Cryptographic Reverse Firewall via Malleable Smooth Projective Hash Functions

Rongmao Chen[1,2(✉)], Yi Mu[1], Guomin Yang[1], Willy Susilo[1], Fuchun Guo[1], and Mingwu Zhang[3]

[1] School of Computing and Information Technology, Centre for Computer and Information Security Research, University of Wollongong, Wollongong, Australia
{rc517,ymu,gyang,wsusilo,fuchun}@uow.edu.au
[2] College of Computer, National University of Defense Technology, Changsha, China
[3] School of Computers, Hubei University of Technology, Wuhan, China
csmwzhang@gmail.com

Abstract. Motivated by the revelations of Edward Snowden, post-Snowden cryptography has become a prominent research direction in recent years. In Eurocrypt 2015, Mironov and Stephens-Davidowitz proposed a novel concept named *cryptographic reverse firewall* (CRF) which can resist exfiltration of secret information from an arbitrarily compromised machine. In this work, we continue this line of research and present generic CRF constructions for several widely used cryptographic protocols based on a new notion named *malleable smooth projective hash function*. Our contributions can be summarized as follows.

- We introduce the notion of malleable smooth projective hash function, which is an extension of the smooth projective hash function (SPHF) introduced by Cramer and Shoup (Eurocrypt'02) with the new properties of *key malleability* and *element rerandomizability*. We demonstrate the feasibility of our new notion using graded rings proposed by Benhamouda et al. (Crypto'13), and present an instantiation from the k-linear assumption.
- We show how to generically construct CRFs via malleable SPHFs in a modular way for some widely used cryptographic protocols. Specifically, we propose generic constructions of CRFs for the unkeyed message-transmission protocol and the oblivious signature-based envelope (OSBE) protocol of Blazy, Pointcheval and Vergnaud (TCC'12). We also present a new malleable SPHF from the linear encryption of valid signatures for instantiating the OSBE protocol with CRFs.
- We further study the two-pass oblivious transfer (OT) protocol and show that the malleable SPHF does not suffice for its CRF constructions. We then develop a new OT framework from graded rings and show how to construct OT-CRFs by modifying the malleable SPHF framework. This new framework encompasses the DDH-based OT-CRF constructions proposed by Mironov and Stephens-Davidowitz (Eurocrypt'15), and yields a new construction under the k-linear assumption.

Keywords: Cryptographic reverse firewall · Malleable smooth projective hash function · Oblivious signature-based envelope · Oblivious transfer

J.H. Cheon and T. Takagi (Eds.): ASIACRYPT 2016, Part I, LNCS 10031, pp. 844–876, 2016.
DOI: 10.1007/978-3-662-53887-6_31

1 Introduction

In the last couple of years, the revelations of Edward Snowden [18,22] showed that the intelligence agencies successfully gained access to a massive collection of user sensitive data by undermining security mechanisms via a broad range of techniques, e.g., by subverting cryptographic protocols and actively deploying security weaknesses in the implementations of cryptosystems. The disclosures of Snowden have reawakened the cryptographic research community to the seriousness of the undermining of cryptographic solutions and standards [6–8,13,23,24], and led to a new research direction known as post-Snowden cryptography. The research problem could be generally summarized by the following question: "How to achieve meaningful security for cryptographic protocols in the presence of an adversary that may arbitrarily tamper with the victim's machine?"

Cryptographic Reverse Firewall. Motivated by the aforementioned question, Mironov and Stephens-Davidowitz [21] recently proposed a novel notion named cryptographic reverse firewall (CRF) aiming at providing strong security against inside vulnerabilities such as security backdoors. Informally, a CRF is a machine that sits at the boundary between the user's computer and the outside world. It plays as the role of an autonomous intermediary that intercepts and modifies the machine's incoming and outgoing messages to provide security protections even if the user's machine is compromised. A cryptographic protocol equipped with a correctly implemented CRF can guarantee that its security is preserved even if it is run on a compromised machine and the CRF could also resist exfiltration of secret information from the tampered machine. More specifically, Mironov and Stephens-Davidowitz defined three desirable properties for an honestly implemented CRF:

- *Functionality Maintaining.* A CRF should not break the functionality (i.e., correctness) of an honestly implemented protocol.
- *Security Preservation.* A protocol with a CRF should provide the same security guarantee as the properly implemented protocol regardless of how the underlying machine behaves.
- *Exfiltration Resistance.* A CRF should resist exfiltration so that a compromised implementation cannot leak any information to the outside world.

The above three properties deserve further interpretation. A good cryptographic protocol should be functional and secure regardless of the existence of the CRF when the protocol implementation is correct. That is, the user does not rely solely on the CRF for security but only requires it to preserve security. In particular, the CRF shares no secret with the protocol party, and thus even if the CRF is not functioning, an honestly implemented protocol would remain secure. This is one significant difference between the CRF and the prior work. On the other hand, when the protocol implementation is tampered but the CRF is implemented correctly, the CRF could provide the user with the desired security guarantee. In short, a protocol with CRF satisfies the security requirement

as long as either the protocol implementation is not tampered or the CRF is implemented correctly.

The CRF could be viewed as a modern take on a line of work that received considerable attention in the 80s and 90s [10,28]. It provides a general framework for building cryptographic schemes that remain secure when run on a compromised machine. The use of rerandomization to "sanitize" messages by the CRF is seemingly similar to the prior work, e.g., divertible protocols [10] and collusion-free protocols [3,19]. As summarized by Mironov and Stephens-Davidowitz in [21], the CRF is a generalization of these prior notions and models.

