A CCA2 Secure PKE Based on McEliece Assumptions in the Standard Model

Jörn Müller-Quade
European Institute for System Security
KIT, Karlsruhe, Germany
04/23/09 | Session ID: CRYP301

Session Classification: Alternative Encryption
Agenda

The McEliece Crypto System

An Attack and IND-CCA2

A First Step: Achieving IND-CCA1

Achieving IND-CCA2
The McEliece Crypto System
The McEliece Public Key Scheme

McEliece Assumption: 1. A public key $G'$ is indistinguishable from random  
2. Syndrome decoding is hard
The McEliece Public Key Scheme

Secret Key

Public Key

Encryption

Decryption

McEliece Assumption: 1. A public key $G'$ is indistinguishable from random
2. Syndrome decoding is hard
Semantic Security (IND-CPA)

Public key $G'$

$A \xrightarrow{\text{challenge}_1, \text{challenge}_2}$

$\text{Enc}_{G'}(\text{challenge}_i)$

Chosen at random

Public key $G'$

$A \xrightarrow{\text{Decision}}$

Kartesiuhe Institute of Technology

RSACONFERENCE2009
Semantic Security (IND-CPA)

\[ \text{Public key } \ G' \]

\[ \text{Enc}_{G'}(\text{challenge}_i) \]

Chosen at random

\[ \text{A cannot distinguish better than guessing} \]

\[ \text{A} \]

Decision

\[ \text{Public key } \ G' \]
Semantic Security (IND-CPA)

Padding the message with a random string yields a semantically secure public key scheme (IND-CPA).

[Nojima, Imai, Kobara, Morozov, DCC 2008]

Diagram:
- $r || m$ (random padding)
- $G'$ (message padding)
- $e$ (encryption)

Random padding
An Attack and IND-CCA2
A Reaction Attack

The password identifies the customer and the nonce avoids replay attacks.

We use text book McEliece to securely send the password and the nonce.
A Reaction Attack

Replay an old $\text{Enc}_\text{VM}(\text{pwd}||\text{nonce})$

Add a new error

Vending Machine
Pay by sending your password and a nonce

- Password not valid
- Nonce not new
A Reaction Attack

Replay an old $Enc_{VM}(pwd||nonce)$

Add a new error

Depending on the message the new error corrected an error from the encryption.
A Reaction Attack

Replay an old $\text{Enc}_{\text{VM}}(\text{pwd}||\text{nonce})$

Add a new error

Vending Machine
Pay by sending your password and a nonce

Locate all the errors from the encryption. Decode the codeword. Extract $\text{pwd}$. Depending on the message the new error corrected an error from the encryption.

- Password not valid
- Nonce not new
IND-CCA2

A

Decryption Oracle

Public key $e$

A

challenge$_1$, challenge$_2$

Enc$_e$(challenge$_i$)

Chosen at random

A

Decision

Enc$_e$(challenge$_i$)

Chosen at random

A

Decision

Public key $e$
IND-CCA2

Decryption Oracle

Public key $e$

$A$

Encryption Oracle

Decryption Oracle

Public key $e$

$A$

$A$ cannot distinguish better than guessing

$Enc_e(challenge_i)$

Chosen at random

$A$

Decision
IND-CCA2

- A decryption oracle yields more information than the reaction of the vending machine. IND-CCA2 security implies security against reaction attacks.

- IND-CCA2 is the strongest established security notion for PKE and it implies universal composability.
A First Step: achieving IND-CCA1
A First Step: IND-CCA1

Decryption Oracle

Public key $e$

$A$ \rightarrow challenge_1, challenge_2$

$Enc_e(\text{challenge}_i)$

Chosen at random

$A$ \rightarrow Decision
A First Step: IND-CCA1

A cannot distinguish better than guessing

Public key $e$

Decryption Oracle

$A$

Public key $e$

Decryption Oracle

$A$

Chosen at random

$\text{Enc}_e(\text{challenge}_i)$

Decision
For a PKE \((\text{KeyGen}, \text{Enc}, \text{Dec})\) we define the \(k\)-repetition \((\text{KeyGen}_k, \text{Enc}_k, \text{Dec}_k)\) as follows:

- \text{KeyGen}_k \) calls \text{KeyGen} \(k\) times.
- \text{Enc}_k \) encrypts with \text{Enc} \(k\) times using the corresponding public keys and correlated randomness.
- \text{Dec}_k \) decrypts all cipher texts with the corresponding secret keys and outputs \(m\) if all individual cipher texts did decrypt to \(m\).
- \text{Soundness}: \text{Dec}_k(\text{Enc}_k(m)) = m\) with overwhelming probability.

This construction is derived from [Rosen, Segev, TCC 2009]
\[ k \]-Repetition McEliece

- \( \text{KeyGen}_k \) calls \( \text{KeyGen} \) \( k \) times
- \( \text{Enc}_k \) encrypts with \( \text{Enc} \) \( k \) times using independent random error vectors, but the same padding \( r\|m \) in all encryptions.
- \( \text{Dec}_k \) decrypts all cipher texts with the corresponding secret keys and outputs \( m \) if all individual cipher texts did decrypt to \( m \).
- Soundness: \( \text{Dec}_k(\text{Enc}_k(m)) = m \) with overwhelming probability.

