Strongly Secure Authenticated Key Exchange from Factoring, Codes, and Lattices

Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, Kazuki Yoneyama

NTT Information Sharing Platform Laboratories
3-9-11 Midori-cho Musashino-shi Tokyo 180-8585, Japan
yoneyama.kazuki@lab.ntt.co.jp

Abstract. An unresolved problem in research on authenticated key exchange (AKE) is to construct a secure protocol against advanced attacks such as key compromise impersonation and maximal exposure attacks without relying on random oracles. HMQV, a state of the art AKE protocol, achieves both efficiency and the strong security model proposed by Krawczyk (we call it the CK^+ model), which includes resistance to advanced attacks. However, the security proof is given under the random oracle model. We propose a generic construction of AKE from a key encapsulation mechanism (KEM). The construction is based on a chosen-ciphertext secure KEM, and the resultant AKE protocol is CK^+ secure in the standard model. The protocol gives the first CK^+ secure AKE protocols based on the hardness of integer factorization problem, code-based problems, or learning problems with errors. In addition, instantiations under the Diffie-Hellman assumption or its variant can be proved to have strong security without non-standard assumptions such as πPRF and KEA1.
Keywords: authenticated key exchange, CK^+ model, key encapsulation mechanism

1 Introduction

1.1 Background

Establishing secure channels is one of the most important areas of cryptographic research. Secure channels provide secrecy and authenticity for both communication parties. When parties can share secret information via a public communication channel, secure channels would be constructed on (symmetric key) encryptions and message authentication codes with the shared secret information called session keys. Public-key cryptography can provide various solutions: one approach uses a *key encapsulation mechanism* (KEM) and another uses *authenticated key exchange* (AKE).

In KEM, a receiver has public information, called a *public key*, and the corresponding secret information, called a *secret key*. The public key is expected to be certified with the receiver's identity through an infrastructure such as a *public key infrastructure* (PKI). A sender who wants to share information, a

session key, with the receiver sends a *ciphertext* of the information and, the receiver decrypts the ciphertext to extract the information. KEM can be easily constructed from *public-key encryption* (PKE) under the reasonable condition that the plaintext space is sufficiently large. The desirable security notion of KEM is formulated as the *indistinguishability against chosen ciphertext attacks* (IND-CCA).

In AKE, each party has public information, called a *static public key*, and the corresponding secret information, called a *static secret key*. The static public key is also expected to be certified with a party's identity through an infrastructure such as PKI. A party who wants to share information with a party exchanges *ephemeral public keys*, generated from the corresponding *ephemeral secret keys*, and computes a *session state* from their static public keys, the corresponding static secret keys, the exchanged ephemeral public keys, and the corresponding ephemeral secret keys. Both parties then derive a *session key* from these values including the session state using a function called the *key derivation function*. Many studies have investigated the security notion of AKE [1–5]. The first security notion of AKE based on indistinguishability was provided by Bellare and Rogaway [1] (BR model). The BR model captures basic security requirements for AKE such as known key security and impersonation resilience. However, the BR model cannot grasp more complicated situations where a static secret key or session state of a party has been leaked. Accordingly, Canetti and Krawczyk [2] defined the first security notion of AKE capturing the leakage of static secret keys and session state and called it the *Canetti-Krawczyk (CK) model*. Though the CK model represents leakage of information other than the target session of the adversary, some advanced attacks such as key compromise impersonation (KCI), the breaking of weak perfect forward secrecy (wPFS) and maximal exposure attacks (MEX) use secret information of the target session; thus, the CK model is not resilient to such attacks. KCI means that when given a static secret key, an adversary will try to impersonate some honest party in order to fool the owner of the leaked secret key. wPFS implies that an adversary cannot recover a session key if the adversary does not modify messages of the target session and the session is executed before the static secret keys are compromised. In MEX, an adversary tries to distinguish the session key from a random value under the disclosure of any pair of secret static keys and ephemeral secret keys of the initiator and the responder in the session except for both the static and ephemeral secret keys of the initiator or the responder. Resistance to MEX requires security against any leakage situation that was not presumed. For example, an implementer of AKE may pretend to generate secret keys in an insecure host machine in order to prevent the randomness generation mechanisms in a tamper-proof module such as a smart card. Additionally, if a pseudo-random number generator implemented in a system is poor, secret keys will be known to the adversary even when the generation of ephemeral secret keys is operated in a tamper-proof module. Most AKE protocols are proved in the CK model; however, it is unclear whether such protocols satisfy resistance to advanced attacks due to the limitations of the CK model. A state of the art AKE protocol

HMQV [3] satisfies all known security requirements for AKE, including resistance to KCI, wPFS[1], and MEX, as well as provable security in the CK model. In this paper, we call this security model the CK^+ model; it is known to be one of the 'strongest' models for AKE. LaMacchia et al. [4] and Sarr et al. [5] also proposed very strong security models for AKE by re-formulating the concept of the CK^+ model; they called them the eCK model and the seCK model, respectively. These models allow an adversary to pose a query that directly reveals the ephemeral secret key of the target session. However, Cremers points out that the CK model and the eCK model are incomparable [9, 10]; thus, the eCK model is not stronger than the CK model while the CK^+ model is. We will briefly show the difference between the CK^+ model and these models. Since MEX includes any non-trivial leakage situation, HMQV (and CK^+ secure protocols) achieves surprisingly strong security.

1.2 Motivating Problem

HMQV is one of the most efficient protocols and satisfies one of the strongest security models (i.e., CK^+ security). However, the security proof is given in the random oracle model (ROM) under a specific number-theoretic assumption (Diffie-Hellman (DH) assumption). Moreover, to prove resistance to MEX, the knowledge-of-exponent assumption (KEA1) [11] (a widely criticized assumption such as [12]) is also necessary. Hence, one of the open problems in research on AKE is to construct a secure scheme in the CK^+ model without relying on random oracles under standard assumptions.