Motivations of This Work. In this work, we further explore the construction of CRFs. Unlike prior work that relies on concrete techniques and thus appears complicated, our goal is to develop generic paradigms for constructing CRFs in a conceptually simple and modular way. From a theoretical point of view, a generic paradigm can modularly explain concrete CRF constructions and their underlying design principles. From a practical point of view, a generic CRF construction based on abstract building blocks enables more concrete instantiations to be built for better security and/or efficiency. In fact, our work (partially) answers an open question raised by Mironov and Stephens-Davidowitz in [21]. Particularly, they stated that "*the "holy grail" would be a full characterization of functionalities and security properties for which reverse firewall exists*".

1.1 Overview of Our Contributions

We introduce the notion of *malleable smooth projective hash function*, which is a new extension of the conventional SPHF. A malleable SPHF is a special SPHF which is of additional properties, namely projection key malleablility and element re-randomizability. Using this notion, we obtain generic CRF constructions for some widely used cryptographic protocols. Before we describe our results, we present an overview of the malleable smooth projective hash function.

Malleable Smooth Projective Hash Function. We first briefly recall the classical definition of the smooth projective hash function (SPHF) (also known as hash proof system) introduced by Cramer and Shoup [12].

Classical Definition. An SPHF requires the existence of a domain $\mathcal{X}$ and an underlying $\mathcal{NP}$ language $\mathcal{L}$, where elements of $\mathcal{L}$ form a subset of $\mathcal{X}$, i.e., $\mathcal{L} \subset \mathcal{X}$. The key property of SPHF is that the hash value of any element $C \in \mathcal{L}$ can be computed by using either a secret hashing key hk, or a public projection key hp with the witness to the fact that $C \in \mathcal{L}$. However, the projection key gives almost no information about the hash value of any element in $\mathcal{X} \backslash \mathcal{L}$. Moreover, we say that the subset membership problem is hard if the distribution of $\mathcal{L}$ is computationally indistinguishable from $\mathcal{X} \backslash \mathcal{L}$.

New Properties. In addition to the above properties of a regular SPHF, we define two new properties for a malleable SPHF as follows.

- **Projection Key Malleability.** This property captures that,
  - *Key Indistinguishability*: any projection key hp can be re-randomized to an independent projection key $\widetilde{\mathsf{hp}}$ using a uniformly chosen randomness $\widetilde{r}$; and
  - *Projection Consistency*: the hash value difference of any element due to the above key re-randomization is computable using $\widetilde{r}$.
- **Element Re-randomizability.** This property captures that,
  - *Element Indistinguishability*: any element C can be re-randomized to another independent element $\widetilde{C}$ using a uniformly chosen witness $\widetilde{w}$; and
  - *Rerandomization Consistency*: the hash value difference between C and $\widetilde{C}$ under the same hashing key is computable using the associated projection key with $\widetilde{w}$; and
  - *Membership Preservation*: the re-randomization of an element does not change its membership (i.e., $\widetilde{C} \in \mathcal{L} \iff C \in \mathcal{L}$).

A Simple Example. We provide a very simple example of our new notion. We remark that such a simple example is just for a quick understanding of the properties captured by our malleable SPHF. The construction would be more complicated from other assumptions. The basic SPHF below is exactly the one of Cramer and Shoup for the DDH language in [12]. Let g_1, g_2 be two generators of a cyclic group $\mathbb{G}$ of prime order p. Let $\mathcal{X} = \mathbb{G}^{1\times 2}$ and $\mathcal{L} = \{(g_1^r, g_2^r) \in \mathcal{X} \mid r \in \mathbb{Z}_p\}$. The hashing key is $\mathsf{hk} = (\alpha_1, \alpha_2) \xleftarrow{\$} \mathbb{Z}_p^2$ and the associated projection key is $\mathsf{hp} = g_1^{\alpha_1} g_2^{\alpha_2}$. For any element $C = (u_1, u_2) \in \mathcal{X}$, the hash value under hk is $\mathsf{hv} = u_1^{\alpha_1} u_2^{\alpha_2}$.

- Choose $\widetilde{r} = (\beta_1, \beta_2) \xleftarrow{\$} \mathbb{Z}_p^2$, and compute $\widetilde{\mathsf{hp}} = \mathsf{hp} \cdot (g_1^{\beta_1} g_2^{\beta_2}) = g_1^{\alpha_1+\beta_1} g_2^{\alpha_2+\beta_2}$. $\widetilde{\mathsf{hp}}$ is independent from hp and its associated hashing key is $\widetilde{\mathsf{hk}} = (\alpha_1+\beta_1, \alpha_2+\beta_2)$. The hash value of element C under $\widetilde{\mathsf{hk}}$ is $\widetilde{\mathsf{hv}} = u_1^{\alpha_1+\beta_1} u_2^{\alpha_2+\beta_2} = \mathsf{hv} \cdot u_1^{\beta_1} u_2^{\beta_2}$, and hence the hash value difference is computable using $\widetilde{r}$.
- Choose $\widetilde{w} = \eta \xleftarrow{\$} \mathbb{Z}_p$ and compute $\widetilde{C} = (u_1 g_1^{\eta}, u_2 g_2^{\eta})$. The hash value of $\widetilde{C}$ under hk is $\widetilde{\mathsf{hv}} = (u_1 g_1^{\eta})^{\alpha_1} (u_2 g_2^{\eta})^{\alpha_2} = \mathsf{hv} \cdot (\mathsf{hp})^{\eta}$, and hence the hash value difference is computable using $\widetilde{w}$ (with hp). One can easily verify that $\widetilde{C} \in \mathcal{L} \iff C \in \mathcal{L}$.

