mathematics

MDPI

Article

Authenticated Key Exchange under Bad Randomness, Revisited

Hui Cui [1,*] and Glen Mudra [2]

[1] Faculty of IT, Monash University, Victoria, VIC 3800, Australia
[2] School of IT, Murdoch University, Perth, WA 6150, Australia; 33683512@student.murdoch.edu.au
* Correspondence: hui.cui@monash.edu

Abstract: A bad randomness may cause catastrophic results in security; thus, it is of importance to make cryptographic systems secure against bad randomness. In this paper, we focus on a practical situation where an adversary is able to force participants in an authenticated key exchange (AKE) system to reuse the random values and the functions of these values, called related randomness attack (RRA). Following the existing randomness resetting security model of AKE and the RRA security model of public-key encryption, we present a model of RRA security for authenticated key exchange, as well as the necessary restrictions on the related randomness functions used to obtain the security definition. Then we show how a related randomness attack adversary breaks the security of some existing AKE protocols, and propose some constructions of RRA-secure authenticated key exchange in the random oracle model and standard model, respectively.

Keywords: related randomness attack; authenticated key exchange; randomness resetting

MSC: 68P27

Citation: Cui, H.; Mudra, G. Authenticated Key Exchange under Bad Randomness, Revisited. *Mathematics* **2023**, *11*, 2721. https://doi.org/10.3390/math11122721

Academic Editor: Jonathan Blackledge

Received: 11 May 2023
Revised: 7 June 2023
Accepted: 14 June 2023
Published: 15 June 2023

1. Introduction

Cryptographic primitives are heavy users of randomness, but due to problems including insufficient estimation of system entropy, poor design of algorithms, bugs in software, and virtual machine randomness resetting, random number generators may fail to generate required randomness in practice [1]. This failure of randomness can cause catastrophic results: private signing keys of digital signatures could be exposed [2], low-entropy plaintexts in public-key encryption schemes might be recovered [3], the procedure of key generation would be severely weakened [4,5], ephemeral Diffie–Hellman keys may become predictable, resulting in the exposure of session keys [3], and electronic wallet security might be compromised [3]. Obviously, standard security notions of indistinguishability under chosen plaintext attacks or chosen ciphertext attacks [6] (IND-CPA or IND-CCA security) are not sufficient when these attacks on randomness are possible. This observation leads the research community to target effort into addressing this problem (e.g., [3,7–9]). However, it is unlikely that the failures of randomness can be completely eliminated [3]. A commonly adopted approach is to try to hedge against randomness failures, which can make cryptographic primitives offer some degree of security when encountering randomness failures.

1.1. Motivation and Contributions

An authenticated key exchange (AKE) protocol allows the communication of two parties to generate a common session key over an insecure network, which has been widely applied in real-world applications (e.g., online banking, virtual private networks (VPNs), wireless communication protocols such as Wi-Fi Protected Access (WPA), etc.) to secure network communications. An AKE protocol is composed of a tuple of randomized algorithms which take random coins produced by pseudorandom number generators (PRNGs) as the input and yield bit-strings computationally indistinguishable from truly

random strings given the truly fresh and random seeds [10]. However, in practice, these seeds are constructed via data collection from an entropy pool which could be controlled by an adversary who might modify the random data, making the randomness become bad. In this context, Yang et al. [11] raised a natural question: would existing well-known AKE protocols still be secure under bad randomness? They defined two security models for reset attacks (RAs) as Reset-1 (where the random coins used by the algorithms are controlled the adversary) and Reset-2 (where a device can be reset by the adversary to force algorithms to reuse certain random coins) to capture such a security, and showed that some widely used AKE protocols become insecure when the randomness becomes bad.

Motivated by the challenge of protecting security in the case of randomness failures, we consider the security for AKE under the setting of related randomness attacks (RRAs) [3], where the adversary not only can force to reuse existing random values as in the RA setting, but also can force to use those random values' functions, i.e., the random bits become predictable in such a way that the adversary is aware of the relations among the randomness in one session and its subsequent sessions. This capability is similar to the ability granted to the adversary in the setting of related-key attacks (RKAs) [6], under which an adversary is capable of tampering with the secret (or private) keys used in cryptographic computations. In actuality, the RA setting can be regarded as a special case of the RRA setting to allow the modelling of RAs such that the adversary cannot reset randomness, but the randomness used is in some way related to that used on previous sessions. These behaviours were discussed in the experimental work in [7], where the bad randomness was divided into reused randomness (the Reset-2 model), exposed randomness, predictable randomness (the RRA model), and chosen randomness (the Reset-1 model). Our RRA setting on authenticated key exchange builds on the bad randomness setting on AKE [11], and it also has interesting connections with related-key attacks for pseudorandom function (PRF) [12], and deterministic digital signature [11].

Contributions. We build an RRA security model for AKE in the RRA setting in this paper, under which the protocol is secure even after the adversary is able to reset a participant to use related random coins in multiple AKE sessions. Based on the security models Bellare–Rogaway (BR) [13] and Canetti–Krawczyk (CK) [14], we define the RRA security by providing additional capabilities to the adversaries in the Reset-2 model in [11]. In our RRA model, the adversary can reset to make the random coins used in multiple AKE sessions related to each other by satisfying some function defined by the adversary, which is called related-randomness deriving (RRD) function. Since the RRD functions can set the random coins in one AKE session identical to the random values in other AKE sessions, it is straightforward that the RRA model is stronger than the Reset-2 model. Different from that in the Reset-1 model [11], the adversary under related randomness attacks does not know the exact values of the used randomness, so our model allows the adversary to corrupt either participant's long-lived key, thereby capturing the weak forward secrecy (FS) [15], which requires that compromising the long-lived secret keys of the participants will not compromise any already established session key. This reflects that, similar to the relations between the Reset-1 and Reset-2 models analysed in [11], the Reset-1 and RRA models are incomparable and we need to preclude an RRA adversary from making queries such that the RRD functions are set to be constants, i.e., the adversary controls the random coins used in the AKE protocol.

Then we show that the AKE protocols in [11] become insecure against RRAs if the adversary can manipulate randomness in a way defined in related randomness attacks. In addition, we present techniques to build RRA-secure AKE protocols from the random oracle model and the standard model, respectively.

- Random oracle construction. This can be simply realised by hashing the output session key.
- Standard model construction. We achieve this by slightly changing the way of applying pseudorandom function (PRF) in Yang et al.'s ISO-R protocol.

The above two constructions mess up the relations of different session keys, thereby building RRA-secure authenticated key exchange protocols with restrictions on the related-randomness deriving functions and additional requirements on the PRFs.

In addition, given the power of the adversary in the RRA setting, a certain set of adversarial queries must be excluded to prevent the adversary from trivially breaking security. For example, constant functions must be disallowed for security in our RRA setting, which could be regarded as the Reset-1 model for the chosen randomness in [11], where the random coins are controlled by the adversary. When the related randomness functions ϕ are restricted to be from some set Φ, we name the functions Φ-restricted RRD functions, and call the corresponding adversary a Φ-restricted adversary.

Other Results. In [11], Yang et al. also presented a generic transformation on a Reset-2 secure AKE protocol to obtain a Reset-1 and Reset-2 secure ABE protocol. Their technique is to make a PRF be a strong randomness extractor (SRE) [11] such that the output of the PRF is close to uniform distribution even when the secret key used in the function is revealed. In our construction in the standard model, we ask the PRF to be RKA-secure to build the RRA-secure AKE protocols. Can our constructions for RRA-secure ABE be extended to cover the Reset-1 model? To our best knowledge, we cannot give an affirmative answer in this paper (this will be explained later with regard to the concrete constructions), and we leave this as an open problem.

1.2. Related Work

Authenticated Key Exchange. In 1993, Bellare and Rogaway [13] gave the first theoretical treatment of the security notion of AKE, which, referred to as the BR model, became the standard for analysing AKE protocols. Later, in 2001, Canetti and Krawczyk [14] gave another security model, known as the CK model, where they showed that AKE protocols composed with symmetric key encryption and authentication functions can be secure in their model to provide secure communication channels. Several popular AKE protocols (e.g., ISO [14], SIGMA [16,17], and HMQV [18]) have been proved to be secure under the CK model. LaMacchia, Lauter, and Mityagin [19] extended the CK model to the eCK model in 2007, where either the long-lived keys or the ephemeral keys of the participants of a protocol session can be comprised by the adversary. Even though there are many comparisons between the CK model and the eCK model [19–21], it is suggested by Boyd et al. [20] that these two models are incomparable.

Bad Randomness. For signatures, there exists a method that can avoid security issues arising from bad randomness while keeping the verification procedure as normal, which simply strengthens the private key in the signature scheme with a key for a PRF, and derives any needed randomness during the signing by applying this PRF to the "to be signed" message.

Regarding the randomness used in symmetric encryption setting, Rogaway [22] proposed nonce-based encryption, Rogaway and Shrimpton [23] proposed the notion of misuse-resistant authenticated-encryption concerning residual security when then nonce is repeated, and Kamara and Katz [24] introduced the security model of such attacks that the random coins are poorly generated, and showed generic transformations for achieving security in this context.

In the public-key setting, Bellare et al. [25] provided the best possible security guarantees for public-key encryption using bad randomness, and gave several public-key encryption schemes achieving this notion. Ristenpart and Yilek [7] studied the use of "hedge" to protect broad classes of randomness failures in already-deployed systems in the random oracle model, and performed this technique in OpenSSL. Yilek [8] focused on the public-key encryption security in a setting where resetting and reusing random numbers are possible, and presented a simple and efficient way to make any existing public-key encryption scheme secure under this model. Paterson, Schuldt, and Sibborn [3], to preserve security under randomness failures, initiated the study of security for public-key encryption in the setting of related randomness attack (RRA).

