Simulation-Based Selective Opening CCA Security for PKE from Key Encapsulation Mechanisms (PKC2015)

Shengli Liu(刘胜利)[1] Kenneth G. Paterson[2]

[1]Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
[2]Information Security Group, Royal Holloway, University of London

March 29, 2015

SOA Security

- PKE and Selective Opening Attack.
- SIM-SO-CCA Security.
- PKE with SIM-SO-CCA Security.
  - Tailored Key Encapsulation Mechanism;
  - Strengthened Cross-Authentication Codes.
- Three constructions of Tailored Key Encapsulation Mechanism.
- Conclusion

Public Key Encryption

Public key encryption PKE= (KeyGen, Enc, Dec).

- KeyGen$(1^\kappa) \to (pk, sk)$.
- Enc$(pk, M) \to C$.
- Dec$(sk, C) \to M/ \perp$.

An PKE scheme has completeness error ϵ if

$$\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, M)) \neq M] \leq \epsilon$$

for all $(pk, sk) \leftarrow$ KeyGen and $M \leftarrow \mathcal{M}$, where the probability is taken over the coins used in encryption.

Public Key Encryption

Public key encryption PKE= (KeyGen, Enc, Dec).

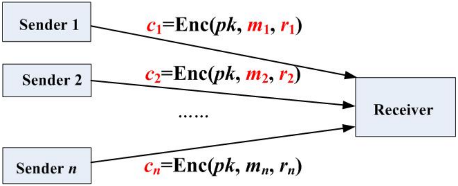
- KeyGen$(1^\kappa) \rightarrow (pk, sk)$.
- Enc$(pk, M) \rightarrow C$.
- Dec$(sk, C) \rightarrow M/ \perp$.

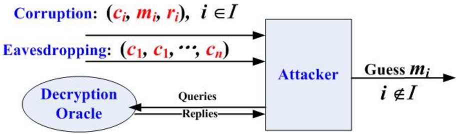
An PKE scheme has completeness error ϵ if

$$\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, M)) \neq M] \leq \epsilon$$

for all $(pk, sk) \leftarrow$ KeyGen and $M \leftarrow \mathcal{M}$, where the probability is taken over the coins used in encryption.

Selective Opening Attack





Selective Opening Attack: a vector of ciphertexts, adaptive corruptions exposing not only some message but also the random coins.

SIM-SO-CCA2 Security: $\mathrm{Exp}^{cca-so-real}_{\mathcal{A},\mathcal{M},R}(1^\kappa)$

The real experiment

Challenger		$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$
(pk, sk)← KeyGen(1^κ)	$\xrightarrow{\text{pk}}$	
	$\xleftarrow{\alpha}$	$\alpha \leftarrow \mathcal{A}_1^{\mathsf{Dec}(\cdot)}(pk)$
$\overrightarrow{M} = (M^{(1)}, \ldots, M^{(n)}) \leftarrow \mathcal{M}(\alpha)$		
$\overrightarrow{R} = (R^{(1)}, \ldots, R^{(n)}) \leftarrow \mathcal{R}$		
$\overrightarrow{C}$ = Enc$(pk, \overrightarrow{M}; \overrightarrow{R})$	$\xrightarrow{\overrightarrow{C}}$	
	$\xleftarrow{I}$	$I \leftarrow \mathcal{A}_2^{\mathsf{Dec}(\cdot)}(\overrightarrow{C})$
	$\xrightarrow{(M^{(i)}, R^{(i)})_{i \in I}}$	out$_A \leftarrow \mathcal{A}_3^{\mathsf{Dec}(\cdot)}\left(\left(M^{(i)}, R^{(i)}\right)_{i \in I}\right)$
	$R(\overrightarrow{M}, I, out_A)$	

SIM-SO-CCA2 Security: $\mathrm{Exp}_{\mathcal{A},\mathcal{M},R}^{cca-so-real}(1^\kappa)$

The real experiment

Challenger		$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$
(pk, sk)$\leftarrow$ KeyGen(1^κ)	$\xrightarrow{\text{pk}}$	
	$\xleftarrow{\alpha}$	$\alpha \leftarrow \mathcal{A}_1^{\text{Dec}(\cdot)}(pk)$
$\vec{M} = (M^{(1)}, \ldots, M^{(n)}) \leftarrow \mathcal{M}(\alpha)$		
$\vec{R} = (R^{(1)}, \ldots, R^{(n)}) \leftarrow \mathcal{R}$		
$\vec{C}$ = Enc$(pk, \vec{M}; \vec{R})$	$\xrightarrow{\vec{C}}$	
	$\xleftarrow{I}$	$I \leftarrow \mathcal{A}_2^{\text{Dec}(\cdot)}(\vec{C})$
	$\xrightarrow{(M^{(i)}, R^{(i)})_{i\in I}}$	$\text{out}_A \leftarrow \mathcal{A}_3^{\text{Dec}(\cdot)}\left(\left(M^{(i)}, R^{(i)}\right)_{i\in I}\right)$
	$R(\vec{M}, I, out_A)$	

SIM-SO-CCA2 Security: $\mathrm{Exp}_{\mathcal{A},\mathcal{M},R}^{cca-so-real}(1^\kappa)$

The real experiment

Challenger		$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$
(pk, sk)$\leftarrow$ KeyGen(1^κ)	$\xrightarrow{\text{pk}}$	
	$\xleftarrow{\alpha}$	$\alpha \leftarrow \mathcal{A}_1^{\mathsf{Dec}(\cdot)}(pk)$
$\vec{M} = (M^{(1)}, \ldots, M^{(n)}) \leftarrow \mathcal{M}(\alpha)$		
$\vec{R} = (R^{(1)}, \ldots, R^{(n)}) \leftarrow \mathcal{R}$		
$\vec{C} = \mathsf{Enc}(pk, \vec{M}; \vec{R})$	$\xrightarrow{\vec{C}}$	
	$\xleftarrow{I}$	$I \leftarrow \mathcal{A}_2^{\mathsf{Dec}(\cdot)}(\vec{C})$
	$\xrightarrow{(M^{(i)},R^{(i)})_{i\in I}}$	$\mathsf{out}_A \leftarrow \mathcal{A}_3^{\mathsf{Dec}(\cdot)}\left(\left(M^{(i)}, R^{(i)}\right)_{i\in I}\right)$
	$R(\vec{M}, I, out_A)$	