More Constructions of Malleable SPHFs. To illustrate the feasibility of our new notion, we propose a generic construction of malleable SPHFs based on graded rings [9], which could be viewed as a common formalization for cyclic groups, bilinear groups, and multilinear groups. We rigorously prove that under some conditions, graded ring implies malleable SPHFs. Particularly, we rely on Katz and Vaikuntanathan [17] type SPHFs (KV-SPHF) where the projection key is independent from the element, as in many cases the linkability between the projection key and the element would make it difficult for a CRF to resist exfiltration and meanwhile maintain functionality. We will make this point clearer in

our CRF constructions. We then provide a malleable SPHF instantiation of our generic framework from the k-linear assumption.

Generic CRF Constructions via Malleable SPHFs. We show how to generically construct CRFs via malleable SPHFs for some widely used protocols. Essentially, our CRF constructions rely on the *key indistinguishability* and the *element indistinguishability* properties of the underlying malleable SPHF for the security preservation and exfiltration resistance, and rely on the *projection consistency*, *rerandomization consistency* and *membership preservation* of the malleable SPHF for the functionality maintaining.

Message Transmission Protocol. We first show as a warm up CRF constructions for the unkeyed message-transmission protocol. That is, both the sender and receiver have neither a shared secret key nor each other's public key. We remark that our framework can be seen as a generic construction of semantically secure public-key encryption scheme (with trusted setup) that is both key malleable and re-randomizable defined in [14], and hence provides a more intuitive way to build two-round message-transmission protocols with CRFs. The idea we illustrate via this simple protocol acts as a steppingstone toward other more complicated protocols.

Oblivious Signature-Based Envelope Protocol. We also study the CRF constructions for another useful protocol, namely Oblivious Signature-Based Envelope (OSBE), which was proposed by Li, Du and Boneh [20] and later enhanced by Blazy, Pointcheval and Vergnaud [11]. An OSBE protocol allows a user Alice to send an envelope, which encapsulates her private message, to another user Bob in such a way that Bob will be able to recover the private message if and only if Bob has possessed a credential, e.g., a signature on an agreed-upon message from the certification authority. OSBE has been found useful in a growing number of protocols and applications such as Secret Handshakes [5] and Password-Based Authenticated Key-Exchange [15]. We show that the SPHF-based construction of OSBE in [11] is CRF-ready if the underlying SPHF is malleable. Surprisingly, we find that their proposed OSBE instantiation from linear encryption of Waters signature [25] could be extended to be malleable for the CRF instantiations. One should note that the extension does not strictly follow the aforementioned generic framework of constructing malleable SPHF from graded rings. This also shows more possibilities for constructing malleable SPHFs.

CRF Constructions for Oblivious Transfer Protocol. Another major contribution of our work is the CRF construction for the oblivious transfer (OT) protocol, which has been widely adopted as a basic tool by many cryptographic systems. Although our CRF constructions are inspired by our generic framework of malleable SPHF from graded rings, there is some substantive difference between them.

In this work, we start with the OT framework of Halevi and Kalai [16], which relies on a special SPHF. The basic idea is that: (1) the receiver picks and sends to the sender two elements $C_b \in \mathcal{L}, C_{1-b} \in \mathcal{X} \backslash \mathcal{L}$ ($b \in \{0,1\}$ is the choice bit);

(2) the sender generates two hashing key pairs and computes the hash values of C_0 and C_1 (using the secret hashing keys) to conceal its two message M_0 and M_1 respectively, and then sends the two concealed messages with projection keys to the receiver; (3) the receiver recovers M_b by computing the hash value of C_b (using the projection key with the witness to the fact $C_b \in \mathcal{L}$). Noting that a malicious receiver might choose both C_b and C_{1-b} from the language $\mathcal{L}$, the underlying SPHF is required to be verifiably smooth such that the sender can verify at least one of (C_0, C_1) is not in the language.

Difficulties. It seems that we could extend the underlying SPHF of the HK-OT construction to be malleable so that the framework could admit CRFs. However, we found that it is actually not the case and the extension is not trivial at all.

- *The required* SPHF *here is not a classical one as it must be verifiably smooth.* Under the HK-OT framework, this is usually guaranteed by the verifiable linkability between C_0 and C_1 chosen by the receiver. However, a tampered implementation of the receiver may leak secret information to the outside world via the linkability. A desirable CRF for the receiver should be able to rerandomize (C_0, C_1) to a uniform tuple $(\widetilde{C_0}, \widetilde{C_1})$ to resist exfiltration. However, the rerandomization would break the linkability of the tuple and lead to protocol failure.
- *The receiver freshly generates the element basis underlying the* SPHF *at the beginning of each protocol session, which means we have to deal with an untrusted setup.* Since the element basis (e.g., $g_1, g_2 \in \mathbb{G}$ for the DDH tuple generation) is chosen by the receiver per session, a tampered receiver may maliciously choose some "bad" basis in order to compromise the security or leak secret information to the outside. Therefore, the CRF should be able to rerandomize the element basis to preserve security and resist exfiltration, while still maintain the protocol functionality. This, unfortunately, could not be trivially realized by the malleable SPHF.