In terms of authenticated key exchange, Aiello et al. [26] discussed the reuse of Diffie–Hellman (DH) exponents in multiple AKE sessions, excluding reusing the same randomness to sign different messages in the authentication and key exchange phase. Yang et al. [11] presented its formal security model under bad randomness where the adversary is given the power of controlling or resetting the random coins used by the stateless AKE algorithms, but their approaches of building AKE protocols cannot be extended to achieve RRA security. Feltz and Cremers also [1] systematically analysed the security of both stateless and stateful AKE protocols under bad randomness, but the maliciously registered public keys are disallowed in their systems.

1.3. Organization

The rest in this paper is structured as follows. In Section 2, we briefly review the notions and definition related to this work. In Section 3, we elaborate the security model of RRA-secure AKE protocol. In Section 4, we point out some simple related randomness attacks on the AKE under bad randomness protocols. In Section 5, we present constructions of AKE with RRA security based on signature. In Section 6, we propose a construction of AKE with RRA security based on encryption. Finally, this paper is summarised in Section 7.

2. Preliminaries

In this section, we recall some basic notions to be used in this paper.

2.1. Pseudorandom Functions

Let $\mathcal{F} : \mathrm{K}_\lambda \times \mathrm{D}_\lambda \to \mathrm{R}_\lambda$ be a set of PRFs [10] with λ being a security parameter, and K_λ, D_λ, and R_λ being arbitrary finite sets. Following the security games in Figure 1, the advantage of a PRF adversary $\mathcal{A}$ against $\mathcal{F}$ is

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathcal{F},\mathcal{A}}(\lambda) = \Pr[\mathrm{REAL}^{\mathcal{A}}_{\mathcal{F}} \Rightarrow 1] - \Pr[\mathrm{RAND}^{\mathcal{A}}_{\mathcal{F}} \Rightarrow 1].$$

proc Initialize	proc Initialize	proc Initialize	proc Initialize
$K \leftarrow \mathrm{K}_\lambda$	CoinTab $\leftarrow \emptyset$	$K \leftarrow \mathrm{K}_\lambda$	$G \leftarrow \mathcal{F}(\mathrm{K}_\lambda, \mathrm{D}_\lambda, \mathrm{R}_\lambda)$
Return 1^λ	Return 1^λ	Return 1^λ	$K \leftarrow \mathrm{K}_\lambda$
proc $\mathcal{F}(x)$	proc $\mathcal{F}(x)$	proc $\mathcal{F}(\phi, x)$	Return 1^λ
Return $\mathcal{F}_K(x)$	If CoinTab$[x] = \perp$ then	Return $\mathcal{F}_{\phi(K)}(x)$	proc $\mathcal{F}(\phi, x)$
proc Finalize(b)	CoinTab$[x] \leftarrow \mathrm{R}_\lambda$	proc Finalize(b)	Return $G_{\phi(K)}(x)$
Return b	Return CoinTab$[x]$	Return b	proc Finalize(b)
	proc Finalize(b)		Return b
	Return b		

Figure 1. Games defining security and related-key attack security for a set of pseudorandom functions $\mathcal{F}$. Left side of ||: Game REAL is on the left while Game RAND is on the right. Right side of ||: Game RKA-REAL is on the left while Game RKA-RAND is on the right.

We say that $\mathcal{F}$ is a secure PRF family if the advantage of any probabilistic polynomial time adversary is negligible in the security parameter λ.

RKA-secure pseudorandom functions. Let Φ be a class of related-key deriving functions $\phi : K \to K$ mapping a key to a related key, which is a finite set of functions with the same domain and range [2]. Let $\mathcal{F} : \mathrm{K}_\lambda \times \mathrm{D}_\lambda \to \mathrm{R}_\lambda$ be a set of PRFs indexed by a security parameter λ with K_λ, D_λ, and R_λ being arbitrary finite sets. The advantage of a Φ-restricted related-key attack secure pseudorandom function (Φ-RKA-PRF) adversary $\mathcal{A}$ against $\mathcal{F}$ is

$$\mathbf{Adv}^{\Phi\text{-rka-prf}}_{\mathcal{F},\mathcal{A}}(\lambda) = \Pr[\text{RKA-REAL}^{\mathcal{A}}_{\mathcal{F}} \Rightarrow 1] - \Pr[\text{RKA-RAND}^{\mathcal{A}}_{\mathcal{F}} \Rightarrow 1],$$

where the security games (following the definitions in [12]) are shown in Figure 1.

We say that $\mathcal{F}$ is a related-key attack secure PRF family if the advantage of any Φ polynomial-time adversary (PPT) is negligible in the security parameter λ.

2.2. Deterministic Digital Signatures

If the signing algorithm $\mathcal{DS}$.Sign of a digital signature scheme $\mathcal{DS} = (\mathcal{DS}.\mathrm{SKG}, \mathcal{DS}.\mathrm{Sign}, \mathcal{DS}.\mathrm{Vf})$ is deterministic, then $\mathcal{DS}$ is deterministic. A randomised digital signature scheme can be transformed into a deterministic one as follows [11].

- Firstly, the signing key is expanded to include a uniformly random key K' from the key space of a PRF family.
- To sign a message m, it computes random coin $r = F'_{K'}(m)$, where F' is a pseudo-random function, and then invokes the randomised signing algorithm $\mathcal{DS}$.Sign with random coin r.

We say that $\mathcal{DS}$ is existentially unforgeable under adaptive chosen message attacks (uf-cma), if for any PPT algorithm $\mathcal{A}$, the advantage function

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{l|l} \mathcal{DS}.\mathrm{Vf}(\mathrm{VK}, m', \sigma') & \\ = \text{true} \wedge & (\mathrm{VK}, \mathrm{SK}) \leftarrow \mathcal{DS}.\mathrm{SKG}(1^\lambda). \\ \mathcal{A} \text{ never queried} & (m', \sigma') \leftarrow \mathcal{A}^{\mathcal{DS}.\mathrm{Sign}(\mathrm{SK},\cdot)}(VK). \\ \mathcal{DS}.\mathrm{Sign}(\mathrm{SK}, m') & \end{array}\right]$$

is negligible in the security parameter λ.

2.3. Complexity Assumption

Decisional Diffie–Hellman (DDH). The DDH problem is that for any PPT algorithm, it is difficult to distinguish (g, g^a, g^b, Z) from (g, g^a, g^b, g^{ab}), where $g, Z \in G$, $a, b \in Z_p^*$ are randomly and independently selected.

3. Modeling RRA Security

In this section, after reviewing the related notions about AKE, we describe its security model in detail.

3.1. Restricted Related-Randomness Deriving Functions

Let Φ be a set of functions that maps from randomness R to randomness R. Let α and β be positive integers. Based on the properties of the Φ-related-key deriving functions described in [12], we exhibit some necessary conditions that the Φ-restricted related-randomness deriving functions must satisfy with the difference that the latter is concerned with the functions executing on the randomness used in the AKE schemes rather than the PRF keys.

1. (α, β)-output-unpredictability for Φ. We say that a set Φ is output-unpredictable if, for all sets $P \subseteq \Phi$, $X \subseteq \mathrm{R}$ over the randomness r, the probability that there exist $\phi \in P$ and $r' \in X$ such that $\phi(r) = r'$, is negligible. This can be formally defined as

$$\mathrm{InSec}^{\mathrm{up}}_{\Phi}(\alpha, \beta) = \max\{\Pr[(\{\phi(r) : \phi \in P\} \cap X) \neq \emptyset \,\Big|\, r \leftarrow \mathrm{R}]\},$$

   where $P \subseteq \Phi$, $X \subseteq \mathrm{R}$, $|P| \leq \alpha$, $|X| \leq \beta$. This restriction guarantees that under related randomness attacks, the adversary has negligible probability of learning the randomness used in the queried session.
2. α-collision-resistance for Φ. We say that a set Φ is collision-resistant if, for all sets $P \subseteq \Phi$ over the randomness r, the probability that there exist two distinct $\phi_1, \phi_2 \in P$ such that $\phi_1(r) = \phi_2(r)$, is negligible. This can be formally defined as

$$\mathrm{InSec}^{\mathrm{cr}}_{\Phi}(\alpha) = \max\{\Pr[|\{\phi(r) : \phi \in P\}| \leq |P| \,\Big|\, r \leftarrow \mathrm{R}]\},$$

where $P \subseteq \Phi$, $|P| \leq \alpha$. This restriction makes sure that the adversary, given the access to the related randomness in the AKE system, has negligible probability of yielding the same session key within two different sessions.

3.2. Protocol Descriptions

An AKE protocol is composed of two PPT algorithms [11]: the long-lived key generation algorithm SKG (we consider the public-key setting in this paper where algorithm SKG returns a private and public key pair for each invocation) and the protocol execution algorithm P.