Boyd et al. [13–15] gave a partial solution to this problem by noting that KEM and AKE are closely related and that it might be natural to construct AKE from KEM. They proposed a generic construction of AKE from KEM (BCGNP construction), and its security is proved in the CK model in the standard model (StdM). Also, the BCGNP construction is shown to satisfy resistance to KCI. However, it is unclear whether the BCGNP construction is secure when leakage of secret information occurs (i.e., resistance to MEX). In fact, the BCGNP construction fails to satisfy CK^+ security when we consider the following attack scenario: Two parties exchange ciphertexts of an IND-CCA secure KEM scheme and generate a session key from these. An adversary who obtains the ephemeral secret keys (randomness used in generating ciphertexts) of the parties can compute the session key and win the game. Though the BCGNP construction can be extended to satisfy wPFS, it is guaranteed under the DH assumption, not a general assumption. It is quite restrictive because it cannot be instantiated from

[1] HMQV does not provide full perfect forward secrecy (fPFS), which is the same as wPFS except that the adversary can modify messages of the target session. Some schemes [6–8] have achieved fPFS. However, the schemes [6, 7] are clearly vulnerable to MEX; that is, the session key is computable if an adversary obtains an ephemeral secret key of parties in the target session. The other scheme [8] is resilient to MEX, but security is proved in the random oracle model. Upgrading wPFS to fPFS is not that difficult; it can be done by simply adding MAC or a signature of ephemeral public keys. Thus, we do not discuss fPFS in this paper.

the hardness of something other than the DH assumption such as an integer factoring problem, code-based problem, or lattice problem. Thus, we still have no AKE protocol that is secure in the 'strongest' model under just a general assumption without relying on random oracles (ROs).

1.3 Our Contribution

We fully solve the open problem by providing a generic construction of AKE from KEM. Our construction is a generalization of the BCGNP construction. The BCGNP construction uses IND-CCA KEM, a strong randomness extractor, and a pseudo-random function (PRF) as building blocks. Our construction effectively follows the design principle of the BCGNP construction. However, we first point out that the security proof of the BCGNP construction is not complete. Specifically, a requirement for KEM has not been formulated. KEM keys must have enough min-entropy in order to make outputs of the strong randomness extractor statistically indistinguishable from a uniformly random chosen element. Thus, the assumption that the KEM scheme satisfies such a property is additionally required. Fortunately, almost all IND-CCA KEM schemes satisfy that. Also, we need an IND-CPA secure KEM in addition to the BCGNP construction. Such an additional KEM can make our scheme wPFS and resilient to MEX. The resultant AKE protocol is CK^+ secure. Its security is proved under the existence of such KEMs, a strong randomness extractor, and a PRF in the StdM. The existence of an IND-CCA secure KEM has been shown from the hardness of integer factoring [16, 17], a code-based problem [18, 19], or a lattice problem [20–26]. To the best of our knowledge, our generic construction provides the first CK^+ secure AKE protocols based on the hardness of the above problems. Regarding the DH assumption or its variant, our generic construction is the first protocol that achieves CK^+ security in the StdM without non-standard assumptions (e.g., πPRF and KEA1).

We also rewrite the CK^+ model before proving the security of our generic construction in order to simplify the original model in [3]. Specifically, the original model is defined as a mix of four definitions (i.e., the CK model, wPFS, and resistance to KCI and MEX); thus, the security proof must also be separated into four theorems, which may reduce the readability. Therefore, we reformulate the CK^+ model as follows: wPFS, resistance to KCI, and resistance to MEX are integrated into the experiment of the extended model by exhaustively classifying leakage patterns. This definition is handy to prove security and rigourously captures all required properties.

We summarize our contributions as follows:

- We propose a two-pass generic CK^+ secure AKE construction from IND-CCA secure KEM and PRF in the StdM.
- We achieve the first CK^+ secure AKE protocols based on the hardness of integer factorization problem, code-based problem, and lattice-based problem in the StdM.
- We achieve the first CK^+ secure AKE protocol based on the DH assumption or its variant in the StdM without knowledge assumptions.

– We reformulate the CK$^+$ model to gain readability of the security proof.

The proposed generic construction can allow a hybrid instantiation; that is, the initiator and the responder can use different KEMs under different assumptions. For example, the initiator uses a factoring-based KEM while the responder uses a lattice-based KEM.

2 Security Model

In this section, we recall the CK$^+$ model that was introduced by [3]. We show a model specified to two pass protocols for simplicity. It can be trivially extended to any round protocol.

2.1 CK$^+$ vs. eCK

As indicated in Table 1, the CK$^+$ model captures all non-trivial patterns of leakage of static and ephemeral secret keys. The eCK model [4], which is a variant of the CK model [2], also captures all non-trivial patterns of leakage, as in Table 1. Since the CK$^+$ model captures all non-trivial patterns of leakage of static and ephemeral secret keys, the CK$^+$ model can theoretically be seen as a completion of the AKE security model.

In Table 1, the six cases in Definition 2 are listed, and these six cases cover wPFS, resistance to KCI, and MEX as follows: Cases 2-(a), 2-(c), and 2-(f) capture KCI, since the adversary obtains the static secret key of one party and the ephemeral secret key of the other party of the test session. Case 2-(e) captures wPFS, since the adversary obtains the static secret keys of both parties of the test session. Cases 2-(b) and 2-(d) capture MEX, since the adversary obtains the ephemeral secret keys of both parties of the test session.

The main difference between the CK$^+$ model and the eCK model is that the CK$^+$ model captures the session state reveal attack, but the eCK model does not. Thus, we adopt the CK$^+$ model, which is stronger than the eCK model from the viewpoint of the session state reveal attack, in this paper.

Notice that the timing of the static and ephemeral key reveal differs in the eCK and CK$^+$ models. In the eCK model, an adversary can issue the static and ephemeral key reveal query adaptively. In contrast, in the CK$^+$ model, an adversary can issue a corrupt query to obtain the static key, and the ephemeral key is given to the adversary when it is determined. We summarize this in Table 2.

2.2 CK$^+$ Security Model

We denote a party by U_i, and party U_i and other parties are modeled as probabilistic polynomial-time (PPT) Turing machines w.r.t. security parameter κ. For party U_i, we denote static secret (public) key by s_i (S_i) and ephemeral secret

Table 1. Classification of attacks and proposed CK^+ model [3] and eCK model [4].

Cases in Def.2	ssk_A	esk_A	ssk_B	esk_B	attack type	CK^+ model [3]	eCK model [4]
2-(a)	r	ok	ok	n	KCI	✓	✓
2-(b)	ok	r	ok	n	MEX	✓	✓
2-(c)	r	ok	ok	r	KCI	✓	✓
2-(d)	ok	r	ok	r	MEX	✓	✓
2-(e)	r	ok	r	ok	wPFS	✓	✓
2-(f)	ok	r	r	ok	KCI	✓	✓

"2-(*)" means the corresponding case in Definition 2. "ssk_A" means the static secret key of owner A of test session sid^*, and "ssk_B" means the static secret key of peer B of test session sid^*. "esk_A" means the ephemeral secret key of test session sid^*, and "esk_B" means the ephemeral secret key of the matching session $\overline{\mathsf{sid}^*}$. "ok" means the secret key is not revealed, "r" means the secret key may be revealed, and "n" means no matching session exists. A ✓ means that the model captures the attack.