Our Solution. In order to resolve the problem, we first propose a special OT construction from graded rings. Particularly, the receiver sends to the sender only one element, based on which the sender could generate an element pair so that the verifiable smoothness can be guaranteed by the sender itself. We then propose CRF constructions for such an OT protocol. Our central idea mainly follows the generic framework of malleable SPHF from graded rings except that we require the receiver's CRF could also rerandomize the element basis chosen by the receiver. We show that the CRF could still achieve all the properties when the transformation matrix for rerandomizing the element basis meets some requirements. The modified semi-generic framework narrows the possible instantiations of the HK-OT framework. However, we show that the CRF construction following our framework not only captures the prior work [21], which is the only known OT-CRF to date, but also can yield new constructions under weaker assumptions. In particular, we present new CRF constructions based on the k-linear assumption, which is weaker than the DDH assumption underlying the OT-CRF construction in [21].

1.2 Related Work

Comparisons with Other SPHF Variants. SPHF was originally introduced by Cramer and Shoup [12]. Since its introduction, it has been widely used for constructions of many cryptographic primitives, including authenticated key exchange [15,17], oblivious transfer [16], zero-knowledge arguments [1,2,9] and so on. Here we mainly introduce the work that are closely related to our notion of malleable SPHF. Hoeteck Wee defined a notion of homomorphic SPHF for achieving key-dependent message security [26]. That is, the combination of hash values of two elements equal to the hash value of the combination of these two elements. One may note that their notion is somewhat similar to the sub-property of *rerandomization consistency* captured by the element re-randomizability of our malleable SPHF. However, their definition is solely based on the secret hashing key while ours uses the projection key to calculate the hash value difference. We should clarify that our defined property is not always the case especially for those SPHFs where the projection key depends on the element. Yang et al. [27] introduced the notion of updatable hash proof system (UHPS) for constructing public key encryption schemes that are secure against continuous memory attacks. The UHPS requires that the secret hashing key could be updated homomorphically. In fact, they mainly consider a special case in which a secret hashing key can be freshly updated while the associated projection key keeps the same.

Other CRF Constructions. Mironov and Stephens-Davidowitz [21] showed how to construct CRFs for a 1-out-of-2 oblivious protocol based on the DDH assumption and also proposed a protocol for private function evaluation. They also provided a generic way to prevent a tampered machine from leaking information to an eavesdropper via any protocol. Ateniese, Magri, and Venturi [4] continued the study on signatures and constructed the CRF to protect signatures schemes against algorithm substitution attacks. Recently, Dodis, Mironov and Stephens-Davidowitz [14] considered CRF constructions for message-transmission protocols. They proposed a rich collection of solutions that vary in efficiency, security, and setup assumptions in the classical setting. It is worth noting that the studied message-transmission protocol in our work belongs to the so-called unkeyed setting in their work. Our framework can be viewed as a generic construction of the semantically secure public-key encryption scheme (with a trusted setup) that is both key malleable and re-randomizable defined in [14].

2 Preliminaries

2.1 Cryptographic Reverse Firewalls

In general, a cryptographic protocol $\mathcal{P}$ must satisfy functionality (i.e., correctness) requirement $\mathcal{F}$, which places constraints on the output of the parties executing $\mathcal{P}$ for particular input, and security requirement $\mathcal{S}$, which places constraints on the message distribution conditioned on specific input. Below we briefly recall the definition of reverse firewalls from [21]. We refer the reader to [21] for more detailed discussions.

Definition 1 (Cryptographic Reverse Firewall (CRF)). *A cryptographic reverse firewall is a stateful algorithm $\mathcal{W}$ that takes as input its state and a message and outputs an updated state and message. For simplicity, we do not write the state of $\mathcal{W}$ explicitly. For a party P and reverse firewall $\mathcal{W}$, we define $\mathcal{W} \circ \mathsf{P}$ as the "composed" party where $\mathcal{W}$ is applied to the incoming and outgoing messages of P. When the composed party engages in a protocol, the state of $\mathcal{W}$ is initialized to the public parameters. If $\mathcal{W}$ is meant to be composed with a party P, we call it a reverse firewall for P.*

One should note that $\mathcal{W}$ has access to all public parameters, but not the private input or the output of P. In reality, $\mathcal{W}$ can be regarded as an "active router" that sits at the boundary between P's private network and the outside world and modifies the messages that P sends and receives. The party P of course does not want a reverse firewall to ruin its protocol's functionality when its internal implementation is correct. Following [21] we require that reverse firewalls should be "stackable", which means the composition of multiple reverse firewalls $\mathcal{W} \circ \mathcal{W} \circ \cdots \circ \mathcal{W} \circ \mathsf{P}$ should still maintain the functionality of the protocol. The following definition captures this property.

Definition 2 (Functionality-maintaining CRFs). *For any reverse firewall $\mathcal{W}$ and any party P, let $\mathcal{W}^1 \circ \mathsf{P} = \mathcal{W} \circ \mathsf{P}$, and for $k \geq 2$, let $\mathcal{W}^k \circ \mathsf{P} = \mathcal{W} \circ (\mathcal{W}^{k-1} \circ \mathsf{P})$. For a protocol $\mathcal{P}$ that satisfies some functionality requirements $\mathcal{F}$, we say that a reverse firewall $\mathcal{W}$ maintains $\mathcal{F}$ for P in $\mathcal{P}$ if $\mathcal{W}^k \circ \mathsf{P}$ maintains $\mathcal{F}$ for P in $\mathcal{P}$ for any polynomial bounded $k \geq 1$. When $\mathcal{F}, \mathsf{P}, \mathcal{P}$ are clear, we simply say that $\mathcal{W}$ maintains functionality.*

Following the notations in [21], we use $\overline{\mathsf{P}}$ to represent arbitrary adversarial implementations of party P and $\widehat{\mathsf{P}}$ to represent the functionality-maintaining adversarial implementations. For a protocol $\mathcal{P}$ with party P, we write $\mathcal{P}_{\mathsf{P} \to \widehat{\mathsf{P}}}$ to represent the protocol where the role of party P is replaced by party $\widehat{\mathsf{P}}$.

