Research Article

Jiageng Chen, Keita Emura and Atsuko Miyaji

SKENO: Secret key encryption with non-interactive opening

Abstract: In this paper, we introduce the notion of *secret key encryption with non-interactive opening* (SKENO). With SKENO, one can make a non-interactive proof π to show that the decryption result of a ciphertext C under a shared secret key K is indeed plaintext M without revealing K itself. SKENO is the secret key analogue of *public key encryption with non-interactive opening* (PKENO). We give a generic construction of SKENO from verifiable random function (VRF) with certain stronger uniqueness, for example, the Hohenberger–Waters VRF and the Berbain–Gilbert IV-dependent stream cipher construction. Although the strong primitive VRF is used, by taking advantage of the features of the stream cipher, we can still achieve good performance without sacrificing much of the efficiency. Though our VRF-based SKENO construction does not require random oracles, we show that SKENO can be constructed from weak VRF (which is strictly weaker primitive than VRF) in the random oracle model.

Keywords: Public/secret key encryption with non-interactive opening, verifiable random function, IV-dependent stream cipher

MSC 2010: 94A60

Jiageng Chen, Atsuko Miyaji: School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan, e-mail: jg-chen@jaist.ac.jp, miyaji@jaist.ac.jp
Keita Emura: Network Security Research Institute, Security Fundamentals Laboratory, National Institute of Information and Communications Technology (NICT), 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan, e-mail: k-emura@nict.go.jp

Communicated by: Ed Dawson

1 Introduction

1.1 Public key encryption with non-interactive opening (PKENO)

Let us first consider the following scenario. Assume that a sender Alice computes a ciphertext C of a message M by using a receiver Bob's public key pk_B. In order to solve the dispute in some circumstances, Bob would like to prove the correctness of the corresponding plaintext-ciphertext pair (C, M), for example, when the information M sent to Bob is not the expected one. The easiest way to prove this is that Bob opens his secret key sk_B, demonstrates the decryption algorithm and shows that the decryption result of C under sk_B is M. However, in this naive way, no other ciphertext encrypted under pk_B remains secure since sk_B has been opened and anyone can decrypt any ciphertext encrypted under pk_B. To securely capture this situation, Damgård, Hofheinz, Kiltz and Thorbek [17] have proposed the notion of *public key encryption with non-interactive opening* (PKENO), where Bob can make a non-interactive proof π which proves that the decryption result of C under sk_B is M without revealing sk_B itself. As a concrete case where PKENO is required, Damgård, Hofheinz, Kiltz and Thorbek [17] mentioned the following scenario: *"Player A sends a secret message to player B who (perhaps at some later time) checks what he receives against some public information. For instance, it may be that the message is supposed to be information for opening a commitment that A established earlier. If the check is OK, B will be able to proceed, but otherwise some 'exception handling' must be done. The standard solution to this is to have B broadcast a complaint and A must then broadcast what he claims to have sent privately, allowing all players to check the information."*

A generic construction of PKENO from identity-based encryption (IBE) has been shown in [10] and also from strongly existentially unforgeable one-time signature (OTS) in [5, 27]. Regarding other generic constructions of PKENO, it is known that PKENO can be constructed from group signature [21] and can be constructed

from enhanced chosen-ciphertext secure public key encryption [16]. Concrete constructions of PKENO have also been proposed in [24, 25, 31].

The main idea behind the PKENO construction is to make a decryption key which only works for the corresponding ciphertext, where the decryption key is set as a proof. For the sake of readability, we introduce the generic construction of PKENO (based on IBE and OTS) proposed by Damgård, Hofheinz, Kiltz and Thorbek [17] as follows: Bob runs $(pk_B, sk_B) \leftarrow \mathsf{IBE.KeyGen}(1^k)$, that is, sk_B is a master key of the underlying IBE scheme (with a soundness property where it is efficiently verifiable if a given user secret key $usk[ID]$ was properly generated for the identity ID). Alice runs $(vk, sk) \leftarrow \mathsf{OTS.KeyGen}(1^k)$ and computes $C \leftarrow \mathsf{IBE.Enc}(pk_B, vk, M)$, that is, a verification key vk is regarded as the identity of the underlying IBE scheme, and computes $\sigma \leftarrow \mathsf{OTS.Sign}(sk, C)$. The PKENO ciphertext is (C, σ, vk). Bob can decrypt (C, σ, vk) such that Bob verifies that $1 = \mathsf{OTS.Verify}(vk, \sigma, C)$ and computes $usk[vk] \leftarrow \mathsf{IBE.Extract}(sk_B, vk)$ and $M \leftarrow \mathsf{IBE.Dec}(usk[vk], C)$. Then, $usk[vk]$ can be set as π, since anyone can prove whether $M = \mathsf{IBE.Dec}(\pi, C)$ or not. Other ciphertexts encrypted by pk_B remain secure assuming that a different vk is chosen in each encryption (this is a reasonable assumption of OTS).

1.2 PKENO with the KEM/DEM framework and its limitations

A PKENO scheme for the KEM/DEM (key encapsulation/decapsulation mechanism) framework has also been proposed by Galindo [24]. To encrypt M, Alice computes $(K, C_1) \leftarrow \mathsf{Encapsulation}(pk_B)$ and $C_2 \leftarrow \mathsf{DEM}(K, M)$, where DEM is a symmetric cipher, and sends $C = (C_1, C_2)$ to Bob. Bob can compute $K \leftarrow \mathsf{Decapsulation}(sk_B, C_1)$ and $M \leftarrow \mathsf{DEM}(K, C_2)$. In Galindo's scheme, Bob can make a non-interactive proof π which proves that the decapsulation result of C_1 under sk_B is K without revealing sk_B itself. The construction methodology of the Galindo PKENO scheme with the KEM/DEM framework is also the same as that of the previous PKENO scheme, where for a proof π anyone can compute $K' \leftarrow \mathsf{Decapsulation}(\pi, C_1)$ and check whether $K = K'$ or not. That is, no verifier can verify π without knowing K. This setting is acceptable if K is an ephemeral value that changes per session (for example, in the KEM/DEM usage). However, in some other applications, where the secret key K is negotiated beforehand among a group of people and it is then used by symmetric cryptosystems to do encryption for a relative period of time,[1] the Galindo scheme does not work since the opening of K will expose unrelated messages as well.

Here, we clarify the situation where a ciphertext is encrypted by a shared key as follows:

- *Let K be a shared key of Alice and Bob, and suppose that Charlie does not have K.*
- *Alice sends C to Bob, where C is a ciphertext of a message M under the key K.*
- *For Charlie, Bob wants to prove that the decryption result of C is M* without revealing K.

We note that the scenario of Damgård, Hofheinz, Kiltz and Thorbek can be considered even in the above setting, where player A and player B share a key. However, PKENO is not capable of handling the above situation even when a PKENO scheme with the KEM/DEM framework has been proposed. Note that the proof itself is the decryption key in the previous PKENO schemes. Since opening the shared key K is not an option anymore, K should not be recognized as a proof. Thus, we need to investigate a new methodology to handle the situation where a ciphertext is encrypted by a shared key which cannot be revealed.

One may think that authenticated encryption, where it is computationally infeasible to produce a ciphertext which is not previously produced by an encryptor, is enough to achieve our goal. More precisely, in the definition of INT-CTXT (integrity of ciphertexts) [4, 8], for a key K, an adversary $\mathcal{A}$ wins if $\mathcal{A}$ (who is given the challenge ciphertext C^*, which is computed by K, and is allowed to issue the decryption query $C \neq C^*$) can produce a ciphertext C', where the decryption result of C' under K is not $\perp$ (that is, C' is a valid ciphertext). That is, by using an INT-CTXT secure DEM, anyone *who has a shared key K* can prove that a ciphertext C is computed by exactly K by executing the decryption algorithm. Moreover, we can achieve INT-CTXT security

1 This is especially the case for an IV-dependent stream cipher, where K has remained unchanged for many sessions while IV is changed for each session and sent in plaintext form for synchronization.

by using a classical "encrypt-then-MAC" methodology (see [4, Theorem 5]) which is also efficient. However, this observation is wrong, since it is hard to satisfy the above scenario, that is, for anyone *who does not have* K, Bob wants to prove that the decryption result of C is M without revealing K by using authenticated encryption only.