- **Protocol participants.** Let $\mathcal{U}$ be a set of parties which is not empty. Each party $U \in \mathcal{U}$ is named by a unique string with some fixed length. Let $\mathcal{MU}$ be a set of malicious parties added by adversary $\mathcal{A}$ to the system after the initialisation stage. Every malicious participant $MU \in \mathcal{MU}$ is also assigned with a distinctive fixed-length string without being used by another party inside the system.
- **Long-lived keys.** Each participant $U \in \mathcal{U}$ has a public key pk_U and private key sk_U created by the SKG algorithm, but each participant $MU \in \mathcal{MU}$'s public key pk_{MU} can be any value as long as pk_{MU} has never been claimed as the public key by another participant inside the system (this is to ensure that pk_{MU} is uniquely possessed by each party).
- **Instances.** One participant may run many instances at the same time. We denote instance i of party U by Π^i_U. When a new instance is built, a unique instance number within the party is selected, a sequence of random coins are created and added to that instance, and the instance is set to the "ready" state.
- **Protocol execution.** A protocol execution algorithm decides how instances of participants behave to respond to messages from their environment. Upon receiving an incoming message M_{in}, an instance runs the protocol P and creates

$$(M_{out}, \text{acc}, \text{term}^i_U, \text{sid}^i_U, \text{pid}^i_U, ssk, St^i_U) \leftarrow \\ \text{P}(1^k, U, pk_U, sk_U, St^i_U, M_{in}),$$

  where M_{out} is the responding message, acc is the decision made by the instance, term^i_U is whether the protocol execution has been terminated, sid^i_U is the session identity and pid^i_u is the partner identity that may be generated during the protocol run, ssk is the session key hold by the instance when the decision is accepted, and St^i_U is the internal state information which is deleted from U's memory once term^i_U is true.
  In this paper, unless otherwise stated explicitly, the session identity will be defined as the concatenation of the messages exchanged between the two participants in the form of (**initiator-message**||**responder-message**), and two matching instances will generate the same session identity.
- **Partnership.** The partnership between two instances is defined via the partner identity (named as) pid with which the instance believes it has an exchanged key, and the session identity sid uniquely labelling the AKE session as an identifier. If $\text{pid}^i_U = V$, $\text{pid}^j_V = U$ and $\text{sid}^i_U = \text{sid}^j_V$, we say that two instances Π^i_U and Π^j_V are partners.

3.3. Security Model

We define the security model RRA-AKE to capture the scenario where the related randomness will be used in an authenticated key exchange protocol, on the basis of the strong corruption Reset-2 security model, under the assumption that the all honest participants' long-lived keys in the set $\mathcal{U}$ are securely yielded with fresh random coins, defined in [11].

RRA Security Model. In this case, we consider that the adversary can launch related randomness attacks but cannot directly set random coins' values. The game RRA-AKE,

defined in Figure 2, is used to define the security of AKE protocols in the related randomness setting, of which the queries are explained as follows.

procedure Initialise
For all $U \in \mathcal{U}$
  $(pk_U, sk_U) \leftarrow \mathrm{SKG}(1^\lambda, U)$; $T_U \leftarrow \emptyset$
$b \leftarrow \{0,1\}$; $\mathcal{MU} \leftarrow \emptyset$
Return $\{pk_U\}_{U \in \mathcal{U}}$

procedure AddUser(U, pk_U)
If $(U \in (\mathcal{U} \cup \mathcal{MU})) \vee$
  $(pk_U \in \{pk_V\}_{V \in (\mathcal{U} \cup \mathcal{MU})})$
  Return Invalid
$\mathcal{MU} \leftarrow \{U\} \cup \mathcal{MU}$
Return true

procedure NewInstance(U, i, i', ϕ)
If $(U \notin \mathcal{U}) \vee (i \in T_U)$ Return Invalid
If $(i' \neq \perp) \wedge (i' \notin T_U)$ Return Invalid
If $(i' = \perp)$ $R^i_U \leftarrow$ Rnd
Else $R^i_U \leftarrow \phi(R^{i'}_U)$
If $R^i_U = \perp$ Return Invalid
$T_U \leftarrow T_U \cup \{i\}$; $St^i_U \leftarrow (R^i_U, \text{ready})$
$\mathrm{acc}^i_U \leftarrow$ false; $\mathrm{term}^i_U \leftarrow$ false
$\mathrm{sid}^i_U \leftarrow \perp$; $\mathrm{pid}^i_U \leftarrow \perp$; $ssk^i_U \leftarrow \perp$
Return true

procedure Send(U, i, M_{in})
If $(U \notin \mathcal{U}) \vee (i \notin T_U) \vee \mathrm{term}^i_U$
  Return Invalid
$(M_{out}, \mathrm{acc}, \mathrm{term}^i_U, \mathrm{sid}^i_u, \mathrm{pid}^i_U, ssk, St^i_U)$
$\leftarrow \mathrm{P}(1^k, U, pk_U, sk_U, St^i_U, M_{in})$
If acc $\wedge$ (not acc^i_U)
  $ssk^i_U \leftarrow ssk$; $\mathrm{acc}^i_U \leftarrow$ true
Return $(M_{out}, \mathrm{acc}, \mathrm{term}^i_U, \mathrm{sid}^i_U, \mathrm{pid}^i_U)$

procedure Reveal(U, i)
If $(U \notin \mathcal{U}) \vee (i \notin T_U)$ Return Invalid
Time[Reveal, (U, i)] $\leftarrow$ true
Return ssk^i_U

procedure RevealState(U, i)
If $(U \notin \mathcal{U}) \vee (i \notin T_U)$ Return Invalid
Time[Reveal, (U, i)] $\leftarrow$ true
Return St^i_U

procedure Corrupt(U)
If $(U \notin \mathcal{U})$ Return Invalid
Time[Corrupt, U] $\leftarrow$ true
Return sk_U

procedure Test(U^*, i^*)
If $(U^* \notin \mathcal{U})$ Return Invalid
If (not $\mathrm{acc}^{i^*}_{U^*}$) Return Invalid
$K \leftarrow$ KeySpace
If $b = 0$ Return K
Else Return $ssk^{i^*}_{U^*}$

procedure Finalize(b')
$V^* \leftarrow \mathrm{pid}^{i^*}_{U^*}$
If $(V^* \notin \mathcal{U})$ Return false
If Time[Reveal, (U^*, i^*)] $\vee$
  Time[RevealState, (U^*, i^*)]
  Return false
If $\exists\, j^* \in T_{V^*}, (\mathrm{pid}^{j^*}_{V^*} = U^*) \wedge$
  $(\mathrm{sid}^{j^*}_{V^*} = \mathrm{sid}^{i^*}_{U^*})$
  If $\exists\, j, j \neq j^* \wedge (R^j_{V^*} = R^{j^*}_{V^*})$
    If $\exists\, i, (i \neq i^*) \wedge (R^i_{U^*} = R^{i^*}_{U^*})$
      Return false
  If Time[Reveal, (V^*, j^*)] $\vee$
    Time[RevealState, (V^*, j^*)]
    Return false
If Time[Corrupt, U^*] $\vee$
    Time[Corrupt, V^*]
    Return false
Return $(b = b')$

Figure 2. Game RRA-AKE in the strong corruption model. If the RevealState queries are removed from the game, it becomes the game of RRA-AKE security in the weak corruption model.

- AddUser(U, pk_U): With this query, adversary $\mathcal{A}$ can add a new user U with public key pk_U. The adversary does not need to prove the knowledge on the secret key corresponding to pk_U. In other words, either the public key pk_U or the user identity U exists in the system.
- NewInstance(U, i, j, ϕ): This query allows adversary $\mathcal{A}$ to initialise a new instance Π^i_U within party U. Adversary $\mathcal{A}$ can specify an existing instance Π^j_U, and Adversary $\mathcal{A}$ can set $j = \perp$ to make Π^i_U use fresh random coins.
- Send(U, i, M_{in}): This query invokes the instance i of U with a message M_{in}. The instance runs $\mathrm{P}(1^k, U, pk_U, sk_U, St^i_U, M_{in})$, and sends the response to adversary $\mathcal{A}$ which includes whether Π^i_U terminates or accepts the session identity sid^i_U and partner identity pid^i_U when they are available.

- Reveal(U, i): The key ssk_U^i is returned to adversary $\mathcal{A}$ if instance Π_U^i has been accepted and a session key ssk_U^i is generated.
- RevealState(U, i): Adversary $\mathcal{A}$ obtains the state information St_U^i from making this query. This is similar to the Canetti–Krawczyk approach [14] where adversary $\mathcal{A}$ is allowed to obtain the secret information stored in the parties memories. Note that adversary $\mathcal{A}$ is prohibited from issuing this query to the target instance $\Pi_{U^*}^{i^*}$ or its partner instance $\Pi_{V^*}^{j^*}$ (if exists).
- Corrupt(U): Adversary $\mathcal{A}$ can access to the party U's long-lived secret key sk_U from this query.
- Test(U^*, i^*): This query is only issued one time in the whole game. Adversary $\mathcal{A}$ in this query chooses a challenge instance $\Pi_{U^*}^{i^*}$. If $\Pi_{U^*}^{i^*}$ is accepted and a session key $ssk_{U^*}^{i^*}$ is created, then ssk_U^* is returned to adversary $\mathcal{A}$ if the coin b flipped in the Initialise phase or a session key randomly selected from the session key space is returned to adversary $\mathcal{A}$ if the coin b is 0.

An adversary's success is determined by its capability to distinguish a random key in the session key space from a real session key. However, some queries could expose the session keys; thus, adversary $\mathcal{A}$ can trivially win the game by asking these queries.

