How to Backdoor (Classic) McEliece and How to Guard Against Backdoors

PQCrypto 2022

Tobias Hemmert [1] Alexander May [2] Johannes Mittmann [1] Carl Richard Theodor Schneider [2]

[1]Bundesamt für Sicherheit in der Informationstechnik, Bonn

[2]Ruhr-University Bochum

Our Contribution

- First backdoor for McEliece-like systems
  - Applicable to Classic McEliece (uncompressed keys)
- Simple countermeasure for this backdoor
- First post-quantum secure backdoor

Backdoor

- Transformation of simplified McEliece-like cryptosystem
- Leaks secret information
  - Only accessible to adversary $\mathcal{A}$
- Indistinguishable from original system
- Adversary substitutes implementation of user $\mathcal{U}$

Backdoor

- Transformation of simplified McEliece-like cryptosystem
- Leaks secret information
  - Only accessible to adversary $\mathcal{A}$
- Indistinguishable from original system
- Adversary substitutes implementation of user $\mathcal{U}$

Goal

Use secret information to recompute the secret key of the user.

Properties of a SETUP Scheme (Young and Yung, 1997)

Transform a cryptosystem into a backdoored cryptosystem.

Properties of a SETUP Scheme (Young and Yung, 1997)

Transform a cryptosystem into a backdoored cryptosystem.

1. Inputs of functions agree with specification.

Properties of a SETUP Scheme (Young and Yung, 1997)

Transform a cryptosystem into a backdoored cryptosystem.

1. Inputs of functions agree with specification.
2. Backdoor remains efficient and calls $\mathsf{Enc}_{\mathsf{pk}_{\mathcal{A}}}$.

Properties of a SETUP Scheme (Young and Yung, 1997)

Transform a cryptosystem into a backdoored cryptosystem.

1. Inputs of functions agree with specification.
2. Backdoor remains efficient and calls $\mathsf{Enc}_{\mathsf{pk}_{\mathcal{A}}}$.
3. Exclusive access for $\mathcal{A}$.

Properties of a SETUP Scheme (Young and Yung, 1997)

Transform a cryptosystem into a backdoored cryptosystem.

1. Inputs of functions agree with specification.
2. Backdoor remains efficient and calls $\mathsf{Enc}_{\mathsf{pk}_{\mathcal{A}}}$.
3. Exclusive access for $\mathcal{A}$.
4. Outputs of functions remain compatible, but contain *additional information*.

Properties of a SETUP Scheme (Young and Yung, 1997)

Transform a cryptosystem into a backdoored cryptosystem.

1. Inputs of functions agree with specification.
2. Backdoor remains efficient and calls $\mathrm{Enc}_{\mathrm{pk}_{\mathcal{A}}}$.
3. Exclusive access for $\mathcal{A}$.
4. Outputs of functions remain compatible, but contain *additional information*.

Strong SETUP

- Outputs are polynomially indistinguishable
  - Exception: $\mathcal{A}$
- Even with knowledge about SETUP

Properties of a SETUP Scheme (Young and Yung, 1997)

Transform a cryptosystem into a backdoored cryptosystem.

1. Inputs of functions agree with specification.
2. Backdoor remains efficient and calls $\mathrm{Enc}_{\mathrm{pk}_{\mathcal{A}}}$.
3. Exclusive access for $\mathcal{A}$.
4. Outputs of functions remain compatible, but contain *additional information*.

Strong SETUP

- Outputs are polynomially indistinguishable
  - Exception: $\mathcal{A}$
- Even with knowledge about SETUP

Weak SETUP

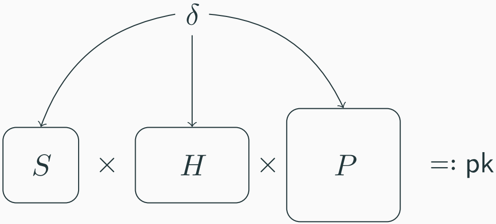
- Outputs are polynomially indistinguishable
  - Exception: $\mathcal{A}, \mathcal{U}$
- Even with knowledge about SETUP