1.3 Our contribution

In this paper, we define the notion of *secret key encryption with non-interactive opening* (SKENO), where Bob can make a non-interactive proof π which proves that the decryption result of C under K is M without revealing K itself and we give a generic construction of SKENO from a verifiable random function (VRF) [1, 2, 18, 19, 29, 33, 35] and the Berbain–Gilbert IV-dependent stream cipher construction [6], where IV is an initial vector.

In the Berbain–Gilbert construction, pseudo-random function (PRF) [26] is regarded as a key scheduling algorithm (KSA) and a pseudo-random number generator (PRNG) [9, 42] is regarded as a pseudo-random generation algorithm (PRGA). From an n-bit initial vector IV and a k-bit secret key K, a PRF (say $F_K(IV)$) outputs an m-bit initial state (say y) which is used as an input of a PRNG. Finally, the PRNG (say $G(y)$) outputs an L-bit keystream Z_{IV}. A ciphertext C is $Z_{IV} \oplus M$, where $\oplus$ is the exclusive-or operation. Berbain and Gilbert give a composition theorem, where the composition $G \circ F_K$ is also a PRF such that $G \circ F : \{0,1\}^n \to \{0,1\}^L$. Therefore, $G \circ F$ can be a secure stream cipher, where no adversary $\mathcal{A}$ can distinguish whether $G(F_K(IV))$ is a truly random number or not, even if $\mathcal{A}$ can select IV.

A VRF is a PRF that provides a non-interactively verifiable proof: Given an input value x and its output $y = F_{SK}(x)$, a proof $\pi(x)$ proves that y is the output of the function F indexed by SK given the input x without revealing the secret key SK itself. Several applications of VRFs have been considered, for example, non-interactive lottery systems used in micropayment schemes [37], resettable zero-knowledge proofs [36], updatable zero knowledge databases [32], set-intersection protocols [28], compact e-cash [3, 13], adaptive oblivious transfer protocols [30], keyword search [23] and so on. We make it clear that, to the best of our knowledge, the usage of a VRF for the SKENO functionality has not appeared in the literature.

We set IV as an input of the VRF (as in the Berbain–Gilbert construction) and set a shared key K as a secret key of the VRF. The VRF leads to an m-bit initial state, which is the input to the PRNG for generating an L-bit keystream Z_{IV}. A ciphertext C is $Z_{IV} \oplus M$. We set $\pi := (\pi' = \Pi_{SK}(x), y = F_K(IV))$ a proof of (IV, C, M), where $K := SK$. Due to the VRF functionality, one can prove that the m-bit initial state is the result of the underlying VRF without revealing K. Moreover, to prevent a chosen-ciphertext attack, we apply message authentication code (MAC), as in the conversion from CPA-secure DEM to CCA-secure DEM [40].

Although the VRF primitive is a relatively expensive assumption, our construction can be considered to be efficient in environments where the key and IV setup phase is not executed frequently, since the execution of the PRNG will play a dominant role in the encryption and decryption process.

We remark that although PKENO implies SKENO (for a pair (PK, SK) of a PKENO scheme we can set $SK = K$), this PKENO-based construction is too strong to construct SKENO. Especially, anyone can compute a ciphertext (as in PKE) and SK is not used for encryption. Again, our goal is to make a non-interactive proof for C which is made by K (as in SKE). Therefore, we do not use this PKENO-based methodology to construct SKENO.

We also have to clarify that we do not propose the SKENO in order to replace PKENO. We just provide another option in the environment where the symmetric key setting is used and efficiency is considered somehow crucial. Since the core part of the encryption scheme is the stream cipher, which is very efficient over any other encryption scheme and most importantly when large amounts of stream data need to be encrypted, our scheme shows a speed advantage over other schemes because in this situation users do not need to change the key so frequently under the premise that the encryption is done under a continuously long keystream generated by the stream cipher. If the WEP protocol implemented RC4 in this way, it would not be broken so badly. So, if we consider an environment where large amounts of video stream data need to be encrypted and supposing that the secret key in SKENO is somehow negotiated beforehand, then our proposed SKENO

is obviously faster than PKENO. Notice that once the key is agreed, we can manage to change the IV instead of the key to achieve a new session key, which will reduce the cost of the key negotiation part. Still someone may question the key negotiation cost itself. It is hard for us to draw a concrete conclusion about this, but what we can say is that for a relatively restricted environment we believe that SKENO provides another very comparable solution to the underlining problem.

As a relationship between VRF and KEM, Abdalla, Catalano and Fiore [1, 2] showed that VRF can be constructed from (VRF-suitable) identity-based key encapsulation mechanism (IB-KEM). By utilizing our result, we can show that SKENO can be constructed from a VRF-suitable IB-KEM. This relation seems to be of independent interest.

Concluding, in comparison to the previous version [15] of the present work, in this full version we give the detailed proof of the security of our SKENO construction under indistinguishability under chosen-ciphertext and prove attacks (IND-CCPA), since in [15] we only gave the construction. Moreover, we discuss whether there is room for constructing SKENO from weaker cryptographic building blocks than VRF, for example, from weak VRF (wVRF) [12]. In Section 6, we show that SKENO can be constructed from wVRF in the random oracle model and discuss the difficulty of wVRF-based SKENO construction in the standard model.

2 Preliminaries

In this section, we define PRNG and VRF. We denote as State the state information transmitted by the adversary to himself across stages of the attack in experiments.

2.1 Pseudo-random number generator (PRNG)

A pseudo-random number generator (PRNG) $G : \{0,1\}^m \to \{0,1\}^L$ is used to expand an m-bit secret seed into an L-bit sequence, $m < L$.

Definition 2.1 (Pseudo-randomness of PRNG). We say that a function $G : \{0,1\}^m \to \{0,1\}^L$ is a PRNG if for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, the following advantage is a negligible function of the security parameter λ:

$$\mathsf{Adv}^{\mathrm{Pseudo}}_{G,\mathcal{A}}(1^\lambda) := \left|\Pr[y^* \xleftarrow{\$} \{0,1\}^m;\ b \xleftarrow{\$} \{0,1\};\ Z_0^* \leftarrow G(y^*);\ Z_1^* \xleftarrow{\$} \{0,1\}^L;\ b' \leftarrow \mathcal{A}(1^\lambda, Z_b^*);\ b = b'] - \frac{1}{2}\right|.$$

2.2 Verifiable random function (VRF)

Next, we define VRF by referring to the Hohenberger–Waters VRF definition [29] as follows: Let $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ be an efficient computable function, where $k = k(\lambda)$, $n = n(\lambda)$ and $m = m(\lambda)$ are polynomials in the security parameter 1^λ. For all $SK \in \{0,1\}^k$, we simply denote $F_{SK} : \{0,1\}^n \to \{0,1\}^m$.

Definition 2.2 ([29]). We say that F is a VRF if there exist algorithms (VRF.Setup, VRF.Prove, VRF.Verify) such that:

- VRF.Setup(1^λ) outputs a pair of keys (PK, SK),
- VRF.Prove(SK, x) outputs a pair (y, π'), where $y = F_{SK}(x)$ is the function value of $x \in \{0,1\}^n$ and $\pi' = \Pi_{SK}(x)$ is the proof of correctness,
- VRF.Verify(PK, x, y, π') outputs 1 if $y = F_{SK}(x)$ and 0 otherwise.

