

Plaintext-Checkable Encryption

Sébastien Canard (Orange Labs)

Georg Fuchsbauer (University of Bristol)

Aline Gouget (Gemalto)

Fabien Laguillaumie (UCBN and ENS Lyon)

CT-RSA 2012



encryption is not always enough

- an encryption scheme permits to hide a confidential information
- but what if one wants to make a search on the encrypted data?
- some practical use cases
 - delegation of keyword search on private databases
 - delegation of search to an email gateway
- different cases
 - case of public vs. private database
 - case of public vs. secret words



related work

- based on symmetric-key cryptography (out of scope)
- decryptable searchable encryption
 - initial work from Ostrovsky and Skeith [JoC07]
 - decryptable version by Fuhr and Paillier [ProvSec07]
 - use of a trapdoor to make the search
 - from c and $\text{Trap}(\text{tk}, m)$, check if $c = \text{Enc}(\text{pk}, m)$
- encryption with equality test
 - proposed by Yang, Tan, Huang and Wong [CT-RSA10]
 - search using a candidate ciphertext
 - from c_1 and c_2 , public check if $\text{Dec}(\text{sk}, c_1) = \text{Dec}(\text{sk}, c_2)$

introduction to PCE

- PCE stands for *plaintext-checkable encryption*
- what do we mean by “plaintext-checkable”?
 - we DO NOT need a trapdoor
 - we DO NOT need a candidate ciphertext
 - we only need a candidate **plaintext**
- from c and m , public **check** if $c = \text{Enc}(\text{pk}, m)$

agenda

- security definition of PCE
- a generic construction in the ROM
- a practical construction in the standard model
- application to VLR group signatures

security definition of PCE

definition for a PCE

- as for a standard encryption scheme
 - $c \leftarrow \text{Encrypt}(pk, m)$
 - $m \leftarrow \text{Decrypt}(c, sk)$
- additional public algorithm: $\text{PCheck}(c, m)$ returns
 - 1 if c is an encryption of m
 - 0 otherwise
- what can we expect for security property?
 - regarding *indistinguishability (IND)*
 - we focus on a *Chosen Plaintext Attack (CPA)* adversary
 - similar work can be done in the Chosen Ciphertext Attack (CCA) case

IND-CPA?

- let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a probabilistic encryption scheme
- experiment $\mathbf{Exp}_{\Pi, \mathcal{A}}^{\text{ind-cpa}}(k)$
 - $b \xleftarrow{\$} \{0, 1\}$
 - $(pk, sk) \leftarrow \mathcal{G}(1^k)$
 - $(m_0, m_1, st) \leftarrow \mathcal{A}_f(1^k, pk)$
 - $c \leftarrow \mathcal{E}(1^k, pk, m_b)$
 - $b' \leftarrow \mathcal{A}_g(1^k, c, st)$
 - return $(b' = b)$
- $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$ can easily win this experiment if Π is a PCE
 - \mathcal{A}_g knows m_0 and m_1 (by st)
 - \mathcal{A}_g can make use of the PCheck procedure with c and e.g. m_0
- what else regarding indistinguishability?

notion of high min-entropy

- the adversary can always use the PCheck procedure to test if a randomly chosen message works
 \implies *the adversary should not be able to retrieve a given unknown message “by chance”*

Definition (High min-entropy)

- we say that an adversary $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$ has *min-entropy* μ if

$$\forall k \in \mathbb{N} \ \forall c \ \forall m : \Pr [m' \leftarrow \mathcal{A}_f(1^k, c) : m' = m] \leq 2^{-\mu(k)} .$$

- \mathcal{A} is said to have *high min-entropy* if it has min-entropy μ with $\mu(k) \in \omega(\log k)$.

IND-DET?

- let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a deterministic encryption scheme
- experiment $\mathbf{Exp}_{\Pi, \mathcal{A}}^{\text{ind-det}}(k)$
 - $b \xleftarrow{\$} \{0, 1\}$
 - $m \leftarrow \mathcal{A}_f(1^k, b)$
 - $(pk, sk) \leftarrow \mathcal{G}(1^k)$
 - $c \leftarrow \mathcal{E}(1^k, pk, m)$
 - $b' \leftarrow \mathcal{A}_g(1^k, pk, c)$
 - return $(b' = b)$
- $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$ should have high min-entropy
- definition* given by Bellare, Fischlin, O'Neill, Ristenpart [Crypto08]
- it seems to work, but this may be not enough
- can we do better?

* Here in the case of a message, and not a vector of messages

a new notion called UNLINK

- infeasibility to decide if two ciphertexts encrypt the same message
- let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an encryption scheme
- experiment $\mathbf{Exp}_{\Pi, \mathcal{A}}^{\text{unlink}}(k)$
 - $b \xleftarrow{\$} \{0, 1\}$
 - $(pk, sk) \leftarrow \mathcal{G}(1^k)$
 - $m_0 \leftarrow \mathcal{A}_f(1^k, pk)$
 - $m_1 \leftarrow \mathcal{A}_f(1^k, pk)$
 - $c_0 \leftarrow \mathcal{E}(1^k, pk, m_b)$
 - $c_1 \leftarrow \mathcal{E}(1^k, pk, m_1)$
 - $b' \leftarrow \mathcal{A}_g(1^k, pk, c_0, c_1)$
 - return $(b' = b)$
- $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$ should have high min-entropy
 - otherwise, \mathcal{A} can easily win the experiment

relation between security properties

- in the paper, we show that
 - every scheme that achieves IND-CPA achieves UNLINK
 - every scheme that achieves UNLINK achieves IND-DET

$$\text{IND-CPA} \subsetneq \text{UNLINK} \subsetneq \text{IND-DET}.$$

- UNLINK is most of time sufficient (see group signature with VLR)
- UNLINK is the best we can hope for a PCE scheme

can we reach the UNLINK property?

- an IND-CPA probabilistic scheme cannot be plaintext-checkable
- an IND-DET deterministic scheme cannot reach UNLINK
- using an encryption scheme with equality test
 - encrypt the putative message m and make use of the “equality test” procedure
 - this scheme does not reach UNLINK since the adversary can do the same
- using a decryptable searchable encryption
 - it seems to work... but can we do better?
- three constructions in the paper
 - one based on any probabilistic encryption scheme, in the ROM
 - one based on any deterministic encryption scheme, in the ROM
 - one based on ElGamal and secure in the standard model

a generic construction in the ROM

in a nutshell

- useful cryptographic tools
 - let $\Pi_p = (\mathcal{G}_p, \mathcal{E}_p, \mathcal{D}_p)$ be an IND-CPA probabilistic encryption scheme
 - let $\mathcal{H} : \{0, 1\}^* \longrightarrow \{0, 1\}^{\ell(k)}$ be a hash function modeled as a random oracle
- high-level idea
 - the message m is encrypted using Π_p
 - the random coin of $\Pi_p \cdot \mathcal{E}_p$ is computed using the message m and some randomly chosen r
 - r is given together with the resulting ciphertext
 - the PCheck procedure consists in re-computing the random coin, using r and the putative message m

in details

Algorithm KeyGen(1^k)

$(\overline{pk}, \overline{sk}) \xleftarrow{\$} \Pi_p.\mathcal{G}_p(1^k)$
 $pk \leftarrow \overline{pk}$
 $sk \leftarrow \overline{sk}$
return (pk, sk)

Algorithm Decrypt($1^k, sk, C$)

$(\overline{c}, r) \leftarrow C$
 $sk \leftarrow sk$
 $m \leftarrow \Pi_p.\mathcal{D}_p(1^k, \overline{sk}, \overline{c})$
return m

Algorithm Encrypt($1^k, pk, m$)

$\overline{pk} \leftarrow pk$
 $r \xleftarrow{\$} \{0, 1\}^{\ell(k)}$
 $\rho \leftarrow \mathcal{H}(m||r)$
 $\overline{c} \leftarrow \Pi_p.\mathcal{E}_p(1^k, \overline{pk}, m; \rho)$
 $C \leftarrow (\overline{c}, r)$
return C

Algorithm PCheck($1^k, pk, C, m$)

$(\overline{c}, r) \leftarrow C$
 $pk \leftarrow pk$
 $\rho \leftarrow \mathcal{H}(m||r)$
 $\tilde{c} \leftarrow \Pi_p.\mathcal{E}_p(1^k, \overline{pk}, m; \rho)$
if $\tilde{c} = \overline{c}$ then return 1
else return 0

Theorem

If Π_p satisfies IND-CPA, then the above PCE scheme satisfies UNLINK, in the random oracle model.

a practical construction in the
standard model

based on ElGamal

- in an asymmetric bilinear group setting
 - p is a prime number
 - \mathbb{G}_1 , \mathbb{G}_2 and \mathbb{G}_T are cyclic groups of order p
 - $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ is a non-degenerated bilinear map
 - $\forall g, h \in \mathbb{G}_1 \times \mathbb{G}_2, \forall a, b \in \mathbb{Z}_p, e(g^a, h^b) = e(g, h)^{ab}$
 - g (resp. h) is a generator of \mathbb{G}_1 (resp. \mathbb{G}_2)
- remember ElGamal
 - secret key $x \in \mathbb{Z}_p^*$, public key $y = g^x$
 - given $m \in \mathbb{G}_1$, choose $r \in_R \mathbb{Z}_p^*$
 - $c = (T_1, T_2)$ with $T_1 = my^r$ and $T_2 = g^r$

in a nutshell

- message m
- random coin $r \in_R \mathbb{Z}_p^*$
- ciphertext $T_1 = my^r$, $T_2 = g^r$
- adding $T_4 = h^r$
 - PCheck becomes possible, using a putative m
 - test if $e(T_1 m^{-1}, h) = e(y, T_4)$
 - but we do not achieved UNLINK
 - given $c_0 = \text{Encrypt}(1^k, pk, m_b)$ and $c_1 = \text{Encrypt}(1^k, pk, m_1)$
 - test whether “ c_0/c_1 ” encrypts 1, using PCheck!

in a nutshell

- message m
- random coin $r \in_R \mathbb{Z}_p^*$
- ciphertext $T_1 = my^r$, $T_2 = g^r$
- we use a random base $T_3 = h^a$
- adding $T_4 = (h^a)^r$
 - PCheck is still possible
 - test if $e(T_1 m^{-1}, T_3) = e(y, T_4)$
 - we achieve UNLINK

in details

Algorithm KeyGen(1^k)