- This randomized \( k \)-repetition McEliece remains IND-CPA secure.
Verifiability of $k$-Repetition

- A $k$-repetition PKE is called verifiable if given a ciphertext $c$, the public key $pk = (pk_1, \ldots, pk_k)$ and any secret key $sk_i$ it is possible to verify if $c$ is a valid ciphertext (i.e., all individual ciphertexts decrypt to the same message).

\[
c = (c_1, \ldots, c_i, \ldots, c_k)\]

\[
\text{Dec} \quad \text{???
}\]

\[
m \quad \text{Dec}_k \quad m
\]
Verifiability of $k$-Repetition

- $k$-repetition McEliece is verifiable:

\[ c = (c_1, \ldots, c_i, \ldots, c_k) \]

Encode with public key matrix and check the hamming distance

\[ Dec \]

\[ m \]

\[ c = (c_1, \ldots, c_i, \ldots, c_k) \]

\[ Dec_k \]

\[ m \]
From $k$-Repetition to IND-CCA1

- **KeyGen$_{CCA1}$** gives $(pk_1^0, pk_1^1, \ldots, pk_k^0, pk_k^1)$ and the secret key $(sk_1^{r_1}, sk_2^{r_2}, \ldots, sk_k^{r_k})$

- **Enc$_{CCA1}$** chooses a random string $s = s_1, \ldots, s_k$, encrypts ($k$-repetition) with $pk_1^{s_1}, pk_1^{s_2}, \ldots, pk_k^{s_k}$, obtains $c = c_1, \ldots, c_k$ and outputs $(s, c)$.

- **Dec$_{CCA1}$** decrypts whenever a $s_i = r_i$ with $sk_{r_i}$ obtains $m$ and outputs $m$ if the ciphertext was valid (verifiability).

This construction follows closely [Rosen, Segev, TCC 2009]
From $k$-Repetition to IND-CCA1

- $\text{KeyGen}_{\text{CCA1}}$ gives $(pk_1^0, pk_1^1, \ldots, pk_k^0, pk_k^1)$ and the secret key $(sk_1^{r_1}, sk_2^{r_2}, \ldots, sk_k^{r_k})$

- $\text{Enc}_{\text{CCA1}}$ chooses a random string $s = s_1, \ldots, s_k$, encrypts ($k$-repetition) with $pk_1^{s_1}, pk_1^{s_2}, \ldots, pk_k^{s_k}$ obtains $c = c_1, \ldots, c_k$ and outputs $(s, c)$.

- $\text{Dec}_{\text{CCA1}}$ decrypts whenever a $s_i = r_i$ with $sk_{r_i}$ obtains $m$ and outputs $m$ if the ciphertext was valid (verifiability).

If $s$ is the complement of the string $r$ it is impossible to decode.
From $k$-Repetition to IND-CCA1

If $s$ is the complement of the string $r$ it is impossible to decode.

Given a successful IND-CCA1 adversary $A$ for $\text{PKE}_{\text{CCA1}}$ we construct an adversary $A'$ breaking the IND-CPA security of $k$-repetition $\text{PKE}_k$:

$A'$ obtains $(pk_1, \ldots, pk_k)$

$A'$ chooses $(pk_1^0, pk_1^1, \ldots, pk_k^0, pk_k^1)$ and $(sk_1^r, sk_2^r, \ldots, sk_k^r)$ then every cipher $(s, c)$ can be deciphered by $A'$, except for $s = \text{complement}(r)$. $A'$ can mimic the oracle for $A$.

$A'$ gives $(pk_1^0, pk_1^1, \ldots, pk_k^0, pk_k^1)$ to $A$. 
From \( k \)-Repetition to IND-CCA1

If \( s \) is the complement of the string \( r \) it is impossible to decode.

Given a successful IND-CCA1 adversary \( A \) for PKE\(_{\text{CCA1}}\) we construct an adversary \( A' \) breaking the IND-CPA security of \( k \)-repetition PKE\(_k\):

- Given \( (pk_1^0,pk_1^1,...,pk_k^0,pk_k^1) \) \( A \) outputs \( m_1, m_2 \)
- \( A' \) outputs \( m_1, m_2 \) and obtains a challenge \( c \).
- \( A' \) gives \((\text{complement}(r), c)\) as a challenge to \( A \).
- \( A \) outputs a decision \( d \).
- \( A' \) outputs this decision \( d \).
Achieving IND-CCA2
This scheme is not yet IND-CCA2 secure: After obtaining the challenge \((s, c)\) just change a bit in \(s\) and give this to the decryption oracle.

- **KeyGen\textsubscript{CCA2}** gives \((pk_1^0, pk_1^1, \ldots, pk_k^0, pk_k^1)\) chooses a signing key and \(r = r_1, \ldots, r_k\) as the complement of a the corresponding signature verification key. The secret key is chosen as \((sk_1^r, sk_2^r, \ldots, sk_k^r)\).
From $k$-Repetition to IND-CCA2

This scheme is not yet IND-CCA2 secure:
After obtaining the challenge $(s,c)$ just change a bit in $s$ and give this to the decryption oracle.

- $Enc_{CCA2}$ chooses a signing key $sk$ and a verification key $vk = vk_1,...,vk_k$, encrypts with $pk_1^{vk_1},pk_1^{vk_2},...,pk_k^{vk_k}$ obtains $c = c_1,...,c_k$ then signs $(sk,c)$ and outputs $c, vk$ and the signature.
This scheme is not yet IND-CCA2 secure: After obtaining the challenge \((s, c)\) just change a bit in \(s\) and give this to the decryption oracle.

- \(\text{Dec}_{\text{CCA1}}\) decrypts whenever \(s_i = r_i\) with \(sk_r\) obtains \(m\) and outputs \(m\) if the ciphertext was valid (verifiability) this includes checking of the signatures.

Successfully changing the ciphertext now amounts to forging a signature.
Conclusion

• Based on the work of Rosen and Segev at TCC 2009 we presented an IND-CCA2 secure PKE based on the McEliece assumptions in the standard model.

• The construction is $O(k^2)$ and finding a more efficient construction is an open problem.