Table 2. Comparison of CK^+ model [3] and eCK model [4].

	CK^+ model [3]	eCK model [4]
All non-trivial key leakage	✓	✓
Session state reveal	✓	χ
Adaptive key leakage	χ	✓

A ✓/χ means that the model does/does not capture the attack.

(public) key by x_i (X_i). Party U_i generates its own keys, s_i and S_i, and the static public key S_i is linked with U_i's identity in some systems like PKI.[2]

Session. An invocation of a protocol is called a *session*. Session activation is done by an incoming message of the forms $(\Pi, \mathcal{I}, U_A, U_B)$ or $(\Pi, \mathcal{R}, U_B, U_A, X_A)$, where we equate Π with a protocol identifier, $\mathcal{I}$ and $\mathcal{R}$ with role identifiers, and U_A and U_B with user identifiers. If U_A is activated with $(\Pi, \mathcal{I}, U_A, U_B)$, then U_A is called the session *initiator*. If U_B is activated with $(\Pi, \mathcal{R}, U_B, U_A, X_A)$, then U_B is called the session *responder*. The initiator U_A outputs X_A, then may receive an incoming message of the forms $(\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$ from the responder U_B, U_A then computes the session key SK if U_A received the message. On the contrary, the responder U_B outputs X_B, and computes the session key SK.

If U_A is the initiator of a session, the session is identified by $\mathsf{sid} = (\Pi, \mathcal{I}, U_A, U_B, X_A)$ or $\mathsf{sid} = (\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$. If U_B is the responder of a session, the ses-

[2] Static public keys must be known to both parties in advance. They can be obtained by exchanging them before starting the protocol or by receiving them from a certificate authority. This situation is common for all PKI-based AKE schemes.

sion is identified by $\mathsf{sid} = (\Pi, \mathcal{R}, U_B, U_A, X_A, X_B)$. We say that U_A is the *owner* of session sid, if the third coordinate of session sid is U_A. We say that U_A is the *peer* of session sid, if the fourth coordinate of session sid is U_A. We say that a session is *completed* if its owner computes the session key. The *matching session* of $(\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$ is session $(\Pi, \mathcal{R}, U_B, U_A, X_A, X_B)$ and vice versa.

Adversary. The adversary $\mathcal{A}$, which is modeled as a probabilistic polynomial-time Turing machine, controls all communications between parties including session activation by performing the following adversary query.

- Send(message): The message has one of the following forms: $(\Pi, \mathcal{I}, U_A, U_B)$, $(\Pi, \mathcal{R}, U_B, U_A, X_A)$, or $(\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$. The adversary $\mathcal{A}$ obtains the response from the party.

To capture leakage of secret information, the adversary $\mathcal{A}$ is allowed to issue the following queries.

- $\mathsf{SessionKeyReveal}(\mathsf{sid})$: The adversary $\mathcal{A}$ obtains the session key SK for the session sid if the session is completed.
- $\mathsf{SessionStateReveal}(\mathsf{sid})$: The adversary $\mathcal{A}$ obtains the session state of the owner of session sid if the session is not completed (the session key is not established yet). The session state includes all ephemeral secret keys and intermediate computation results except for immediately erased information but does not include the static secret key.
- $\mathsf{Corrupt}(U_i)$: This query allows the adversary $\mathcal{A}$ to obtain all information of the party U_i. If a party is corrupted by a $\mathsf{Corrupt}(U_i, S_i)$ query issued by the adversary $\mathcal{A}$, then we call the party U_i *dishonest*. If not, we call the party *honest*.

Freshness. For the security definition, we need the notion of freshness.

Definition 1 (Freshness). *Let* $\mathsf{sid}^* = (\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$ *or* $(\Pi, \mathcal{R}, U_A, U_B, X_B, X_A)$ *be a completed session between honest users* U_A *and* U_B*. If the matching session exists, then let* $\overline{\mathsf{sid}^*}$ *be the matching session of* sid^**. We say session* sid^* *is* fresh *if none of the following conditions hold:*

1. *The adversary* $\mathcal{A}$ *issues* $\mathsf{SessionKeyReveal}(\mathsf{sid}^*)$*, or* $\mathsf{SessionKeyReveal}(\overline{\mathsf{sid}^*})$ *if* $\overline{\mathsf{sid}^*}$ *exists,*
2. $\overline{\mathsf{sid}^*}$ *exists and the adversary* $\mathcal{A}$ *makes either of the following queries*
   - $\mathsf{SessionStateReveal}(\mathsf{sid}^*)$ *or* $\mathsf{SessionStateReveal}(\overline{\mathsf{sid}^*})$*,*
3. $\overline{\mathsf{sid}^*}$ *does not exist and the adversary* $\mathcal{A}$ *makes the following query*
   - $\mathsf{SessionStateReveal}(\mathsf{sid}^*)$*.*

Security Experiment. For the security definition, we consider the following security experiment. Initially, the adversary $\mathcal{A}$ is given a set of honest users and makes any sequence of the queries described above. During the experiment, the adversary $\mathcal{A}$ makes the following query.

- $\mathsf{Test}(\mathsf{sid}^*)$: Here, sid^* must be a fresh session. Select random bit $b \in_U \{0,1\}$, and return the session key held by sid^* if $b = 0$, and return a random key if $b = 1$.

The experiment continues until the adversary $\mathcal{A}$ makes a guess b'. The adversary $\mathcal{A}$ *wins* the game if the test session sid^* is still fresh and if the guess of the adversary $\mathcal{A}$ is correct, i.e., $b' = b$. The advantage of the adversary $\mathcal{A}$ in the AKE experiment with the PKI-based AKE protocol Π is defined as

$$\mathrm{Adv}_{\Pi}^{\mathrm{AKE}}(\mathcal{A}) = \Pr[\mathcal{A}\ wins] - \frac{1}{2}.$$

We define the security as follows.