$x \xleftarrow{\$} \mathbb{Z}_p^*$
 $y \leftarrow g^x$
 $(pk, sk) \leftarrow (y, x)$
return (pk, sk)

Algorithm Decrypt($1^k, sk, C$)

$x \leftarrow sk$
 $(T_1, T_2, T_3, T_4) \leftarrow C$
if $e(g, T_4) \neq e(T_2, T_3)$ then return \perp
 $m \leftarrow T_1/T_2^x$
return m

Algorithm Encrypt($1^k, pk, m$)

$y \leftarrow pk$
 $r, a \xleftarrow{\$} \mathbb{Z}_p^*$
 $C \leftarrow (my^r, g^r, h^a, h^{ar})$
return C

Algorithm PCheck($1^k, pk, C, m$)

$y \leftarrow pk$
 $(T_1, T_2, T_3, T_4) \leftarrow C$
if $e(g, T_4) \neq e(T_2, T_3)$ then return 0
if $e(T_1/m, T_3) = e(y, T_4)$ then return 1
else return 0

Theorem

Under a new assumption, the proposed construction is a PCE scheme which is UNLINK against adversaries outputting the uniform distribution.*

*This assumption, related to both DDH and DLIN, is secure in the generic-group model.

application to VLR group signatures

VLR group signatures

- group signatures
 - introduced by Chaum and van Heyst in 1991
 - permit group members to anonymously sign messages on behalf of the group
 - anonymity revocation by a designated authority
- membership revocation
 - not easy as the signer is anonymous!
 - based on the verifier local revocation (VLR)
 - introduced by Boneh-Shacham [ACM-CCS04]
 - the verifier has to test each entry of a revocation list before accepting a group signature
- our contributions
 - PCE is a new building block for VLR group signatures
 - instantiation by a very efficient scheme

adding the VLR property

- backward unlinkability
 - the revocation of a member should not compromise the anonymity of her previously generated group signatures
 - we use the idea of Nakanishi and Funabiki [Asiacrypt05]
⇒ time is divided into periods
- construction of one revocation token $tk[i, j]$ per group member i and time period j
 - revocation at j_0 ⇒ publication of the revocation tokens for all $j > j_0$
 - used by the verifier to check the revocation
 - $tk[i, j]$ cannot be revealed as it compromised the anonymity
 - idea: output a PCE of $tk[i, j]$
- instantiation using the Abe *et al.* group signatures [Crypto10]
 - based on automorphic signatures
 - based on Groth-Sahai NIWI proof system

additional remarks

- security
 - standard model (under the assumption that the PCE scheme is UNLINK)
 - with backward unlinkability
 - (with anonymity revocation)
- efficiency (comparison with the Libert-Vergnaud (LV) scheme)
 - group signature size = $12|\mathbb{G}_1| + 18|\mathbb{G}_2|$ (better than LV)
 - signer's work: 6 modular exponentiations, 1 quadratic GS proof, 5 linear GS proofs (better than LV)
 - revocation check: 2 pairing computations per element in the revocation list (LV is better)

conclusion

- we have introduced a new cryptographic tool
- we have provided several concrete instantiations
 - generic construction in the ROM
 - a practical construction in the standard model
- we have proved its usability
 - in the case of data search in databases or in cloud storage
 - in the case of VLR group signature schemes

thank you



Generic Construction of Chosen Ciphertext Secure Proxy Re-Encryption

Yutaka Kawai
The University of Tokyo

Session ID: CRYP-401


Session Classification: Advanced

RSACONFERENCE2012

Table of Contents

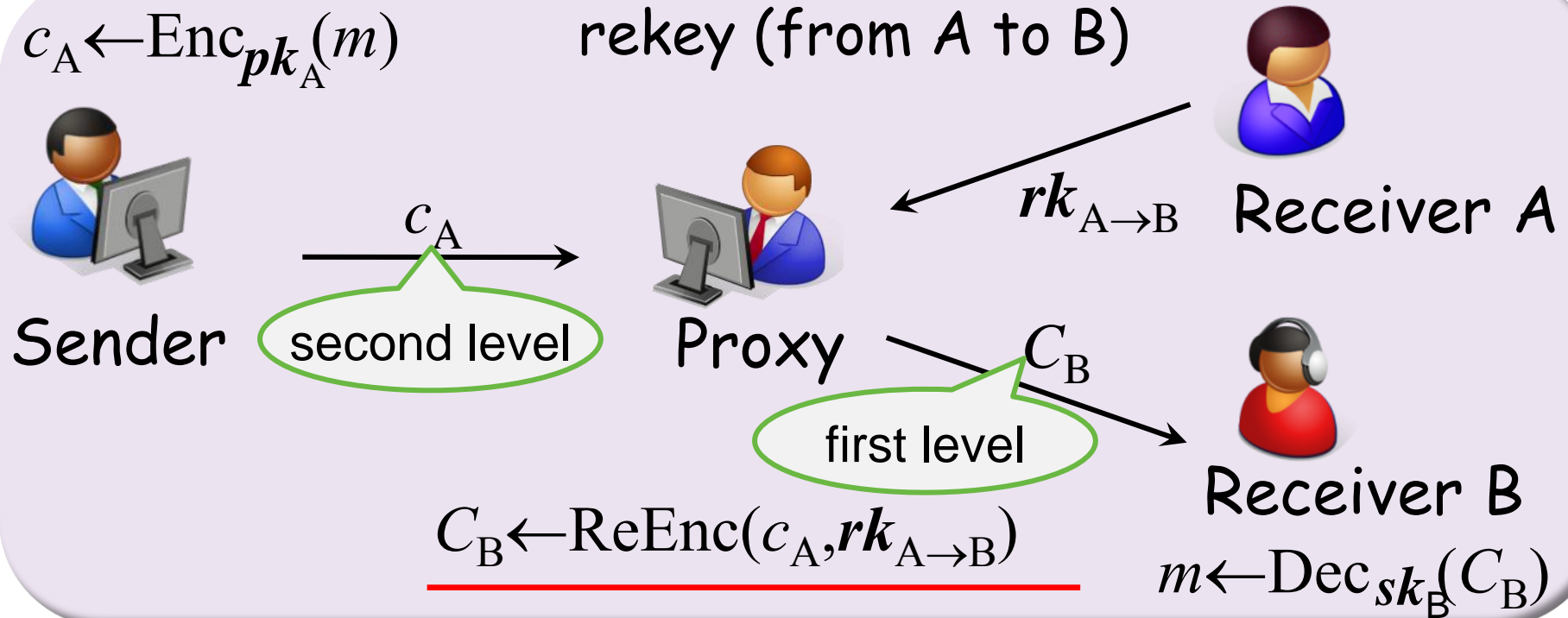
- Single Use Unidirectional Proxy Re-Encryption (SUPRE)
- Main difficulty of CCA secure SUPRE
- Generic construction of single use proxy re-encryption





Single Use Unidirectional Proxy Re-Encryption

Single Use Proxy Re-Encryption



PRE allows a proxy to convert a ciphertext encrypted under one key into an encryption of the same message under another key.



Previous Works

Scheme	Uni/Bi	Security model	ROM/ STM	Pairing computation
[AFGH06]	Uni	CPA	ROM	✓
[HRSV07]	Uni	CPA	STM	✓
[CH07]	Bi	CCA	STM	✓
[LV08]	Uni	RCCA (weak CCA)	STM	✓
[AABH09]	Uni	CPA	ROM	✓
[CWYD10]	Uni	CCA + several restrictions	ROM	
Ours	Uni	CCA	STM	✓



Motivations

- In previous CCA security of SUPRE, CCA security definition **is not strong enough.**
- CCA secure SUPRE **in the standard model** was not proposed in previous works.



Our Contribution

- We define **CCA** security of SUPRE.
- We present the **first generic construction** of CCA secure SUPRE.
 - there are three building blocks in our generic construction: **CCA secure PKE**, **strong unforgeable digital signature** and **Resplittable CCA secure Threshold PKE**.
 - Resplittable TPKE is a new primitive.

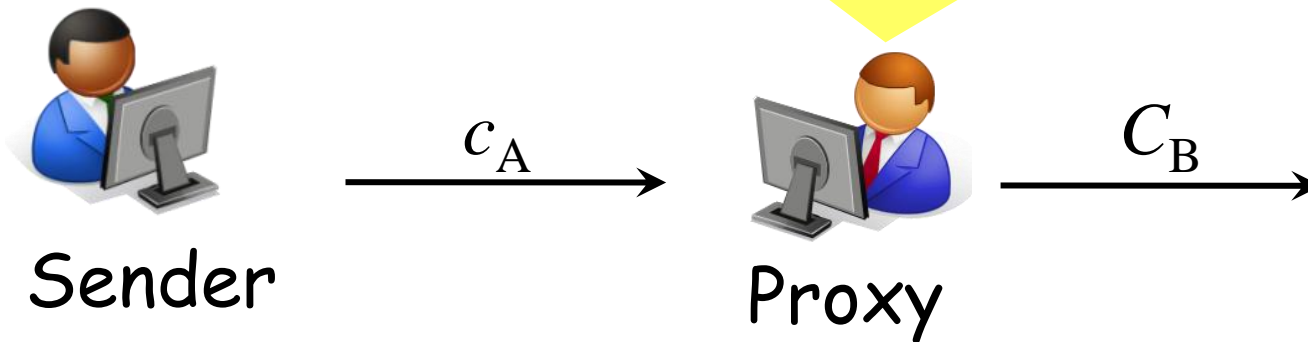




Main Difficulty of CCA secure SUPRE

Main Difficulty of CCA secure SUPRE

When proxy computes first level ciphertext C_B from C_A , he should determine **whether a ciphertext is valid**.



Proxy should check the validity of C_A
without the secret key of A.



Main Difficulty of CCA secure SUPRE

- CCA Security (>Non-Malleability):

A ciphertext is **not converted** meaningfully.