A reverse firewall should also preserve the security of the underlying protocol, even in the presence of compromise. The strongest notion requires that the protocol in which party P is replaced with $\mathcal{W} \circ \overline{\mathsf{P}}$ for an arbitrarily corrupted party $\overline{\mathsf{P}}$ still preserves the security while the weaker notion only considers tampered implementations that maintain functionality. The below definition captures this property.

Definition 3 (Security-preserving CRFs). *For a protocol $\mathcal{P}$ that satisfies some security requirements $\mathcal{S}$ and functionality $\mathcal{F}$ and a reverse firewall $\mathcal{W}$,*

- *$\mathcal{W}$ strongly preserves $\mathcal{S}$ for P in $\mathcal{P}$ if the protocol $\mathcal{P}_{\mathsf{P} \to \mathcal{W} \circ \overline{\mathsf{P}}}$ satisfies $\mathcal{S}$; and*
- *$\mathcal{W}$ weakly preserves $\mathcal{S}$ for P in $\mathcal{P}$ if the protocol $\mathcal{P}_{\mathsf{P} \to \mathcal{W} \circ \widehat{\mathsf{P}}}$ satisfies $\mathcal{S}$.*

When $\mathcal{P}, \mathcal{F}, \mathcal{S}, \mathsf{P}$ are clear, we simple say that $\mathcal{W}$ strongly preserves security or weakly preserves security.

As introduced in [21], we also need the notion of exfiltration resistance. Intuitively, a reverse firewall is exfiltration resistant if "no corrupted implementation

of P can leak information through the firewall." We define this notion using the game LEAK which is presented in Fig. 1. Intuitively, the game asks the adversary to distinguish between a tampered implementation and an honest implementation. An exfiltration-resistant reverse firewall therefore prevents an adversary from even learning whether a party has been compromised, let alone leaking information.

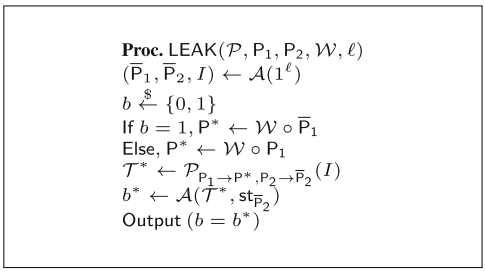

Fig. 1. $\mathsf{LEAK}(\mathcal{P}, \mathsf{P}_1, \mathsf{P}_2, \mathcal{W}, \ell)$, the exfiltration resistance security game for a reverse firewall $\mathcal{W}$ for party P_1 in protocol $\mathcal{P}$ against party P_2. $\mathcal{A}$ is the adversary, ℓ the security parameter, $\mathsf{st}_{\overline{\mathsf{P}}_2}$ the state of $\overline{\mathsf{P}}_2$ after the run of the protocol, I valid input for $\mathcal{P}$, and $\mathcal{T}^*$ is the transcript of running protocol $\mathcal{P}_{\mathsf{P}_1 \to \mathsf{P}^*, \mathsf{P}_2 \to \overline{\mathsf{P}}_2}(I)$.

The advantage of any adversary $\mathcal{A}$ in the game LEAK is defined as

$$\mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell) = \Pr[\mathsf{LEAK}(\mathcal{P}, \mathsf{P}_1, \mathsf{P}_2, \mathcal{W}, \ell) = 1] - 1/2.$$

Definition 4 (Exfiltration-resistant CRFs). *For a protocol $\mathcal{P}$ that satisfies functionality $\mathcal{F}$ and a reverse firewall $\mathcal{W}$,*

- *$\mathcal{W}$ is strongly exfiltration-resistant for party P_1 against party P_2 in protocol $\mathcal{P}$ if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$ is negligible in the security parameter ℓ; and*
- *$\mathcal{W}$ is weakly exfiltration-resistant for party P_1 against party P_2 in protocol $\mathcal{P}$, if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$ is negligible in the security parameter ℓ provided that $\overline{\mathsf{P}}_1$ maintains $\mathcal{F}$ for P_1.*

When $\mathcal{P}, \mathcal{F}, \mathsf{P}_1$ are clear, we simple say that $\mathcal{W}$ is strongly exfiltration-resistant against P_2 or weakly exfiltration-resistant against P_2. In the special case when P_2 is empty, we say that $\mathcal{W}$ is exfiltration-resistant against eavesdroppers.