In addition, three security notions are required:

- Provability: This guarantees that an honestly generated proof is always accepted by the Verify algorithm, that is, for all $(PK, SK) \leftarrow$ VRF.Setup(1^λ) and $x \in \{0,1\}^n$, if $(y, \pi') \leftarrow$ VRF.Prove(SK, x), then VRF.Verify$(PK, x, y, \pi') = 1$.

- Uniqueness: This guarantees that no proof is accepted for different values $y_1 \neq y_2$ and a common x, that is, for all $(PK, SK) \leftarrow \mathsf{VRF.Setup}(1^\lambda)$ and $x \in \{0,1\}^n$, there does not exist a tuple $(y_1, y_2, \pi_1', \pi_2')$ such that $y_1 \neq y_2$ and $\mathsf{VRF.Verify}(PK, x, y_1, \pi_1') = \mathsf{VRF.Verify}(PK, x, y_2, \pi_2') = 1$. Note that no adversary can break uniqueness, even if SK is opened, since such a tuple does not exist. This property is used for realizing the proof soundness property (which is defined in Section 3).
- Pseudo-randomness: This guarantees that no adversary can distinguish whether an output of F is truly random or not, that is, for all PPT adversaries $\mathcal{A}$ the following advantage is a negligible function of the security parameter λ:

$$\mathsf{Adv}^{\mathrm{Pseudo}}_{F,\mathcal{A}}(1^\lambda) := \Big|\Pr\big[(PK, SK) \leftarrow \mathsf{VRF.Setup}(1^\lambda);\ (x^*, \mathrm{State}) \leftarrow \mathcal{A}^{\mathsf{VRF.Prove}(SK,\cdot)}(1^\lambda, PK);$$
$$b \xleftarrow{\$} \{0,1\};\ y_0^* \leftarrow F_{SK}(x^*);\ y_1^* \xleftarrow{\$} \{0,1\}^m;\ b' \leftarrow \mathcal{A}^{\mathsf{VRF.Prove}(SK,\cdot)}(y_b^*, \mathrm{State});\ b = b'\big] - \frac{1}{2}\Big|,$$

  where $\mathsf{VRF.Prove}(SK, \cdot)$ is the prove oracle that takes as input $x \in \{0,1\}^n$, $x \neq x^*$, and outputs $(F_{SK}(x), \pi')$.

We remark that as a similar cryptographic primitive of VRF, verifiable pseudo-random bit generators (VPRGs) have been introduced by Dwork and Naor [20], where the holder of the seed can generate proofs of consistency for some parts of the sequence without hurting the unpredictability of the remaining bits. Note that in our SKENO construction we do not have to use VPRG (instead of PRNG) by assuming that the underlying PRNG satisfies the soundness property, where no adversary can find $(y_1, y_2) \in \{0,1\}^m \times \{0,1\}^m$ such that $G(y_1) = G(y_2)$ and $y_1 \neq y_2$. The advantage $\mathsf{Adv}^{\mathrm{Sound}}_{\mathrm{PRNG},\mathcal{A}}(1^\lambda)$ is simply defined as the probability that an adversary $\mathcal{A}$ produces such y_1 and y_2. Note that this requirement is natural, for example, Bertoni, Daemen, Peeters and Assche [7] have mentioned that '*loading different seeds into the PRNG shall result in different output sequences. In this respect, a PRNG is similar to a cryptographic hash function that should be collision-resistant*".

2.3 Message authentication code (MAC)

Here, we introduce strongly existentially unforgeable against a one-time chosen-message attack (one-time sEU-CMA) MAC [11, 34]. A MAC consists of two algorithms, MAC and Vrfy. The MAC algorithm takes as inputs a secret key $K \in \{0,1\}^{L_2}$ and a message $M \in \{0,1\}^{L_1}$, $L_1, L_2 \in \mathbb{N}$, and outputs a tag $t \leftarrow \mathsf{MAC}_K(M)$. The Vrfy algorithm takes as inputs K and t and outputs 0 (reject) or 1 (accept). We require the following correctness property that for all K and M we have $\mathsf{Vrfy}_K(M, \mathsf{MAC}_K(M)) = 1$. As in [11], we assume that MAC and Vrfy are deterministic.

Definition 2.3 (One-time sEU-CMA). We say that a MAC scheme Π is one-time sEU-CMA secure if for all PPT adversaries $\mathcal{A}$ the following advantage is negligible in the security parameter:

$$\mathsf{Adv}^{\text{ot-sEU-CMA}}_{\Pi,\mathcal{A}}(1^\lambda) := \Pr\big[K \xleftarrow{\$} \{0,1\}^{L_2};\ (M^*, \mathrm{State}) \leftarrow \mathcal{A}(1^\lambda);\ t^* \leftarrow \mathsf{MAC}_K(M^*);$$
$$(M, t) \leftarrow \mathcal{A}(t^*, \mathrm{State});\ (M, t) \neq (M^*, t^*);\ \mathsf{Vrfy}_K(M, t) = 1\big].$$

We additionally require that for all K and M we have $\mathsf{Vrfy}_{K'}(M, \mathsf{MAC}_K(M)) = 0$, where $K \neq K'$, with overwhelming probability. This naturally holds, for example, a PRF-based MAC construction [38], where a tag is computed by $t \leftarrow \mathsf{PRF}_K(M)$ and (t, M) is verified as $t \stackrel{?}{=} \mathsf{PRF}_K(M)$. Since the output of a PRF can be regarded as random, the probability that $t = \mathsf{PRF}_{K'}(M)$ holds is estimated as $1/|\mathcal{R}|$, where $\mathcal{R}$ is a range of the underlying PRF, and is negligible.[2]

2 Ran Canetti [14] stroke a note of warning against digital signature: "Is it OK for a signature scheme to allow an adversary, given a verification key K_v, a message M, and a signature σ, to come up with a verification public key $K_v' \neq K_v$ such that σ is a valid signature for M with respect to K_v'?" Our requirement is the MAC case of his investigation.

3 Secret key encryption with non-interactive opening (SKENO)

In this section, we give the definition of (IV-dependent) SKENO. We assume that each IV is randomly chosen for each encryption. A SKENO consists of five algorithms: KeyGen, Enc, Dec, Prove and Verify.

Definition 3.1 (System operation of SKENO).
- $\mathsf{KeyGen}(1^\lambda)$: This algorithm takes as inputs a security parameter $\lambda \in \mathbb{N}$ and returns a public verification key VK and a secret key K.
- $\mathsf{Enc}(K, IV, M)$: This algorithm takes as inputs K, an initial vector $IV \in \{0,1\}^n$ and a message $M \in \{0,1\}^L$ and returns a ciphertext C. We assume that IV is also sent to a decryptor.
- $\mathsf{Dec}(K, IV, C)$: This algorithm takes as inputs K and C and returns M or $\perp$.
- $\mathsf{Prove}(K, IV, C)$: This algorithm takes as inputs K, IV and C and returns a proof π.
- $\mathsf{Verify}(VK, IV, C, M, \pi)$: This algorithm takes as inputs VK, IV, C, M and π and returns 1 if C is the ciphertext of M and IV under K and 0 otherwise.

We require correctness and completeness as follows:

Definition 3.2 (Correctness). For all $(VK, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $IV \in \{0,1\}^n$ and $M \in \{0,1\}^L$, there holds $\mathsf{Dec}(K, IV, \mathsf{Enc}(K, IV, M)) = M$.

Definition 3.3 (Completeness). For all $(VK, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $IV \in \{0,1\}^n$ and for any ciphertext C, we have that for $M \leftarrow \mathsf{Dec}(K, IV, C)$ there holds $\mathsf{Verify}(VK, IV, C, M, \mathsf{Prove}(K, IV, C)) = 1$ with overwhelming probability (note that M may be $\perp$).