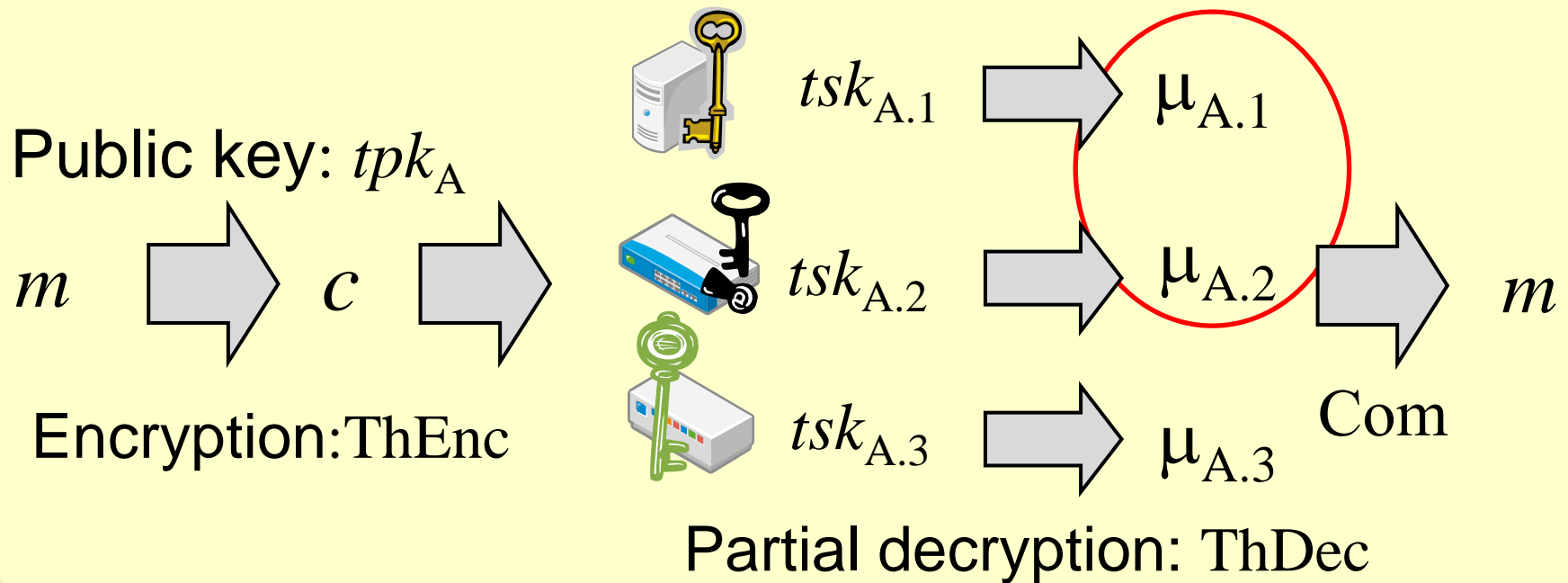
- Proxy Re-Encryption

A second-level ciphertext **can be converted** another ciphertext of the same message under another key.



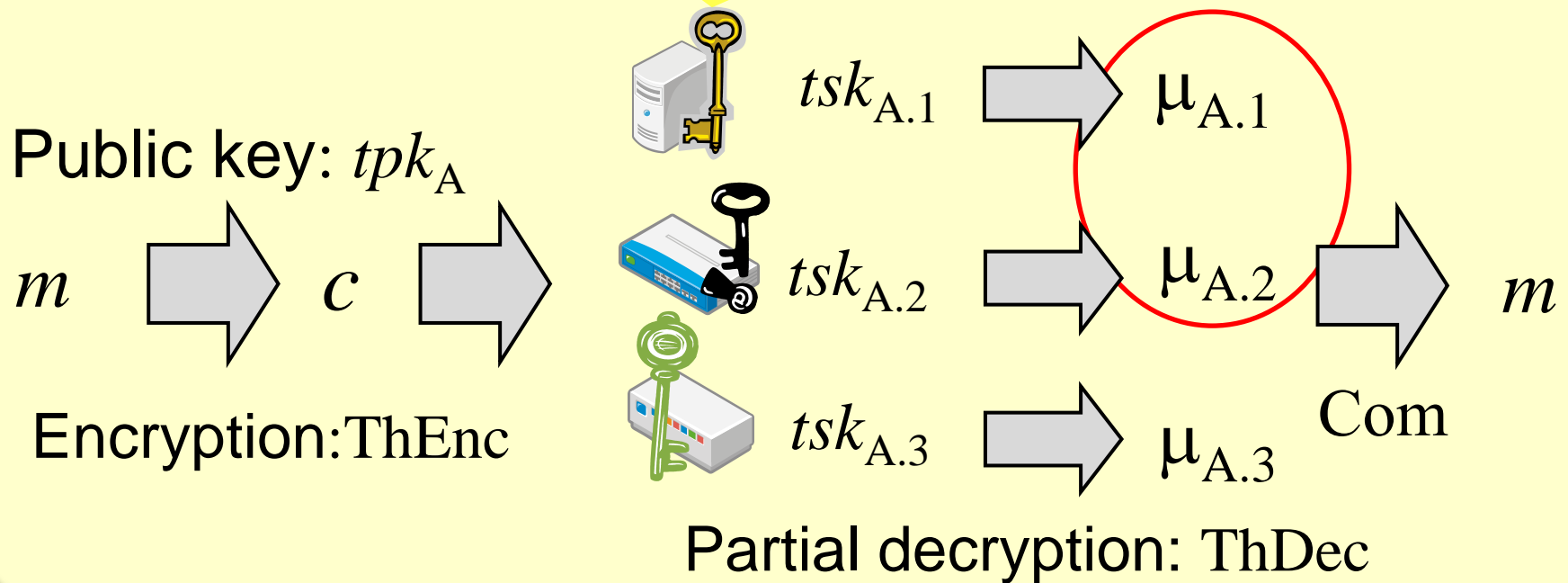
Our Main Idea

We focus on **CCA secure**
Threshold Public Key Encryption (TPKE).



Our Main Idea

Each decryption server should determine
whether a ciphertext is valid without decrypting c .

