dopisat utok
As it was discussed at [Komara and Imai], the original version of MECS is susceptible to partially-known plaintext, related plaintext, reaction attacks and also does not provide security against malleability. However, Kobara and Imai presented two slightly different CCA conversion of MECS, which are considered to be secure against already mentioned attacks. These conversions require the use of random oracles (hash functions) [2] to randomize the input and break relations of the plaintext and ciphertext. Unfortunately, conversions cause an overhand since some redundant data are necessary. As is depicted on

the figure [referencie], an algorithm for encryption is straightforward and can be described as follow: The input of this process is a public key (G', t), a message m and constant $Const$, which sum of its $(m||Const)$ length. The output of encryption is a ciphertext. The decryption process is described respectively by following algorithms.

Algorithm 6 Kobara-Imai CCA2 gamma conversion - encryption

```
1. Let $m$ be a message, a constant $Const$ and a random $r$-bit number $Rand$,
2. Calculate $y_1 = Prng(Rand) + (m||Const)$,
3. Calculate $y_2 = HASH(y_1) + Rand$,
4. Let $y_1 = y_1'||y_4||y_3$,
5. Let $y_5 = y_2||y_1'$,
6. Encode error vector $e = CONV(y_4)$,
7. Encrypt $y_3$ using MECS such that $y_6 = MECS_{enc}(e, y_3) = y_3G' + CONV(y_4)$,
8. A ciphertext is $c = y_5||y_6$.
```

Algorithm 7 Kobara-Imai CCA2 gamma conversion - decryption

```
1. Let $c$ be the received message, $c = y_5||y_6$ and the constant $Const$,
2. Decrypt $y_6$ using MECS $(e, y_3) = MECS_{dec}(y_6)$, where $e$ is the error vector,
3. Decode $y_4 = CONV^{-1}(e)$,
4. Let $y_5 = y_2||y_1'$, we reconstruct $y_2$ and $y_1$ such that $y_1 = y_1'||y_4||y_3$,
5. Calculate $Rand = y_2 + HASH(y_1)$,
6. Calculate $(m||Const') = PRNG(Rand) + y_1$,
7. If $Const'$ is equal to $Const$, the integrity of message is not corrupted,
8. The plaintext is $m$.
```

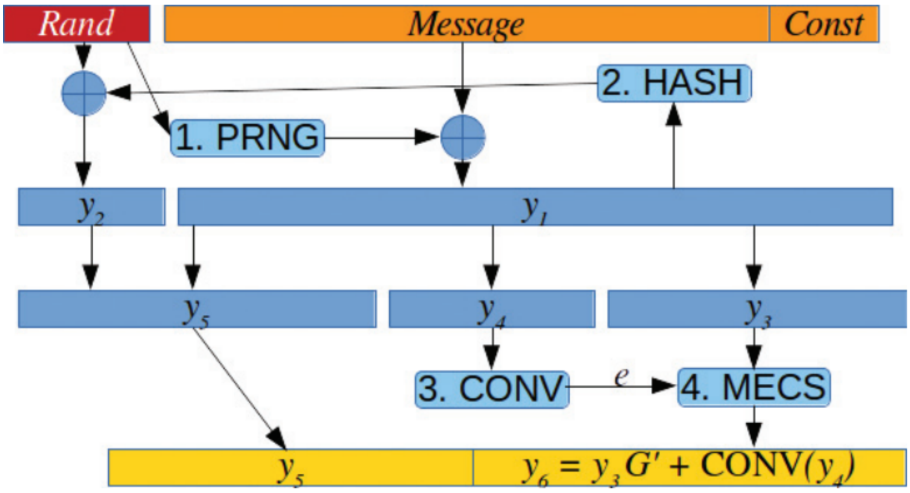

Figure 2: An illustration of the Kobara-Imai CCA-2 secure conversion of MECS.

- *Rand* is a random session key (with size given by the required security, level for symmetric encryption).
- $PRNG$ is a cryptographically secure pseudorandom generator (can be a symmetric encryption algorithm in practice).
- $Const$ is a public constant (with size given by the required security level for authentication).,
- $HASH$ is a cryptographically secure hash function.
- $CONV$ is a bijection function, which converts an integer into the corresponding error vector.
- $MECS$ is a standard MECS routine (with e as an additional input).

3 Our contribution

Algorithm 8 algoritmu

```
/* Hello World program */

#include<stdio.h>

struct cpu_info {
    long unsigned utime, ntime, stime, itime;
    long unsigned iowtime, irqtime, sirqtime;
};

main()
{
    printf("Hello World");
}
```

Conclusion

Cieľom práce bola analýza anonymizačných modulov, identifikačných prvkov prehliadača a vytvorenie anonymizačného modulu pre internetový prehliadač.

Analýzou najpoužívanejších modulov a vlastností prehliadača, ktoré slúžia na identifikáciu používateľa, sme zistili aktuálny stav a funkcionalitu rozšírení, ktorými je možné anonymizovať prístup na internet. Väčšina týchto rozšírení modifikuje len časť vlastností prehliadača, ktoré sú odosielané na server, alebo úplne blokuje ich odosielanie. Nami vytvorené rozšírenie dokáže modifikovať väčšinu identifikačných prvkov rozšírenia, pričom dodržiava súvislosti medzi vlastnosťami (používateľský agent odosielaný v hlavičke dopytu je totožný s používateľským agentom zisťovaním pomocou JavaScript príkazu, súvislosť medzi šírkou a dĺžkou rozšírenia obrazovky). Dokáže blokovať údaje, ktoré sú posielané v otvorenej podobe na server a obsahujú informácie o identifikačných údajoch prehliadača, ktoré sa nedajú na úrovni rozšírení modifikovať.

Testovanie rozšírenia nám overilo funkčnosť a správnosť implementácie. Rozšírenie dokáže buď vždy, alebo v časových intervaloch modifikovať väčšinu charakteristických prvkov prehliadača odsielaných na server, a tým zvyšuje anonymitu používateľa.

Resumé

Ciel'om práce bolo zmapovat' súčasný stav v oblasti...

Bibliography

[1] McEliece, R. J. A public-key cryptosystem based on algebraic coding theory. *DSN progress report 42*, 44 (1978), 114–116.

Appendix

A Structure of attached medium

\

\Bakalarska_praca.pdf

\FEIk_Identuty.xpi

\FEIkIdentity

\FEIkIdentity\chrome.manifest

\FEIkIdentity\install.rdf

\FEIkIdemtity\content

\FEIkIdemtity\content \function.js

\FEIkIdemtity\content \options.xul

\FEIkIdemtity\content \overlay.xul

\FEIkIdemtity\content \window.js

\FEIkIdemtity\content \window.xul

\FEIkIdemtity\defaults

\FEIkIdemtity\defaults\preferences

\FEIkIdemtity\defaults\preferences \prefs.js

\FEIkIdemtity\locale

\FEIkIdemtity\locale \sk-SK

\FEIkIdemtity\locale \sk-SK\options.dtd

\FEIkIdemtity\locale \sk-SK\window.dtd

\FEIkIdemtity\skin

B Algoritmus

Algorithm B.1 Ukážka algoritmu

```c
/* Hello World program */

#include<stdio.h>

struct cpu_info {
    long unsigned utime, ntime, stime, itime;
    long unsigned iowtime, irqtime, sirqtime;
};

main()
{
    printf("Hello World");
}
```