Next, we define security notions of SKENO, called indistinguishability under chosen-ciphertext and prove attacks (IND-CCPA) and proof soundness.

Definition 3.4 (IND-CCPA). We say that a SKENO scheme Π is IND-CCPA secure if for all PPT adversaries $\mathcal{A}$ the following advantage is negligible in the security parameter:

$$\mathsf{Adv}^{\text{Ind-ccpa}}_{\Pi,\mathcal{A}}(1^\lambda) := \Big|\Pr\big[(VK, K) \leftarrow \mathsf{KeyGen}(1^\lambda);\ (IV^*, M_0^*, M_1^*, \mathrm{State}) \leftarrow \mathcal{A}^{\mathcal{O}}(1^\lambda, VK);\ b \xleftarrow{\$} \{0,1\};\ C^* \leftarrow \mathsf{Enc}(K, IV^*, M_b^*);\ b' \leftarrow \mathcal{A}^{\mathcal{O}}(C^*, \mathrm{State});\ b = b'\big] - \frac{1}{2}\Big|$$

where $\mathcal{O} := \{\mathsf{Enc}(K,\cdot,\cdot), \mathsf{Dec}(K,\cdot,\cdot), \mathsf{Prove}(K,\cdot,\cdot)\}$. $\mathsf{Enc}(K,\cdot,\cdot)$ is the encryption oracle which takes as input $IV \in \{0,1\}^n$ and $M \in \{0,1\}^L$, where $IV \neq IV^*$, and outputs $C \leftarrow \mathsf{Enc}(K, IV, M)$. $\mathsf{Dec}(K,\cdot,\cdot)$ is the decryption oracle which takes as input $IV \in \{0,1\}^n$ and $C \in \{0,1\}^L$, where $(IV, C) \neq (IV^*, C^*)$, and outputs $M/\perp \leftarrow \mathsf{Dec}(K, IV, M)$. $\mathsf{Prove}(K,\cdot,\cdot)$ is the prove oracle which takes as input $IV \in \{0,1\}^n$ and $C \in \{0,1\}^L$, where $IV \neq IV^*$, and outputs $\pi \leftarrow \mathsf{Prove}(K, IV, C)$.

IND-CCPA is stronger than the conventional IND-CCA since an adversary is allowed to issue the Prove oracle in addition to the Dec oracle. Note that there is a restriction of the Prove oracle compared to the Dec oracle, where IV^* is not allowed to input the oracle. (This restriction is due to the restriction of the VRF.Prove oracle.) We would like to indicate this as future work to be revisited.

Definition 3.5 (Proof soundness). We say that a SKENO scheme Π satisfies proof soundness if for all PPT adversaries $\mathcal{A}$ the following advantage is negligible in the security parameter:

$$\mathsf{Adv}^{\text{sound}}_{\Pi,\mathcal{A}}(1^\lambda) := \Pr\big[(VK, K) \leftarrow \mathsf{KeyGen}(1^\lambda);\ (IV, M, \mathrm{State}) \leftarrow \mathcal{A}(1^\lambda, VK, K);\ C \leftarrow \mathsf{Enc}(K, IV, M);\ (\tilde{M}, \tilde{\pi}) \leftarrow \mathcal{A}(C, \mathrm{State});\ \mathsf{Verify}(VK, IV, C, \tilde{M}, \tilde{\pi}) = 1;\ M \neq \tilde{M}\big].$$

Note that $M \in \{0,1\}^L$ and $\perp \notin \{0,1\}^L$ and that $\mathcal{A}$ wins if $\tilde{\pi}$ is a valid proof for $\tilde{M} = \perp$.

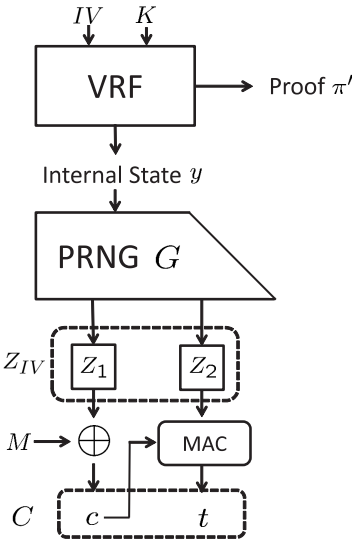

Figure 1. IND-CCPA secure SKENO construction.

4 Our SKENO constructions

In this section, we give our IND-CCPA secure SKENO scheme. As in the conversion from CPA-secure DEM to CCA-secure DEM using MAC [40], we can construct an IND-CCPA secure SKENO scheme. Let (VRF.Setup, VRF.Prove, VRF.Verify) be a VRF and G be a PRNG. Briefly, $Z_{IV} = G(F_K(IV))$ is divided into two parts, say $Z_1 \in \{0,1\}^{L_1}$ and $Z_2 \in \{0,1\}^{L_2}$, where $L_1 + L_2 = L$ and L_2 is the key length of the underlying MAC algorithm (MAC, Vrfy).

Protocol 4.1 (Proposed IND-CCPA secure SKENO).

- KeyGen(1^λ): Run $(PK, SK) \leftarrow$ VRF.Setup(1^λ) and output $(VK, K) = (PK, SK)$.
- Enc(K, IV, M): Run $(y, \pi') \leftarrow$ VRF.Prove(K, IV) and $Z_{IV} \leftarrow G(y)$, divide $Z_{IV} = Z_1||Z_2$, compute $c = M \oplus Z_1$ and $t = \mathsf{MAC}_{Z_2}(c)$ and output $C = (c, t)$.
- Dec(K, IV, C): Parse $C = (c, t)$. Run $(y, \pi') \leftarrow$ VRF.Prove(K, IV) and $Z_{IV} \leftarrow G(y)$, divide $Z_{IV} = Z_1||Z_2$, compute $M = c \oplus Z_1$ and output M if $\mathsf{Vrfy}_{Z_2}(c, t) = 1$ holds and $\perp$ otherwise.
- Prove(K, IV, C): Parse $C = (c, t)$. Run $(y, \pi') \leftarrow$ VRF.Prove(K, IV) and $Z_{IV} \leftarrow G(y)$ and divide $Z_{IV} = Z_1||Z_2$.
  If $\mathsf{Vrfy}_{Z_2}(c, t) = 1$ holds, then set $\pi = (y, \pi', 0)$.
  If $\mathsf{Vrfy}_{Z_2}(c, t) = 0$ holds, then set $\pi = (y, \pi', \perp)$.
- Verify(VK, IV, C, M, π): Parse $C = (c, t)$ and $\pi = (y, \pi', \pi'')$. Run $Z_{IV} \leftarrow G(y)$ and divide $Z_{IV} = Z_1||Z_2$.
  If $\pi'' = \perp$, VRF.Verify(VK, IV, y, π') $= 1$, $\mathsf{Vrfy}_{Z_2}(c, t) = 0$ and $M = \perp$, then output 1.
  If $\pi'' = 0$, VRF.Verify(VK, IV, y, π') $= 1$, $\mathsf{Vrfy}_{Z_2}(c, t) = 1$ and $c \oplus Z_1 = M$, then output 1.
  Otherwise, output 0.

We further demonstrate our construction by Figure 1 in order to give a direct visualization.

Here, we want to clarify some facts about the performance of our structure. The VRF we used here is a very strong primitive and its speed is far from the traditional symmetric key ciphers. However, due to the special property of the stream cipher, we put the VRF in the key scheduling algorithm which fortunately will not affect much of the encryption and decryption speed, since the speed of the PRNG is to some degree equal to the speed of the stream cipher. Take the final portfolio of eSTREAM project for example, HC-128 [41] is the fastest among all the candidates; however, it becomes the slowest if the key scheduling part is taken into consideration. In a real world environment where a stream cipher is used, the computational cost of a PRNG is dominant and depending on the applications. If the key scheduling part is not performed so frequently, then our scheme can still be treated as efficient compared with traditional symmetric key cryptography given the additional functionalities, and more so with public key cryptography.