Definition 2 (Security). *We say that a PKI-based AKE protocol Π is* secure in the CK^+ model *if the following conditions hold:*

1. *If two honest parties complete matching sessions, then, except with negligible probability, they both compute the same session key.*
2. *For any PPT bounded adversary $\mathcal{A}$, $\mathrm{Adv}_{\Pi}^{\mathrm{AKE}}(\mathcal{A})$ is negligible in security parameter κ for the test session sid^*,*
   (a) *if $\overline{\mathsf{sid}^*}$ does not exist, and the static secret key of the owner of sid^* is given to $\mathcal{A}$.*
   (b) *if $\overline{\mathsf{sid}^*}$ does not exist, and the ephemeral secret key of sid^* is given to $\mathcal{A}$.*
   (c) *if $\overline{\mathsf{sid}^*}$ exists, and the static secret key of the owner of sid^* and the ephemeral secret key of $\overline{\mathsf{sid}^*}$ are given to $\mathcal{A}$.*
   (d) *if $\overline{\mathsf{sid}^*}$ exists, and the ephemeral secret key of sid^* and the ephemeral secret key of $\overline{\mathsf{sid}^*}$ are given to $\mathcal{A}$.*
   (e) *if $\overline{\mathsf{sid}^*}$ exists, and the static secret key of the owner of sid^* and the static secret key of the peer of sid^* are given to $\mathcal{A}$.*
   (f) *if $\overline{\mathsf{sid}^*}$ exists, and the ephemeral secret key of sid^* and the static secret key of the peer of sid^* are given to $\mathcal{A}$.*

Note that the items 2.a, 2.c, and 2.f correspond to resistance to KCI, item 2.e corresponds to wPFS, and items 2.b and 2.d correspond to resistance to MEX.

3 Generic AKE Construction from KEM without Random Oracles

In this section, we propose a generic construction of CK^+-secure AKE from KEM.

3.1 Preliminaries

Security Notions of KEM Schemes. Here, we recall the definition of IND-CCA and IND-CPA security for KEM, and min-entropy of KEM keys as follows.

Definition 3 (Model for KEM Schemes). *A KEM scheme consists of the following 3-tuple* (KeyGen, EnCap, DeCap)*:*

- $(ek, dk) \leftarrow \mathsf{KeyGen}(1^\kappa, r_g)$ *: a key generation algorithm which on inputs* 1^κ *and* $r_g \in \mathcal{RS}_G$*, where* κ *is the security parameter and* $\mathcal{RS}_G$ *is a randomness space, outputs a pair of keys* (ek, dk)*.*
- $(K, CT) \leftarrow \mathsf{EnCap}_{ek}(r_e)$ *: an encryption algorithm which takes as inputs encapsulation key* ek *and* $r_e \in \mathcal{RS}_E$*, outputs session key* $K \in \mathcal{KS}$ *and ciphertext* $CT \in \mathcal{CS}$*, where* $\mathcal{RS}_E$ *is a randomness space,* $\mathcal{KS}$ *is a session key space, and* $\mathcal{CS}$ *is a ciphertext space.*
- $K \leftarrow \mathsf{DeCap}_{dk}(CT)$ *: a decryption algorithm which takes as inputs decapsulation key* dk *and ciphertext* $CT \in \mathcal{CS}$*, and outputs session key* $K \in \mathcal{KS}$*.*

Definition 4 (IND-CCA and IND-CPA Security for KEM). *A KEM scheme is IND-CCA-secure for KEM if the following property holds for security parameter* κ*; For any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*,* $\mathbf{Adv}^{\mathrm{ind-cca}} = |\Pr[r_g \leftarrow \mathcal{RS}_G; (ek, dk) \leftarrow \mathsf{KeyGen}(1^\kappa, r_g); (state) \leftarrow \mathcal{A}_1^{\mathcal{DO}(dk,\cdot)}(ek); b \leftarrow \{0,1\}; r_e \leftarrow \mathcal{RS}_E; (K_0^*, CT_0^*) \leftarrow \mathsf{EnCap}_{ek}(r_e); K_1^* \leftarrow \mathcal{K}; b' \leftarrow \mathcal{A}_2^{\mathcal{DO}(dk,\cdot)}(ek, (K_b^*, CT_0^*), state); b' = b] - 1/2| \leq negl$*, where* $\mathcal{DO}$ *is the decryption oracle,* $\mathcal{K}$ *is the space of session key and state is state information that* $\mathcal{A}$ *wants to preserve from* $\mathcal{A}_1$ *to* $\mathcal{A}_2$*.* $\mathcal{A}$ *cannot submit the ciphertext* $CT = CT_0^*$ *to* $\mathcal{DO}$*.*

We say a KEM scheme is IND-CPA-secure for KEM if $\mathcal{A}$ *does not access* $\mathcal{DO}$*.*

Definition 5 (Min-Entropy of KEM Key). *A KEM scheme is* k*-min-entropy KEM if for any* ek*, for distribution* $D_{\mathcal{KS}}$ *of variable* K *defined by* $(K, CT) \leftarrow \mathsf{EnCap}_{ek}(r_e)$ *and random* $r_e \in \mathcal{RS}_E$*,* $H_\infty(D_{\mathcal{KS}}) \geq k$ *holds, where* H_∞ *denotes min-entropy.*

Security Notions of Randomness Extractor and Pseudo-Random Function. Let $Ext : S \times X \rightarrow Y$ be a function with finite seed space S, finite domain X, and finite range Y.

Definition 6 (Strong Randomness Extractor). *We say that function Ext is a strong randomness extractor, if for any distribution* D_X *over* X *with* $H_\infty(D_X) \geq k$*,* $\Delta((U_S, Ext(U_S, D_X)), (U_S, U_Y)) \leq negl$ *holds, where both* U_S *in* $(U_S, Ext(U_S, D_X))$ *denotes the same random variable,* Δ *denotes statistical distance,* U_S, U_X, U_Y *denotes uniform distribution over* S, X, Y *respectively,* $|X| = n \geq k$*,* $|Y| = k$*, and* $|S| = d$*.*

Let κ be a security parameter and $\mathsf{F} = \{F_\kappa : Dom_\kappa \times \mathcal{FS}_\kappa \rightarrow Rng_\kappa\}_\kappa$ be a function family with a family of domains $\{\mathsf{Dom}_\kappa\}_\kappa$, a family of key spaces $\{\mathsf{FS}_\kappa\}_\kappa$ and a family of ranges $\{\mathsf{Rng}_\kappa\}_\kappa$.

Definition 7 (Pseudo-Random Function). *We say that function family* $\mathsf{F} = \{F_\kappa\}_\kappa$ *is the PRF family, if for any PPT distinguisher* $\mathcal{D}$, $\mathbf{Adv}^{\mathrm{prf}} = |\Pr[\mathcal{D}^{F_\kappa(\cdot)} \to 1] - \Pr[\mathcal{D}^{RF_\kappa(\cdot)} \to 1]| \leq negl$, *where* $RF_\kappa : Dom_\kappa \to Rng_\kappa$ *is a truly random function.*

3.2 Construction

Our construction (GC) is based on an IND-CCA secure KEM, an IND-CPA secure KEM, PRFs, and strong randomness extractors. While the requirements for the underlying building blocks are not stronger than those for the previous generic construction [13, 14], GC achieves stronger security (i.e., CK^+ security) without random oracles.

