Code-based Public-key Encryption Resistant to Key Leakage

Edoardo Persichetti

Warsaw University

10 June 2013

Outline of the talk

- Preliminaries

Outline of the talk

- Preliminaries
- Hash Proof Systems

Outline of the talk

- Preliminaries
- Hash Proof Systems
- The Construction

Outline of the talk

- Preliminaries
- Hash Proof Systems
- The Construction
- Conclusions

Part I

PRELIMINARIES

Leakage-Resilient Public-Key Encryption

Key-leakage attacks: adversary obtains partial information about the private key.

Leakage-Resilient Public-Key Encryption

Key-leakage attacks: adversary obtains partial information about the private key.

Leakage Oracle queries: submit any function f with $|f(sk)| \leq \lambda$.

Leakage-Resilient Public-Key Encryption

Key-leakage attacks: adversary obtains partial information about the private key.

Leakage Oracle queries: submit any function f with $|f(sk)| \leq \lambda$.

Semantic security against key-leakage

- Get public key pk.
- Perform leakage queries.
- Choose messages m_0 and m_1. Challenge ciphertext: $c^* = \text{Enc}(pk, m_b)$ for $b \in \{0, 1\}$.
- Return b^*.

Leakage-Resilient Public-Key Encryption

Initially modelled by Akavia, Goldwasser and Vaikuntanathan [1].

Leakage-Resilient Public-Key Encryption

Initially modelled by Akavia, Goldwasser and Vaikuntanathan [1].

Work by Naor and Segev [8] provides general construction.

Leakage-Resilient Public-Key Encryption

Initially modelled by Akavia, Goldwasser and Vaikuntanathan [1].

Work by Naor and Segev [8] provides general construction.

Based on Hash Proof Systems + randomness extractors.

Leakage-Resilient Public-Key Encryption

Initially modelled by Akavia, Goldwasser and Vaikuntanathan [1].

Work by Naor and Segev [8] provides general construction.

Based on Hash Proof Systems + randomness extractors.

Constructions given for DDH assumption, and variant of Cramer-Shoup cryptosystem.

Part II

Hash Proof Systems

Hash Proof Systems

Introduced by Cramer and Shoup [3] as a theoretical tool.

Hash Proof Systems

Introduced by Cramer and Shoup [3] as a theoretical tool.

Subsequently revisited and used in various settings, for example Kiltz et al. [7] for KEM.

Hash Proof Systems

Introduced by Cramer and Shoup [3] as a theoretical tool.

Subsequently revisited and used in various settings, for example Kiltz et al. [7] for KEM.

We adapt the “simplified” definition of Alwen et al. [2] given for the Identity-Based setting.

Hash Proof Systems

Introduced by Cramer and Shoup [3] as a theoretical tool.

Subsequently revisited and used in various settings, for example Kiltz et al. [7] for KEM.

We adapt the "simplified" definition of Alwen et al. [2] given for the Identity-Based setting.

HPS

Setup: sets public parameters.

KeyGen: generates public key pk and private key sk.

Encap: produces a ciphertext/key pair (c_0, K).

Encap*: produces an "invalid" ciphertext c_0.

Decap: given sk and c_0 outputs a key K'.

Hash Proof Systems

Introduced by Cramer and Shoup [3] as a theoretical tool.

Subsequently revisited and used in various settings, for example Kiltz et al. [7] for KEM.

We adapt the "simplified" definition of Alwen et al. [2] given for the Identity-Based setting.

HPS

Setup: sets public parameters.

KeyGen: generates public key pk and private key sk.

Encap: produces a ciphertext/key pair (c_0, K).

Encap*: produces an "invalid" ciphertext c_0.

Decap: given sk and c_0 outputs a key K'.

Three requirements for the scheme.

I - Correctness

Valid ciphertexts should decapsulate correctly.

I - Correctness

Valid ciphertexts should decapsulate correctly.

Correctness

If $(c_0, K) = \mathsf{Encap}(pk)$ and $K' = \mathsf{Decap}(sk, c_0)$, then

$$pr[K \neq K'] = \mathsf{negl}(\theta).$$

I - Correctness

Valid ciphertexts should decapsulate correctly.

Correctness

If $(c_0, K) = \text{Encap}(pk)$ and $K' = \text{Decap}(sk, c_0)$, then

$$pr[K \neq K'] = \text{negl}(\theta).$$

For our purposes, a relaxation of the above is sufficient.

I - Correctness

Valid ciphertexts should decapsulate correctly.