5 Security analysis

In this section, we give the security proofs of our SKENO construction.

Theorem 5.1. *Our SKENO construction is IND-CCPA secure if the underlying VRF satisfies pseudorandomness, the underlying PRNG satisfies soundness and pseudorandomness and the underlying MAC is one-time sEU-CMA.*

To prove Theorem 5.1, we use sequences of games [39]. Consider the following games:

- Game 1: IND-CCPA experiment
- Game 2: The same as Game 1, except if $\mathcal{A}$ queries $IV \neq IV^*$ such that either $F_K(IV) = F_K(IV^*)$ or $F_K(IV) = y_1^*$, then abort.
- Game 3: The same as Game 2, except if $\mathcal{A}$ queries $IV \neq IV^*$ such that $G(F_K(IV)) = G(y^*)$, then abort.
- Game 4: The same as Game 3, except if $\mathcal{A}$ queries $(IV, C) \neq (IV^*, C^*)$ such that $Z_2 = Z_2^*$, then abort.
- Game 5: The same as Game 4, except if $\mathcal{A}$ queries $(IV, C) \neq (IV^*, C^*)$ such that $IV = IV^*$ and $\mathsf{Vrfy}_{Z_2^*}(c, t^*) = 1$, where $C = (c, t)$ and $C^* = (c^*, t^*)$ (then, $c \neq c^*$ and $t = t^*$ hold), then abort.
- Game 6: The same as Game 5, except all Z_{IV} is randomly chosen instead of using PRNG G.
- Game 7: The same as Game 6, except the challenge ciphertext C^* is computed by using $y' \xleftarrow{\$} \{0,1\}^m$ (which is randomly chosen instead of $y^* = F_{SK}(x^*)$).
  Let Succ_i be the event that $\mathcal{A}$ wins in Game i. Then $\mathcal{A}$'s advantage can be estimated as

$$\mathsf{Adv}^{\text{Ind-ccpa}}_{\mathsf{SKENO},\mathcal{A}}(1^\lambda) = \left|\Pr[\mathrm{Succ}_1] - \frac{1}{2}\right| \leq \sum_{i=1}^{4} |\Pr[\mathrm{Succ}_i] - \Pr[\mathrm{Succ}_{i+1}]| + \left|\Pr[\mathrm{Succ}_5] - \frac{1}{2}\right|$$

and thus it remains to upperbound the right-hand side of the above inequality, a task we carry out in the following claims:

Claim 5.2. *There exists a PPT adversary $\mathcal{B}$ such that* $|\Pr[\mathrm{Succ}_1] - \Pr[\mathrm{Succ}_2]| \leq \mathsf{Adv}^{\text{Pseudo}}_{F,\mathcal{B}}(1^\lambda) + 1/2^m$.

Proof. It is straightforward to verify that if $\mathcal{A}$ queries $IV \neq IV^*$ such that $F_K(IV) = F_K(IV^*)$, then $\mathcal{B}$ can immediately break pseudo-randomness of the underlying VRF as follows: $\mathcal{B}$ can query IV to the VRF.Prove oracle, $\mathcal{B}$ obtains $F_K(IV)$ and outputs 0 if $y^* = F_K(IV)$ and 1 otherwise. Note that $F_K(IV) = y_1^*$ happens with probability at most $1/2^m$ since $y_1^* \xleftarrow{\$} \{0,1\}^m$ according to the definition of pseudo-randomness and is negligible. □

Claim 5.3. *There exists a PPT adversary $\mathcal{B}$ such that* $|\Pr[\mathrm{Succ}_2] - \Pr[\mathrm{Succ}_3]| \leq \mathsf{Adv}^{\text{Sound}}_{\mathrm{PRNG},\mathcal{B}}(1^\lambda)$.

Proof. Note that $F_K(IV) \notin \{F_K(IV^*), y_1^*\}$ holds in this game. So, if $\mathcal{A}$ queries $IV \neq IV^*$ such that $G(F_K(IV)) = G(y^*)$, then $\mathcal{B}$ immediately breaks soundness of the underlying PRNG. □

Claim 5.4. *There exists a PPT adversary $\mathcal{B}$ such that* $|\Pr[\mathrm{Succ}_3] - \Pr[\mathrm{Succ}_4]| \leq \mathsf{Adv}^{\text{ot-sEU-CMA}}_{\Pi,\mathcal{A}}(1^\lambda)$.

Proof. Let $\mathcal{C}_{\mathrm{Sig}}$ be the challenger of the underlying OTS. $\mathcal{B}$ runs $(VK, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and gives VK to $\mathcal{A}$. In the challenge phase, $\mathcal{A}$ sends (IV^*, M_0^*, M_1^*) to $\mathcal{B}$. $\mathcal{B}$ randomly chooses $c^* \xleftarrow{\$} \{0,1\}^{L_1}$ and sends c^* to $\mathcal{C}_{\mathrm{Sig}}$. $\mathcal{C}_{\mathrm{Sig}}$ computes $t^* \leftarrow \mathsf{MAC}_{Z_2^*}(c^*)$. $\mathcal{B}$ sends (c^*, t^*) to $\mathcal{A}$. If $\mathcal{A}$ queries $(IV, C) \neq (IV^*, C^*)$ such that $Z_2 = Z_2^*$, then $\mathcal{B}$ immediately breaks the underlying MAC. □

Claim 5.5. *There exists a PPT adversary $\mathcal{B}$ such that* $|\Pr[\mathrm{Succ}_4] - \Pr[\mathrm{Succ}_5]| \leq \mathsf{Adv}^{\text{ot-sEU-CMA}}_{\Pi,\mathcal{A}}(1^\lambda)$.

Proof. Let $\mathcal{C}_{\mathrm{Sig}}$ be the challenger of the underlying OTS. $\mathcal{B}$ runs $(VK, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and gives VK to $\mathcal{A}$. In the challenge phase, $\mathcal{A}$ sends (IV^*, M_0^*, M_1^*) to $\mathcal{B}$. $\mathcal{B}$ randomly chooses $c^* \xleftarrow{\$} \{0,1\}^{L_1}$ and sends c^* to $\mathcal{C}_{\mathrm{Sig}}$. $\mathcal{C}_{\mathrm{Sig}}$ computes $t^* \leftarrow \mathsf{MAC}_{Z_2^*}(c^*)$. $\mathcal{B}$ sends (c^*, t^*) to $\mathcal{A}$. If $\mathcal{A}$ queries $(IV, C) \neq (IV^*, C^*)$ such that $IV = IV^*$ and $\mathsf{Vrfy}_{Z_2^*}(c, t^*) = 1$, then $\mathcal{B}$ outputs (c, t) as a forged MAC, since $(c, t) \neq (c^*, t^*)$ holds. □

The following claim clearly holds:

Claim 5.6. *There exists a PPT adversary $\mathcal{B}$ such that* $|\Pr[\mathrm{Succ}_5] - \Pr[\mathrm{Succ}_6]| \leq \mathsf{Adv}^{\text{Pseudo}}_{\mathrm{PRNG},\mathcal{B}}(1^\lambda)$.

Claim 5.7. *There exists a PPT adversary $\mathcal{B}$ such that* $|\Pr[\mathrm{Succ}_6] - \Pr[\mathrm{Succ}_7]| \leq \mathsf{Adv}^{\text{Pseudo}}_{F,\mathcal{B}}(1^\lambda)$.

Proof. Let $\mathcal{A}$ be an adversary who can break the IND-CCPA security of our SKENO construction. We construct an algorithm $\mathcal{B}$ that breaks the pseudorandomness of the underlying VRF. Let $\mathcal{C}_{\mathrm{VRF}}$ be the challenger of the underlying VRF. $\mathcal{C}_{\mathrm{VRF}}$ sends PK to $\mathcal{B}$. $\mathcal{B}$ sets $VK = PK$ and sends VK to $\mathcal{A}$.