SIM-SO-CCA2 Security: $\mathrm{Exp}^{cca-so-ideal}_{\mathcal{A},\mathcal{M},R}(1^\kappa)$

The ideal experiment

Challenger		$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$
	$\xleftarrow{\alpha}$	$\alpha \leftarrow \mathcal{S}_1(1^\kappa)$
$\vec{M} = (M^{(1)}, \ldots, M^{(n)}) \leftarrow \mathcal{M}(\alpha)$		
	$\xleftarrow{I \subsetneq [n]}$	$I \leftarrow \mathcal{S}_2(1^{\lvert M^{(i)} \rvert})$
	$\xrightarrow{(M^{(i)})_{i \in I}}$	$\text{out}_{\mathcal{S}} \leftarrow \mathcal{S}_3\left(\left(M^{(i)}\right)_{i \in I}\right)$
	$R(\vec{M}, I, out_{\mathcal{S}})$	

SIM-SO-CCA2 Security: $\forall$ PPT $\mathcal{A}$, $\forall$ PPT R, $\forall$ PPT $\mathcal{M}$, $\exists$ $\mathcal{S}$ such that

$$\left|\Pr\left[R(\vec{M}, I, out_{\mathcal{A}}) = 1\right] - \Pr\left[R(\vec{M}, I, out_{\mathcal{S}}) = 1\right]\right| \quad \text{is negligible.}$$

SIM-SO-CCA2 Security: $\mathrm{Exp}^{cca-so-ideal}_{\mathcal{A},\mathcal{M},R}(1^\kappa)$

The ideal experiment

Challenger		$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$
	$\xleftarrow{\alpha}$	$\alpha \leftarrow \mathcal{S}_1(1^\kappa)$
$\vec{M} = (M^{(1)}, \ldots, M^{(n)}) \leftarrow \mathcal{M}(\alpha)$		
	$\xleftarrow{I \subsetneq [n]}$	$I \leftarrow \mathcal{S}_2(1^{\lvert M^{(i)}\rvert})$
	$\xrightarrow{(M^{(i)})_{i \in I}}$	$\mathrm{out}_{\mathcal{S}} \leftarrow \mathcal{S}_3\left(\left(M^{(i)}\right)_{i \in I}\right)$
	$R(\vec{M}, I, out_{\mathcal{S}})$	

SIM-SO-CCA2 Security: $\forall$ PPT $\mathcal{A}$, $\forall$ PPT R, $\forall$ PPT $\mathcal{M}$, $\exists$ $\mathcal{S}$ such that

$$\left|\Pr\left[R(\vec{M}, I, out_{\mathcal{A}}) = 1\right] - \Pr\left[R(\vec{M}, I, out_{\mathcal{S}}) = 1\right]\right|$$ is negligible.

Related work

- [BHY2009] formalized IND-SOA, SIM-SOA security.
- SIM-SOA security is harder to achieve than IND-SOA security.
- [FHKW2010] proposed the first construction of PKE with SIM-SO-CCA2 Security.
- [BWY2011] proposed the first construction of IBE with SIM-SO-CPA Security.
- [LDLWZ2014] showed to construct IBE with SIM-SO-CCA2 Security.
- [HJKS2015] will present PKE schemes with SIM-SO-CCA2 Security in the Random Oracle model.

Related work

- [BHY2009] formalized IND-SOA, SIM-SOA security.
- SIM-SOA security is harder to achieve than IND-SOA security.
- [FHKW2010] proposed the first construction of PKE with SIM-SO-CCA2 Security.
- [BWY2011] proposed the first construction of IBE with SIM-SO-CPA Security.
- [LDLWZ2014] showed to construct IBE with SIM-SO-CCA2 Security.
- [HJKS2015] will present PKE schemes with SIM-SO-CCA2 Security in the Random Oracle model.

Related work

- [BHY2009] formalized IND-SOA, SIM-SOA security.
- SIM-SOA security is harder to achieve than IND-SOA security.
- [FHKW2010] proposed the first construction of PKE with SIM-SO-CCA2 Security.
- [BWY2011] proposed the first construction of IBE with SIM-SO-CPA Security.
- [LDLWZ2014] showed to construct IBE with SIM-SO-CCA2 Security.
- [HJKS2015] will present PKE schemes with SIM-SO-CCA2 Security in the Random Oracle model.

Related work

- [BHY2009] formalized IND-SOA, SIM-SOA security.
- SIM-SOA security is harder to achieve than IND-SOA security.
- [FHKW2010] proposed the first construction of PKE with SIM-SO-CCA2 Security.
- [BWY2011] proposed the first construction of IBE with SIM-SO-CPA Security.
- [LDLWZ2014] showed to construct IBE with SIM-SO-CCA2 Security.
- [HJKS2015] will present PKE schemes with SIM-SO-CCA2 Security in the Random Oracle model.

Related work

- [BHY2009] formalized IND-SOA, SIM-SOA security.
- SIM-SOA security is harder to achieve than IND-SOA security.
- [FHKW2010] proposed the first construction of PKE with SIM-SO-CCA2 Security.
- [BWY2011] proposed the first construction of IBE with SIM-SO-CPA Security.
- [LDLWZ2014] showed to construct IBE with SIM-SO-CCA2 Security.
- [HJKS2015] will present PKE schemes with SIM-SO-CCA2 Security in the Random Oracle model.

Related work

- [BHY2009] formalized IND-SOA, SIM-SOA security.
- SIM-SOA security is harder to achieve than IND-SOA security.
- [FHKW2010] proposed the first construction of PKE with SIM-SO-CCA2 Security.
- [BWY2011] proposed the first construction of IBE with SIM-SO-CPA Security.
- [LDLWZ2014] showed to construct IBE with SIM-SO-CCA2 Security.
- [HJKS2015] will present PKE schemes with SIM-SO-CCA2 Security in the Random Oracle model.

Our Contribution

To achieve SIM-SO-CCA2 Security

[FHKW2010]: PKE = Extended HPS + Strong XAC+ CR-Hash.

- We generalize the black-box PKE construction of [FHKW2010]:

  PKE = tailored KEM + strengthened XAC.

- We characterize the properties of tailored KEM.

- We give three constructions for tailored KEM, including
  - Hash Proof System.
  - n-Linear Assumption.
  - indistinguishability Obfuscation (iO).

  SIM-SO-CCA security follows from the existence of iO and OWF.