2.2 Smooth Projective Hash Function

An SPHF is based on a domain $\mathcal{X}$ and an $\mathcal{NP}$ language $\mathcal{L}$, where $\mathcal{L}$ contains a subset of the elements of the domain $\mathcal{X}$, i.e., $\mathcal{L} \subset \mathcal{X}$. An SPHF system over a language $\mathcal{L} \subset \mathcal{X}$, onto a set $\mathcal{Y}$, is defined by the following five algorithms ($\mathsf{SPHFSetup}$, HashKG, ProjKG, Hash, $\mathsf{ProjHash}$):

- $\mathsf{SPHFSetup}(1^\ell)$: The $\mathsf{SPHFSetup}$ algorithm takes as input a security parameter ℓ, generates the *global parameters* param and the description of an $\mathcal{NP}$ language $\mathcal{L}$, outputs $\mathsf{pp} = (\mathcal{L}, \mathsf{param})$ as the public parameter.
- $\mathsf{HashKG}(\mathsf{pp})$: The HashKG algorithm generates a *hashing key* hk;
- $\mathsf{ProjKG}(\mathsf{pp}, \mathsf{hk}, C)$: The ProjKG algorithm derives the *projection key* hp from the hashing key hk and possibly an element C;
- $\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C)$: The Hash algorithm takes as input an element C and the hashing key hk, outputs the hash value $\mathsf{hv} \in \mathcal{Y}$;
- $\mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, w)$: The $\mathsf{ProjHash}$ algorithm takes as input the projection key hp and an element C with the witness w to the fact that $C \in \mathcal{L}$, outputs the hash value $\mathsf{hv} \in \mathcal{Y}$.

SPHFs could be classified into two types according to whether ProjKG takes an element as input. The Gennaro and Lindell [15] type (GL-SPHF) allows hp to depend on C while the Katz and Vaikuntanathan [17] type (KV-SPHF) does not. As shown later, our proposed new SPHF falls in the KV-SPHF category.

An SPHF should satisfy the following two properties.

Correctness. Formally, for any element $C \in \mathcal{L}$ with w the witness, we have

$$\Pr\left[\mathsf{hv} \neq \mathsf{hv}' : \begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{SPHFSetup}(1^\ell); \\ \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{pp}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{pp}, \mathsf{hk}); \\ \mathsf{hv} \leftarrow \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C); \\ \mathsf{hv}' \leftarrow \mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, w) \end{array}\right] \leq \mathsf{negl}(\ell).$$

Smoothness. For any $C \in \mathcal{X} \backslash \mathcal{L}$, the following two distributions are statistically indistinguishable,

$$\mathcal{V}_1 = \{(\mathsf{pp}, C, \mathsf{hp}, \mathsf{hv}) | \mathsf{hv} = \mathsf{Hash}(\mathsf{hk}, C')\}, \mathcal{V}_2 = \{(\mathsf{pp}, C, \mathsf{hp}, \mathsf{hv}) | \mathsf{hv} \xleftarrow{\$} \mathcal{Y}\}.$$

That is, $\mathsf{Adv}^{\mathsf{smooth}}_{\mathsf{SPHF}}(\ell) = \sum_{v \in \mathcal{Y}} |\Pr_{\mathcal{V}_1}[\mathsf{hv} = v] - \Pr_{\mathcal{V}_2}[\mathsf{hv} = v]| \leq \mathsf{negl}(\ell)$.

It is required that one could efficiently sample elements from the set $\mathcal{X}$. That is, one could run a polynomial time algorithm $\mathsf{SampYes}(\mathsf{pp})$ to sample an element (C, w) from $\mathcal{L}$ where w is the witness to the membership $C \in \mathcal{L}$ and another polynomial time algorithm $\mathsf{SampNo}(\mathsf{pp})$ to sample an element C from $\mathcal{X} \backslash \mathcal{L}$. The subset membership problem between $\mathcal{L}$ and $\mathcal{X}$ is usually required to be difficult, which is defined as follows.

Definition 5 (Hard Subset Membership Problem). *The subset membership problem* (SMP) *is hard on* $(\mathcal{X}, \mathcal{L})$ *for an* SPHF *that consists of* (SPHFSetup, HashKG, ProjKG, Hash, ProjHash)*, if for any PPT adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{SMP}}_{\mathcal{A},\mathsf{SPHF}}(\ell) = \Pr\left[b' = b : \begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{SPHFSetup}(1^\ell); \\ \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{pp}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{pp}, \mathsf{hk}); \\ b \xleftarrow{\$} \{0,1\}; (C_0, w) \xleftarrow{\$} \mathsf{SampYes}(\mathsf{pp}); \\ C_1 \xleftarrow{\$} \mathsf{SampNo}(\mathsf{pp}); \\ b' \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{hk}, \mathsf{hp}, C_b) \end{array}\right] - \frac{1}{2} \leq \mathsf{negl}(\ell).$$

3 Malleable Smooth Projective Hash Function

3.1 Definition

A malleable SPHF is defined by a tuple of algorithms ($\mathsf{SPHFSetup}$, HashKG, ProjKG, Hash, $\mathsf{ProjHash}$, MaulK, MaulH, ReranE, ReranH) which work as follows:

- $\mathsf{SPHFSetup}$, HashKG, ProjKG, Hash, $\mathsf{ProjHash}$ are the same as in the classical SPHF;
- $\mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}, \widetilde{r})$. The MaulK algorithm takes as input a projection key hp and randomness $\widetilde{r}$, outputs a new projection key $\widetilde{\mathsf{hp}}$;
- $\mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, \widetilde{r}, C)$. The MaulH algorithm takes as input a projection key hp, the randomness $\widetilde{r}$ and an element C, outputs the hash value $\widetilde{\mathsf{hv}}$;
- $\mathsf{ReranE}(\mathsf{pp}, C, \widetilde{w})$. The ReranE algorithm takes as input an element C and the randomness $\widetilde{w}$, outputs a new element $\widetilde{C}$;
- $\mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{w})$. The ReranH algorithm takes as input the projection key hp, an element C and the randomness $\widetilde{w}$, outputs the hash value $\widetilde{\mathsf{hv}}$;

We describe two randomness sampling algorithms named SampR and SampW. One could run $\mathsf{SampR}(\mathsf{pp})$ to sample $\widetilde{r}$ from the distribution of randomness using which we generate the hashing key. The algorithm $\mathsf{SampW}(\mathsf{pp})$ can be used to sample $\widetilde{w}$ from the witness distribution of the language.