Necessity of Min-Entropy of KEM Key. In the BCGNP construction, a KEM scheme is only assumed to be IND-CCA. However, it is not enough to prove the security. Both parties derive the session key by applying decapsulated KEM keys to a strong randomness extractor before applying them to PRFs. This extractor guarantees to output a statistically indistinguishable value from a uniform randomly chosen element from the same space. It requires as input a seed and a KEM key with min-entropy κ, where κ is a security parameter. IND-CCA states that no PPT adversary can distinguish the KEM key from a random element, but this is "only" computational indistinguishability. What we need is statistical indistinguishability. Thus, we must also assume that min-entropy of the KEM key is equal or larger than κ. This property is not very strong; almost all IND-CCA secure schemes satisfy it. We will discuss later about this property of concrete KEM schemes.

Design Principle. The main ideas to achieve CK^+ security are to use the *twisted PRF* trick and *session-specific* key generation.

First, we have to consider resistance to MEX. The most awkward pattern of MEX is the disclosure of ephemeral secret keys of the initiator and the responder. If we use KEM naturally, all randomness used to generate ciphertexts is leaked as ephemeral secret keys; thus, the adversary can obtain encrypted messages without knowing secret keys. Hence, we have to avoid using ephemeral secret keys as randomness of KEM directly. A possible solution is to generate randomness from the static secret key as well as the ephemeral secret key by using a technique such as the ordinary NAXOS trick [4]. Though this trick leads to security against leakage of ephemeral secret keys, the trick must apply an RO to the concatenation of the static and ephemeral secret keys, and it uses the output as a quasi-ephemeral secret key. It is unsuitable for our purpose to construct secure protocols in the StdM. Thus, we use a trick to achieve the same properties as the NAXOS trick but without ROs. We call it the twisted PRF trick.[3] This trick uses two PRFs (F, F') with reversing keys; we choose two ephemeral

[3] A similar trick is used in the Okamoto AKE scheme [27].

keys (r, r') and compute $F_\sigma(r) \oplus F'_{r'}(\sigma)$, where σ is the static secret key. The twisted PRF trick is especially effective in the following two scenarios: leakage of both ephemeral secret keys of the initiator and the responder, and leakage of the static secret key of the initiator and the ephemeral secret key of the responder (i.e., corresponding to KCI). If (r, r') is leaked, $F_\sigma(r)$ cannot be computed without knowing σ. Similarly, if σ is leaked, $F'_{r'}(\sigma)$ cannot be computed without knowing r'. In our KEM-based generic construction, the output of the twisted PRF is used as randomness for the encapsulation algorithm.

Next, we have to consider the scenario in which static secret keys are leaked as the attack scenario in wPFS. We cannot achieve a CK^+ secure scheme by any combination of KEMs using static secret keys as decapsulation keys against leakage of both static secret keys of the initiator and the responder because an adversary can obtain all information the parties can obtain by using static secret keys. Our solution is to generate session-specific decapsulation and encapsulation keys. The initiator sends the temporary encapsulation key to the responder, the responder encapsulates a KEM key with the temporary encapsulation key, and the initiator decapsulates the ciphertext. Since this procedure does not depend on the static secret keys, the KEM key is hidden even if both static secret keys of the initiator and the responder are leaked. Note that security of KEM for temporary use only requires IND-CPA. The session-specific key generation is effective for achieving wPFS.

As the BCGNP construction [13, 14], we use IND-CCA secure KEM schemes to exchange ciphertexts. CCA security is necessary to simulate SessionStateReveal queries in the security proof. When we prove security in the case where ephemeral secret keys are leaked, the simulator needs to embed the challenge ciphertext in the ephemeral public key in the test session. Then, the static secret key to decrypt the challenge ciphertext is not known; that is, the simulator must respond to the SessionStateReveal query for a session owned by the same parties as the test session without knowing the static secret key. Hence, the simulator needs the power of the decryption oracle to obtain intermediate computation results corresponding to the SessionStateReveal query.

Generic Construction GC. The protocol of GC from KEMs (KeyGen, EnCap, DeCap) and (wKeyGen, wEnCap, wDeCap) is as follows.

Public Parameters. Let κ be the security parameter, $F : \{0,1\}^* \times \mathcal{FS} \to \mathcal{RS}_E$, $F' : \{0,1\}^* \times \mathcal{FS} \to \mathcal{RS}_E$, and $G : \{0,1\}^* \times \mathcal{FS} \to \{0,1\}^\kappa$ be pseudo-random functions, where $\mathcal{FS}$ is the key space of PRFs ($|\mathcal{FS}| = \kappa$), $\mathcal{RS}_E$ is the randomness space of encapsulation algorithms, and $\mathcal{RS}_G$ is the randomness space of key generation algorithms, and let $Ext : \mathcal{SS} \times \mathcal{KS} \to \mathcal{FS}$ be a strong randomness extractor with randomly chosen seed $s \in \mathcal{SS}$, where SS is the seed space and $\mathcal{KS}$ is the KEM key space. These are provided as some of the public parameters.

Secret and Public Keys. Party U_I randomly selects $\sigma_I \in \mathcal{FS}$ and $r_I \in \mathcal{RS}_G$, and runs the key generation algorithm $(ek_{I,1}, dk_{I,1}) \leftarrow \mathsf{KeyGen}(1^\kappa, r_I)$, where $\mathcal{RS}_G$

is the randomness space of KeyGen. Party U_I's static secret and public keys are $((dk_{I,1}, \sigma_I), ek_{I,1})$.

Key Exchange. Party U_A with secret and public keys $((dk_{A,1}, \sigma_A), ek_{A,1})$, and who is the initiator, and party U_B with secret and public keys $((dk_{B,1}, \sigma_B), ek_{B,1})$, and who is the responder, perform the following two-pass key exchange protocol.