Our Contribution

To achieve SIM-SO-CCA2 Security

[FHKW2010]: PKE = Extended HPS + Strong XAC+ CR-Hash.

- We generalize the black-box PKE construction of [FHKW2010]:

  PKE = tailored KEM + strengthened XAC.

- We characterize the properties of tailored KEM.
- We give three constructions for tailored KEM, including
  - Hash Proof System.
  - n-Linear Assumption.
  - indistinguishability Obfuscation (iO).

  SIM-SO-CCA security follows from the existence of iO and OWF.

Our Contribution

To achieve SIM-SO-CCA2 Security

[FHKW2010]: PKE = Extended HPS + Strong XAC+ CR-Hash.

- We generalize the black-box PKE construction of [FHKW2010]:

  PKE = tailored KEM + strengthened XAC.

- We characterize the properties of tailored KEM.

- We give three constructions for tailored KEM, including
  - Hash Proof System.
  - n-Linear Assumption.
  - indistinguishability Obfuscation (iO).

  SIM-SO-CCA security follows from the existence of iO and OWF.

Our Contribution

To achieve SIM-SO-CCA2 Security

[FHKW2010]: PKE = Extended HPS + Strong XAC+ CR-Hash.

- We generalize the black-box PKE construction of [FHKW2010]:

  PKE = tailored KEM + strengthened XAC.

- We characterize the properties of tailored KEM.
- We give three constructions for tailored KEM, including
  - Hash Proof System.
  - n-Linear Assumption.
  - indistinguishability Obfuscation ($i\mathcal{O}$).

  SIM-SO-CCA security follows from the existence of $i\mathcal{O}$ and OWF.

How to get SIM-SO-CCA2 Security: the idea

Challenger		$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$
	$\xleftarrow{\alpha}$	$\alpha \leftarrow \mathcal{S}_1(1^\kappa)$
$\vec{M} = (M^{(1)}, \ldots, M^{(n)})$	$\xleftarrow{I}$	$I \leftarrow \mathcal{S}_2(1^{\lvert M^{(i)} \rvert})$
	$\xrightarrow{(M^{(i)})_{i \in I}}$	$\text{Out}_S \leftarrow \mathcal{S}_3\left(\left(M^{(i)}\right)_{i \in I}\right)$

Aim: $\left(\vec{M}, I, out_A\right) \approx_c \left(\vec{M}, I, out_S\right)$

How to get SIM-SO-CCA2 Security: the idea

Challenger		$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$
	$\xleftarrow{\alpha}$	$\alpha \leftarrow \mathcal{S}_1(1^\kappa)$
		$\{$ (pk, sk) $\leftarrow$ KeyGen(1^κ)
$\vec{M} \leftarrow \mathcal{M}(\alpha)$		$\alpha \leftarrow \mathcal{A}_1^{Dec(\cdot)}(PK)$ $\}$
$\vec{M} = (M^{(1)}, \ldots, M^{(n)})$	$\xleftarrow{I}$	$I \leftarrow \mathcal{S}_2(1^{\lvert M^{(i)} \rvert})$
	$\xrightarrow{\left(M^{(i)}\right)_{i \in I}}$	$\text{Out}_{\mathcal{S}} \leftarrow \mathcal{S}_3\left(\left(M^{(i)}\right)_{i \in I}\right)$

Aim: $\left(\vec{M}, I, out_A\right) \approx_c \left(\vec{M}, I, out_S\right)$

How to get SIM-SO-CCA2 Security: the idea

Challenger		$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$
	$\xleftarrow{(\alpha, \overrightarrow{ID})}$	$(\alpha, \overrightarrow{ID}) \leftarrow \mathcal{S}_1(1^\kappa)$
		$\{ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
$\vec{M} \leftarrow \mathcal{M}(\alpha)$		$\alpha \leftarrow \mathcal{A}_1^{\mathsf{Dec}(\cdot)}(PK) \}$
$\vec{M} = (M^{(1)}, \ldots, M^{(n)})$	$\xleftarrow{I}$	$I \leftarrow \mathcal{S}_2(1^{\lvert M^{(i)} \rvert})$
		$\{ I \leftarrow \mathcal{A}_2^{\mathsf{Dec}(\cdot)}(\vec{C}) \}$
	$\xrightarrow{(M^{(i)})_{i \in I}}$	$\mathsf{Out}_S \leftarrow \mathcal{S}_3\left(\left(M^{(i)}\right)_{i \in I}\right)$
		$\{ \mathsf{Out}_A \leftarrow \mathcal{A}_3^{\mathsf{Dec}(\cdot)}\left(\left(M^{(i)}, R^{(i)}\right)_{i \in I}\right) \}$

Aim: $\left(\vec{M}, I, out_A\right) \approx_c \left(\vec{M}, I, out_S\right)$

The challenging job for the simulator

How to create a fake ciphertext vector $\overrightarrow{C}$ s.t.

- $\boxed{\text{(fake) } \overrightarrow{C} \approx_c \overrightarrow{C} \text{ (real)}}$.
- C can be opened to any messages.

Following the techniques of non-committing and deniable encryption, we can build Single-bit PKE from a KEM=(KEM.Kg, KEM.Enc, KEM.Dec):

KEM.Kg$(1^\kappa) \to (pk, sk)$; KEM.Encap$(pk) \to (K, \phi)$; KEM.Decap$(sk, \phi) \to K/\perp$.

Single-bit PKE from KEM

$$\text{Enc}_{pk}(M)\text{: Ciphertext } C = \begin{cases} (K^R, \phi^R) \leftarrow (\mathcal{K}, \mathcal{C}) & \text{if} \quad M = 0 \\ (K, \phi) \leftarrow \text{KEM.Encap}(pk) & \text{if} \quad M = 1 \end{cases}$$

$$\text{Dec}_{sk}(C)\text{: Return } M = \begin{cases} 0 & \text{if} \quad \text{KEM.Decap}(sk, \phi) = \perp \\ 1 & \text{if} \quad \text{KEM.Decap}(sk, \phi) = K \end{cases}$$

The challenging job for the simulator

How to create a fake ciphertext vector $\vec{C}$ s.t.

- $\boxed{\text{(fake) } \vec{C} \approx_c \vec{C} \text{ (real)}}$.
- C can be opened to any messages.