We begin with the phase 1 query.

- Enc: Let (IV, M) be an encryption query issued by $\mathcal{A}$. $\mathcal{B}$ sends IV to $\mathcal{C}_{\mathrm{VRF}}$ as a VRF.Prove query, obtains (y, π'), chooses $Z_{IV} \xleftarrow{\$} \{0,1\}^{L_1+L_2}$, divides $Z_{IV} = Z_1||Z_2$, computes $c = M \oplus Z_1$ and $t = \mathsf{MAC}_{Z_2}(c)$ and returns $C = (c, t)$.
- Prove: Let (IV, C) be a prove query issued by $\mathcal{A}$. Parse $C = (c, t)$. $\mathcal{B}$ sends IV to $\mathcal{C}_{\mathrm{VRF}}$ as a VRF.Prove query, obtains (y, π'), chooses $Z_{IV} \xleftarrow{\$} \{0,1\}^{L_1+L_2}$ and divides $Z_{IV} = Z_1||Z_2$.
  If $\mathsf{Vrfy}_{Z_2}(c, t) = 1$ holds, then returns $\pi = (y, \pi', 0)$.
  If $\mathsf{Vrfy}_{Z_2}(c, t) = 0$ holds, then returns $\pi = (y, \pi', \perp)$.
- Dec: Let (IV, C) be a decryption query issued by $\mathcal{A}$. $\mathcal{B}$ works as in the simulation of the Prove oracle, except $\mathcal{B}$ returns $M = c \oplus Z_1$ (if $\mathsf{Vrfy}_{Z_2}(c, t) = 1$) or $\perp$ (if $\mathsf{Vrfy}_{Z_2}(c, t) = 0$).

In the challenge phase, $\mathcal{A}$ sends (IV^*, M_0^*, M_1^*) to $\mathcal{B}$ and $\mathcal{B}$ sends IV^* to $\mathcal{C}_{\mathrm{VRF}}$. $\mathcal{C}_{\mathrm{VRF}}$ selects $b \in \{0,1\}$ and computes y^* such that $y^* = F_K(IV^*)$ (if $b = 0$) and $y^* \xleftarrow{\$} \{0,1\}^m$ (if $b = 1$) and sends y^* to $\mathcal{B}$. $\mathcal{B}$ randomly chooses $b'' \in \{0,1\}$, chooses $Z_{IV^*} \xleftarrow{\$} \{0,1\}^{L_1+L_2}$, divides $Z_{IV}^* = Z_1^*||Z_2^*$, computes $c^* = M_{b''}^* \oplus Z_1^*$ and $t^* = \mathsf{MAC}_{Z_2^*}(c^*)$ and sends $C^* = (c^*, t^*)$ to $\mathcal{A}$.

We continue with the phase 2 query.

- Enc: The same as the phase 1 query. Note that $\mathcal{A}$ never issues IV^* as an encryption query.
- Prove: Let (IV, C) be a prove query issued by $\mathcal{A}$, where $IV \neq IV^*$. Parse $C = (c, t)$. We consider four cases as follows:
  - $C = C^*$: $\mathcal{B}$ sends IV to $\mathcal{C}_{\mathrm{VRF}}$ as a VRF.Prove query and obtains (y, π'). $\mathcal{B}$ chooses $Z_{IV} \xleftarrow{\$} \{0,1\}^{L_1+L_2}$ and divides $Z_{IV} = Z_1||Z_2$. Note that the case $Z_2 = Z_2^*$ has been excluded in Game 4. Then, since $Z_2 \neq Z_2^*$, $\mathsf{Vrfy}_{Z_2}(c^*, t^*) = 0$ holds due to the property assumed in Section 2.3. Therefore, $\mathcal{B}$ sets $\pi = (y, \pi', \perp)$ and returns π.
  - $c = c^* \wedge t \neq t^*$: $\mathcal{B}$ sends IV to $\mathcal{C}_{\mathrm{VRF}}$ as a VRF.Prove query and obtains (y, π'). $\mathcal{B}$ chooses $Z_{IV} \xleftarrow{\$} \{0,1\}^{L_1+L_2}$ and divides $Z_{IV} = Z_1||Z_2$. If $\mathsf{Vrfy}_{Z_2}(c^*, t) = 1$ holds, then $\mathcal{B}$ sets $\pi = (y, \pi', 0)$ and returns π. Otherwise, since $\mathsf{Vrfy}_{Z_2}(c^*, t) = 0$ holds, $\mathcal{B}$ sets $\pi = (y, \pi', \perp)$ and returns π.
  - $c \neq c^* \wedge t = t^*$: $\mathcal{B}$ sends IV to $\mathcal{C}_{\mathrm{VRF}}$ as a VRF.Prove query and obtains (y, π'). $\mathcal{B}$ chooses $Z_{IV} \xleftarrow{\$} \{0,1\}^{L_1+L_2}$ and divides $Z_{IV} = Z_1||Z_2$. If $\mathsf{Vrfy}_{Z_2}(c, t^*) = 1$ holds, then $\mathcal{B}$ sets $\pi = (y, \pi', 0)$ and returns π. Otherwise, since $\mathsf{Vrfy}_{Z_2}(c, t^*) = 0$ holds, $\mathcal{B}$ sets $\pi = (y, \pi', \perp)$ and returns π.
  - $c \neq c^* \wedge t \neq t^*$: In this case, $\mathcal{B}$ works as in phase 1.
- Dec: Let $(IV, C) \neq (IV^*, C^*)$ be a decryption query. Parse $C = (c, t)$. First, we estimate the case $IV = IV^*$. Note that the case $\mathsf{Vrfy}_{Z_2^*}(c, t^*) = 1$ has been excluded in Game 5. Therefore, since $\mathsf{Vrfy}_{Z_2^*}(c, t^*) = 0$ holds, $\mathcal{B}$ returns $\perp$. Next, let $IV \neq IV^*$. $\mathcal{B}$ works as in the simulation of the Prove oracle, except $\mathcal{B}$ returns $M = c \oplus Z_1$ (if $\mathsf{Vrfy}_{Z_2}(c, t) = 1$) or $\perp$ (if $\mathsf{Vrfy}_{Z_2}(c, t) = 0$).

Finally, in the guessing phase, $\mathcal{A}$ outputs $b' \in \{0,1\}$. Note that if $b = 0$ (that is, $y^* = F_K(IV^*)$), then C^* is a valid ciphertext of $M_{b''}^*$ (that is, in Game 6). So, $\mathcal{A}$ has an advantage. Otherwise, let $b = 1$ (that is, in Game 7; then y^* is a random value and is independent of IV^*, therefore, $\Pr[\mathrm{Succ}_7] = 1/2$ holds). Then, the probabilistic distribution of $M_0^* \oplus Z_1^*$ and the probabilistic distribution of $M_1^* \oplus Z_1^*$ are identical. So, $\mathcal{A}$ has no advantage in the case when $b = 1$. Therefore, if $b' = b''$, then $\mathcal{B}$ outputs 0, and 1 otherwise. □

Theorem 5.8. *Our SKENO construction satisfies proof soundness if the underlying VRF satisfies uniqueness and the underlying MAC satisfies correctness.*

Proof. Let $(M, \tilde{M})$ be a message pair and $C \leftarrow \mathsf{Enc}(K, IV, M)$ (these have appeared in the proof soundness definition). We consider two cases.