1. Party U_A randomly chooses ephemeral secret keys $r_{A,1}, r'_{A,1} \in \mathcal{FS}$ and $r_{A,2} \in \mathcal{RS}_G$. Party U_A computes $(CT_{A,1}, K_{A,1}) \leftarrow \mathsf{EnCap}_{ek_{B,1}}(F_{\sigma_A}(r_{A,1}) \oplus F'_{r'_{A,1}}(\sigma_A))$ and $(ek_{A,2}, dk_{A,2}) \leftarrow \mathsf{wKeyGen}(1^\kappa, r_{A,2})$ and sends $(U_A, U_B, CT_{A,1}, ek_{A,2})$ to party U_B.
2. Upon receiving $(U_A, U_B, CT_{A,1}, ek_{A,2})$, party U_B chooses the ephemeral secret keys $r_{B,1}, r'_{B,1} \in \mathcal{FS}$ and $r_{B,2} \in \mathcal{RS}_E$, computes $(CT_{B,1}, K_{B,1}) \leftarrow \mathsf{EnCap}_{ek_{A,1}}(F_{\sigma_B}(r_{B,1}) \oplus F'_{r'_{B,1}}(\sigma_B))$ and $(CT_{B,2}, K_{B,2}) \leftarrow \mathsf{wEnCap}_{ek_{A,2}}(r_{B,2})$, and sends $(U_A, U_B, CT_{B,1}, CT_{B,2})$ to party U_A.
   Party U_B computes $K_{A,1} \leftarrow \mathsf{DeCap}_{dk_{B,1}}(CT_{A,1})$, $K'_1 \leftarrow Ext(s, K_{A,1})$, $K'_2 \leftarrow Ext(s, K_{B,1})$ and $K'_3 \leftarrow Ext(s, K_{B,2})$, sets the session transcript $\mathsf{ST} = (U_A, U_B, ek_{A,1}, ek_{B,1}, CT_{A,1}, ek_{A,2}, CT_{B,1}, CT_{B,2})$ and the session key $SK = G_{K'_1}(\mathsf{ST}) \oplus G_{K'_2}(\mathsf{ST}) \oplus G_{K'_3}(\mathsf{ST})$, completes the session, and erases all session states.
3. Upon receiving $(U_A, U_B, CT_{B,1}, CT_{B,2})$, party U_A computes $K_{B,1} \leftarrow \mathsf{DeCap}_{dk_{A,1}}(CT_{B,1})$, $K_{B,2} \leftarrow \mathsf{wDeCap}_{dk_{A,2}}(CT_{B,2})$, $K'_1 \leftarrow Ext(s, K_{A,1})$, $K'_2 \leftarrow Ext(s, K_{B,1})$ and $K'_3 \leftarrow Ext(s, K_{B,2})$, sets the session transcript $\mathsf{ST} = (U_A, U_B, ek_{A,1}, ek_{B,1}, CT_{A,1}, ek_{A,2}, CT_{B,1}, CT_{B,2})$ and the session key $SK = G_{K'_1}(\mathsf{ST}) \oplus G_{K'_2}(\mathsf{ST}) \oplus G_{K'_3}(\mathsf{ST})$, completes the session, and erases all session states.

The session state of a session owned by U_A contains ephemeral secret keys $(r_{A,1}, r'_{A,1}, r_{A,2})$, KEM keys $(K_{A,1}, K_{B,1}, K_{B,2})$, outputs of the extractor (K'_1, K'_2, K'_3) and outputs of PRFs $(F_{\sigma_A}(r_{A,1}), F'_{r'_{A,1}}(\sigma_A), G_{K'_1}(\mathsf{ST}), G_{K'_2}(\mathsf{ST}), G_{K'_3}(\mathsf{ST}))$. Similarly, the session state of a session owned by U_B contains ephemeral secret keys $(r_{B,1}, r'_{B,1}, r_{B,2})$, decapsulated KEM keys $(K_{A,1}, K_{B,1}, K_{B,2})$, outputs of the extractor (K'_1, K'_2, K'_3) and outputs of PRFs $(F_{\sigma_B}(r_{B,1}), F'_{r'_{B,1}}(\sigma_B), G_{K'_1}(\mathsf{ST}), G_{K'_2}(\mathsf{ST}), G_{K'_3}(\mathsf{ST}))$.

Remark 1. Obviously, we can use arbitrary combinations of KEM schemes in the generic construction. This means that each party can rely on a different assumption from the peer. Since our construction does not contain any direct operation between derivatives of KEM schemes, it is no problem that randomness spaces, public keys, or ciphertext are distinct from each other.

Security. We show the following theorem.

Theorem 1. *If KEM* $(\mathsf{KeyGen}, \mathsf{EnCap}, \mathsf{DeCap})$ *is IND-CCA secure and is* κ*-min-entropy KEM, KEM* $(\mathsf{wKeyGen}, \mathsf{wEnCap}, \mathsf{wDeCap})$ *is IND-CPA secure and is* κ*-min-entropy KEM,* F, F', G *are PRFs, and Ext is a strong randomness extractor, then AKE scheme* GC *is* CK^+*-secure.*

Due to space limitations we defer the proof of Theorem 1 to the full version. Here, we give an overview of the security proof.

We have to consider the following four leakage patterns in the CK$^+$ security model (matching cases):

2-(c) the static secret key of the initiator and the ephemeral secret key of the responder
2-(d) both ephemeral secret keys
2-(e) both static secret keys
2-(f) the ephemeral secret key of the initiator and the static secret key of the responder

In case 2-(c), $K_{A,1}$ is protected by the security of $CT_{A,1}$ because $r'_{A,1}$ is not leaked; therefore, $F'_{r'_{A,1}}(\sigma_A)$ is hidden and $dk_{B,1}$ is not leaked. In case 2-(d), $K_{A,1}$ and $K_{B,1}$ are protected by the security of $CT_{A,1}$ and $CT_{B,1}$ because σ_A and σ_B are not leaked; therefore, $F_{\sigma_A}(r_{A,1})$ and $F_{\sigma_B}(r_{B,1})$ are hidden and $dk_{A,1}$ and $dk_{B,1}$ are not leaked. In case 2-(e), $K_{B,2}$ is protected by the security of $CT_{B,2}$ because $dk_{A,2}$ and $r_{B,2}$ are not leaked. In case 2-(f), $K_{B,1}$ is protected by the security of $CT_{B,1}$ because $r'_{B,1}$ is not leaked; therefore, $F'_{r'_{B,1}}(\sigma_B)$ is hidden and $dk_{A,1}$ is not leaked. Then, we transform the CK$^+$ security game since the session key in the test session is randomly distributed. First, we change part of the doubled PRF in the test session into a random function because the key of part of the doubled PRF is hidden from the adversary; therefore, the randomness of the protected KEM can be randomly distributed. Second, we change the protected KEM key into a random key for each pattern; therefore, the input of Ext is randomly distributed and has sufficient min-entropy. Third, we change the output of Ext into randomly chosen values. Finally, we change one of the PRFs (corresponding to the protected KEM) into a random function. Therefore, the session key in the test session is randomly distributed; thus, there is no advantage to the adversary. We can show a similar proof in non-matching cases.