Now we are ready to describe the properties of a malleable SPHF. In addition to the properties captured by a classical SPHF, a malleable SPHF also satisfies the following new properties which are essential in our constructions of CRFs.

Definition 6 (Projection Key Malleability). *A smooth projective hash function is projection key-malleable if the following properties hold.*

- ***Key Indistinguishability.*** *For any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\mathsf{Adv}^{\mathit{Key\text{-}Ind}}_{\mathcal{A},\mathsf{MSPHF}}(\ell) = \Pr\left[b' = b : \begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{SPHFSetup}(1^\ell); \\ (\mathsf{hp}_1, \mathsf{hp}_2, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pp}); \\ b \xleftarrow{\$} \{0,1\}; \widetilde{r} \xleftarrow{\$} \mathsf{SampR}(\mathsf{pp}); \\ \widetilde{\mathsf{hp}} \leftarrow \mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}_b, \widetilde{r}); \\ b' \leftarrow \mathcal{A}_2(\mathsf{pp}, \mathsf{st}, \mathsf{hp}_1, \mathsf{hp}_2, \widetilde{\mathsf{hp}}) \end{array}\right] - \frac{1}{2} \leq \mathsf{negl}(\ell).$$

- ***Projection Consistency.*** *For any element* $C \in \mathcal{X}$,

$$\Pr\left[\mathsf{hv} \neq \mathsf{hv}' : \begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{SPHFSetup}(1^\ell); \\ \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{pp}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{pp}, \mathsf{hk}); \\ \widetilde{r} \xleftarrow{\$} \mathsf{SampR}(\mathsf{pp}); \widetilde{\mathsf{hp}} \leftarrow \mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}, \widetilde{r}); \\ \mathsf{hv} \leftarrow \mathsf{Hash}(\mathsf{pp}, \widetilde{\mathsf{hk}}, C); \\ \widetilde{\mathsf{hv}} \leftarrow \mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, \widetilde{r}, C); \\ \mathsf{hv}' \leftarrow \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) * \widetilde{\mathsf{hv}} \end{array}\right] \leq \mathsf{negl}(\ell).$$

  where $\widetilde{\mathsf{hk}}$ *is the associated hashing key of* $\widetilde{\mathsf{hp}} \leftarrow \mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}, \widetilde{r})$ *and* $*$ *denotes the operation between two hash values in* $\mathcal{Y}$.

Definition 7 (Element Re-randomizability). *A smooth projective hash function is element-rerandomizable if the followings hold.*

– ***Element Indistinguishability.*** *For any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\mathsf{Adv}^{\textit{Element-Ind}}_{\mathcal{A},\mathsf{MSPHF}}(\ell) = \Pr\left[b' = b : \begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{SPHFSetup}(1^\ell); \\ (C_1, C_2, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pp}); \\ b \xleftarrow{\$} \{0,1\}; \widetilde{w} \xleftarrow{\$} \mathsf{SampW}(\mathsf{pp}); \\ \widetilde{C} \leftarrow \mathsf{ReranE}(\mathsf{pp}, C_b, \widetilde{w}); \\ b' \leftarrow \mathcal{A}_2(\mathsf{pp}, \mathsf{st}, C_1, C_2, \widetilde{C}) \end{array}\right] - \frac{1}{2} \leq \mathsf{negl}(\ell).$$

– ***Rerandomization Consistency.*** *For any element* $C \in \mathcal{X}$,

$$\Pr\left[\mathsf{hv} \neq \mathsf{hv}' : \begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{SPHFSetup}(1^\ell); \\ \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{pp}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{pp}, \mathsf{hk}); \\ \widetilde{w} \xleftarrow{\$} \mathsf{SampW}(\mathsf{pp}); \widetilde{C} \leftarrow \mathsf{ReranE}(\mathsf{pp}, C, \widetilde{w}); \\ \mathsf{hv} \leftarrow \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, \widetilde{C}); \\ \widetilde{\mathsf{hv}} \leftarrow \mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{w}); \\ \mathsf{hv}' \leftarrow \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) * \widetilde{\mathsf{hv}} \end{array}\right] \leq \mathsf{negl}(\ell).$$

– ***Membership Preservation.*** *For any element* $C \in \mathcal{X}$, *let* $\widetilde{C} \leftarrow \mathsf{ReranE}(\mathsf{pp}, C, \widetilde{w})$ *where* $\widetilde{w} \xleftarrow{\$} \mathsf{SampW}(\mathsf{pp})$, *we have* $\widetilde{C} \in \mathcal{L}$ *if and only if* $C \in \mathcal{L}$.

Definition 8 (Malleable SPHF). *An* SPHF *is malleable if it is projection key-malleable and element-rerandomizable.*

3.2 Malleable SPHFs from Graded Rings

In this section, we show that under some conditions, the SPHF framework from graded rings proposed by Benhamouda et al. [9] could be extended into malleable SPHF. The main goal of this part is to demonstrate the feasibility of our definition. We remark that malleable SPHFs can be constructed using other approaches.