Vanilla McEliece

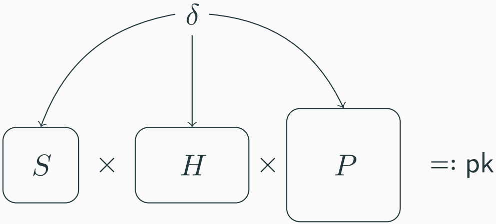
Key Generation

$$S \times H \times P =: \mathsf{pk}$$

Key Generation



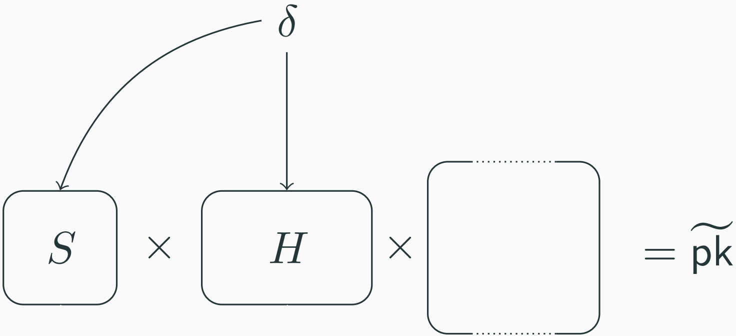
Key Generation



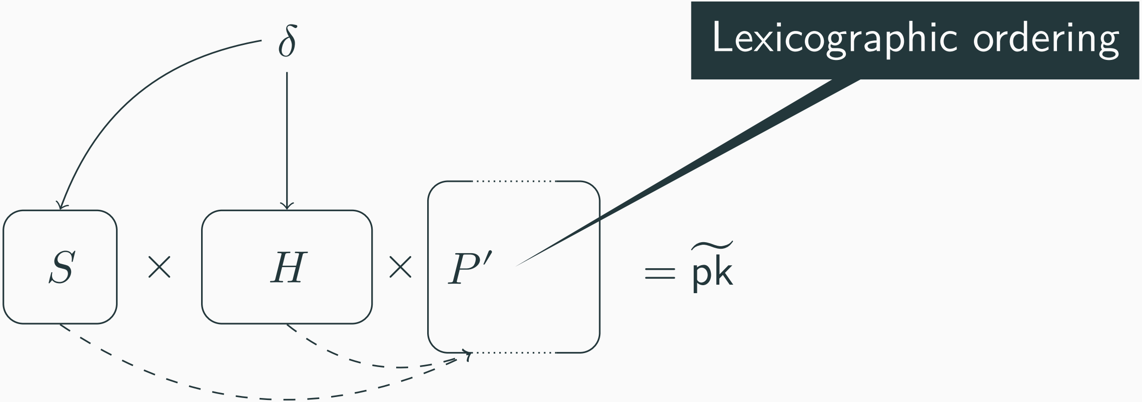
Goal of the Backdoor

Leak δ to $\mathcal{A}$, embedded in pk

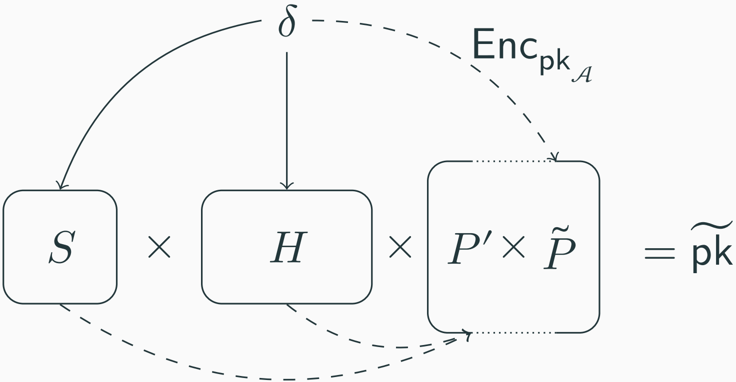
Backdoor for Vanilla McEliece



Backdoor for Vanilla McEliece



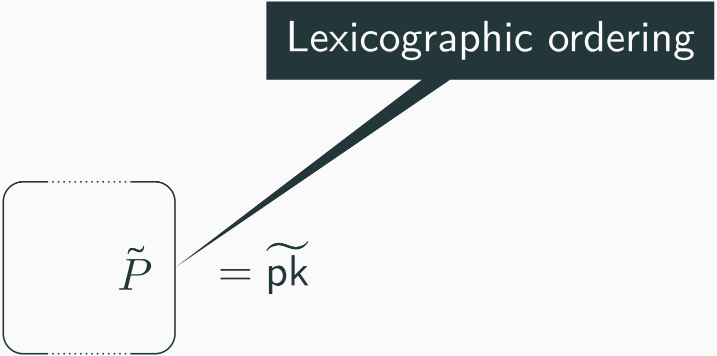
Backdoor for Vanilla McEliece



Recovering δ (View of $\mathcal{A}$)