Following the techniques of non-committing and deniable encryption, we can build Single-bit PKE from a KEM=(KEM.Kg, KEM.Enc, KEM.Dec):

KEM.Kg$(1^\kappa) \to (pk, sk)$; KEM.Encap$(pk) \to (K, \phi)$; KEM.Decap$(sk, \phi) \to K/\perp$.

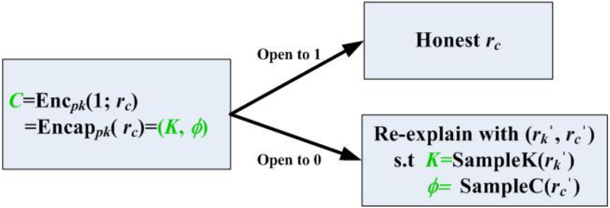
Single-bit PKE from KEM

$$\text{Enc}_{pk}(M)\text{: Ciphertext } C = \begin{cases} (K^R, \phi^R) \leftarrow (\mathcal{K}, \mathcal{C}) & \text{if} \quad M = 0 \\ (K, \phi) \leftarrow \text{KEM.Encap}(pk) & \text{if} \quad M = 1 \end{cases}$$

$$\text{Dec}_{sk}(C)\text{: Return } M = \begin{cases} 0 & \text{if} \quad \text{KEM.Decap}(sk, \phi) = \perp \\ 1 & \text{if} \quad \text{KEM.Decap}(sk, \phi) = K \end{cases}$$

SIM-SO-CCA Security for single-bit PKE

The equivocable ciphertext is $C = Enc_{pk}(1) = (K, \phi)$.



Requirement for KEM=(KEM.Kg, KEM.Enc, KEM.Dec)

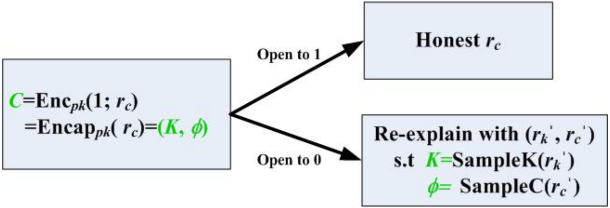
$(K, \phi) \approx_c (K^R, \phi^R)$: $(K, \phi) \leftarrow$ KEM.Encap(pk)

$(K^R, \phi^R) \leftarrow (\mathcal{K}, \mathcal{C})$.

$\mathcal{K}, \mathcal{C}$: Efficiently samplable and explainable (ESE) domains. Any (K, ϕ) can be explained with a randomness as if they are randomly chosen.

SIM-SO-CCA Security for single-bit PKE

The equivocable ciphertext is $C = Enc_{pk}(1) = (K, \phi)$.



Requirement for KEM=(KEM.Kg, KEM.Enc, KEM.Dec)

$(K, \phi) \approx_c (K^R, \phi^R)$: $(K, \phi) \leftarrow$ KEM.Encap(pk)

$(K^R, \phi^R) \leftarrow (\mathcal{K}, \mathcal{C})$.

$\mathcal{K}, \mathcal{C}$: Efficiently samplable and explainable (ESE) domains. Any (K, ϕ) can be explained with a randomness as if they are randomly chosen.

Constructing multi-bit PKE

Cross Authentication Codes ℓ-XAC $= (\mathsf{XAuth}, \mathsf{XVer})$ (due to Fehr et al.):

- **Authentication and Verification.** If $T \leftarrow \mathsf{XAuth}(K_1, \ldots, K_\ell)$, then $\mathsf{XVer}(K_i, T) = 1$.
- **Security against impersonation attacks.** $\forall T \in \mathcal{T}$,

$$\Pr\left[\mathsf{XVer}(K, T) = 0 \;:\; K \leftarrow \mathcal{T}\right] = 1 - neg(\kappa).$$

- **Security against substitution attacks.** Given T ($T = \mathsf{XAuth}(K_1, \ldots, K_\ell)$) and $(K_j)_{j\in[\ell], j\neq i}$, as long as K_i is uniformly chosen, then it is hard for an adversary to forge $T' \neq T$, such that $\mathsf{XVer}(K_i, T') = 0$.
- **Strongness.** $\exists$ a ppt ReSample such that, given $K_i \leftarrow \mathcal{K}$, values of $(K_j)_{j\in[\ell], j\neq i}$ and the tag $T = \mathsf{XAuth}(K_1, \ldots, K_\ell)$,

$$\hat{K}_i \leftarrow \mathsf{ReSample}((K_j)_{j\in[\ell], j\neq i}, T)$$

and $\hat{K}_i \mid_{T,(K_j)_{j\in[\ell], j\neq i}} \approx_s \hat{K} \mid_{T,(K_j)_{j\in[\ell], j\neq i}}$.

- **Semi-uniqueness.** $K = (K_x, K_y) \in \mathcal{K}_x \times \mathcal{K}_y$. $\forall T, \forall K_x, \exists! K_y \in \mathcal{K}_y$ s. t. $\mathsf{XVer}(K, T) = 1$

Constructing multi-bit PKE

Cross Authentication Codes ℓ-XAC = (XAuth, XVer) (due to Fehr et al.):

- **Authentication and Verification.** If $T \leftarrow \mathrm{XAuth}(K_1, \ldots, K_\ell)$, then $\mathrm{XVer}(K_i, T) = 1$.
- **Security against impersonation attacks.** $\forall T \in \mathcal{T}$,

$$\Pr\left[\mathrm{XVer}(K, T) = 0 \ : \ K \leftarrow \mathcal{T}\right] = 1 - neg(\kappa).$$

- **Security against substitution attacks.** Given T ($T = \mathrm{XAuth}(K_1, \ldots, K_\ell)$) and $(K_j)_{j \in [\ell], j \neq i}$, as long as K_i is uniformly chosen, then it is hard for an adversary to forge $T' \neq T$, such that $\mathrm{XVer}(K_i, T') = 0$.
- **Strongness.** $\exists$ a ppt ReSample such that, given $K_i \leftarrow \mathcal{K}$, values of $(K_j)_{j \in [\ell], j \neq i}$ and the tag $T = \mathrm{XAuth}(K_1, \ldots, K_\ell)$,

$$\hat{K}_i \leftarrow \mathrm{ReSample}((K_j)_{j \in [\ell], j \neq i}, T)$$

and $\hat{K}_i \mid_{T, (K_j)_{j \in [\ell], j \neq i}} \approx_s \hat{K} \mid_{T, (K_j)_{j \in [\ell], j \neq i}}$.