If $\tilde{M} = \perp$, since $\mathsf{Verify}(VK, IV, C, \tilde{M}, \tilde{\pi}) = 1$, $\tilde{\pi} = (\tilde{y}, \tilde{\pi}', \tilde{\pi}'')$ satisfies $\tilde{\pi}'' = \perp$, $\mathsf{VRF.Verify}(VK, IV, \tilde{y}, \tilde{\pi}') = 1$, and $\mathsf{Vrfy}_{Z_2}(c, t) = 0$, where $Z_1||Z_2 := \tilde{Z_{IV}} = G(\tilde{y})$. Since $C = (c, t)$ is an honestly computed ciphertext of M (such as $C \leftarrow \mathsf{Enc}(K, IV, M)$), for $(y, \pi') \leftarrow \mathsf{VRF.Prove}(K, IV)$, $y = \tilde{y}$ and $\mathsf{VRF.Verify}(VK, IV, y, \pi') = 1$ hold (the former holds due to the uniqueness property of the VRF). Therefore, $\mathsf{Vrfy}_{Z_2}(c, t) = 1$ holds due to the correctness property of the MAC. Therefore $\mathsf{Verify}(VK, IV, C, \tilde{M}, \tilde{\pi}) = 1$ never holds.

If $\tilde{M} \neq \perp$, since $\mathsf{Verify}(VK, IV, C, \tilde{M}, \tilde{\pi}) = 1$, $\tilde{\pi} = (\tilde{y}, \tilde{\pi}', \tilde{\pi}'')$ satisfies $\tilde{\pi}'' = 0$, $\mathsf{VRF.Verify}(VK, IV, \tilde{y}, \tilde{\pi}') = 1$, $\mathsf{Vrfy}_{Z_2}(c, t) = 1$, and $c \oplus Z_1 = \tilde{M}$, where $Z_1 || Z_2 := \tilde{Z_{IV}} = G(\tilde{y})$. Since $C = (c, t)$ is an honestly computed ciphertext of M, $\mathsf{Verify}(VK, IV, C, M, \pi) = 1$ holds, where $\pi = (y, \pi', 0)$ and $(y, \pi') \leftarrow \mathsf{VRF.Prove}(K, IV)$. Since $M \neq \tilde{M}$, $c \oplus Z_1 \neq \tilde{M}$ holds and therefore $\mathsf{Verify}(VK, IV, C, \tilde{M}, \tilde{\pi}) = 1$ never holds. □

6 Discussion

One may think that SKENO can be constructed from weaker primitives, for example, wVRFs. Actually, Brakerski, Goldwasser, Rothblum and Vaikuntanathan [12] showed that there is no black-box construction of wVRF from one-way permutation, whereas it is possible to construct wVRF from enhanced trapdoor permutation (TDP) in a black-box manner. By combining the results of [22] and [12], wVRF is a strictly weaker primitive than VRF, that is, there is no black-box construction of VRF from wVRF.

Briefly, wVRF has similar functionality to the VRF, but in the security definition an adversary is given an output/proof pair $\{(y_i, \pi_i)\}$ for randomly chosen $\{x_i\}$. So, in the following we can explain how difficult it is to achieve the functionality of SKENO if we replace the VRF with a wVRF. To answer encryption queries, the simulator $\mathcal{B}$ (which is an adversary against the pseudo-randomness of the wVRF) needs to compute $F_K(IV)$. However, recall that $\mathcal{B}$ is only given an output/proof pair $\{(y_i, \pi_i)\}$ for randomly chosen $\{x_i\}$ and cannot obtain a VRF value that is of $\mathcal{B}$'s choice. Since IV is chosen by an adversary $\mathcal{A}$ (that is, $IV \notin \{x_i\}_{i=1}^{\mathrm{poly}(\lambda)}$ holds with overwhelming probability), it is hard to answer the query in the wVRF setting. Meanwhile, such a problem does not occur if we use a (non-weak-)VRF, because the simulator $\mathcal{B}$ (which is an adversary against the pseudo-randomness of the VRF) can query IV to the VRF challenger and get the corresponding VRF value.

To bridge the initially chosen inputs $\{x_i\}_{i=1}^{\mathrm{poly}(\lambda)}$ with adversarially chosen IV values, we can employ a random oracle (say H) such that the value $H(IV)$ is set as x. In other words, it seems hard to achieve the SKENO functionality by using weaker primitives than VRF in the standard model. Constructing SKENO from weaker assumptions in the standard model is an interesting question for future work.

7 Conclusion

The previously proposed non-interactive opening ciphertext techniques were aimed at public key cryptosystems but cannot perform in the situation where a symmetric key cryptosystem is being used and the secret key is shared among a group of people, since the key itself is not appropriate for opening in order to behave as a proof. This paper fills the above gaps by proposing the first stream cipher based SKENO scheme that can be proved to be IND-CCPA secure, which can provide non-interactive opening services in the shared key environment.

Although we have to mention that the VRF primitive is a relatively expensive assumption, our construction can be considered to be efficient in the environments where the key and IV setup phase is not executed frequently, since the execution of a PRNG will play a dominant role in the encryption and decryption process.

Acknowledgement: The authors would like to thank Dr. Takahiro Matsuda (AIST, Japan) for his invaluable comments. The preliminary version of this paper appeared in the 13th International Conference on Information and Communications Security, ICICS 2011 [15]; this is the full version.

References

[1] M. Abdalla, D. Catalano and D. Fiore, Verifiable random functions from identity-based key encapsulation, in: *Advances in Cryptology – EUROCRYPT 2009* (Cologne 2009), Lecture Notes in Comput. Sci. 5479, Springer, Berlin (2009), 554–571.

[2] M. Abdalla, D. Catalano and D. Fiore, Verifiable random functions: Relations to identity-based key encapsulation and new constructions, *J. Cryptology* **27** (2013), 544–593.

[3] M. Belenkiy, M. Chase, M. Kohlweiss and A. Lysyanskaya, Compact e-cash and simulatable VRFs revisited, in: *Pairing-Based Cryptography – Pairing 2009* (Palo Alto 2009), Lecture Notes in Comput. Sci. 5671, Springer, Berlin (2009), 114–131.

[4] M. Bellare and C. Namprempre, Authenticated encryption: Relations among notions and analysis of the generic composition paradigm, in: *Advances in Cryptology – ASIACRYPT 2000* (Kyoto 2000), Lecture Notes in Comput. Sci. 1976, Springer, Berlin (2000), 531–545.

[5] M. Bellare and S. Shoup, Two-tier signatures, strongly unforgeable signatures, and Fiat–Shamir without random oracles, in: *Public Key Cryptography – PKC 2007* (Beijing 2007), Lecture Notes in Comput. Sci. 4450, Springer, Berlin (2007), 201–216.

[6] C. Berbain and H. Gilbert, On the security of IV dependent stream ciphers, in: *Fast Software Encryption* (Luxembourg 2007), Lecture Notes in Comput. Sci. 4593, Springer, Berlin (2007), 254–273.

[7] G. Bertoni, J. Daemen, M. Peeters and G. Van Assche, Sponge-based pseudo-random number generators, in: *Cryptographic Hardware and Embedded Systems – CHES 2010* (Santa Barbara 2010), Lecture Notes in Comput. Sci. 6225, Springer, Berlin (2010), 33–47.

[8] T. E. Bjørstad, A. W. Dent and N. P. Smart, Efficient KEMs with partial message recovery, in: *Cryptography and Coding* (Cirencester 2007), Lecture Notes in Comput. Sci. 4887, Springer, Berlin (2007), 233–256.

[9] M. Blum and S. Micali, How to generate cryptographically strong sequences of pseudo-random bits, *SIAM J. Comput.* **13** (1984), 850–864.

[10] D. Boneh and M. K. Franklin, Identity-based encryption from the Weil pairing, in: *Advances in Cryptology – CRYPTO 2001* (Santa Barbara 2001), Lecture Notes in Comput. Sci. 2139, Springer, Berlin (2001), 213–229.