- Adversary $\mathcal{A}$ can obtain a session key if adversary $\mathcal{A}$ itself is one of the parties involved in that session.
- Adversary $\mathcal{A}$ can know the value of a session key from a Reveal query. Under related randomness attacks, adversary $\mathcal{A}$ can also learn a session key through reset-and-reply attacks [11] where adversary $\mathcal{A}$ first invokes a protocol execution between instance Π_U^i with random value R_U^i and instance Π_V^j with random value R_V^j, and then it activates another instance $\Pi_U^{i'}$ with random value $\phi(R_U^i) = R_U^i$. Thus, adversary $\mathcal{A}$ can make $ssk_U^i = ssk_U^{i'}$ by replaying the same message from Π_V^j. In this case, revealing ssk_U^i (or $ssk_U^{i'}$) will simultaneously disclose $ssk_U^{i'}$ (or ssk_U^i). Such attacks tell that if the randomness of instance Π_U^i is used by instance $\Pi_U^{i'}$ in a way that $R_U^{i'} = \phi(R_U^i) = R_U^i$ and their partner instances $\Pi_V^{j'}$ and Π_V^j share the same randomness (i.e., the random tapes at the sides of the initiator and the responder are reset to the identical ones for a previous session), it is impossible to guarantee the security on the session keys generated during these two instances. Therefore, when defining the freshness of an instance, it is necessary to require that either its randomness or its partner's randomness will never be used by another instance in the same format. In this paper, our goal is to design AKE protocols secure against related-randomness attacks such that the security of session keys generated by those one-side reset and un-reset instances will not be affected.

We do not consider these trivial attacks, and adversary $\mathcal{A}$ is said to be successful only if it can specify a fresh one in the Test query [11].

This model can be used to achieve forward secrecy (fs), which requires that the adversary does not have an advantage in revealing any (already) created session key by compromising the long-lived secret keys of two users. If an instance Π_U^i $(U \in \mathcal{U})$ is true in any of the conditions below, we say that it is fs-unfresh in the RRA model.

1. pid_U^i is generated by adversary $\mathcal{A}$ from an AddUser query.
2. Adversary $\mathcal{A}$ exposes the session key of either Π_U^i or its partner instance Π_V^j (if it exists).
3. There exists another instance of U whose session key equals that of either Π_U^i (ssk_U^i) or Π_U^i's partner instance Π_V^j $(ssk_V^j$, if it exists), i.e., related randomness attacks $\phi(R_U^i) = R_U^i$ against Π_U^i and $\phi(R_V^j) = R_V^j$ against Π_U^i's partner instance Π_V^j (if it exists) have happened.
4. Adversary $\mathcal{A}$ corrupts pid_U^i if Π_U^i does not have a partner instance.

Otherwise, Π_U^i is said to be fs-fresh.

Definition 1. *We denote $\mathcal{AKE}$ as an AKE protocol, $\mathcal{A}$ as a Φ-restricted RRA adversary against $\mathcal{AKE}$, and λ as a security parameter. Adversary $\mathcal{A}$'s advantage is defined to be*

$$\boldsymbol{Adv}^{rra\text{-}ake}_{\mathcal{AKE},\mathcal{A}}(\lambda) = Pr[RRA\text{-}AKE_{\mathcal{AKE},\mathcal{A}}(\lambda) \Rightarrow true] - 1/2.$$

The protocol $\mathcal{AKE}$ is said to be RRA-secure if

1. *Two partnering instances generate the same session key when a benign adversary honestly transmits messages;*
2. *For any PPT adversary $\mathcal{A}$, $\boldsymbol{Adv}^{rra\text{-}ake}_{\mathcal{AKE},\mathcal{A}}(\lambda)$ is negligible in the security parameter λ.*

4. Security Analysis on Yang's Authenticated Key Exchange Protocols

In this section, we point out that Yang's authenticated key exchange protocols under bad randomness are vulnerable in our security model.

4.1. A Related Randomness Attack on Yang's ISO-R2 Protocol

We define G as a prime-order (which is p) group with a generator g and we let $\mathcal{DS} = (\mathcal{DS}.\text{SKG}, \mathcal{DS}.\text{Sign}, \mathcal{DS}.\text{Vf})$ be a deterministic digital signature scheme. We revisit the AKE protocol under bad randomness ISO-R2 of Yang et al. [11] between two entities A and B in Figure 3.

A		B
$(pk_A, sk_A) \leftarrow \mathcal{DS}.\text{SKG}(1^\lambda)$		$(pk_B, sk_B) \leftarrow \mathcal{DS}.\text{SKG}(1^\lambda)$
$x \in Z_p^*, X \leftarrow g^x$		
	$\xrightarrow{A, X}$	$y \in Z_p^*, Y \leftarrow g^y, \text{sid} \leftarrow X\|\|Y$
	$\xleftarrow{B, Y, \sigma_B}$	$\sigma_B \leftarrow \mathcal{DS}.\text{Sign}(sk_B, X, Y, A, 0)$
$\text{sid} \leftarrow X\|\|Y$		
$\sigma_A \leftarrow \mathcal{DS}.\text{Sign}(sk_A, Y, X, B, 1)$	$\xrightarrow{A, \sigma_A}$	
$ssk \leftarrow g^{xy}$		$ssk \leftarrow g^{xy}$

Figure 3. The ISO-R2 protocol between the participants A and B.

The attack. We show that under the related randomness attacks, the security of this scheme can be broken. Let the related-randomness deriving functions $\phi : Z_p^* \to Z_p^*$ be indexed by $\triangle \in Z_p^*$ such that $\phi(r) = r * \triangle \neq r$, where $*$ could be addition $(+)$ or multiplication $(\cdot)$.

1. The adversary activates a new session of A. After receiving (A, X) for $X = g^x$ from A, the adversary sends (A, X) to B.
2. * The adversary activates A another session of using the related randomness $x' = \phi(x) = x * \triangle$. After receiving (A, X') for $X' = g^{x'}$ from A, the adversary sends (A, X') to B.
3. After receiving (B, Y, σ_B) for $Y = g^y$ from B, the adversary sends back (B, Y, σ_B) to A.
4. * The adversary activates B to use the related randomness $y' = \phi(y) = y$. After receiving (B, Y', σ'_B) for $Y' = g^y, \sigma'_B \leftarrow \mathcal{DS}.\text{Sign}(sk_B, X, Y', A, 0)$ from B, the adversary sends back (B, Y', σ'_B) to A.
5. After receiving σ_A from A, the adversary sends back σ_A to B.
6. * The adversary receives σ'_A for $\sigma'_A \leftarrow \mathcal{DS}.\text{Sign}(sk_A, Y', X, B, 1)$ from A, and sends back σ'_A to B.

In the above process, the adversary builds two sessions between A and B with $\text{sid} = X||Y$ and $\text{sid}' = X'||Y$, respectively. In this case, once the adversary knows the session key sid', it obtains the session key sid from

$$g^{x'y} = g^{xy}Y^{\triangle} \text{ or } g^{x'y} = (g^{xy})^{\triangle}.$$

Notice that since the reset attack has happened to B, the result of $\phi(x)$ cannot be x. Otherwise, the session $\text{sid} = X||Y$ between A and B will not regarded as a fresh session.

4.2. A Related Randomness Attack on Yang's ISO-R Protocol

Let G be a group of prime order p with a generator g. Let $\mathbb{F} = \{F_K : \{0,1\}^{\rho(\lambda)} \to Z_p^* \mid K \in Z_p^*\}$ be a pseudorandom function family and a strong randomness extractor (SRE) [27], where $\rho(\lambda)$ is the polynomial of λ. Let $\mathcal{DS} = (\mathcal{DS}.\text{SKG}, \mathcal{DS}.\text{Sign}, \mathcal{DS}.\text{Vf})$ be a deterministic digital signature scheme. We revisit the AKE protocol under bad randomness ISO-R between two entities A and B of Yang et al. [11] in Figure 4.

A		B
$a \leftarrow Z_p^*$		$b \leftarrow Z_p^*$
$(pk, sk) \leftarrow \mathcal{DS}.\text{SKG}(1^\lambda)$		$(pk', sk') \leftarrow \mathcal{DS}.\text{SKG}(1^\lambda)$
$pk_A \leftarrow pk, sk_A \leftarrow (sk, a)$		$pk_B \leftarrow pk', sk_B \leftarrow (sk', b)$
$\widetilde{x} \leftarrow \{0,1\}^{\rho(\lambda)}$		
$x \leftarrow F_a(\widetilde{x})$	$A, X \longrightarrow$	
$X \leftarrow g^x$		$\widetilde{y} \leftarrow \{0,1\}^{\rho(\lambda)}$
		$y \leftarrow F_b(\widetilde{y})$
	$\leftarrow B, Y, \sigma_B$	$Y \leftarrow g^y$
		$\text{sid} \leftarrow X\|\|Y$
		$\sigma_B \leftarrow \mathcal{DS}.\text{Sign}(sk', X, Y, A, 0)$
$\text{sid} \leftarrow X\|\|Y$		
$\sigma_A \leftarrow \mathcal{DS}.\text{Sign}(sk, Y, X, B, 1)$	$\sigma_A \longrightarrow$	
$ssk \leftarrow g^{xy}$		
		$ssk \leftarrow g^{xy}$

Figure 4. The ISO-R protocol between two participants A and B.

The attack. We show that under the related randomness attacks, the security of this scheme can be broken. Let the related-randomness deriving functions $\phi : Z_p^* \to Z_p^*$ be indexed by $\triangle \in Z_p^*$ such that $\phi(r) \neq r$. The attack mostly follows that ofISO-R2 protocol. After the attack, the adversary builds two sessions between A and B with $\text{sid} = X||Y$ and $\text{sid}' = X'||Y$, respectively. As the adversary is given A's long-lived secret key a, it is highly possible that it controls the pseudorandom function F_a and thereby inferring the relationship between $x' = F_a(\phi(\widetilde{x}))$ and $x = F_a(\widetilde{x})$ (if F_a is poorly built in the form such as $g^{a\widetilde{x}}$). Denote the relation between x and x' as $x' = x * \triangle$. If $*$ is addition $(+)$ or multiplication $(\cdot)$, then once the adversary knows the session key of sid', it obtains the session key of sid from

$$g^{x'y} = g^{xy}Y^{\triangle} \text{ or } g^{x'y} = (g^{xy})^{\triangle}.$$

Likewise, $\phi(\widetilde{x})$ cannot be equal $\widetilde{x}$. Otherwise, the session between A and B for $\text{sid} = X||Y$ is not a fresh session.