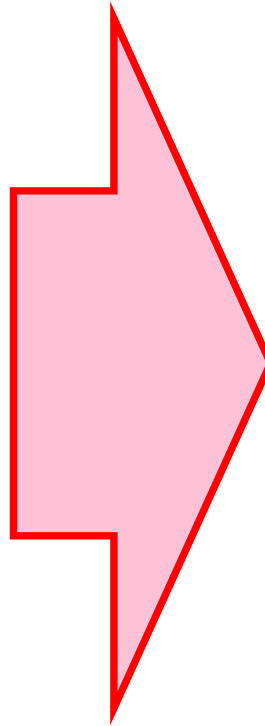


Our Contribution

CCA secure PKE

Strongly Unforgeable
Signature

Resplittable CCA
secure TPKE



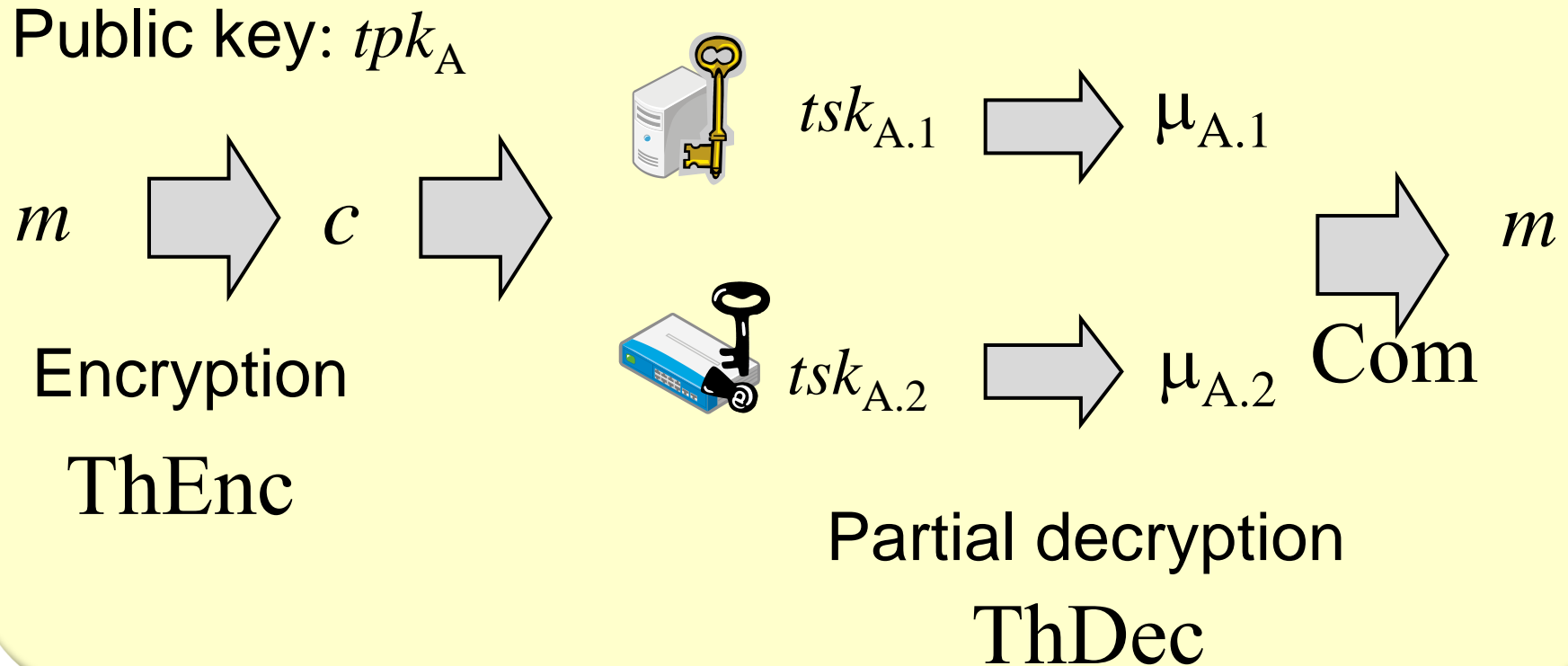
CCA secure
SUPRE



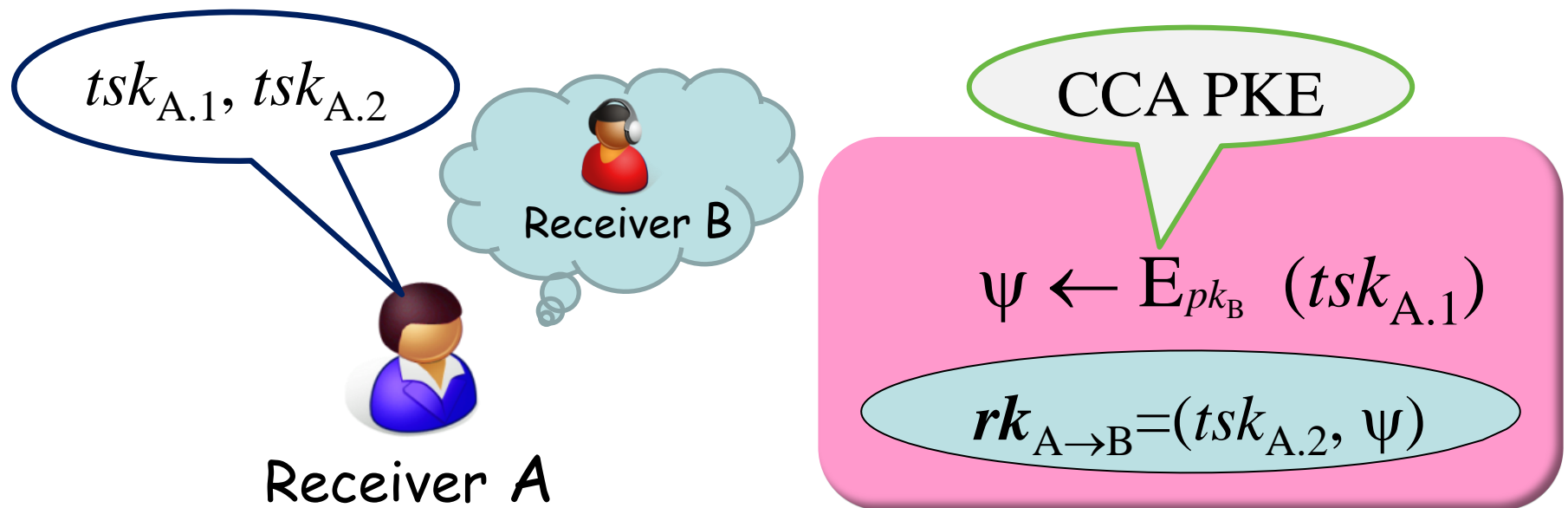


Generic Construction

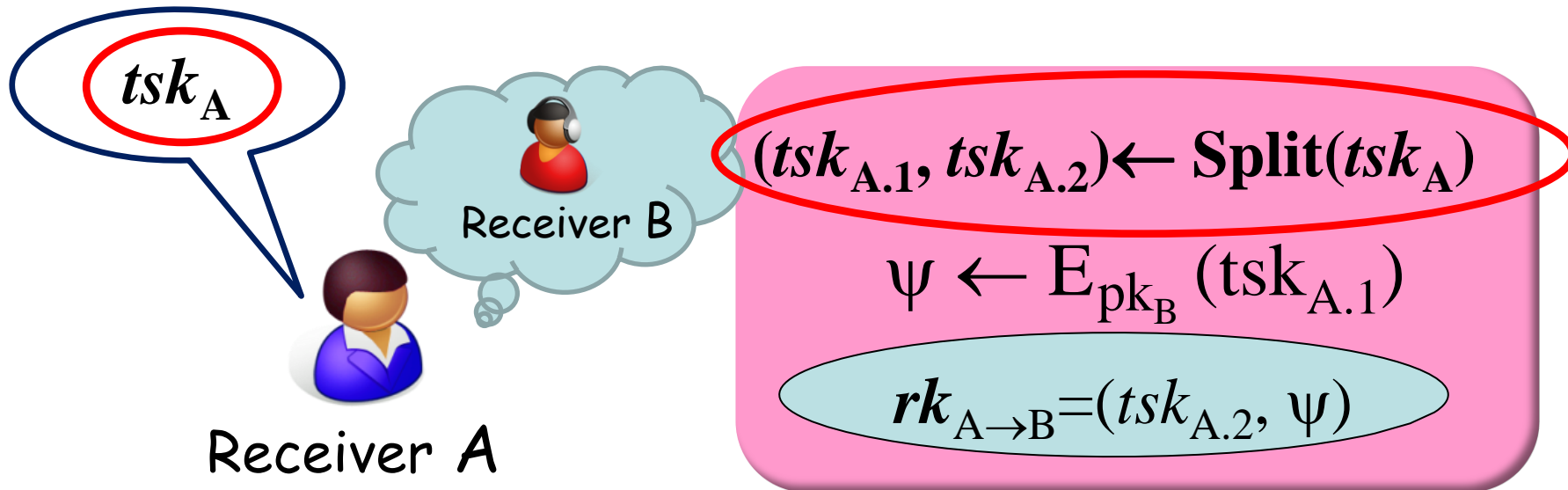
Main Building Block : 2-out-of-2 Threshold Public Key Encryption



Basic Idea: Rekey Generation Algorithm



Resplittability of TPKE



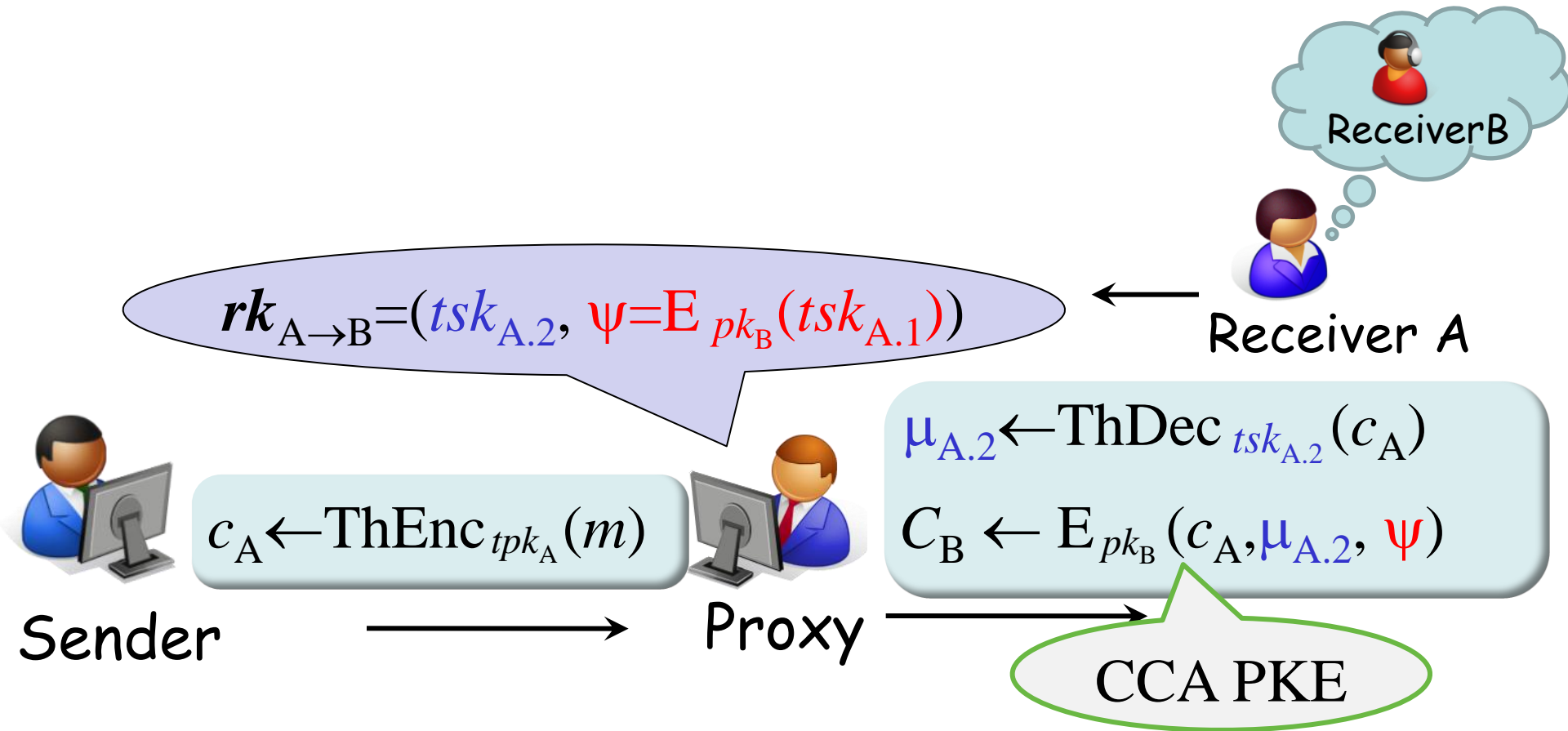
We define Resplittable TPKE and its security requirements.

Boneh, Boyen, and Halevi
(CT-RSA 2006)
Arita and Tsurudome
(ACNS 2009)

are examples of
Resplittable TPKE.



Basic Idea: Encryption and Re-Encryption



Basic Idea: First-Level Decryption



$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi = E_{pk_B}(tsk_{A.1}))$$

$$\mu_{A.2} \leftarrow \text{ThDec}_{tsk_{A.2}}(c_A)$$

$$C_B \leftarrow E_{pk_B}(c_A, \mu_{A.2}, \psi)$$

$$dk_B$$

C_B → Receiver B

$$\langle c_A, \mu_{A.2}, \psi \rangle \leftarrow D_{dk_B}(C_B)$$

$$tsk_{A.1} \leftarrow D_{dk_B}(\psi)$$

$$\mu_{A.1} \leftarrow \text{ThDec}_{tsk_{A.1}}(c_A)$$

$$m \leftarrow \text{Com}(\mu_{A.1}, \mu_{A.2})$$





Proxy

The malicious proxy might encrypt another (invalid) μ .

$$\mu_{A.2} \leftarrow \text{ThDec}_{tsk_{A.2}}(c_A)$$

$$C_B \leftarrow E_{pk_B}(c_A, \mu'_{A.2}, \psi)$$

B have to check the validity of μ .