Correctness

If $(c_0, K) = \text{Encap}(pk)$ and $K' = \text{Decap}(sk, c_0)$, then

$$pr[K \neq K'] = \text{negl}(\theta).$$

For our purposes, a relaxation of the above is sufficient.

t-Approximate Correctness

If $(c_0, K) = \text{Encap}(pk)$ and $K' = \text{Decap}(sk, c_0)$, then

$$pr[d(K, K') > t] = \text{negl}(\theta).$$

II - UNIVERSALITY/SMOOTHNESS

Invalid ciphertexts should decapsulate to strings that are almost uniformly distributed.

II - Universality/Smoothness

Invalid ciphertexts should decapsulate to strings that are almost uniformly distributed.

Universality

An HPS is (η, ν)-universal if

- $\tilde{H}_\infty(SK|PK) \geq \eta$
- $pr[\text{Decap}(sk, c_0) = \text{Decap}(sk', c_0)] \leq \nu$

where $c_0 = \text{Encap}^*(pk)$ and $sk \neq sk'$.

II - Universality/Smoothness

Invalid ciphertexts should decapsulate to strings that are almost uniformly distributed.

Universality

An HPS is (η, ν)-universal if

- $\tilde{H}_\infty(SK|PK) \geq \eta$
- $pr[\text{Decap}(sk, c_0) = \text{Decap}(sk', c_0)] \leq \nu$

where $c_0 = \text{Encap}^*(pk)$ and $sk \neq sk'$.

Smoothness

An HPS is smooth if

$$\Delta((c_0, K), (c_0, K')) = \text{negl}(\theta).$$

It is λ-leakage smooth if

$$\Delta((c_0, f(sk), K), (c_0, f(sk), K')) = \text{negl}(\theta),$$

for $c_0 = \text{Encap}^*(pk)$, $K = \text{Decap}(sk, c_0)$, $K' \leftarrow \mathbb{U}$ and $|f(sk)| \leq \lambda$.

III - Ciphertext Indistinguishability

Invalid ciphertexts should be computationally indistinguishable from valid ones.

III - Ciphertext Indistinguishability

Invalid ciphertexts should be computationally indistinguishable from valid ones.

Ciphertext Indistinguishability

- Query the challenger for public key/private key pairs (pk, sk).
- Challenge ciphertext: c_0 computed either from $\text{Encap}(pk^*)$ $(b = 0)$ or $\text{Encap}^*(pk^*)$ $(b = 1)$, for a fixed public key pk^*.
- Keep performing queries as above.
- Return b^*.

III - Ciphertext Indistinguishability

Invalid ciphertexts should be computationally indistinguishable from valid ones.

Ciphertext Indistinguishability

- Query the challenger for public key/private key pairs (pk, sk).
- Challenge ciphertext: c_0 computed either from $\text{Encap}(pk^*)$ $(b = 0)$ or $\text{Encap}^*(pk^*)$ $(b = 1)$, for a fixed public key pk^*.
- Keep performing queries as above.
- Return b^*.

No restrictions are placed on the queries hence an adversary is allowed to even see the whole of sk^*.

Part III

The Construction

HPS

Setup: public parameters are a $A \xleftarrow{\$} \mathbb{F}_2^{k \times n}$ and integers k, n, ℓ with $k < n$, $\ell > k$. Let δ be the minimum distance of the code having A as generator matrix, $\rho = \delta/n$ and $\tau = \gamma\rho$ for $\gamma > 0$.
The set of encapsulated keys is $\mathbb{F}_2^{\ell}$.

KeyGen: selects matrices $M \xleftarrow{\$} \mathbb{F}_2^{\ell \times k}$ and $E \leftarrow \chi_\rho^{\ell \times n}$ and outputs $sk = M$ and $pk = MA + E$.

Encap: chooses $s \leftarrow \chi_\tau^n$ and returns $(c_0, K) = (As^T, pk \cdot s)$.

Encap*: chooses $r \xleftarrow{\$} \mathbb{F}_2^k$ and returns $c_0 = r$.

Decap: takes as input sk and c_0 and computes $K' = sk \cdot c_0$.

HPS

Setup: public parameters are a $A \xleftarrow{\$} \mathbb{F}_2^{k \times n}$ and integers k, n, ℓ with $k < n$, $\ell > k$. Let δ be the minimum distance of the code having A as generator matrix, $\rho = \delta/n$ and $\tau = \gamma\rho$ for $\gamma > 0$.
The set of encapsulated keys is $\mathbb{F}_2^{\ell}$.