5. Authenticated Key Exchange under Related Randomness Attacks Based on Signature

In the previous section, we show that Yang's authenticated key exchange protocols are insecure under the related randomness attack model. Here, we modify Yang's authenticated

key exchange protocols based on signature to make it resistant against related randomness attacks. For brevity, we omit all the related verification algorithms in the protocols.

5.1. Construction in the Random Oracle Model

In Figure 5, we present a slightly modified protocol to Yang's ISO-R2 protocol [8], and call it ISO-RR2. We denote G as a group of prime order p with a generator g, and H as a collision resistant hash function. We let $\mathcal{DS} = (\mathcal{DS}.\text{SKG}, \mathcal{DS}.\text{Sign}, \mathcal{DS}.\text{Vf})$ be a deterministic digital signature scheme.

A		B
$(pk_A, sk_A) \leftarrow \mathcal{DS}.\text{SKG}(1^k)$		$(pk_B, sk_B) \leftarrow \mathcal{DS}.\text{SKG}(1^k)$
$x \in Z_p^*, X \leftarrow g^x$		
	$\xrightarrow{A, X}$	$y \in Z_p^*, Y \leftarrow g^y, \text{sid} \leftarrow X\|\|Y$
	$\xleftarrow{B, Y, \sigma_B}$	$\sigma_B \leftarrow \mathcal{DS}.\text{Sign}(sk_B, X, Y, A, 0)$
$\text{sid} \leftarrow X\|\|Y$		
$\sigma_A \leftarrow \mathcal{DS}.\text{Sign}(sk_A, Y, X, B, 1)$	$\xrightarrow{A, \sigma_A}$	
$ssk \leftarrow H(g^{xy})$		$ssk \leftarrow H(g^{xy})$

Figure 5. The ISO-RR2 protocol between two participants A and B.

Related Randomness Attack. The ISO-RR2 protocol is similar to Yang's ISO-R2 protocol with the difference that a hash function is used to generate the session key. The modification can prevent the related randomness attack described in Section 4, because the relation between two session keys is messed up by the hash function, thereby disabling the adversary to learn one session key from the other session key.

Interleaving attacks [11]. An interleaving attack occurs when the session identity is denoted as the concatenation of the initiator's and the responder's random group elements. To resist such an attack, a role indicator ('1' for initiator and '0' for responder) can be added into the signed message of each party. In addition to adding a role indicator, there are other ways to resolve the problem [11]: (1) denoting "(self-message||peer-message)" as the session identity to make different session identities for two matching instances; (2) using an explicit session identity rather than the concatenation of exchanged messages between two entities.

Theorem 1. *The ISO-RR2 protocol is RRA-secure for a Φ-restricted adversary in the random oracle model if DDH assumption holds in the underlying group and the deterministic digital signature $\mathcal{DS}$ is a uf-cma secure.*

Proof. The proof of this part is very similar to that in [11] except that in the last game, the session key outputted by the simulator is from a random oracle controlled by the simulator itself. It is not difficult to see that the random oracle plays a very important role here, which prevents the adversary, given the relation between the random coins in different instances, from learning one session key from another session key. We omit the details of the proof. □

Reset-1 Security. In the Reset-1 model, the adversary controls the randomness and is not allowed to corrupt the long-lived key of either participant in the protocol. Thus, in the ISO-RR2 protocol, x is chosen by the adversary. Similar to the analysis about the ISO-R2 protocol in [11], it is not difficult to see that the ISO-RR2 protocol cannot achieve the Reset-1 security.

5.2. Construction from RKA-PRFs

In Figure 6, we present a slightly modified protocol to Yang's ISO-R protocol [11] (denoted by ISO-RR), and prove its RRA security. We denote G as a group of prime order p wiht a generator of g. We denote $\mathbb{F} = \{F_K : \{0,1\}^{\rho(\lambda)} \to Z_p^* \mid K \in Z_p^*\}$ as a related-key attack secure PRF family, where $\rho(\lambda)$ is the polynomial of λ. We assume that $\mathcal{DS} = (\mathcal{DS}.\text{SKG}, \mathcal{DS}.\text{Sign}, \mathcal{DS}.\text{Vf})$ is a deterministic digital signature scheme.

A		B
$a \leftarrow Z_p^*$		$b \leftarrow Z_p^*$
$(pk, sk) \leftarrow \mathcal{DS}.\text{SKG}(1^\lambda)$		$(pk', sk') \leftarrow \mathcal{DS}.\text{SKG}(1^\lambda)$
$pk_A \leftarrow pk, sk_A \leftarrow (sk, a)$		$pk_B \leftarrow pk', sk_B \leftarrow (sk', b)$
$\widetilde{x} \leftarrow \{0,1\}^{\rho(\lambda)}$		
$x \leftarrow F_{\widetilde{x}}(a \oplus \widetilde{x})$	A, X	
$X \leftarrow g^x$	$\longrightarrow$	$\widetilde{y} \leftarrow \{0,1\}^{\rho(\lambda)}$
		$y \leftarrow F_{\widetilde{y}}(b \oplus \widetilde{y})$
	B, Y, σ_B	$Y \leftarrow g^y$
	$\longleftarrow$	$\text{sid} \leftarrow X\|\|Y$
		$\sigma_B \leftarrow \mathcal{DS}.\text{Sign}(sk', X, Y, A, 0)$
$\text{sid} \leftarrow X\|\|Y$		
$\sigma_A \leftarrow \mathcal{DS}.\text{Sign}(sk, Y, X, B, 1)$	σ_A	
$ssk \leftarrow g^{xy}$	$\longrightarrow$	
		$ssk \leftarrow g^{xy}$

Figure 6. The ISO-RR protocol between two participants A and B.

The ISO-RR protocol is similar to Yang's ISO-R protocol with the difference that the pseudorandom function F is related-key attack secure, and it takes the randomness, rather than the long-lived key, as the key. The reason that Yang's ISO-R protocol fails to prevent related randomness attacks is that when the long-lived key and the underlying relationship of the random coins are known to the adversary, it is hard to assume that the outputs of the strong randomness extractor SRE function [27] under different random coins are still independent of each other. Our modification is to prevent the related randomness attack described in Section 4, as now, due to the RKA security, the outputs of the pseudorandom function become unknown to the adversary.

Theorem 2. *The ISO-RR protocol is secure in the RRA-AKE model for a Φ-restricted adversary if the deterministic digital signature $\mathcal{DS}$ is a uf-cma secure, $\mathbb{F}$ is a Φ-restricted related-key attack secure pseudorandom function family, and DDH assumption holds in the underlying group.*

Proof. If an adversary in the Test query outputs an instance (U^*, i^*), then there should be a partner instance (V^*, j^*) for (U^*, i^*). Otherwise, the security of the signature scheme $\mathcal{DS}$ is broken. We prove it as follows.

Let $\mathcal{A}_1$ be a restricted RRA adversary that adversary $\mathcal{A}_1$ creates an instance (U^*, i^*) with a partner instance (V^*, j^*) in the Test query. Given an adversary $\mathcal{A}$ against the RRA security model in the ISO-RR protocol, we construct adversary $\mathcal{A}_1$ to answer all adversary $\mathcal{A}$'s queries using its own oracle. If adversary $\mathcal{A}$ generates an instance (U^*, i^*) without a partner instance, adversary $\mathcal{A}_1$ halts. Otherwise, adversary $\mathcal{A}_1$ generates (U^*, i^*) in the Test query, and returns the received response to adversary $\mathcal{A}$. When adversary $\mathcal{A}$ outputs a bit b' and halts, adversary $\mathcal{A}_1$ outputs b' and halts.

We denote Forge by the event that adversary $\mathcal{A}$ in the game generates a pair (m^*, σ^*) such that a party $I \in \mathcal{U}$, which is not corrupted when adversary $\mathcal{A}$ outputs (m^*, σ^*), exists such that true $\leftarrow \mathcal{DS}.\text{Vf}(pk_I, m^*, \sigma^*)$, and the party I has never created a signature on message m^*.

We denote E as the event that adversary $\mathcal{A}$ outputs an instance (U^*, i^*) (without a partner instance) in the Test query. A Forge event occurs if the event E occurs. Adversary $\mathcal{A}_1$

and adversary $\mathcal{A}$ will be the same if the event E does not occur. Therefore, we conclude that

$$\mathbf{Adv}^{\text{rra-ake}}_{\text{ISO-RR},\mathcal{A}}(\lambda) - \mathbf{Adv}^{\text{rra-ake}}_{\text{ISO-RR},\mathcal{A}_1}(\lambda) \leq \Pr[\text{E}] \leq \Pr[\text{Forge}].$$

Below, we prove that there is a negligible probability for the event Forge to occur, or the signature $\mathcal{DS}$'s uf-cma security is broken. Given adversary $\mathcal{A}$ in the original RRA-AKE game, we can build a signature forger $\mathcal{S}$ which is given a public key pk created by $(pk, sk) \leftarrow \mathcal{DS}.\text{SKG}(1^k)$, and the access to the signing oracle $\mathcal{DS}.\text{Sign}(sk, \cdot)$. Forger $\mathcal{S}$ randomly chooses an entity $U \in \mathcal{U}$, sets $pk_U = pk$, and then creates the long-lived keys for all entities in the set $\mathcal{U} \setminus \{U\}$ for $|\mathcal{U}| = n$.