4 Instantiations

4.1 Diffie-Hellman-based

We can achieve various AKE schemes as concrete instantiations based on the hardness of the DH problem and its variants. These are derived from the generic construction GC in Section 3. For example, we can apply efficient IND-CCA KEM schemes to GC from the decisional DH [28, 29] (DDH), computational DH [30, 31], hashed DH [32] and bilinear DH assumptions [33].

We can easily show that these schemes have κ-min-entropy KEM keys. The KEM part of the Cramer-Shoup PKE consists of $g_1^{zr} \in G$, where G is a finite cyclic group of order prime p, g_1^z is part of ek, and r is uniformly chosen randomness, and $|r|$ is 2κ. Thus, g_1^{zr} has min-entropy larger than κ. Similarly, other schemes also have κ-min-entropy KEM keys.

The significant advantage of our instantiations in the StdM is reasonable assumption. First, HMQV satisfies the same security model as our construction. However, it requires the KEA1 assumption and relies on ROs. Since it has been criticised, in particular because the KEA1 assumption does not appear to be "efficiently falsifiable" as Naor put it [12], this assumption is quite undesirable. Also, it was shown that there exist some protocols that are secure in the ROM but are insecure if ROs are replaced by any specific function [34]. A disadvantage of our construction to HMQV is that HMQV is a one-round protocol but our scheme is not. One-round protocols mean that the initiator and the responder can send their messages independently and simultaneously. Conversely, in our scheme, the responder must wait to receive the message from the initiator. Next, the AKE scheme by Okamoto [27] is secure in the StdM. However, it is not proved in the CK^+ model and needs to assume existence of πPRF. πPRF is a stronger primitive than ordinary PRF, and it is not known how to construct πPRF concretely. On the contrary, our instantiations only require the standard notions of KEM and pseudo-random function security. Moreover, the BCGNP construction [13, 14] is secure in the StdM with standard assumption. However, the security is not proved in the CK^+ model.[4] Thus, DH-based AKE schemes from GC are first CK^+ secure schemes in the StdM with standard assumptions.

For example, our scheme can be instantiated with the Cramer-Shoup KEM [35] as an IND-CCA KEM, and with the ElGamal KEM as an IND-CPA KEM under the DDH assumption. Communication complexity (for two parties) of this instantiation is $8|p|$, where $|p|$ is the length of a group element. Computational complexity (for two parties) of this instantiation is 4 multi-exponentiations and 12 regular exponentiations (all symmetric operations such as hash function/KDF/PRF and multiplications are ignored). We show a comparison between this instantiation and previous schemes in Table 3.

4.2 Factoring-based

We can achieve several new AKE protocols as concrete instantiations based on the hardness of integer factorization and its variants such as the RSA problem.

Some instantiations in the StdM are based on the hardness of the integer factorization problem. By applying the Hofheinz-Kiltz PKE [16] and the Mei-Li-Lu-Jia PKE [17], which are IND-CCA secure in the StdM under the factoring assumption to GC, we can obtain first CK^+ secure AKE protocols in the StdM under the integer factorization assumption. Also, we have other instantiations based on the hardness of RSA inversion. By applying the Chevallier-Mames-Joye PKE [36] and the Kiltz-Mohassel-O'Neill PKE [37], which are IND-CCA

[4] The BCGNP construction with an additional exchange of a DH value (called Protocol 2 in [13, 14]) can be proved in the CK model, and it satisfies wPFS and resistance to KCI. We can extend the security of Protocol 2 to the CK^+ security with the twisted PRF trick. If IND-CPA KEM in GC is instantiated with the ElGamal KEM, our scheme is the same as Protocol 2 with the twisted PRF trick. Thus, our scheme can also be seen as a generalization of the BCGNP construction.

Table 3. Comparison of previous DH-based schemes and an instantiation of our scheme

	Model	Resource	Assumption	Computation (#[multi,regular]-exp)	Communication complexity	
[3]	CK^+	ROM	gap DH & KEA1	$[2, 2]$	$2\|p\|$	512
[27]	eCK	StdM	DDH & πPRF	$[6, 6]$	$9\|p\|$	2304
[14]	CK & KCI	StdM	DDH	$[4, 8]$	$6\|p\|$	1536
Ours	CK^+	StdM	DDH	$[4, 12]$	$8\|p\|$	2048

For concreteness the expected ciphertext overhead for a 128-bit implementation is also given. Note that computational costs are estimated without any pre-computation technique.

secure in the StdM under the instance-independent RSA assumption to GC, we can obtain first CK^+ secure AKE protocols in the StdM under the RSA-type assumption.

We can regard a message in PKE as a KEM key when the message space is larger than κ and messages are uniformly chosen randomness. In this case, it is obvious that such a KEM scheme has κ-min-entropy KEM keys.

4.3 Code-based

We can achieve new AKE protocols as concrete instantiations based on code-based problems.

For the AKE protocol in the StdM, we can apply Dowsley et al.'s PKE [19] that is IND-CCA secure in the StdM under the McEliece and LPN assumptions to GC. (See Ref. [19] for definitions of these assumptions.) This is the first CK^+ secure AKE protocol without ROs based on a code-based problem.

As for factoring-based PKE, code-based PKE schemes also have κ-min-entropy KEM keys when the message space is larger than κ and messages are uniformly chosen randomness.

Remark 2. Bernstein et al. [38] estimated the size of a public key of the original McEliece at about 2 Mbits for 128-bit security. If we employ "wild" McEliece by Bernstein et al. [39] rather than the original McEliece PKE, the size of the public key is reduced to 750K bits. Our generic construction contains the public key of the KEM from the temporary key generation in the first round message. If the randomized McEliece PKE by Nojima et al. [40] is employed as the IND-CPA secure KEM, which is IND-CPA secure and requires the same size for the public key as the original, the communication complexity of the resultant AKE scheme is high. However, the way to construct an efficient and CK^+ secure AKE scheme from codes is an open problem.

4.4 Lattice-based

We also achieve new concrete AKE protocols based on the worst-case hardness of the (ring-)LWE problems derived from our generic constructions.

PKE schemes [20–26, 41] which are IND-CCA secure in the StdM are easily converted into IND-CCA secure KEM schemes. Also, PRFs are obtained from one-way functions [42–45] and directly constructed from the (ring-)LWE assumptions with sub-exponential parameters [46]. Thus, by applying these building blocks to GC, we can obtain first CK^+ secure AKE protocols in the StdM under the (ring-)LWE assumption. Unfortunately, the obtained AKE protocols are still theoretical since these PKE schemes require huge keys, say, of the quadratic or cubic order of the security parameter, and thus, an efficient and direct construction of PRFs from the (ring-)LWE assumption with polynomial parameters has not yet been achieved.