- **Semi-uniqueness.** $K = (K_x, K_y) \in \mathcal{K}_x \times \mathcal{K}_y$. $\forall T, \forall K_x, \exists! K_y \in \mathcal{K}_y$ s. t. $\mathrm{XVer}(K, T) = 1$

Constructing multi-bit PKE

Cross Authentication Codes ℓ-XAC $= (\mathsf{XAuth}, \mathsf{XVer})$ (due to Fehr et al.):

- Authentication and Verification. If $T \leftarrow \mathsf{XAuth}(K_1, \ldots, K_\ell)$, then $\mathsf{XVer}(K_i, T) = 1$.
- Security against impersonation attacks. $\forall T \in \mathcal{T}$,

$$\Pr\left[\mathsf{XVer}(K, T) = 0 \ : \ K \leftarrow \mathcal{T}\right] = 1 - neg(\kappa).$$

- Security against substitution attacks. Given T ($T = \mathsf{XAuth}(K_1, \ldots, K_\ell)$) and $(K_j)_{j \in [\ell], j \neq i}$, as long as K_i is uniformly chosen, then it is hard for an adversary to forge $T' \neq T$, such that $\mathsf{XVer}(K_i, T') = 0$.
- Strongness. $\exists$ a ppt ReSample such that, given $K_i \leftarrow \mathcal{K}$, values of $(K_j)_{j \in [\ell], j \neq i}$ and the tag $T = \mathsf{XAuth}(K_1, \ldots, K_\ell)$,

$$\hat{K}_i \leftarrow \mathsf{ReSample}((K_j)_{j \in [\ell], j \neq i}, T)$$

and $\quad \hat{K}_i \mid_{T, (K_j)_{j \in [\ell], j \neq i}} \approx_s \hat{K} \mid_{T, (K_j)_{j \in [\ell], j \neq i}}$.

- Semi-uniqueness. $K = (K_x, K_y) \in \mathcal{K}_x \times \mathcal{K}_y$. $\forall T, \forall K_x, \exists! K_y \in \mathcal{K}_y$ s. t. $\mathsf{XVer}(K, T) = 1$

Constructing multi-bit PKE

Cross Authentication Codes ℓ-XAC = (XAuth, XVer) (due to Fehr et al.):

- Authentication and Verification. If $T \leftarrow \text{XAuth}(K_1, \ldots, K_\ell)$, then $\text{XVer}(K_i, T) = 1$.
- Security against impersonation attacks. $\forall T \in \mathcal{T}$,

$$\Pr[\text{XVer}(K, T) = 0 \ : \ K \leftarrow \mathcal{T}] = 1 - neg(\kappa).$$

- Security against substitution attacks. Given T ($T = \text{XAuth}(K_1, \ldots, K_\ell)$) and $(K_j)_{j \in [\ell], j \neq i}$, as long as K_i is uniformly chosen, then it is hard for an adversary to forge $T' \neq T$, such that $\text{XVer}(K_i, T') = 0$.
- Strongness. $\exists$ a ppt ReSample such that, given $K_i \leftarrow \mathcal{K}$, values of $(K_j)_{j \in [\ell], j \neq i}$ and the tag $T = \text{XAuth}(K_1, \ldots, K_\ell)$,

$$\hat{K}_i \leftarrow \text{ReSample}((K_j)_{j \in [\ell], j \neq i}, T)$$

and $$\hat{K}_i \mid_{T, (K_j)_{j \in [\ell], j \neq i}} \approx_s \hat{K} \mid_{T, (K_j)_{j \in [\ell], j \neq i}}.$$

- Semi-uniqueness. $K = (K_x, K_y) \in \mathcal{K}_x \times \mathcal{K}_y$. $\forall T, \forall K_x, \exists! K_y \in \mathcal{K}_y$ s. t. $\text{XVer}(K, T) = 1$

Constructing multi-bit PKE

Cross Authentication Codes ℓ-XAC = (XAuth, XVer) (due to Fehr et al.):

- Authentication and Verification. If $T \leftarrow \text{XAuth}(K_1, \ldots, K_\ell)$, then $\text{XVer}(K_i, T) = 1$.
- Security against impersonation attacks. $\forall T \in \mathcal{T}$,

$$\Pr[\text{XVer}(K, T) = 0 \ : \ K \leftarrow \mathcal{T}] = 1 - neg(\kappa).$$

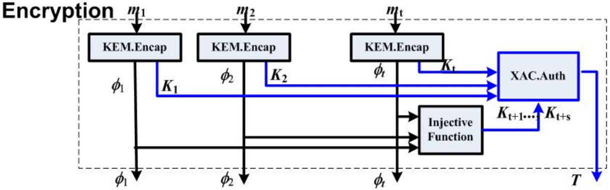
- Security against substitution attacks. Given T ($T = \text{XAuth}(K_1, \ldots, K_\ell)$) and $(K_j)_{j\in[\ell], j\neq i}$, as long as K_i is uniformly chosen, then it is hard for an adversary to forge $T' \neq T$, such that $\text{XVer}(K_i, T') = 0$.
- Strongness. $\exists$ a ppt ReSample such that, given $K_i \leftarrow \mathcal{K}$, values of $(K_j)_{j\in[\ell], j\neq i}$ and the tag $T = \text{XAuth}(K_1, \ldots, K_\ell)$,

$$\hat{K}_i \leftarrow \text{ReSample}((K_j)_{j\in[\ell], j\neq i}, T)$$

and $$\hat{K}_i \mid_{T,(K_j)_{j\in[\ell], j\neq i}} \approx_s \hat{K} \mid_{T,(K_j)_{j\in[\ell], j\neq i}}.$$

- Semi-uniqueness. $K = (K_x, K_y) \in \mathcal{K}_x \times \mathcal{K}_y$. $\forall T, \forall K_x$, $\exists! K_y \in \mathcal{K}_y$ s. t. $\text{XVer}(K, T) = 1$

Constructing multi-bit PKE

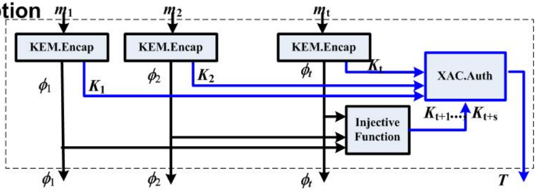


PKE.Enc$(pk_{kem}, m_1\|\ldots\|m_\ell)$: Ciphertext $C = (\phi_1, \phi_2, \ldots, \phi_\ell, T)$.