C_B

Receiver B

$$\langle c_A, \mu'_{A.2}, \psi \rangle \leftarrow D_{dk_B}(C_B)$$

$$tsk_{A.1} \leftarrow D_{dk_B}(\psi)$$

$$\mu_{A.1} \leftarrow \text{ThDec}_{tsk_{A.1}}(c_A)$$

$$m' \leftarrow \text{Com}(\mu_{A.1}, \mu'_{A.2})$$



Robustness of TPKE

$\text{ThV}(c, \mu, tvk) \rightarrow (\text{in})\text{valid}$

Boneh, Boyen, and Halevi
(CT-RSA 2006)
Arita and Tsurudome
(ACNS 2009)

are examples of
Robustness TPKE.

using Paring Computation



Modified Scheme



Proxy

$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi = E_{pk_B}(tsk_{A.1}), tvk)$$

$$\mu_{A.2} \leftarrow \text{ThDec}_{tsk_{A.2}}(c_A)$$

$$C_B \leftarrow E_{pk_B}(c_A, \mu_{A.2}, \psi, tvk)$$



$C_B \rightarrow$ Receiver B

$$\langle c_A, \mu_{A.2}, \psi, tvk \rangle \leftarrow D_{dk_B}(C_B)$$

$$\text{If valid} \leftarrow \text{ThV}(c_A, \mu_{A.2}, tvk)$$

$$tsk_{A.1} \leftarrow D_{dk_B}(\psi)$$

$$\mu_{A.1} \leftarrow \text{ThDec}_{tsk_{A.1}}(c_A)$$

$$m \leftarrow \text{Com}(\mu_{A.1}, \mu_{A.2})$$



Modified Scheme



Proxy

$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi = E_{pk_B}(tsk_{A.1}), tvk)$$

$$\mu_{A.2} \leftarrow \text{ThDec}_{tsk_{A.2}}(c_A)$$

$$C_B \leftarrow E_{pk_B}(c_A, \mu_{A.2}, \psi, tvk' \neq tvk)$$

B cannot check whether *tvk* is generated by the **original receiver**.



$C_B \rightarrow$ Receiver B

$$\langle c_A, \mu_{A.2}, \psi, tvk' \rangle \leftarrow D_{dk_B}(C_B)$$

If invalid $\leftarrow \text{ThV}(c_A, \mu_{A.2}, tvk')$

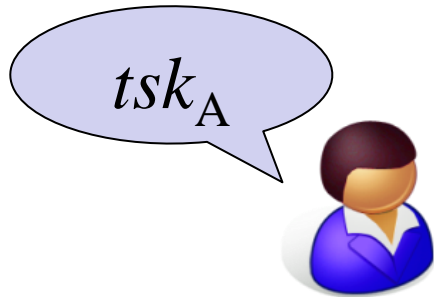
$$tsk_{A.1} \leftarrow D_{dk_B}(\psi)$$

$$\mu_{A.1} \leftarrow \text{ThDec}_{tsk_{A.1}}(c_A)$$

$$m \leftarrow \text{Comb}(\mu_{A.1}, \mu_{A.2})$$



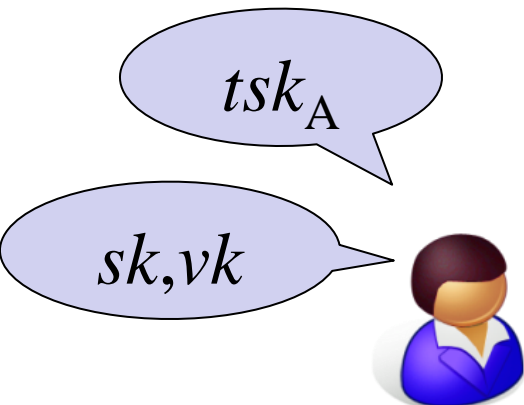
Rekey Generation Algorithm



Receiver A

$$(tvk, tsk_{A.1}, tsk_{A.2}) \leftarrow \text{Split}(tsk_A)$$
$$\psi \leftarrow E_{pk_B}(tsk_{A.1})$$

$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi, tvk)$$



Receiver A

$$(tvk, tsk_{A.1}, tsk_{A.2}) \leftarrow \text{Split}(tsk_A)$$
$$\psi \leftarrow E_{pk_B}(tsk_{A.1})$$

$$\sigma \leftarrow \text{Sig}_{sk_A}(\psi, tvk)$$

$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi, tvk, \sigma)$$



$$(tsk_{A.2}, \psi = E_{pk_A}(tsk_{A.1}), tvk, \sigma)$$

$$c_A \leftarrow \text{ThEnc}_{tpk_A}(m)$$

$$\begin{aligned} \mu_{A.2} &\leftarrow \text{ThDec}_{tsk_{A.2}}(c_A) \\ C_B &\leftarrow E_{pk_B}(c_A, \mu_{A.2}, \psi, tvk, \sigma) \end{aligned}$$

$$\begin{aligned} \langle c_A, \mu_{A.2}, \psi, tvk, \sigma \rangle &\leftarrow D_{dk_B}(C_B) \\ \text{If } \text{valid} &\leftarrow \text{Ver}_{vk_A}(\langle \psi, tvk \rangle, \sigma) \\ \text{valid} &\leftarrow \text{ThV}(c_A, \mu_{A.2}, tvk) \\ tsk_{A.1} &\leftarrow D_{dk_B}(\psi) \\ \mu_{A.1} &\leftarrow \text{ThDec}_{tsk_{A.1}}(c_A) \\ m &\leftarrow \text{Com}(\mu_{A.1}, \mu_{A.2}) \end{aligned}$$

$$dk_B$$

$$C_B$$

Receiver B



Conclusion

- We define **CCA** security of PRE.
- We present the **first generic construction** of CCA secure Single Use PRE.
- Via our generic construction, we present first construction which is **CCA secure in the standard model.**






Apply Slide

Apply Slide

- In order to use CCA secure SUPRE in cloud computing services, we should construct specific and efficient scheme by reference to our proposed generic construction.
- We should discuss whether our generic construction is secure under other security requirements.





Generic Construction of Chosen Ciphertext Secure Proxy Re-Encryption

Goichiro Hanaoka¹, **Yutaka Kawai**²,
Noboru Kunihiro², Takahiro Matsuda¹, Jian Weng³,
Rui Zhang⁴, and Yunlei Zhao⁵

¹National Institute of Advance Industrial Science and Technology

²**The University of Tokyo**

³Jinan University

⁴Chinese Academy of Sciences

⁵Fudan University

Session ID: CRYPT-401


Session Classification: Advanced

RSACONFERENCE2012

Agenda

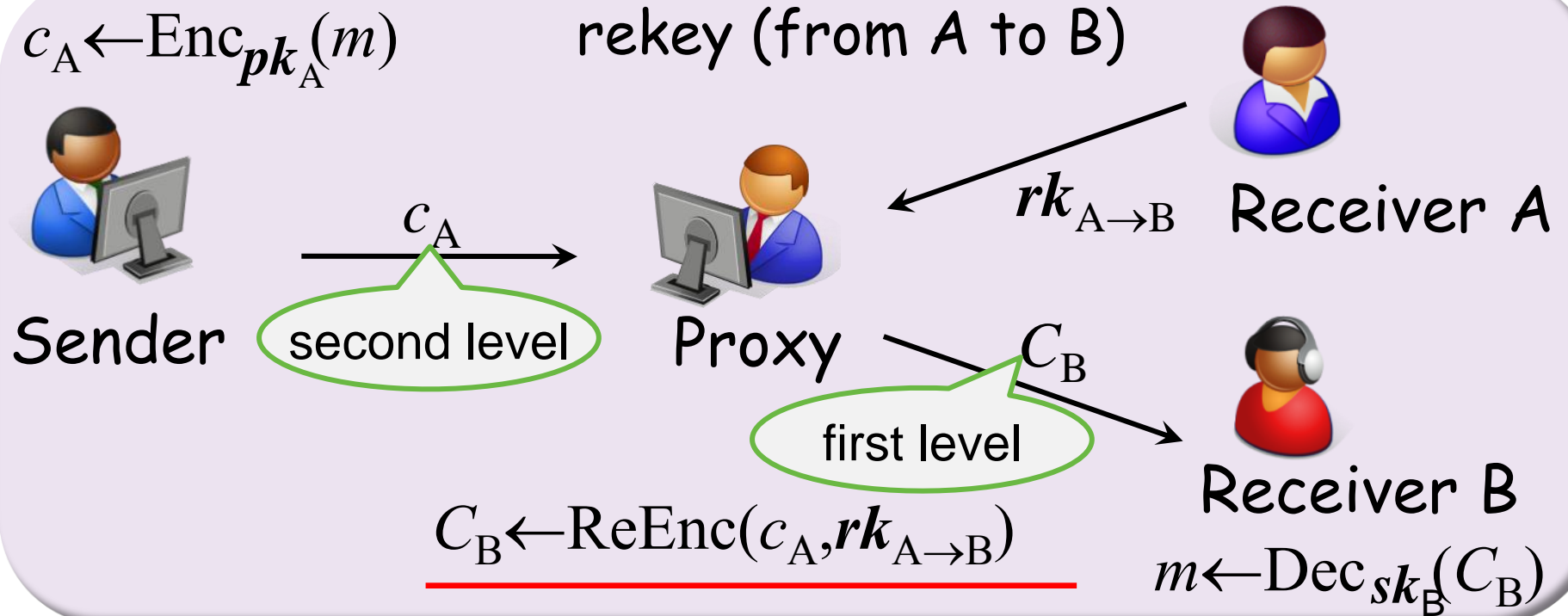
- Single Use Unidirectional Proxy Re-Encryption (SUPRE)
- Main difficulty of CCA secure SUPRE
- Main Idea
- Generic construction of CCA secure SUPRE





Single Use Unidirectional Proxy Re-Encryption

Single Use Proxy Re-Encryption



PRE allows a proxy to convert a ciphertext encrypted under one key into an encryption of the same message under another key.