KeyGen: selects matrices $M \xleftarrow{\$} \mathbb{F}_2^{\ell \times k}$ and $E \leftarrow \chi_\rho^{\ell \times n}$ and outputs $sk = M$ and $pk = MA + E$.

Encap: chooses $s \leftarrow \chi_\tau^n$ and returns $(c_0, K) = (As^T, pk \cdot s)$.

Encap*: chooses $r \xleftarrow{\$} \mathbb{F}_2^k$ and returns $c_0 = r$.

Decap: takes as input sk and c_0 and computes $K' = sk \cdot c_0$.

Choice of parameters important: rate $R = k/n$ needs to be high enough for ρ to be less than $1/\sqrt{n}$.

PROPERTIES

The scheme satisfies the three required properties.

PROPERTIES

The scheme satisfies the three required properties.

t-Approximate Correctness: follows from a result of Döttling et al. [4]. K and K' differ by a factor of Es^T and this string has weight bounded by t with high probability.

Properties

The scheme satisfies the three required properties.

t-Approximate Correctness: follows from a result of Döttling et al. [4]. K and K' differ by a factor of Es^T and this string has weight bounded by t with high probability.

Universality: the first part uses a result from Dumer et al. [5] on the expected number of codewords in a ball of radius δ, and the fact that A defines a random linear code, so δ is on the GV bound with high probability. The second part is a direct consequence of the fact that $\ell > k$ and that matrices chosen uniformly at random are of full rank with overwhelming probability.

PROPERTIES

The scheme satisfies the three required properties.

t-Approximate Correctness: follows from a result of Döttling et al. [4]. K and K' differ by a factor of Es^T and this string has weight bounded by t with high probability.

Universality: the first part uses a result from Dumer et al. [5] on the expected number of codewords in a ball of radius δ, and the fact that A defines a random linear code, so δ is on the GV bound with high probability. The second part is a direct consequence of the fact that $\ell > k$ and that matrices chosen uniformly at random are of full rank with overwhelming probability.

Ciphertext Indistinguishability: since $\rho = O(n^{-1/2-\varepsilon})$, we expect s to have weight below the GV bound. As proved by Fischer and Stern in [6], the vector $c_0 = As^T$ is therefore pseudorandom. The property is satisfied since the private key M doesn't carry information about the ciphertext.

The Encryption Scheme

The HPS just described can be used in a “natural” way for public-key encryption.

The Encryption Scheme

The HPS just described can be used in a “natural” way for public-key encryption.

Need to incorporate an error-correcting code $\mathcal{C}$ into the framework to deal with the error coming from approximate correctness.

The Encryption Scheme

The HPS just described can be used in a "natural" way for public-key encryption.

Need to incorporate an error-correcting code $\mathcal{C}$ into the framework to deal with the error coming from approximate correctness.

Encryption

- Get input m and public-key pk.
- Run Encap(pk) to obtain (c_0, K).
- Set $c_1 = K \oplus \text{Encode}_{\mathcal{C}}(m)$.
- Output $c = (c_0, c_1)$.

The Encryption Scheme

The HPS just described can be used in a "natural" way for public-key encryption.

Need to incorporate an error-correcting code $\mathcal{C}$ into the framework to deal with the error coming from approximate correctness.

Encryption

- Get input m and public-key pk.
- Run Encap(pk) to obtain (c_0, K).
- Set $c_1 = K \oplus \text{Encode}_{\mathcal{C}}(m)$.
- Output $c = (c_0, c_1)$.

Decryption

- Get input sk and $c = (c_0, c_1)$.
- Calculate K' as Decap(sk, c_0).
- Return $m = \text{Decode}_{\mathcal{C}}(K' \oplus c_1)$.

SECURITY

We make use of a result from Alwen et al. [2, Theorem 3.1].

Security

We make use of a result from Alwen et al. [2, Theorem 3.1].

Theorem

Let $\mathcal{H}$ be an (η, ν)-universal HPS with key space $\{0,1\}^\ell$. Then $\mathcal{H}$ is also λ-leakage smooth as long as $\lambda \leq \eta - \ell - \omega(\log \theta)$ and $\nu \leq 2^{-\ell}(1 + \mathit{negl}(\theta))$.

Security

We make use of a result from Alwen et al. [2, Theorem 3.1].

Theorem

Let $\mathcal{H}$ be an (η, ν)-universal HPS with key space $\{0,1\}^\ell$. Then $\mathcal{H}$ is also λ-leakage smooth as long as $\lambda \leq \eta - \ell - \omega(\log \theta)$ and $\nu \leq 2^{-\ell}(1 + \mathit{negl}(\theta))$.