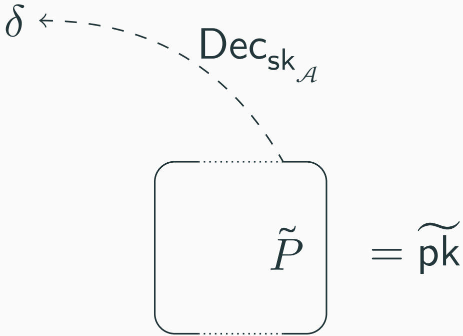
$$= \widetilde{\mathsf{pk}}$$

Recovering δ (View of $\mathcal{A}$)

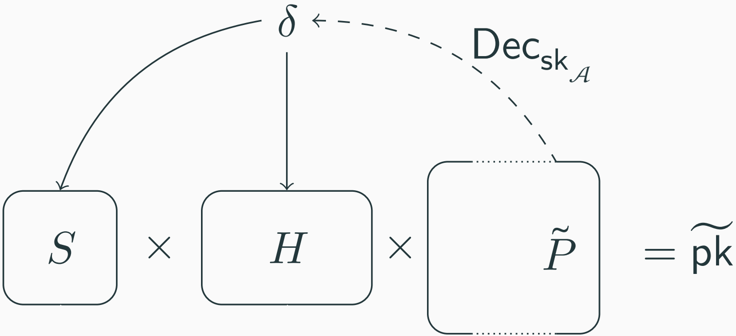


$= \widetilde{\mathsf{pk}}$

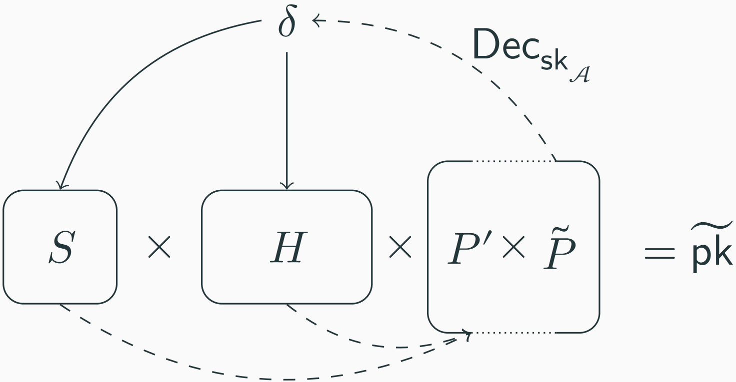
Recovering δ (View of $\mathcal{A}$)



Recovering δ (View of $\mathcal{A}$)



Recovering δ (View of $\mathcal{A}$)



Is this a SETUP?

Is this a SETUP?

1. Inputs are identical ✓

Is this a SETUP?

1. Inputs are identical ✓
2. $\widetilde{\mathsf{KGen}}_{\mathtt{V}}(1^n, \mathsf{pk}_{\mathcal{A}})$ remains efficient ✓

Is this a SETUP?

1. Inputs are identical ✓
2. $\widetilde{\mathsf{KGen}}_{\mathtt{V}}(1^n, \mathsf{pk}_{\mathcal{A}})$ remains efficient ✓
3. Exclusive access due to asymmetric cryptography ✓

Is this a SETUP?

1. Inputs are identical ✓
2. $\widetilde{\mathsf{KGen}}_{\mathtt{V}}(1^n, \mathsf{pk}_{\mathcal{A}})$ remains efficient ✓
3. Exclusive access due to asymmetric cryptography ✓
4. Outputs remain compatible ✓

Is this a SETUP?

1. Inputs are identical ✓
2. $\widetilde{\mathsf{KGen}}_{\mathtt{V}}(1^n, \mathsf{pk}_{\mathcal{A}})$ remains efficient ✓
3. Exclusive access due to asymmetric cryptography ✓
4. Outputs remain compatible ✓

Type of SETUP (Theorem 1)

This is a **strong SETUP**, assuming the ciphertext is indistinguishable from random ($\mathsf{IND\$} - \mathsf{CPA}$).

Countermeasure

- Skip δ, sample directly from randomness?

Countermeasure

- Skip δ, sample directly from randomness?
  - SETUP can resort to PRG
  - We lost reproducibility

Countermeasure

- Skip δ, sample directly from randomness?
  - SETUP can resort to PRG
  - We lost reproducibility
- Reliance on δ
  - Key generation is deterministic in δ

Countermeasure

- Skip δ, sample directly from randomness?
  - SETUP can resort to PRG
  - We lost reproducibility
- Reliance on δ
  - Key generation is deterministic in δ
  - User can verify key generation with a *different* implementation

Countermeasure

- Skip δ, sample directly from randomness?
  - SETUP can resort to PRG
  - We lost reproducibility
- Reliance on δ
  - Key generation is deterministic in δ
  - User can verify key generation with a *different* implementation

Advice for Implementors (Theorem 2)

If δ is part of sk, only weak SETUPs are possible.