As for factoring-based PKE, lattice-based PKE schemes also have κ-min-entropy KEM keys when the message space is larger than κ and messages are uniformly chosen randomness.

References

1. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: CRYPTO 1993. (1993) 232–249
2. Canetti, R., Krawczyk, H.: Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In: EUROCRYPT 2001. (2001) 453–474
3. Krawczyk, H.: HMQV: A High-Performance Secure Diffie-Hellman Protocol. In: CRYPTO 2005. (2005) 546–566
4. LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger Security of Authenticated Key Exchange. In: ProvSec 2007. (2007) 1–16
5. Sarr, A.P., Elbaz-Vincent, P., Bajard, J.C.: A New Security Model for Authenticated Key Agreement. In: SCN 2010. (2010) 219–234
6. Jeong, I.R., Katz, J., Lee, D.H.: One-Round Protocols for Two-Party Authenticated Key Exchange. In: ACNS 2004. (2004) 220–232
7. Gennaro, R., Krawczyk, H., Rabin, T.: Okamoto-Tanaka Revisited: Fully Authenticated Diffie-Hellman with Minimal Overhead. In: ACNS 2010. (2010) 309–328
8. Cremers, C.J.F., Feltz, M.: One-round Strongly Secure Key Exchange with Perfect Forward Secrecy and Deniability. In: Cryptology ePrint Archive: 2011/300. (2011)
9. Cremers, C.J.F.: Session-state Reveal Is Stronger Than Ephemeral Key Reveal: Attacking the NAXOS Authenticated Key Exchange Protocol. In: ACNS 2009. (2009) 20–33
10. Cremers, C.J.F.: Examining Indistinguishability-Based Security Models for Key Exchange Protocols: The case of CK, CK-HMQV, and eCK. In: ASIACCS 2011. (2011) 80–91
11. Damgård, I.: Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks. In: CRYPTO 1991. (1991) 445–456
12. Naor, M.: On Cryptographic Assumptions and Challenges. In: CRYPTO 2003. (2003) 96–109
13. Boyd, C., Cliff, Y., González Nieto, J.M., Paterson, K.G.: Efficient One-Round Key Exchange in the Standard Model. In: ACISP 2008. (2008) 69–83
14. Boyd, C., Cliff, Y., González Nieto, J.M., Paterson, K.G.: One-round key exchange in the standard model. In: IJACT 1(3). (2009) 181–199
15. Gorantla, M.C., Boyd, C., González Nieto, J.M., Manulis, M.: Generic One Round Group Key Exchange in the Standard Model. In: ICISC 2009. (2009) 1–15

16. Hofheinz, D., Kiltz, E.: Practical Chosen Ciphertext Secure Encryption from Factoring. In: EUROCRYPT 2009. (2009) 313–332
17. Mei, Q., Li, B., Lu, X., Jia, D.: Chosen Ciphertext Secure Encryption under Factoring Assumption Revisited. In: Public Key Cryptography 2011. (2011) 210–227
18. McEliece, R.J.: A Public-Key Cryptosystem Based on Algebraic Coding Theory. In: Deep Space Network progress Report. (1978)
19. Dowsley, R., Müller-Quade, J., Nascimento, A.C.A.: A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model. In: CT-RSA 2009. (2009) 240–251
20. Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. In: STOC 2008. (2008) 187–196
21. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009. (2009) 333–342
22. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: EUROCRYPT 2010. (2010) 523–552
23. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: EUROCRYPT 2010. (2010) 553–572
24. Agrawal, S., Boneh, D., Boyen, X.: Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In: CRYPTO 2010. (2010) 98–115
25. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient Public Key Encryption Based on Ideal Lattices. In: ASIACRYPT 2009. (2009) 617–635
26. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: EUROCRYPT 2010. (2010) 1–23
27. Okamoto, T.: Authenticated Key Exchange and Key Encapsulation in the Standard Model. In: ASIACRYPT 2007. (2007) 474–484
28. Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In: CRYPTO 1998. (1998) 13–25
29. Kurosawa, K., Desmedt, Y.: A New Paradigm of Hybrid Encryption Scheme. In: CRYPTO 2004. (2004) 426–442
30. Hanaoka, G., Kurosawa, K.: Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. In: ASIACRYPT 2008. (2008) 308–325
31. Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and Efficient Public-Key Encryption from Computational Diffie-Hellman in the Standard Model. In: Public Key Cryptography 2010. (2010) 1–18
32. Kiltz, E.: Chosen-Ciphertext Secure Key-Encapsulation Based on Gap Hashed Diffie-Hellman. In: Public Key Cryptography 2007. (2007) 282–297
33. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: ACM Conference on Computer and Communications Security 2005. (2005) 320–329
34. Canetti, R., Goldreich, O., Halevi, S.: The Random Oracle Methodology, Revisited (Preliminary Version). In: STOC 1998. (1998) 131–140
35. Cramer, R., Shoup, V.: Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. In: SIAM Journal on Computing 33. (2004) 167–226
36. Chevallier-Mames, B., Joye, M.: Chosen-Ciphertext Secure RSA-Type Cryptosystems. In: ProvSec 2009. (2009) 32–46
37. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive Trapdoor Functions and Chosen-Ciphertext Security. In: EUROCRYPT 2010. (2010) 673–692

38. Bernstein, D.J., Lange, T., Peters, C.: Smaller Decoding Exponents: Ball-Collision Decoding. In: CRYPTO 2011. (2011) 743–760
39. Bernstein, D.J., Lange, T., Peters, C.: Wild McEliece. In: SAC 2010. (2010) 143–158
40. Nojima, R., Imai, H., Kobara, K., Morozov, K.: Semantic security for the McEliece cryptosystem without random oracles. Designs, Codes and Cryptography **49**(1-3) (2008) 289–305
41. Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller (2012) To appear in *EUROCRYPT 2012*.
42. Ajtai, M.: Generating Hard Instances of Lattice Problems (Extended Abstract). In: STOC 1996. (1996) 99–108 See also ECCC TR96-007.
43. Micciancio, D., Regev, O.: Worst-Case to Average-Case Reductions Based on Gaussian Measures. SIAM Journal on Computing **37**(1) (2007) 267–302 Preliminary version in *FOCS 2004*, 2004.
44. Lyubashevsky, V., Micciancio, D.: Generalized Compact Knapsacks are Collision Resistant. In: ICALP (2) 2006. (2006) 144–155
45. Peikert, C., Rosen, A.: Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In: TCC 2006. (2006) 145–166
46. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom Functions and Lattices (2012) To appear in *EUROCRYPT 2012*.