Security is proved using a sequence of games.

Security

We make use of a result from Alwen et al. [2, Theorem 3.1].

Theorem

Let $\mathcal{H}$ be an (η, ν)-universal HPS with key space $\{0,1\}^\ell$. Then $\mathcal{H}$ is also λ-leakage smooth as long as $\lambda \leq \eta - \ell - \omega(\log \theta)$ and $\nu \leq 2^{-\ell}(1 + \mathit{negl}(\theta))$.

Security is proved using a sequence of games.

Semantic security against key-leakage

- Game 0: the semantic security game with leakage.

Security

We make use of a result from Alwen et al. [2, Theorem 3.1].

Theorem

Let $\mathcal{H}$ be an (η, ν)-universal HPS with key space $\{0,1\}^\ell$. Then $\mathcal{H}$ is also λ-leakage smooth as long as $\lambda \leq \eta - \ell - \omega(\log \theta)$ and $\nu \leq 2^{-\ell}(1 + \mathit{negl}(\theta))$.

Security is proved using a sequence of games.

Semantic security against key-leakage

- Game 0: the semantic security game with leakage.

  Ciphertext indistinguishability

- Game 1: replace valid challenge ciphertext with invalid one.

Security

We make use of a result from Alwen et al. [2, Theorem 3.1].

Theorem

Let $\mathcal{H}$ be an (η, ν)-universal HPS with key space $\{0,1\}^\ell$. Then $\mathcal{H}$ is also λ-leakage smooth as long as $\lambda \leq \eta - \ell - \omega(\log \theta)$ and $\nu \leq 2^{-\ell}(1 + negl(\theta))$.

Security is proved using a sequence of games.

Semantic security against key-leakage

- Game 0: the semantic security game with leakage.

  Ciphertext indistinguishability

- Game 1: replace valid challenge ciphertext with invalid one.

  Leakage smoothness

- Game 2: replace c_1^* with a uniformly random string.

Security

We make use of a result from Alwen et al. [2, Theorem 3.1].

Theorem

Let $\mathcal{H}$ be an (η, ν)-universal HPS with key space $\{0,1\}^\ell$. Then $\mathcal{H}$ is also λ-leakage smooth as long as $\lambda \leq \eta - \ell - \omega(\log \theta)$ and $\nu \leq 2^{-\ell}(1 + negl(\theta))$.

Security is proved using a sequence of games.

Semantic security against key-leakage

- Game 0: the semantic security game with leakage.

  Ciphertext indistinguishability

- Game 1: replace valid challenge ciphertext with invalid one.

  Leakage smoothness

- Game 2: replace c_1^* with a uniformly random string.

The advantage in Game 2 is 0 since independent from bit b. □

Part IV

Conclusions

Conclusions

First code-based Hash Proof System.

Conclusions

First code-based Hash Proof System.

First step towards efficient leakage-resilient code-based encryption schemes.

Conclusions

First code-based Hash Proof System.

First step towards efficient leakage-resilient code-based encryption schemes.

Achieves semantic security against leakage attacks without using randomness extractors.

Conclusions

First code-based Hash Proof System.

First step towards efficient leakage-resilient code-based encryption schemes.

Achieves semantic security against leakage attacks without using randomness extractors.

CCA security?

Thank you

References

A. Akavia, S. Goldwasser, and V. Vaikuntanathan.

Simultaneous hardcore bits and cryptography against memory attacks.
In O. Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009.

J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs.

Public-key encryption in the bounded-retrieval model.
In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 113–134. Springer, 2010.

R. Cramer and V. Shoup.

Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption.
In L. R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002.

N. Döttling, J. Müller-Quade, and A. C. A. Nascimento.

Ind-cca secure cryptography based on a variant of the lpn problem.
In X. Wang and K. Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 485–503. Springer, 2012.

I. Dumer, D. Micciancio, and M. Sudan.

Hardness of approximating the minimum distance of a linear code.
IEEE Transactions on Information Theory, 49(1):22–37, 2003.

J.-B. Fischer and J. Stern.

An efficient pseudo-random generator provably as secure as syndrome decoding.
In U. M. Maurer, editor, *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 245–255. Springer, 1996.

E. Kiltz, K. Pietrzak, M. Stam, and M. Yung.

A new randomness extraction paradigm for hybrid encryption.
In A. Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 590–609. Springer, 2009.

M. Naor and G. Segev.

Public-key cryptosystems resilient to key leakage.
In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2009.