Classic McEliece

Relevant Differences to Vanilla McEliece

Relevant Differences to Vanilla McEliece

1. S is chosen so that $SH = \left[I_{n-k} \| T \right]$
   - Focus on systematic form

Relevant Differences to Vanilla McEliece

1. S is chosen so that $SH = \left[I_{n-k} \| T\right]$
   - Focus on systematic form
2. δ is part of sk
   - Only weak SETUP

Relevant Differences to Vanilla McEliece

1. S is chosen so that $SH = \left[I_{n-k} \| T\right]$
   - Focus on systematic form
2. δ is part of sk
   - Only weak SETUP
3. No explicit P

Relevant Differences to Vanilla McEliece

1. S is chosen so that $SH = \left[I_{n-k} \| T\right]$
   - Focus on systematic form
2. δ is part of sk
   - Only weak SETUP
3. No explicit P
4. Fixed to Binary Goppa codes

Relevant Differences to Vanilla McEliece

1. S is chosen so that $SH = \left[I_{n-k} \| T \right]$
   - Focus on systematic form
2. δ is part of sk
   - Only weak SETUP
3. No explicit P
4. Fixed to Binary Goppa codes

Goppa Codes

$$H = \begin{pmatrix} \frac{1}{g(\alpha_1)} & \frac{1}{g(\alpha_2)} & \cdots & \frac{1}{g(\alpha_n)} \\ \frac{\alpha_1}{g(\alpha_1)} & \frac{\alpha_2}{g(\alpha_2)} & \cdots & \frac{\alpha_n}{g(\alpha_n)} \\ \vdots & & \ddots & \\ \frac{\alpha_1^{t-1}}{g(\alpha_1)} & \frac{\alpha_2^{t-1}}{g(\alpha_2)} & \cdots & \frac{\alpha_n^{t-1}}{g(\alpha_n)} \end{pmatrix}$$

Instances for $\mathsf{Enc}_{\mathsf{pk}_{\mathcal{A}}}(\delta)$

Instances for $\mathsf{Enc}_{\mathsf{pk}_{\mathcal{A}}}(\delta)$

- Randomized Niederreiter Cryptosystem (Nojima et al., 2008)
  - Provides $\mathsf{IND\$-CPA}$
  - Instantiate with Category 5 code parameters
  - Ciphertext size: 1664 bit

Instances for $\mathsf{Enc}_{\mathsf{pk}_{\mathcal{A}}}(\delta)$

- Randomized Niederreiter Cryptosystem (Nojima et al., 2008)
  - Provides $\mathsf{IND\$-CPA}$
  - Instantiate with Category 5 code parameters
  - Ciphertext size: 1664 bit

Data Rate

Target instance	Category	n	k	$\lceil \log_2(k!) \rceil$
`kem/mceliece348864`	1	3488	2720	27117
`kem/mceliece460896`	3	4608	3360	34520
`kem/mceliece6688128`	5	6688	5024	54528
`kem/mceliece6960119`	5	6960	5413	59332
`kem/mceliece8192128`	5	8192	6528	73316

Conclusion

Summary

- Backdoor for McEliece-like cryptosystems
  - (Ab)uses deterministic key generation
  - Prevented by knowledge of the seed
  - Instantiable with McEliece itself

Summary

- Backdoor for McEliece-like cryptosystems
  - (Ab)uses deterministic key generation
  - Prevented by knowledge of the seed
  - Instantiable with McEliece itself

Open Question

What about other code-based schmes?

Advice for Implementors

Store δ to make keys verifiable.

Bibliography

Kreher, D. L., and Stinson, D. R. (1999). *Combinatorial Algorithms: Generation, Enumeration, and Search*. CRC Press.

Nojima, R., Imai, H., Kobara, K., and Morozov, K. (2008). "Semantic security for the McEliece cryptosystem without random oracles." *Des. Codes Cryptography*, 49, 289–305.

Young, A., and Yung, M. (1997). "Kleptography: Using Cryptography Against Cryptography." *EUROCRYPT'97*, LNCS, W. Fumy, ed., Springer, Heidelberg, 62–74.