Previous Works

Scheme	Uni/Bi	Security model	ROM/ STM	Pairing computation
[AFGH06]	Uni	CPA	ROM	✓
[HRSV07]	Uni	CPA	STM	✓
[CH07]	Bi	CCA	STM	✓
[LV08]	Uni	RCCA (weak CCA)	STM	✓
[AABH09]	Uni	CPA	ROM	✓
[CWYD10]	Uni	CCA + several restrictions	ROM	
Ours	Uni	CCA	STM	✓



Motivations

- In previous CCA security of SUPRE, CCA security definition **is not strong enough.**
- CCA secure SUPRE **in the standard model** was not proposed in previous works.



Our Contribution

- We define **CCA** security of SUPRE.
- We present the **first generic construction** of CCA secure SUPRE.
 - There are three building blocks in our generic construction:
CCA secure PKE,
Strong unforgeable digital signature and
Resplittable CCA secure Threshold PKE.
 - Resplittable TPKE is a new primitive.





Main Difficulty of CCA secure SUPRE

Main Difficulty of CCA secure SUPRE

➤ CCA Security:

A ciphertext is **not converted** meaningfully.

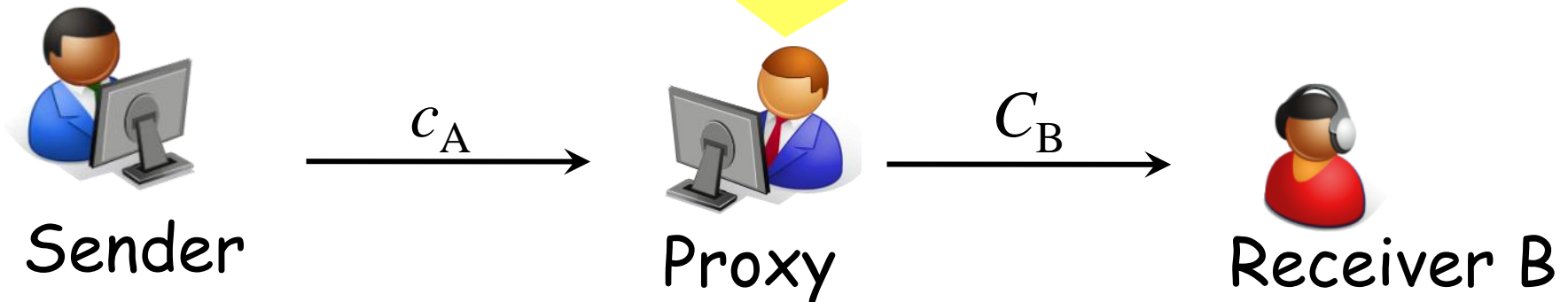
➤ Proxy Re-Encryption

A second-level ciphertext **can be converted** another ciphertext of the same message under another key.



Main Difficulty of CCA secure SUPRE

When a proxy computes first level ciphertext C_B from c_A , he should determine **whether a ciphertext is valid**.

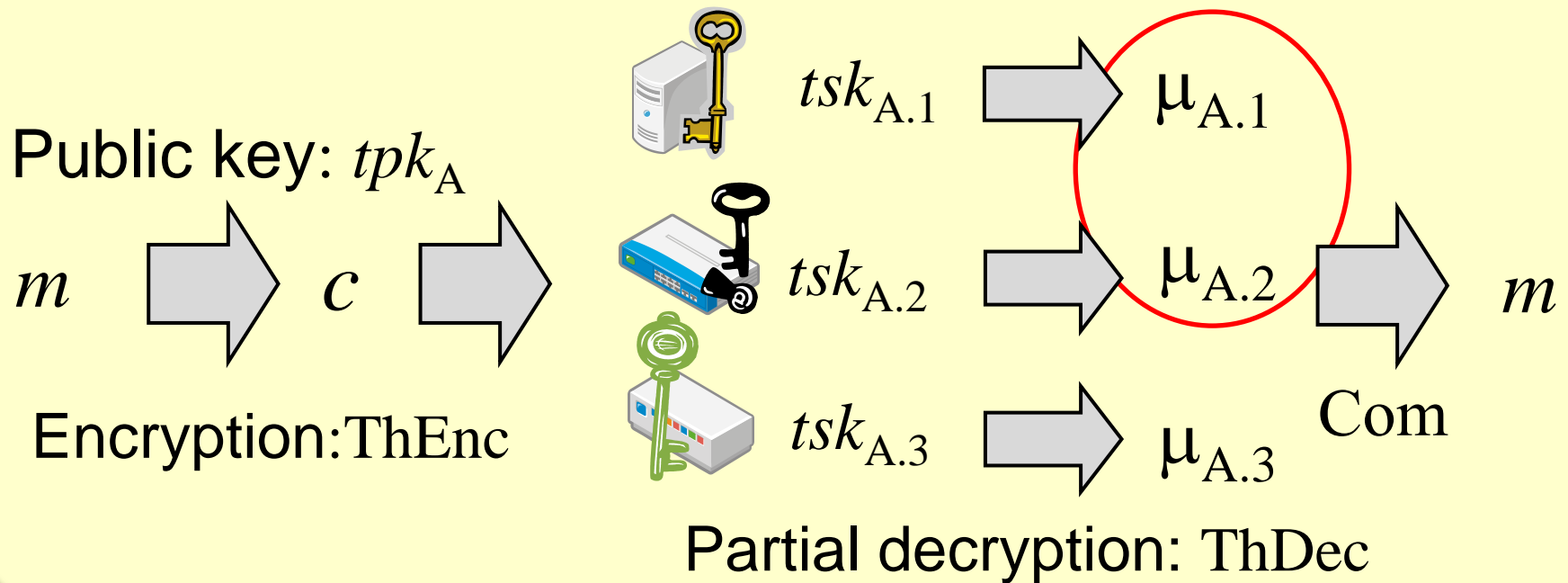


Proxy should check the validity of c_A
without the secret key of A.



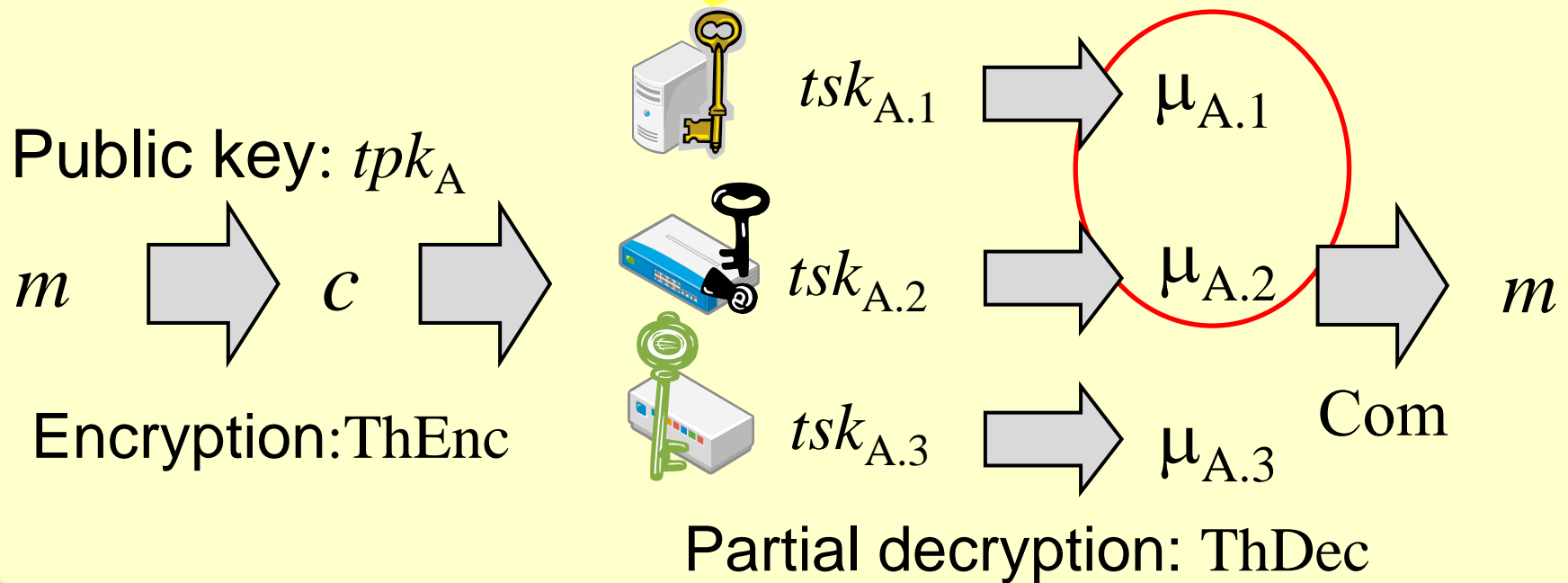
Our Main Idea

We focus on **CCA secure**
Threshold Public Key Encryption (TPKE).



Our Main Idea

Each decryption server should determine
whether a ciphertext is valid without decrypting c .