Forger $\mathcal{S}$ simulates the original RRA-AKE game for adversary $\mathcal{A}$. If a Forge event occurs in the simulation and $I = U$, then Forger $\mathcal{S}$ outputs the forgery by adversary $\mathcal{A}$ and halts. Thus, we conclude that

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},\mathcal{S}}(\lambda) \geq \frac{1}{n}\Pr[\text{Forge}].$$

Therefore, we have

$$\mathbf{Adv}^{\text{rra-ake}}_{\text{ISO-RR},\mathcal{A}_1}(\lambda) \geq \mathbf{Adv}^{\text{rra-ake}}_{\text{ISO-RR},\mathcal{A}}(\lambda) - n \cdot \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},\mathcal{S}}(\lambda).$$

Given adversary $\mathcal{A}_1$ with advantage $\mathbf{Adv}^{\text{rr-ake}}_{\text{ISO-RR},\mathcal{A}_1}(k)$, we define another restricted adversary, $\mathcal{A}_2$, which outputs two integers l and l' after the Initialise phase. Adversary $\mathcal{A}_2$ assumes that the Test session outputted by adversary $\mathcal{A}_1$ is between the l-th and l'-th instances. Given adversary $\mathcal{A}_1$ making at most q_I NewInstance queries, adversary $\mathcal{A}_2$ can be constructed as

$$\mathbf{Adv}^{\text{rra-ake}}_{\text{ISO-RR},\mathcal{A}_2}(\lambda) \geq \frac{1}{q_I(q_I-1)}\mathbf{Adv}^{\text{rr-ake}}_{\text{ISO-RR},\mathcal{A}_1}(\lambda).$$

Game$_1$**.** Let $F_{K_{U^*}}(\cdot)$ be the PRF with the key K_{U^*} used by the party U^* in the RRA-AKE game, and $F_{K_{V^*}}(\cdot)$ be the PRF with the key K_{V^*} used by the party V^* in the RRA-AKE game. We modify the RRA-AKE game for adversary $\mathcal{A}_2$ to a game Game_1 such that the output of the function $F_{K_{U^*}}(\cdot)$ in the l-th instance (or (U^*, i^*)) is a random string, and the output of the function $F_{K_{V^*}}(\cdot)$ in l'-th instance (or (V^*, j^*)) is another random string. Adversary $\mathcal{A}_2$ has similar advantages in the original RRA-AKE game and game Game_1, or we can construct an adversary $\mathcal{D}$ against the RKA security of the PRF.

Adversary $\mathcal{D}$ has access to an oracle $\mathcal{O}$ which returns either a true result of $F_{\phi(K)}(\cdot)$ or a random output of $G_{\phi(K)}(\cdot)$, where $G \in \mathbb{F}$. Adversary $\mathcal{D}$ simulates the RRA-AKE game by honestly running all operations except that adversary $\mathcal{D}$ simulates the PRFs $F_{K_{U^*}}(\cdot)$ of party U^* and $F_{K_{V^*}}(\cdot)$ of party V^* by asking its own oracle. Finally, when adversary $\mathcal{A}_2$ outputs a bit b' and halts, adversary $\mathcal{D}$ outputs the same b' and halts. We can then conclude that

$$\begin{aligned}
2 \cdot \mathbf{Adv}^{\text{rka-prf}}_{\mathbb{F},\mathcal{D}}(\lambda) &= \Pr[\mathcal{D}^{F_{\phi(K)}(\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{D}^{G_{\phi(K)}(\cdot)}(1^\lambda) = 1] \\
&= \Pr[\mathcal{A}_2 \text{ wins the game} | \mathcal{O} = F_{\phi(K)}(\cdot)] - \\
&\quad \Pr[\mathcal{A}_2 \text{ wins the game} | \mathcal{O} = G_{\phi(K)}(\cdot)] \\
&= \Pr[\text{RRA-AKE}^{\text{ISO-RR},\mathcal{A}_2}(\lambda) \Rightarrow \text{true}] - \\
&\quad \Pr[\text{Game}_1^{\text{ISO-RR},\mathcal{A}_2}(\lambda) \Rightarrow \text{true}] \\
&= \mathbf{Adv}^{\text{rra-ake}}_{\text{ISO-RR},\mathcal{A}_2}(\lambda) - \mathbf{Adv}^{\text{G}_1}_{\text{ISO-RR},\mathcal{A}_2}(\lambda).
\end{aligned}$$

Game$_2$**.** We modify the game Game_1 to a game Game_2 such that the simulator randomly chooses a key and sets it as the session key of the l-th and the l'-th instances. Adversary $\mathcal{A}_2$ has similar advantages in game Game_1 and game Game_2, or we can construct an adversary $\mathcal{B}$ breaking the DDH assumption.

Given a tuple $\{g, g^a, g^b, Z\}$, adversary $\mathcal{B}$'s goal is to guess whether Z is a random group element or $Z = g^{ab}$. Following the procedure of the game Game_1, adversary $\mathcal{B}$ simulates the game Game_2 for adversary $\mathcal{A}_2$ except that the ephemeral public key in the l-th instance is set to be X by adversary $\mathcal{B}$, the ephemeral public key in the l'-th instance is set to be Y by adversary $\mathcal{B}$, and the session key of the l-th and the l'-th instances is set to be Z by adversary $\mathcal{B}$. We then conclude that

$$\begin{aligned}\mathbf{Adv}_{\mathcal{B}}^{\text{DDH}}(\lambda) &= \Pr[\mathcal{A}_2 \text{ wins the game} | Z = g^{ab}] - \Pr[\mathcal{A}_2 \text{ wins the game} | Z = g^r] \\ &= \Pr[\text{Game}_1^{\text{ISO-RR},\mathcal{A}_2}(\lambda) \Rightarrow \text{true}] - \Pr[\text{Game}_2^{\text{ISO-RR},\mathcal{A}_2}(\lambda) \Rightarrow \text{true}] \\ &= \mathbf{Adv}_{\text{ISO-RR},\mathcal{A}_2}^{\text{G}_1}(\lambda) - \mathbf{Adv}_{\text{ISO-RR},\mathcal{A}_2}^{\text{G}_2}(\lambda).\end{aligned}$$

Since in Game_2, adversary $\mathcal{A}_2$ has no advantage in winning the game, i.e., $\mathbf{Adv}_{\text{ISO-RR},\mathcal{A}_2}^{\text{G}_2}(k) = 0$. Combining all the above results, we have

$$\begin{aligned}\mathbf{Adv}_{\text{ISO-RR},\mathcal{A}}^{\text{rra-ake}}(\lambda) \leq n \cdot \mathbf{Adv}_{\mathcal{DS},\mathcal{S}}^{\text{uf-cma}}(\lambda) + q_I(q_I - 1)(\mathbf{Adv}_{\mathcal{B}}^{\text{DDH}}(\lambda) \\ + 2 \cdot \mathbf{Adv}_{\mathbb{F},\mathcal{D}}^{\text{rka-prf}}(\lambda)).\end{aligned}$$

□

Reset-1 Security. In the Reset-1 model, the adversary controls the randomness and is not allowed to corrupt the long-lived key of either participant in the protocol. Thus, in the ISO-RR protocol, $\widetilde{x}$ is known to the adversary. Therefore, according to the approach used to obtain the Reset-1 and Reset-2 security from a Reset-2 secure AKE protocol in [11], the pseudorandom function should be a strong extractor. However, the PRF in the ISO-RR protocol is already set to be RKA-secure. Do such kind of pseudorandom functions, which are RKA-secure and can be strong extractors as well, exist? To the best of our knowledge, we cannot affirmatively answer this.

6. RRA-Secure Authenticated Key Exchange Based on Encryption

In the full version of [11], there are also authenticated key exchange constructions on the basis of public-key encryption with message authentication code. If we modify the PKEDH-R2 and PKEDH-R protocols in [11] following the modifications to the ISO-R2 and ISO-R protocols, and require the underlying public-key encryption scheme to be secure against related randomness attacks [3], we can achieve weak corruption RRA security in both of them. To show this, below, we will take the modified PKEDH-R protocol (which is named as PKEDH-RR) as an instance.

Public-Key Encryption (PKE). A PKE scheme $\mathcal{PKE}$ is composed of three algorithms [11]: a key generation algorithm $\mathcal{PKE}.\text{SKG}(1^\lambda)$ outputting a public key pk and private key sk on inputting a security parameter, an encryption algorithm $\mathcal{PKE}.\text{Enc}(pk, m)$ outputting a ciphertext c on inputting a message m and the public key pk, and a decryption algorithm $\mathcal{PKE}.\text{Dec}(sk, c)$ outputting a failure symbol $\perp$ or a message m on inputting the private key sk and a ciphertext c. The encryption algorithm can also be denoted as $\mathcal{PKE}.\text{Enc}(pk, m; r)$, meaning that message m is encrypted under public key pk using randomness r.

If for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage function for the scheme $\mathcal{PKE}$

$$\mathbf{Adv}_{\mathcal{PKE},\mathcal{A}}^{\text{ind-cca}}(\lambda) = \Pr\left[b' = b \;\middle|\; \begin{array}{l}(pk, sk) \leftarrow \mathcal{PKE}.\text{SKG},\ b \leftarrow \{0,1\} \\ (m_0, m_1, state) \leftarrow \mathcal{A}_1^{\mathcal{PKE}.\text{Dec}(sk,\cdot)},\ |m_0| \neq |m_1| \\ C^* \leftarrow \mathcal{PKE}.\text{Enc}(pk, m_b) \\ b' \leftarrow \mathcal{A}_2^{\mathcal{PKE}.\text{Dec}(sk,\cdot)}(pk, m_0, m_1, state, C^*)\end{array}\right] - 1/2$$

is negligible in the security parameter λ and adversary $\mathcal{A}_2$ is excluded from making a decryption query on the ciphertext C^*, then the PKE scheme $\mathcal{PKE}$ is IND-CCA secure.