$$\left.\begin{array}{l}
m_1 = \left\{\begin{array}{ccc} 1 & \Rightarrow & \text{KEM.Enc}(pk_{kem}) \\ 0 & \Rightarrow & \text{random pair} \end{array}\right\} \Rightarrow (\phi_1, K_1) \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\
m_\ell = \left\{\begin{array}{ccc} 1 & \Rightarrow & \text{KEM.Enc}(pk_{kem}) \\ 0 & \Rightarrow & \text{random pair} \end{array}\right\} \Rightarrow (\phi_\ell, K_\ell) \\
\text{F}(\phi_1, \phi_2, \ldots, \phi_\ell) \quad \Rightarrow \quad (K_{\ell+1}, \ldots, K_{\ell+s})
\end{array}\right\} \Rightarrow \left\{\begin{array}{c} T = \text{XAuth}(K_1, \ldots, K_\ell, \ldots, K_{\ell+s}) \\ C = (\phi_1, \phi_2, \ldots, \phi_\ell, T) \end{array}\right.$$

Constructing multi-bit PKE

Encryption



PKE.Enc$(pk_{kem}, m_1 \| \ldots \| m_\ell)$: Ciphertext $C = (\phi_1, \phi_2, \ldots, \phi_\ell, T)$.

$$m_1 = \left\{ \begin{array}{ccc} 1 & \Rightarrow & \text{KEM.Enc}(pk_{kem}) \\ 0 & \Rightarrow & \text{random pair} \end{array} \right\} \Rightarrow (\phi_1, K_1)$$

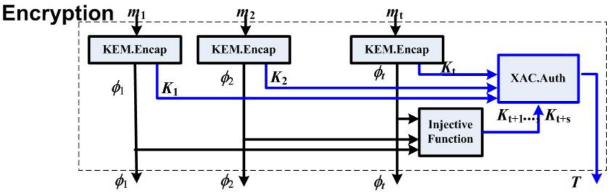
$$\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots$$

$$m_\ell = \left\{ \begin{array}{ccc} 1 & \Rightarrow & \text{KEM.Enc}(pk_{kem}) \\ 0 & \Rightarrow & \text{random pair} \end{array} \right\} \Rightarrow (\phi_\ell, K_\ell)$$

$$\mathsf{F}(\phi_1, \phi_2, \ldots, \phi_\ell) \quad \Rightarrow \quad (K_{\ell+1}, \ldots, K_{\ell+s})$$

$$\Rightarrow \left\{ \begin{array}{c} T = \text{XAuth}(K_1, \ldots, K_\ell, \ldots, K_{\ell+s}) \\ C = (\phi_1, \phi_2, \ldots, \phi_\ell, T) \end{array} \right.$$

Constructing multi-bit PKE

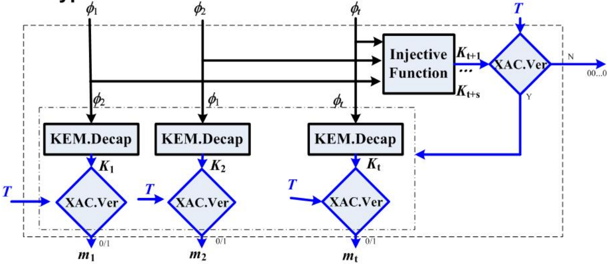


PKE.Enc$(pk_{kem}, m_1\|\ldots\|m_\ell)$: Ciphertext $C = (\phi_1, \phi_2, \ldots, \phi_\ell, T)$.

$$
\left.\begin{array}{l}
m_1 = \left\{\begin{array}{ccc} 1 & \Rightarrow & \text{KEM.Enc}(pk_{kem}) \\ 0 & \Rightarrow & \text{random pair} \end{array}\right\} \Rightarrow (\phi_1, K_1) \\
\quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\
m_\ell = \left\{\begin{array}{ccc} 1 & \Rightarrow & \text{KEM.Enc}(pk_{kem}) \\ 0 & \Rightarrow & \text{random pair} \end{array}\right\} \Rightarrow (\phi_\ell, K_\ell) \\
\quad \text{F}(\phi_1, \phi_2, \ldots, \phi_\ell) \quad \Rightarrow \quad (K_{\ell+1}, \ldots, K_{\ell+s})
\end{array}\right\}
\Rightarrow
\left\{\begin{array}{l}
T = \text{XAuth}(K_1, \ldots, K_\ell, \ldots, K_{\ell+s}) \\
C = (\phi_1, \phi_2, \ldots, \phi_\ell, T)
\end{array}\right.
$$

Constructing multi-bit PKE

Decryption



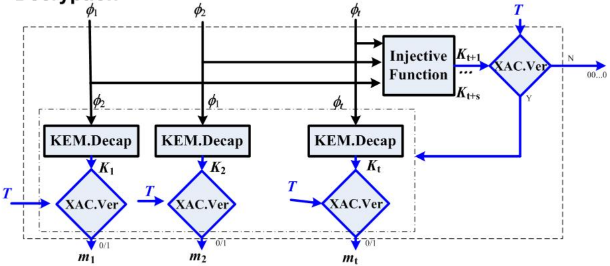
$$\textsf{Dec}(sk, C = (\phi_1, \phi_2, \ldots, \phi_\ell, T)) : F(\phi_1, \phi_2, \ldots, \phi_\ell) \quad \Rightarrow \quad (K_{\ell+1}, \ldots, K_{\ell+s})$$

If $\left(\bigwedge_{j=1}^{s} \text{XVer}(K'_{\ell+j}, T) = 0\right)$ Return($00\ldots0$); Else

$$\begin{cases} \phi_1 & \Rightarrow & \text{KEM.Decap}_{sk_{kem}}(\phi_1) & \Rightarrow & K_1 & \Rightarrow & m_1 := \text{XVer}(K_1, T) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_\ell & \Rightarrow & \text{KEM.Decap}_{sk_{kem}}(\phi_\ell) & \Rightarrow & K_\ell & \Rightarrow & m_\ell := \text{XVer}(K_\ell, T) \end{cases}$$