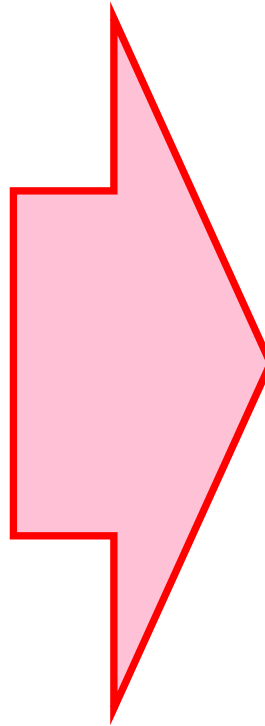


Our Contribution

CCA secure PKE

Strongly Unforgeable
Signature

Resplittable CCA
secure TPKE



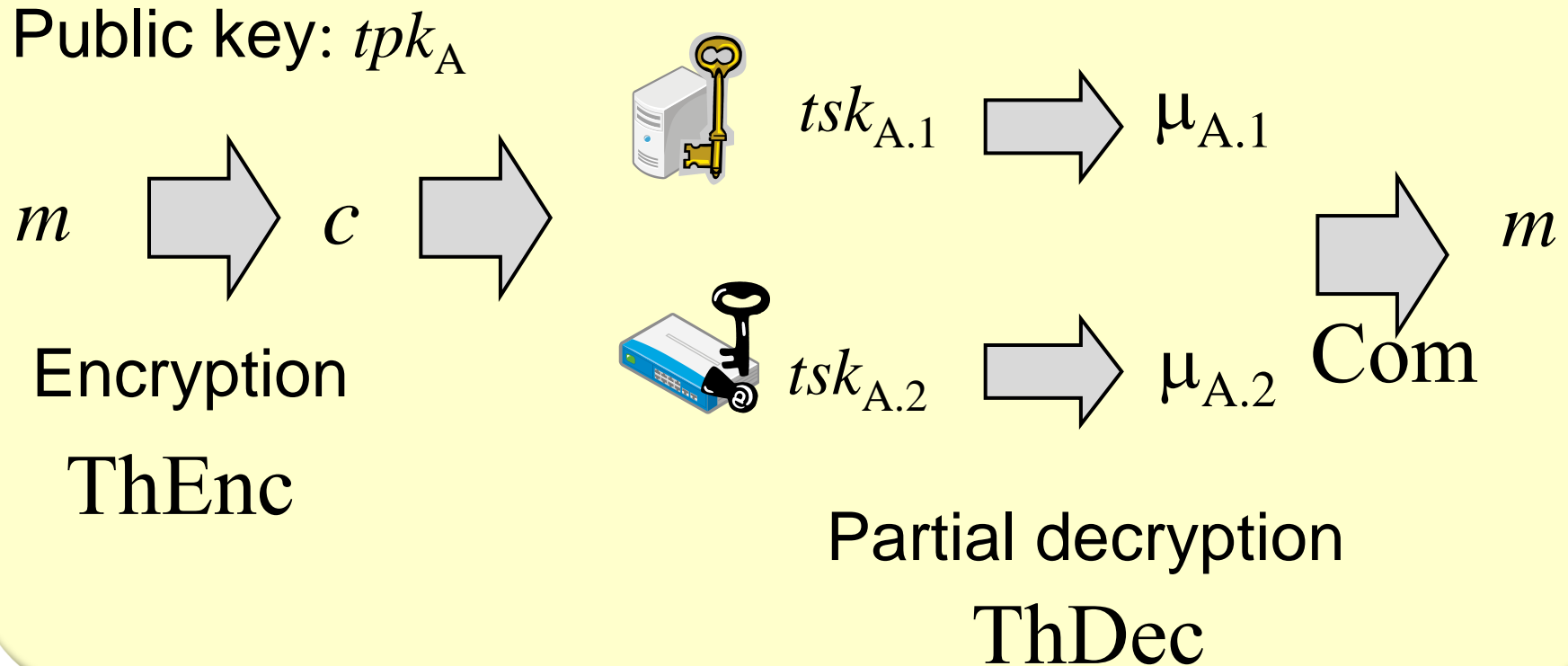
CCA secure
SUPRE



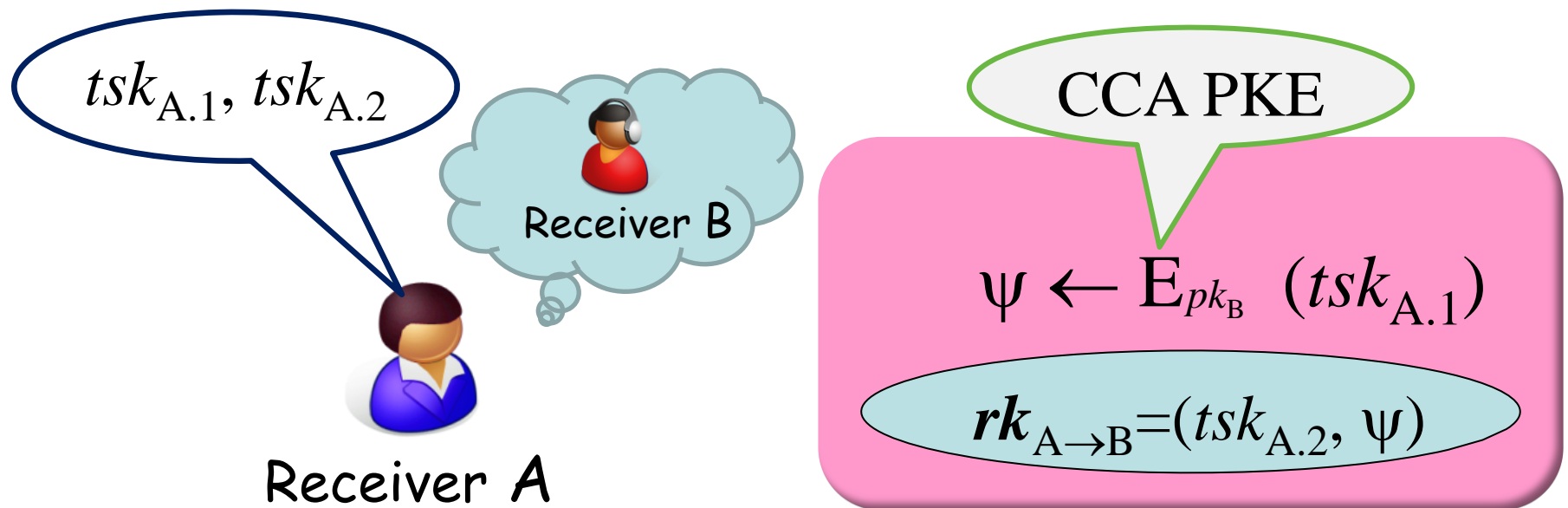


Generic Construction

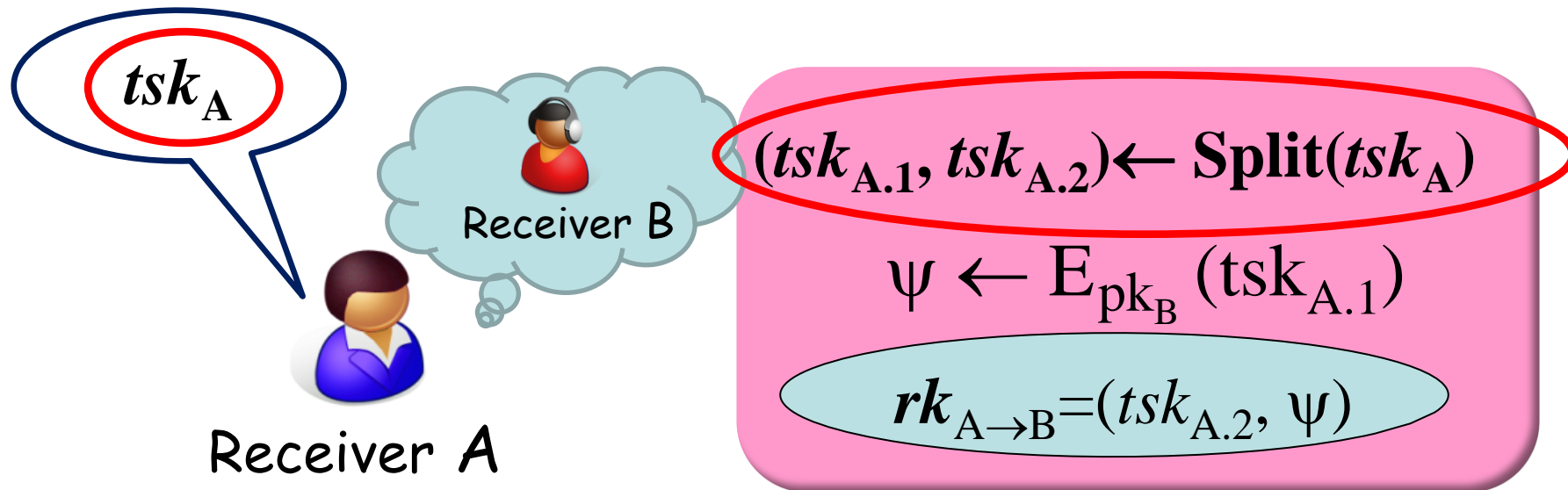
Main Building Block : 2-out-of-2 Threshold Public Key Encryption



Basic Idea: Rekey Generation Algorithm



Resplittability of TPKE



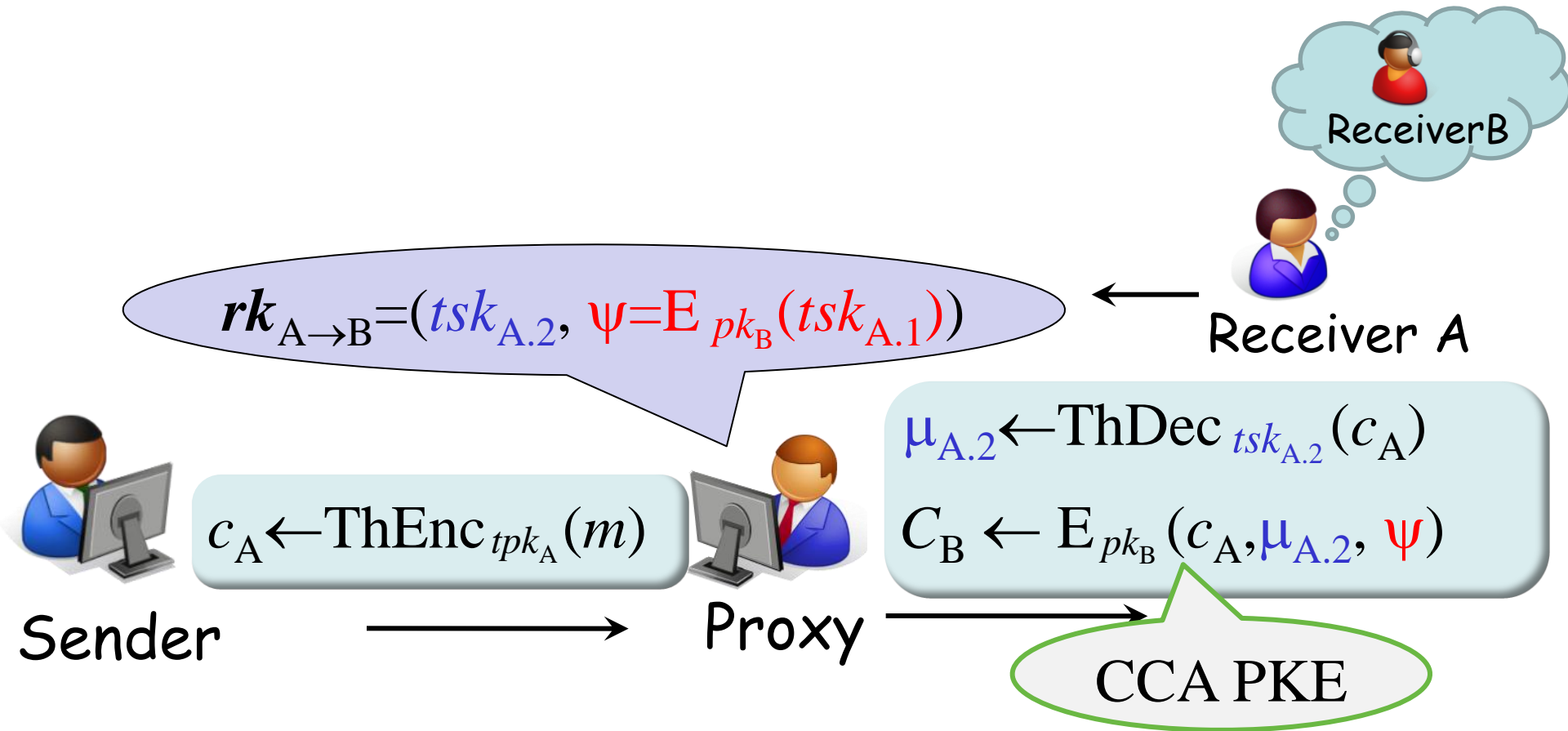
We define Resplittable TPKE and its security requirements.