Message Authentication Code (MAC). An MAC scheme $\mathcal{MAC}$ with key space $\mathcal{K}$ [11] consists of a message authentication algorithm $\text{MAC}_K(m)$ outputting an authentication tag τ on inputting a message m and a key $K \in \mathcal{K}$, and a verification algorithm $\text{MAV}_K(m, \tau)$ outputting 0 or 1 on inputting a message and tag pair (m, τ) and a key $K \in \mathcal{K}$.

If for any PPT adversary $\mathcal{A}$, the advantage function for the scheme $\mathcal{MAC}$

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{MAC},\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{l|l} \text{MAV}_K(m^*, \tau^*) = 1 \wedge & K \leftarrow \mathcal{K} \\ \mathcal{A} \text{ has never queried } \text{MAC}_K(m^*) & (m^*, \tau^*) \leftarrow \mathcal{A}^{\text{MAC}_K(\cdot)(1^\lambda)} \end{array}\right]$$

is negligible in the security parameter λ, then the MAC scheme $\mathcal{MAC}$ is secure under chosen message attacks.

6.1. A PKEDH-RR Protocol

Let $\mathbb{F} = \{F_K : \{0,1\}^{\rho(\lambda)} \to Z_p^* \mid K \in Z_p^*\}$ be a related-key attack secure PRF family where $\rho(\lambda)$ is the polynomial of λ. Let $\mathcal{PKE} = (\mathcal{PKE}.\text{SKG}, \mathcal{PKE}.\text{Enc}, \mathcal{PKE}.\text{Dec})$ be a PKE scheme. Let $\mathcal{MAC} = (\text{MAC}, \text{MAV})$ be an MAC scheme. We present the PKEDH-RR protocol in Figure 7, where G is a group of prime order p, and g is a generator of G.

A		B
$a \leftarrow Z_p^*$		$b \leftarrow Z_p^*$
$(pk, sk) \leftarrow \mathcal{PKE}.\text{SKG}(1^\lambda)$		$(pk', sk') \leftarrow \mathcal{PKE}.\text{SKG}(1^\lambda)$
$pk_A \leftarrow pk, sk_A \leftarrow (sk, a)$		$pk_B \leftarrow pk', sk_B \leftarrow (sk', b)$
$\widetilde{x} \leftarrow \{0,1\}^{\rho(\lambda)}$		
$x, K_A, r \leftarrow F_{\widetilde{x}}(a \oplus \widetilde{x})$		
$X \leftarrow g^x$	X, c_A	
$c_A \leftarrow \mathcal{PKE}.\text{Enc}(pk_B, A, K_A; r)$	$\longrightarrow$	$\widetilde{y} \leftarrow \{0,1\}^{\rho(\lambda)}$
		$y, K_B, r' \leftarrow F_{\widetilde{y}}(b \oplus \widetilde{y})$
		$Y \leftarrow g^y$
		$\text{sid} \leftarrow X \| Y$
	c_B, Y, τ_B	$c_B \leftarrow \mathcal{PKE}.\text{Enc}(pk_A, B, K_B; r')$
	$\longleftarrow$	$\tau_B \leftarrow \text{MAC}_{K_A}(X, Y, B, A, 0)$
$\text{sid} \leftarrow X \| Y$		
$\tau_A \leftarrow \text{MAC}_{K_B}(Y, X, A, B, 1)$	τ_A	
$ssk \leftarrow g^{xy}$	$\longrightarrow$	
		$ssk \leftarrow g^{xy}$

Figure 7. The PKEDH-RR protocol between two participants A and B. Related decryption and verification algorithms in the protocol are omitted due to the brevity consideration.

Similarly, the PKEDH-RR protocol is similar to Yang's PKEDH-R protocol with the difference that the pseudorandom function F is related-key attack secure, and it takes the randomness rather than the long-lived key as the key. The reason that Yang's PKEDH-R protocol fails to prevent related randomness attacks is that when the long-lived key and the underlying relationship of the random coins are known to the adversary, it is hard to assume that the outputs of the strong randomness extractor SRE function [27] under different random coins are still independent of each other. Our modification is to prevent the related randomness attack described in Section 4, as now, due to the RKA security, the outputs of the pseudorandom function become unknown to the adversary.

6.2. Security Proof

Theorem 3. *The PKEDH-RR protocol is secure in the weak corruption RRA-AKE model for a* Φ*-restricted adversary if* $\mathcal{PKE}$ *is an IND-CCA secure PKE scheme,* $\mathcal{MAC}$ *is an MAC scheme secure*

under adaptive chosen message attacks, $\mathbb{F}$ is a Φ-restricted related-key attack secure pseudorandom function family, and DDH assumption holds in the underlying group.

Proof. Similar to the proof of Theorem 2, there must be a partner instance (V^*, j^*) corresponding to an instance (U^*, i^*) output by an adversary in the Test query, or the security of the public-key encryption scheme $\mathcal{PKE}$ or the message authentication code $\mathcal{MAC}$ could be broken. □

Following the proof in [11], we define an encryption forger $\mathcal{B}$. Let $(pk, sk) \leftarrow \mathcal{PKE}.\text{SKG}(1^\lambda)$, and $c^* \leftarrow \mathcal{PKE}.\text{Enc}(pk, S, N^*)$, where S is a randomly selected string by forger $\mathcal{B}$, and N^* is a randomly selected key from $\mathcal{MAC}$'s key space (unknown to forger $\mathcal{B}$). Given pk, c^*, and access to an oracle $\mathcal{O}_{sk}(\cdot)$ decrypting ciphertexts unequal to c^*, and an oracle $\mathcal{O}_{N^*}(\cdot)$ returning $\text{MAC}_{N^*}(m)$ on an input m, the goal of forger $\mathcal{B}$ is to output m^*, $\text{MAC}_{N^*}(m^*)$ with forger $\mathcal{B}$ never querying to the oracle $\mathcal{O}_{N^*}(\cdot)$ on m^*.

We denote E by the event that the instance (U^*, i^*) in the Test query created by adversary $\mathcal{A}$ has no partner instance. We can build an encryption forger $\mathcal{B}$ in the RRA-AKE game if the event E occurs.

We denote q_I as the maximum number of NewInstance queries sent by adversary $\mathcal{A}$. Forger $\mathcal{B}$ randomly chooses two entities U^*, V^* from $\mathcal{U}$ ($|\mathcal{U}| = n$), and creates all long-lived keys for other entities in $\mathcal{U} \setminus \{V^*\}$. Forger $\mathcal{B}$ then chooses an integer $l \leftarrow [1, q_I]$, and requests the challenger to return the challenge $c^* = \mathcal{PKE}.\text{Enc}(pk, U^*, N^*)$ on input U^* under pk, and then it sets $pk_{V^*} = pk$, and simulates the RRA-AKE game with adversary $\mathcal{A}$ with an exception in cases below.

- Forger $\mathcal{B}$ halts if adversary $\mathcal{A}$ does not make a Test query with an instance of U^*.
- Forger $\mathcal{B}$ halts if $\text{pid}^{i^*}_{U^*} \neq V^*$.
- Forger $\mathcal{B}$ halts if (U^*, i^*) is not the l-th instance.
- Forger $\mathcal{B}$ halts if adversary $\mathcal{A}$ makes a corrupt query with input V^*.
- Forger $\mathcal{B}$ sets $c_{U^*} = c^*$ in the l-th instance, generates the ephemeral DH public and private key pair for (U^*, i^*), utilises sk_{U^*} to obtain $N \leftarrow \mathcal{PKE}.\text{Dec}(sk_{U^*}, c_{V^*})$, and honestly yields the tag τ_{U^*} with N.
- If adversary $\mathcal{A}$ forwards a message $(c, \ldots)$ to V^* with $c = c^*$, forger $\mathcal{B}$ queries c to its decryption oracle $\mathcal{O}_{sk}(\cdot)$, and proceeds as usual after receiving from $\mathcal{O}_{N^*}(\cdot)$.
- If adversary $\mathcal{A}$ forwards a message $(c^*, \ldots)$ to V^*, forger $\mathcal{B}$ queries to its oracle $\mathcal{O}_{N^*}$ to obtain the response tag.
- If adversary $\mathcal{A}$ forwards the MAC tag to the l-th instance, forger $\mathcal{B}$ outputs its forgery with the message and MAC tag pair and halts.