Constructing multi-bit PKE

Decryption

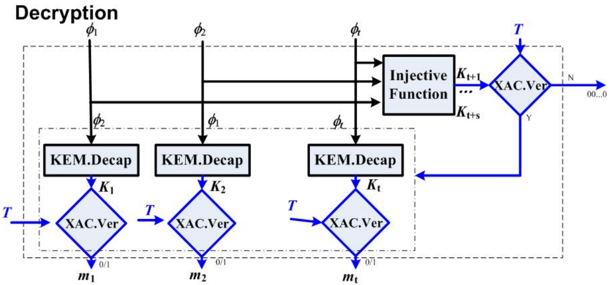


Dec$(sk, C = (\phi_1, \phi_2, \ldots, \phi_\ell, T)) : F(\phi_1, \phi_2, \ldots, \phi_\ell) \quad \Rightarrow \quad (K_{\ell+1}, \ldots, K_{\ell+s})$

If $\left(\bigwedge_{j=1}^{s} \text{XVer}(K'_{\ell+j}, T) = 0\right)$ Return(00...0); Else

$$\begin{cases} \phi_1 \Rightarrow \text{KEM.Decap}_{sk_{kem}}(\phi_1) \Rightarrow K_1 \Rightarrow m_1 := \text{XVer}(K_1, T) \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ \phi_\ell \Rightarrow \text{KEM.Decap}_{sk_{kem}}(\phi_\ell) \Rightarrow K_\ell \Rightarrow m_\ell := \text{XVer}(K_\ell, T) \end{cases}$$

Constructing multi-bit PKE



Dec$(sk, C = (\phi_1, \phi_2, \ldots, \phi_\ell, T)) : F(\phi_1, \phi_2, \ldots, \phi_\ell) \quad \Rightarrow \quad (K_{\ell+1}, \ldots, K_{\ell+s})$

If $\left(\bigwedge_{j=1}^{s} \text{XVer}(K'_{\ell+j}, T) = 0\right)$ Return(00...0); Else

$$\begin{cases} \phi_1 & \Rightarrow & \text{KEM.Decap}_{sk_{kem}}(\phi_1) & \Rightarrow & K_1 & \Rightarrow & m_1 := \text{XVer}(K_1, T) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_\ell & \Rightarrow & \text{KEM.Decap}_{sk_{kem}}(\phi_\ell) & \Rightarrow & K_\ell & \Rightarrow & m_\ell := \text{XVer}(K_\ell, T) \end{cases}$$

Construction of PKE

Correctness:

$m_i = 1$: $(K_i, \phi) \leftarrow$ KEM.Encap, and $T = \mathsf{XAuth}(\ldots, K_i, \ldots)$.

$K_i =$ KEM.Decap(ϕ) and $m_i = \mathsf{XVer}(K_i, T) = 1$.

$m_i = 0$: KEM.Decap$(\phi') = \perp$ or a random key K^R, hence $m_i = \mathsf{XVer}(K^R, T) = 0$.

Requirements for Tailored KEM:

1. Tailored decapsulation: used for correctness. $\forall (pk, sk) \leftarrow$ KEM.Kg(1^κ),

$$\text{KEM.Decap}(\phi') = \begin{cases} \perp & \phi' \leftarrow C \\ K^R & \phi' \leftarrow C \end{cases}$$

2. ESE domains: $\mathcal{K}, C$ are Efficienly Samplable and Explainable domains.

3. Tailored constrained CCA2 security: $(K, \phi) \approx_c (K^R, \phi^R)$ even if adversary $\mathcal{A}$ is given a constrained decryption oracle $\widetilde{\mathsf{Decap}}(\mathsf{XVer}(\cdot, T), \phi)$

$$\widetilde{\mathsf{Decap}}(\mathsf{XVer}(\cdot, T), \phi) = \mathsf{XVer}(\mathsf{Decap}(sk, \phi), T).$$

Construction of PKE

Correctness:

$m_i = 1$: $(K_i, \phi) \leftarrow$ KEM.Encap, and $T =$ XAuth$(\ldots, K_i, \ldots)$.

$K_i =$ KEM.Decap(ϕ) and $m_i =$ XVer$(K_i, T) = 1$.

$m_i = 0$: KEM.Decap$(\phi') = \perp$ or a random key K^R, hence $m_i =$ XVer$(K^R, T) = 0$.

Requirements for Tailored KEM:

1. Tailored decapsulation: used for correctness. $\forall (pk, sk) \leftarrow$ KEM.Kg(1^κ),

$$\text{KEM.Decap}(\phi') = \begin{cases} \perp & \phi' \leftarrow C \\ K^R & \phi' \leftarrow C \end{cases}$$

2. ESE domains: $\mathcal{K}, C$ are Efficienly Samplable and Explainable domains.

3. Tailored constrained CCA2 security: $(K, \phi) \approx_c (K^R, \phi^R)$ even if adversary $\mathcal{A}$ is given a constrained decryption oracle $\widetilde{\text{Decap}}(\text{XVer}(\cdot, T), \phi)$

$$\widetilde{\text{Decap}}(\text{XVer}(\cdot, T), \phi) = \text{XVer}(\text{Decap}(sk, \phi), T).$$

Construction of PKE

Correctness:

$m_i = 1$: $(K_i, \phi) \leftarrow$ KEM.Encap, and T = XAuth$(\ldots, K_i, \ldots)$.

K_i = KEM.Decap(ϕ) and m_i = XVer$(K_i, T) = 1$.

$m_i = 0$: KEM.Decap$(\phi') = \perp$ or a random key K^R, hence m_i = XVer$(K^R, T) = 0$.

Requirements for Tailored KEM:

1. Tailored decapsulation: used for correctness. $\forall(pk, sk) \leftarrow$ KEM.Kg(1^κ),

$$\text{KEM.Decap}(\phi') = \begin{cases} \perp & \phi' \leftarrow C \\ K^R & \phi' \leftarrow C \end{cases}$$

2. ESE domains: $\mathcal{K}, C$ are Efficiently Samplable and Explainable domains.

3. Tailored constrained CCA2 security: $(K, \phi) \approx_c (K^R, \phi^R)$ even if adversary $\mathcal{A}$ is given a constrained decryption oracle $\widetilde{\text{Decap}}(\text{XVer}(\cdot, T), \phi)$

$$\widetilde{\text{Decap}}(\text{XVer}(\cdot, T), \phi) = \text{XVer}(\text{Decap}(sk, \phi), T).$$

Construction of the Simulator

Simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$

- $\mathcal{S}_1$ for $\mathcal{A}_1$: Generate public key with $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$.