[11] D. Boneh and J. Katz, Improved efficiency for CCA-secure cryptosystems built using identity-based encryption, in: *Topics in Cryptology – CT-RSA 2005* (San Francisco 2005), Lecture Notes in Comput. Sci. 3376, Springer, Berlin (2005), 87–103.

[12] Z. Brakerski, S. Goldwasser, G. N. Rothblum and V. Vaikuntanathan, Weak verifiable random functions, in: *Theory of Cryptography* (San Francisco 2009), Lecture Notes in Comput. Sci. 5444, Springer, Berlin (2009), 558–576.

[13] J. Camenisch, S. Hohenberger and A. Lysyanskaya, Compact e-cash, in: *Advances in Cryptology – EUROCRYPT 2005* (Aarhus 2005), Lecture Notes in Comput. Sci. 3494, Springer, Berlin (2005), 302–321.

[14] R. Canetti, Universally composable signature, certification, and authentication, in: *Computer Security Foundations Workshop* (Pacific Grove 2004), IEEE Computer Society, Washington (2004), 219–233.

[15] J. Chen, K. Emura and A. Miyaji, Non-interactive opening for ciphertexts encrypted by shared keys, in: *Information and Communications Security* (Beijing 2011), Lecture Notes in Comput. Sci. 7043, Springer, Berlin (2011), 57–68.

[16] D. Dachman-Soled, G. Fuchsbauer, P. Mohassel and A. O'Neill, Enhanced chosen-ciphertext security and applications, in: *Public-Key Cryptography – PKC 2014* (Buenos Aires 2014), Lecture Notes in Comput. Sci. 8383, Springer, Berlin (2014), 329–344.

[17] I. Damgård, D. Hofheinz, E. Kiltz and R. Thorbek, Public-key encryption with non-interactive opening, in: *Topics in Cryptology – CT-RSA 2008* (San Francisco 2008), Lecture Notes in Comput. Sci. 4964, Springer, Berlin (2008), 239–255.

[18] Y. Dodis, Efficient construction of (distributed) verifiable random functions, in: *Public Key Cryptography – PKC 2003* (Miami 2003), Lecture Notes in Comput. Sci. 2567, Springer, Berlin (2003), 1–17.

[19] Y. Dodis and A. Yampolskiy, A verifiable random function with short proofs and keys, in: *Public Key Cryptography – PKC 2005* (Les Diablerets 2005), Lecture Notes in Comput. Sci. 3386, Springer, Berlin (2005), 416–431.

[20] C. Dwork and M. Naor, Zaps and their applications, *SIAM J. Comput.* **36** (2007), 1513–1543.

[21] K. Emura, G. Hanaoka, Y. Sakai and J. C. N. Schuldt, Group signature implies public-key encryption with non-interactive opening, *Int. J. Inf. Secur.* **13** (2014), 51–62.

[22] D. Fiore and D. Schröder, Uniqueness Is a different story: Impossibility of verifiable random functions from trapdoor permutations, in: *Theory of Cryptography* (Taormina 2012), Lecture Notes in Comput. Sci. 7194, Springer, Berlin (2012), 636–653.

[23] M. J. Freedman, Y. Ishai, B. Pinkas and O. Reingold, Keyword search and oblivious pseudorandom functions, in: *Theory of Cryptography* (Cambridge 2005), Lecture Notes in Comput. Sci. 3378, Springer, Berlin (2005), 303–324.

[24] D. Galindo, Breaking and repairing Damgård et al. public key encryption scheme with non-interactive opening, in: *Topics in Cryptology – CT-RSA 2009* (San Francisco 2009), Lecture Notes in Comput. Sci. 5473, Springer, Berlin (2009), 389–398.

[25] D. Galindo, B. Libert, M. Fischlin, G. Fuchsbauer, A. Lehmann, M. Manulis and D. Schröder, Public-key encryption with non-interactive opening: New constructions and stronger definitions, in: *Progress in Cryptology – AFRICACRYPT 2010* (Stellenbosch 2010), Lecture Notes in Comput. Sci. 6055, Springer, Berlin (2010), 333–350.

[26] O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions, *J. ACM* **33** (1986), 792–807.

[27] J. Groth, Simulation-sound NIZK proofs for a practical language and constant size group signatures, in: *Advances in Cryptology – ASIACRYPT 2006* (Shanghai 2006), Lecture Notes in Comput. Sci. 4284, Springer, Berlin (2006), 444–459.
[28] C. Hazay and Y. Lindell, Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries, in: *Theory of Cryptography* (New York 2008), Lecture Notes in Comput. Sci. 4948, Springer, Berlin (2008), 155–175.
[29] S. Hohenberger and B. Waters, Constructing verifiable random functions with large input spaces, in: *Advances in Cryptology – EUROCRYPT 2010* (French Riviera 2010), Lecture Notes in Comput. Sci. 6110, Springer, Berlin (2010), 656–672.
[30] S. Jarecki and X. Liu, Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection, in: *Theory of Cryptography* (San Francisco 2009), Lecture Notes in Comput. Sci. 5444, Springer, Berlin (2009), 577–594.
[31] J. Lai, R. H. Deng, S. Liu and W. Kou, Efficient CCA-secure PKE from identity-based techniques, in: *Topics in Cryptology – CT-RSA 2010* (San Francisco 2010), Lecture Notes in Comput. Sci. 5985, Springer, Berlin (2010), 132–147.
[32] M. Liskov, Updatable zero-knowledge databases, in: *Advances in Cryptology – ASIACRYPT 2005* (Chennai 2005), Lecture Notes in Comput. Sci. 3788, Springer, Berlin (2005), 174–198.
[33] A. Lysyanskaya, Unique signatures and verifiable random functions from the DH-DDH separation, in: *Advances in Cryptology – CRYPTO 2002* (Santa Barbara 2002), Lecture Notes in Comput. Sci. 2442, Springer, Berlin (2002), 597–612.
[34] T. Matsuda, G. Hanaoka, K. Matsuura and H. Imai, An efficient encapsulation scheme from near collision resistant pseudorandom generators and its application to IBE-to-PKE transformations, in: *Topics in Cryptology – CT-RSA 2009* (San Francisco 2009), Lecture Notes in Comput. Sci. 5473, Springer, Berlin (2009), 16–31.
[35] S. Micali, M. O. Rabin and S. P. Vadhan, Verifiable random functions, in: *Foundations of Computer Science* (New York 1999), IEEE Computer Society, Washington (1999), 120–130.
[36] S. Micali and L. Reyzin, Soundness in the public-key model, in: *Advances in Cryptology – CRYPTO 2001* (Santa Barbara 2001), Lecture Notes in Comput. Sci. 2139, Springer, Berlin (2001), 542–565.
[37] S. Micali and R. L. Rivest, Micropayments revisited, in: *Topics in Cryptology – CT-RSA 2002* (San Francisco 2002), Lecture Notes in Comput. Sci. 2271, Springer, Berlin (2002), 149–163.
[38] C. Peikert, *Theoretical Foundations of Cryptography*, Lecture 11: Message authentication, 2010, https://wiki.cc.gatech.edu/theory/images/9/9e/Lec11.pdf.
[39] V. Shoup, Sequences of games: A tool for taming complexity in security proofs, preprint (2004), http://eprint.iacr.org/2004/332.
[40] V. Shoup, *ISO/IEC 18033-2. Encryption Algorithms – Part 2: Asymmetric Ciphers*, International Organization for Standardization, 2006.
[41] H. Wu, The stream cipher HC-128, in: *New Stream Cipher Designs*, Lecture Notes in Comput. Sci. 4986, Springer, Berlin (2008), 39–47.
[42] A. C.-C. Yao, Theory and applications of trapdoor functions, in: *Foundations of Computer Science* (Chicago 1982), IEEE Computer Society, Washington, (1982), 80–91.

Received April 1, 2014; revised June 16, 2014; accepted November 12, 2014.