Boneh, Boyen, and Halevi
(CT-RSA 2006)
Arita and Tsurudome
(ACNS 2009)

are examples of
Resplittable TPKE.



Basic Idea: Encryption and Re-Encryption



Basic Idea: First-Level Decryption



$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi = E_{pk_B}(tsk_{A.1}))$$

$$\mu_{A.2} \leftarrow \text{ThDec}_{tsk_{A.2}}(c_A)$$

$$C_B \leftarrow E_{pk_B}(c_A, \mu_{A.2}, \psi)$$

$$dk_B$$

$C_B \rightarrow$ Receiver B

$$\langle c_A, \mu_{A.2}, \psi \rangle \leftarrow D_{dk_B}(C_B)$$

$$tsk_{A.1} \leftarrow D_{dk_B}(\psi)$$

$$\mu_{A.1} \leftarrow \text{ThDec}_{tsk_{A.1}}(c_A)$$

$$m \leftarrow \text{Com}(\mu_{A.1}, \mu_{A.2})$$





Proxy

The malicious proxy might encrypt another (invalid) μ .

$$\mu_{A.2} \leftarrow \text{ThDec}_{tsk_{A.2}}(c_A)$$

$$C_B \leftarrow E_{pk_B}(c_A, \mu'_{A.2}, \psi)$$

B has to check the validity of μ .



C_B

Receiver B

$$\langle c_A, \mu'_{A.2}, \psi \rangle \leftarrow D_{dk_B}(C_B)$$

$$tsk_{A.1} \leftarrow D_{dk_B}(\psi)$$

$$\mu_{A.1} \leftarrow \text{ThDec}_{tsk_{A.1}}(c_A)$$

$$m' \leftarrow \text{Com}(\mu_{A.1}, \mu'_{A.2})$$



Robustness of TPKE

$\text{ThV}(c, \mu, tvk) \rightarrow (\text{in})\text{valid}$

Boneh, Boyen, and Halevi
(CT-RSA 2006)
Arita and Tsurudome
(ACNS 2009)

are examples of
Robustness TPKE.

using Paring Computation



Modified Scheme



Proxy

$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi = E_{pk_B}(tsk_{A.1}), tvk)$$

$$\mu_{A.2} \leftarrow \text{ThDec}_{tsk_{A.2}}(c_A)$$

$$C_B \leftarrow E_{pk_B}(c_A, \mu_{A.2}, \psi, tvk)$$



C_B → Receiver B →

$$\langle c_A, \mu_{A.2}, \psi, tvk \rangle \leftarrow D_{dk_B}(C_B)$$

$$\text{If valid} \leftarrow \text{ThV}(c_A, \mu_{A.2}, tvk)$$

$$tsk_{A.1} \leftarrow D_{dk_B}(\psi)$$

$$\mu_{A.1} \leftarrow \text{ThDec}_{tsk_{A.1}}(c_A)$$

$$m \leftarrow \text{Com}(\mu_{A.1}, \mu_{A.2})$$



Modified Scheme



Proxy

$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi = E_{pk_B}(tsk_{A.1}), tvk)$$

$$\mu_{A.2} \leftarrow \text{ThDec}_{tsk_{A.2}}(c_A)$$

$$C_B \leftarrow E_{pk_B}(c_A, \mu_{A.2}, \psi, tvk' \neq tvk)$$

B cannot check whether *tvk* is generated by the **original receiver**.



$C_B \rightarrow$ Receiver B

$$\langle c_A, \mu_{A.2}, \psi, tvk' \rangle \leftarrow D_{dk_B}(C_B)$$

If invalid $\leftarrow \text{ThV}(c_A, \mu_{A.2}, tvk')$

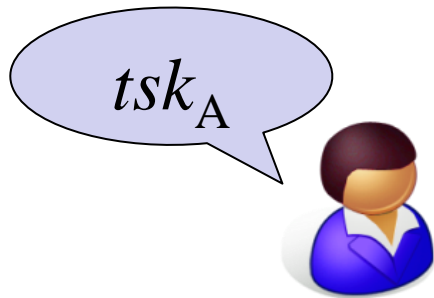
$$tsk_{A.1} \leftarrow D_{dk_B}(\psi)$$

$$\mu_{A.1} \leftarrow \text{ThDec}_{tsk_{A.1}}(c_A)$$

$$m \leftarrow \text{Comb}(\mu_{A.1}, \mu_{A.2})$$



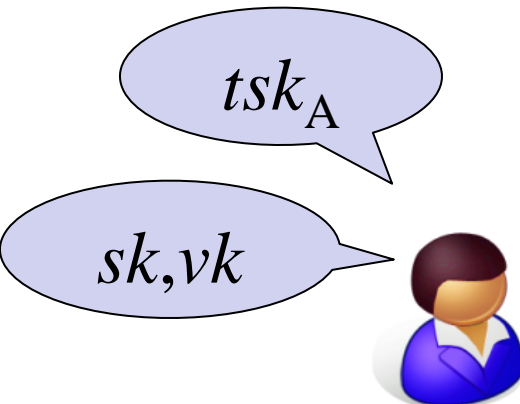
Rekey Generation Algorithm



Receiver A

$$(tvk, tsk_{A.1}, tsk_{A.2}) \leftarrow \text{Split}(tsk_A)$$
$$\psi \leftarrow E_{pk_B}(tsk_{A.1})$$

$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi, tvk)$$

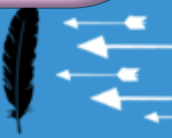


Receiver A

$$(tvk, tsk_{A.1}, tsk_{A.2}) \leftarrow \text{Split}(tsk_A)$$
$$\psi \leftarrow E_{pk_B}(tsk_{A.1})$$

$$\underline{\sigma \leftarrow \text{Sig}_{sk_A}(\psi, tvk)}$$

$$rk_{A \rightarrow B} = (tsk_{A.2}, \psi, tvk, \sigma)$$



$$(tsk_{A.2}, \psi = E_{pk_A}(tsk_{A.1}), tvk, \sigma)$$

$$c_A \leftarrow \text{ThEnc}_{tpk_A}(m)$$

$$\begin{aligned} \mu_{A.2} &\leftarrow \text{ThDec}_{tsk_{A.2}}(c_A) \\ C_B &\leftarrow E_{pk_B}(c_A, \mu_{A.2}, \psi, tvk, \sigma) \end{aligned}$$

$$\begin{aligned} \langle c_A, \mu_{A.2}, \psi, tvk, \sigma \rangle &\leftarrow D_{dk_B}(C_B) \\ \text{If } \text{valid} &\leftarrow \text{Ver}_{vk_A}(\langle \psi, tvk \rangle, \sigma) \\ \text{valid} &\leftarrow \text{ThV}(c_A, \mu_{A.2}, tvk) \\ tsk_{A.1} &\leftarrow D_{dk_B}(\psi) \\ \mu_{A.1} &\leftarrow \text{ThDec}_{tsk_{A.1}}(c_A) \\ m &\leftarrow \text{Com}(\mu_{A.1}, \mu_{A.2}) \end{aligned}$$

$$dk_B$$

$$C_B$$

Receiver B

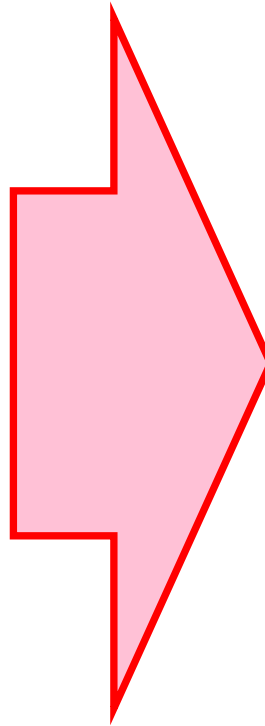


Our Contribution

CCA secure PKE

Strongly Unforgeable
Signature

Resplittable CCA
secure TPKE



CCA secure
SUPRE



Previous Works and Our work

Scheme	Uni/Bi	Security model	ROM/ STM	Pairing computation
[AFGH06]	Uni	CPA	ROM	✓
[HRSV07]	Uni	CPA	STM	✓
[CH07]	Bi	CCA	STM	✓
[LV08]	Uni	RCCA (weak CCA)	STM	✓
[AABH09]	Uni	CPA	ROM	✓
[CWYD10]	Uni	CCA + several restrictions	ROM	
Ours	Uni	CCA	STM	✓



Conclusion and Remark

- We define **CCA** security of PRE.
- We present the **first generic construction** of CCA secure Single Use PRE.
- Via our generic construction, we present first construction which is **CCA secure in the standard model.**
 - We should construct specific and efficient scheme by reference to our proposed generic construction.





Thanks for your
attention