  Use sk to answer decryption queries.

- $\mathcal{S}_2$ for $\mathcal{A}_2$: Generate equivocable ciphertexts $C^{(i)} = \mathsf{Enc}(\mathsf{pk}, \overbrace{1 \cdots 1}^{\ell}), i \in [n]$.

- $\mathcal{S}_3$ for $\mathcal{A}_3$: Open equivocable ciphertexts $C^{(i)}$ according to the real message.

  - If $m_j^{(i)} = 1$ open honestly;
  - If $m_j^{(i)} = 0$, $\hat{K}_j^{(i)} \leftarrow \mathsf{ReSamp}(K_{\neq j}^{(i)}, T^{(i)})$ Explain $\left(\phi_j^{(i)}, \hat{K}_j^{(i)}\right)$ as randomly chosen.

$out_{\mathcal{S}} := out_{\mathcal{A}}$.

Security Proof: Hybrid Argument

Tailored constrained CCA2 security of KEM: $(K, \phi) \approx_c (K^R, \phi^R)$ even if adversary $\mathcal{A}$ is given a constrained decryption oracle $\widetilde{\mathsf{Decap}}(\mathsf{XVer}(\cdot, T), \phi)$ and

$$\widetilde{\mathsf{Decap}}(\mathsf{XVer}(\cdot, T), \phi) = \mathsf{XVer}(K, T).$$

Suppose that the first challenger ciphertext is $C = (\phi_1, \phi_2, \phi_3, T)$.

Game 0:	$\phi_1[m_1]$	$\phi_2[m_2]$	$\phi_3[m_3]$	$T = \mathsf{XAuth}(K_1, K_2, K_3, K_4)$
Game 1:	$\phi_1[1]$	$\phi_2[m_2]$	$\phi_3[m_3]$	$T = \mathsf{XAuth}(K_1, K_2, K_3, K_4)$
Game 2:	$\phi_1[1]$	$\phi_2[1]$	$\phi_3[m_3]$	$T = \mathsf{XAuth}(K_1, K_2, K_3, K_4)$
Game 3:	$\phi_1[1]$	$\phi_2[1]$	$\phi_3[1]$	$T = \mathsf{XAuth}(K_1, K_2, K_3, K_4)$

The green parts will use ReSample and Explain to open to 0.

With Hybrid Argument, we have

$$\text{Game 0} \approx_c \text{Game 1} \approx_c \text{Game 2} \approx_c \text{Game 3}.$$

Security Proof: Hybrid Argument

Tailored constrained CCA2 security of KEM: $(K, \phi) \approx_c (K^R, \phi^R)$ even if adversary $\mathcal{A}$ is given a constrained decryption oracle $\widetilde{\mathsf{Decap}}(\mathsf{XVer}(\cdot, T), \phi)$ and

$$\widetilde{\mathsf{Decap}}(\mathsf{XVer}(\cdot, T), \phi) = \mathsf{XVer}(K, T).$$

Suppose that the first challenger ciphertext is $C = (\phi_1, \phi_2, \phi_3, T)$.

Game 0:	$\phi_1[m_1]$	$\phi_2[m_2]$	$\phi_3[m_3]$	$T = \mathsf{XAuth}(K_1, K_2, K_3, K_4)$
Game 1:	$\phi_1[1]$	$\phi_2[m_2]$	$\phi_3[m_3]$	$T = \mathsf{XAuth}(K_1, K_2, K_3, K_4)$
Game 2:	$\phi_1[1]$	$\phi_2[1]$	$\phi_3[m_3]$	$T = \mathsf{XAuth}(K_1, K_2, K_3, K_4)$
Game 3:	$\phi_1[1]$	$\phi_2[1]$	$\phi_3[1]$	$T = \mathsf{XAuth}(K_1, K_2, K_3, K_4)$

The green parts will use ReSample and Explain to open to 0.

With Hybrid Argument, we have

$$\text{Game 0} \approx_c \text{Game 1} \approx_c \text{Game 2} \approx_c \text{Game 3}.$$

Instantiations of Tailored KEM

KEM from Strongly Universal$_2$ Hash Proof Systems:

- Tailored Constrained CCA2 security: $(K, \phi) \approx_c (K^R, \phi^R)$
  - Constrained CCA2 security [HK06]: $(K, \phi) \approx_c (K^R, \phi)$;
  - Subset Membership Problem: $(K^R, \phi) \approx_c (K^R, \phi^R)$
- If HPS has a sparse valid ciphertext set, then a randomly chosen ciphertext will decapsulate to a random key.
- $\mathcal{C}$ and $\mathcal{K}$ can be ESE with some HPS.

Instantiations of Tailored KEM

KEM from the n-Linear assumption: Hofheinz-Kiltz KEM [HK06].

- Tailored Constrained CCA2 security: $(K, \phi) \approx_c (K^R, \phi^R)$.
- Tailored Decapsulation: A sparse valid ciphertext set, and a randomly chosen ciphertext will decapsulate to a random key.
- $\mathcal{C}$ and $\mathcal{K}$ can be ESE with proper groups.

Instantiations of Tailored KEM

KEM from Indistinguishability Obfuscation, a PRG and a PRF [SW2014].

- Tailored Constrained CCA2 security: $(K, \phi) \approx_c (K^R, \phi^R)$.
    - CCA2 security [SW14]: $(K, \phi) \approx_c (K^R, \phi)$;
    - $\text{PRG}(r) = \phi \approx_c \phi^R \in \{0,1\}^{2\kappa} \Rightarrow (K^R, \phi) \approx_c (K^R, \phi^R)$.
- Tailored Decapsulation: A randomly chosen ciphertext will decapsulate to a random key, if the PRF is an extracting one.
- $C = \{0,1\}^{2\kappa}$ and $\mathcal{K} = \{0,1\}^{\kappa}$ are ESE.

Conclusion

- Tailored KEM: we characterise the properties needed of a KEM for our PKE construction to be SIM-SO-CCA secure.
- Three constructions of Tailored KEM: HPS, the n-Linear assumption, and $i\mathcal{O}$.
- We have
    - PKE with SIM-SO-CCA security from HPS and strengthened XACs.
    - PKE with SIM-SO-CCA security from the n-Linear assumption in a way that differs from our HPS-based construction.
    - PKE with SIM-SO-CCA security assuming only the existence of $i\mathcal{O}$ and one-way functions.

Thank You