Therefore, we can conclude that

$$\epsilon = \Pr[\mathcal{B} \text{ succeeds}] \geq \frac{1}{n(n-1)q_I} \Pr[\text{E}].$$

Given an encryption forger $\mathcal{B}$, we can build another adversary $\mathcal{D}$ against the PKE scheme in the IND-CCA security game. Adversary $\mathcal{D}$ is given a public key pk and can access both encryption and decryption oracles. When forger $\mathcal{B}$ requests a challenge on the input S, adversary $\mathcal{D}$ randomly chooses numbers N_0 and N_1, and requests its challenger with inputs $S||N_0$ and $S||N_1$. Adversary $\mathcal{D}$ sets pk, c^* as forger $\mathcal{B}$'s challenge after obtaining the challenge c^*. When forger $\mathcal{B}$ queries to the encryption oracle on a ciphertext $c \neq c^*$, adversary $\mathcal{D}$ queries to the decryption oracle on the input c to its challenger. When forger $\mathcal{B}$ queries to the MAC oracle on a message m, adversary $\mathcal{D}$ responds to forger $\mathcal{B}$ $\text{MAC}_{N_0}(m)$. Lastly, adversary $\mathcal{D}$ outputs 0 if forger $\mathcal{B}$ makes a successful forgery $\text{MAC}_{N_0}(m^*)$, meaning that c^* is a ciphertext for $S||N_0$. Otherwise, adversary $\mathcal{D}$ outputs 0 if forger $\mathcal{B}$'s forgery fails, meaning that c^* is a ciphertext for $S||N_1$. Hence, we can conclude that

$$\begin{aligned}\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{PKE},\mathcal{D}}(\lambda) &= \Pr[\mathcal{D} \text{ outputs } 0|b=0]\Pr[b=0]+ \\ &\quad \Pr[\mathcal{D} \text{ outputs } 1|b=1]\Pr[b=1] - \frac{1}{2} \\ &= \frac{1}{2}\Pr[\mathcal{B} \text{ succeeds}|b=0] + \frac{1}{2}(1 - \Pr[\mathcal{B} \text{ succeeds}|b=1]) - \frac{1}{2} \\ &= \frac{1}{2}(\Pr[\mathcal{B} \text{ succeeds}|b=0] - \Pr[\mathcal{B} \text{ succeeds}|b=1]) \\ &= \frac{1}{2}(\epsilon - \mathbf{Adv}^{\text{cma}}_{\mathcal{MAC},\mathcal{B}}(\lambda)),\end{aligned}$$

where the last line is summarised from the fact that forger $\mathcal{B}$ is in the encryption forger game when $b = 0$, and forger $\mathcal{B}$ is in the chosen message attack game when $b = 1$, c^* is independent of N_0.

Integrating all previous results, we can conclude that

$$\Pr[\mathrm{E}] \leq n(n-1)q_I(2 \cdot \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{PKE},\mathcal{D}}(k) + \mathbf{Adv}^{\text{cma}}_{\mathcal{MAC},\mathcal{B}}(\lambda)).$$

The rest of the proof is similar to that in Theorem 2, so we omit the details.

Reset-1 Security. The method used to construct a PKEDH-RR protocol that is RRA-secure is similar to that used in the ISO-RR protocol. Therefore, for the same reason, we are not sure whether the PKEDH-RR protocol can be extended to cover the Reset-1 model defined in [11].

7. Conclusions

Several recent incidents caused by the various kinds of randomness failures make the research community begin to find methods hedging cryptographic primitives against such failures. In this paper, we focus on a special attack, called related randomness attack (RRA), executed on the randomness used in authenticated key exchange, where an adversary is able to force the participants of an authenticated key exchange scheme to reuse the random values and the functions of these values. We start from formalising the RRA security model for an authenticated key exchange protocol. Following the RRA security model of public-key encryption and the randomness resetting security model of authenticated key exchange, we present our model of RRA security for authenticated key exchange. After pointing out the related randomness attacks on the authenticated key exchange protocols in [11], we propose several constructions of authenticated key exchange under related randomness attacks.

Author Contributions: Methodology, H.C.; Formal analysis, G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number 62072369.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Feltz, M.; Cremers, C. Strengthening the security of authenticated key exchange against bad randomness. *Des. Codes Cryptogr.* **2018**, *86*, 481–516. [CrossRef]
2. Cui, H.; Qin, B.; Susilo, W.; Nepal, S. Robust digital signature revisited. *Theor. Comput. Sci.* **2020**, *844*, 87–96. [CrossRef]
3. Paterson, K.G.; Schuldt, J.C.N.; Sibborn, D.L. Related Randomness Attacks for Public Key Encryption. In *Public Key Cryptography*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8383, pp. 465–482.
4. Heninger, N.; Durumeric, Z.; Wustrow, E.; Halderman, J.A. Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices. In Proceedings of the USENIX Security Symposium, Bellevue, WA, USA, 8–10 August 2012; Volume 2012, pp. 205–220.

5. Lenstra, A.K.; Hughes, J.P.; Augier, M.; Bos, J.W.; Kleinjung, T.; Wachter, C. Public Keys. In Proceedings of the CRYPTO, Santa Barbara, CA, USA, 19–23 August 2012; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7417, pp. 626–642.
6. Cui, H.; Mu, Y.; Au, M.H. Relations between robustness and RKA security under public-key encryption. *Theor. Comput. Sci.* **2016**, *628*, 78–91. [CrossRef]
7. Ristenpart, T.; Yilek, S. When Good Randomness Goes Bad: Virtual Machine Reset Vulnerabilities and Hedging Deployed Cryptography. In Proceedings of the NDSS, San Diego, CA, USA, 28 February–3 March 2010; Volume 2010.
8. Yilek, S. Resettable Public-Key Encryption: How to Encrypt on a Virtual Machine. In Proceedings of the CT-RSA, San Francisco, CA, USA, 1–5 March 2010; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 5985, pp. 41–56.
9. Yuen, T.H.; Zhang, C.; Chow, S.S.M.; Yiu, S. Related Randomness Attacks for Public Key Cryptosystems. In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '15, Singapore, 14–17 April 2015; Bao, F., Miller, S., Zhou, J., Ahn, G., Eds.; ACM: New York, NY, USA, 2015; pp. 215–223. [CrossRef]
10. Blackman, D.; Vigna, S. Scrambled Linear Pseudorandom Number Generators. *ACM Trans. Math. Softw.* **2021**, *47*, 36:1–36:32. [CrossRef]
11. Yang, G.; Duan, S.; Wong, D.S.; Tan, C.H.; Wang, H. Authenticated Key Exchange under Bad Randomness. In Proceedings of the Financial Cryptography and Data Security-15th International Conference, FC 2011, Gros Islet, Saint Lucia, 28 February–4 March 2011; Revised Selected Papers; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7035, pp. 113–126.
12. Bellare, M.; Kohno, T. A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications. In Proceedings of the EUROCRYPT; Lecture Notes in Computer Science, Warsaw, Poland, 4–8 May 2003; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2656, , pp. 491–506.
13. Bellare, M.; Rogaway, P. Entity Authentication and Key Distribution. In Proceedings of the Advances in Cryptology-CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, CA, USA, 22–26 August 1993; Springer: Berlin/Heidelberg, Germany, 1993; Volume 773, pp. 232–249.
14. Canetti, R.; Krawczyk, H. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In Proceedings of the Advances in Cryptology-EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, 6–10 May 2001; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2045, pp. 453–474.
15. Becerra, J.; Ostrev, D.; Skrobot, M. Forward Secrecy of SPAKE2. In Proceedings of the Provable Security-12th International Conference, ProvSec 2018, Jeju, Republic of Korea, 25–28 October 2018; Baek, J., Susilo, W., Kim, J., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11192, pp. 366–384. [CrossRef]
16. Canetti, R.; Krawczyk, H. Security Analysis of IKE's Signature-Based Key-Exchange Protocol. In Proceedings of the Advances in Cryptology-CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2002; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2442, pp. 143–161.
17. Krawczyk, H. SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols. In Proceedings of the Advances in Cryptology-CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 2003; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2729, pp. 400–425.
18. Krawczyk, H. HMQV: A High-Performance Secure Diffie-Hellman Protocol. In Proceedings of the Advances in Cryptology-CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2005; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3621, pp. 546–566.
19. LaMacchia, B.A.; Lauter, K.E.; Mityagin, A. Stronger Security of Authenticated Key Exchange. In Proceedings of the Provable Security, First International Conference, ProvSec 2007, Wollongong, Australia, 1–2 November 2007; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4784, pp. 1–16.
20. Boyd, C.; Cliff, Y.; Nieto, J.M.G.; Paterson, K.G. Efficient One-Round Key Exchange in the Standard Model. In Proceedings of the Information Security and Privacy, 13th Australasian Conference, ACISP 2008, Wollongong, Australia, 7–9 July 2008; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5107, pp. 69–83.
21. Okamoto, T. Authenticated Key Exchange and Key Encapsulation in the Standard Model. In Proceedings of the Advances in Cryptology-ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, 2–6 December 2007; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4833, pp. 474–484.
22. Rogaway, P. Nonce-Based Symmetric Encryption. In Proceedings of the FSE, Delhi, India, 5–7 February 2004; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3017, pp. 348–359.
23. Rogaway, P.; Shrimpton, T. A Provable-Security Treatment of the Key-Wrap Problem. In Proceedings of the EUROCRYPT, St. Petersburg, Russia, 28 May–1 June 2006; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4004, pp. 373–390.
24. Kamara, S.; Katz, J. How to Encrypt with a Malicious Random Number Generator. In Proceedings of the FSE, Lausanne, Switzerland, 10–13 February 2008; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5086, pp. 303–315.
25. Bellare, M.; Brakerski, Z.; Naor, M.; Ristenpart, T.; Segev, G.; Shacham, H.; Yilek, S. Hedged Public-Key Encryption: How to Protect against Bad Randomness. In Proceedings of the ASIACRYPT, Tokyo, Japan, 6–10 December 2009; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5912, pp. 232–249.

26. Aiello, W.; Bellovin, S.M.; Blaze, M.; Canetti, R.; Ioannidis, J.; Keromytis, A.D.; Reingold, O. Just fast keying: Key agreement in a hostile internet. *ACM Trans. Inf. Syst. Secur.* **2004**, *7*, 242–273. [CrossRef]
27. Feng, H.; Tang, Q. Computational Robust (Fuzzy) Extractors for CRS-Dependent Sources with Minimal Min-entropy. In Proceedings of the Theory of Cryptography-19th International Conference, TCC 2021, Raleigh, NC, USA, 8–11 November 2021; Part II; Nissim, K.; Waters, B., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2021; Volume 13043, pp. 689–717. [CrossRef]