Graded Rings. Benhamouda et al. [9] proposed a generic framework for SPHFs using a new notion named graded rings, which is a common formalization for cyclic groups, bilinear groups, and even multilinear groups. The graded ring provides a practical way to manipulate elements of various groups involved in pairings and more generally, in multi-linear maps. Before describing their SPHF framework, we briefly recall the notion of graded rings. The notation $\oplus$ and $\odot$ correspond to the addition operation and the multiplication operation, respectively. For simplicity, here we focus on cyclic groups and symmetric bilinear groups. Let $\mathbb{G}, \mathbb{G}_T$ be two multiplicative groups with the same prime order p with a symmetric bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

- For any $a, b \in \mathbb{Z}_p$, $a \oplus b = a + b$, $a \odot b = a \cdot b$;
- For any $u_1, v_1 \in \mathbb{G}$, $u_1 \oplus v_1 = u_1 \cdot v_1, u_1 \ominus v_1 = u_1 \cdot v_1^{-1}$, and for any $c \in \mathbb{Z}_p$, $c \odot u_1 = u_1^c$;
- For any $u_T, v_T \in \mathbb{G}_T$, $u_T \oplus v_T = u_T \cdot v_T, u_T \ominus v_T = u_T \cdot v_T^{-1}$, and for any $c \in \mathbb{Z}_p$, $c \odot u_T = u_T^c$;
- For any $u_1, v_1 \in \mathbb{G}$, $u_1 \odot v_1 = e(u_1, v_1) \in \mathbb{G}_T$.

That is, $\oplus$ and $\odot$ correspond to the addition and the multiplication of the exponents. The notations could be extended in a natural way when it comes to the case of vectors and matrices.

We are now ready to describe the framework of SPHF introduced in [9]. For a language $\mathcal{L}$ which is specified by the parameter aux, suppose there exist two positive integers m and n, a function $\Gamma : \mathcal{X} \longmapsto \mathbb{G}^{m \times n}$ (for generating the element basis) and a function $\Theta_{\mathsf{aux}} : \mathcal{X} \longmapsto \mathbb{G}^{1 \times n}$, such that for any element $C \in \mathcal{X}$,

$$(C \in \mathcal{L}) \Longleftrightarrow (\exists \boldsymbol{\lambda} \in \mathbb{Z}_p^{1 \times m} \text{ s.t., } \Theta_{\mathsf{aux}}(C) = \boldsymbol{\lambda} \odot \Gamma(C)).$$

In other words, $C \in \mathcal{L}$ if and only if $\Theta_{\mathsf{aux}}(C)$ is a linear combination of the rows in $\Gamma(C)$. Here it is required that the one who knows the witness w of the membership $C \in \mathcal{L}$ can efficiently compute the above linear combination $\boldsymbol{\lambda}$. This requirement seems somewhat strong but is actually verified by very expressive languages [9].

With the above notations, the hashing key in an SPHF is a vector $\mathsf{hk} := \boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)^\mathsf{T} \xleftarrow{\$} \mathbb{Z}_p^n$ and the projection key for an element C is $\mathsf{hp} := \boldsymbol{\gamma}(C) = \Gamma(C) \odot \boldsymbol{\alpha} \in \mathbb{G}^k$. Then the hash value computation for an element C is:

$$\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) := \Theta_{\mathsf{aux}}(C) \odot \boldsymbol{\alpha}, \qquad \mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, w) := \boldsymbol{\lambda} \odot \boldsymbol{\gamma}(C).$$

Intuitively, if $C \in \mathcal{L}$ with $\boldsymbol{\lambda}$, then we have,

$$\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) = \Theta_{\mathsf{aux}}(C) \odot \boldsymbol{\alpha} = \boldsymbol{\lambda} \odot \Gamma(C) \odot \boldsymbol{\alpha} = \boldsymbol{\lambda} \odot \boldsymbol{\gamma}(C) = \mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, w).$$

This guarantees the correctness of the SPHF. As for the smoothness property, we can see that for any element $C \notin \mathcal{L}$ and a projection key $\mathsf{hp} = \boldsymbol{\gamma}(C) = \Gamma(C) \odot \boldsymbol{\alpha}$, the vector $\Theta_{\mathsf{aux}}(C)$ is not in the linear span of $\Gamma(C)$, and thus its hash value $\mathsf{hv} = \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) = \Theta_{\mathsf{aux}}(C) \odot \boldsymbol{\alpha}$ is independent from $\mathsf{hp} = \Gamma(C) \odot \boldsymbol{\alpha}$. We refer the readers to [9] for a more detailed analysis. One can note that if the function $\Gamma : \mathcal{X} \longmapsto \mathbb{G}^{m \times n}$ is a constant function, the corresponding SPHF is of KV-SPHF type, otherwise it is of GL-SPHF type.

A Simple Example. We illustrate this framework for the DDH language. Let g_1, g_2 be two generators of a cyclic group $\mathbb{G}$ of prime order p. Let $\mathcal{X} = \mathbb{G}^{1 \times 2}$ and $\mathcal{L} = \{(u_1, u_2) \mid r \in \mathbb{Z}_p, \text{ s.t., } u_1 = g_1^r, u_2 = g_2^r\}$. For any $C = (u_1, u_2) \in \mathcal{L}$, $\Theta_{\mathsf{aux}}(C) = C$, $\Gamma(C) = (g_1, g_2)$ and the witness for $C \in \mathcal{L}$ is $w = r$ and here $\boldsymbol{\lambda} = w = r$. The hashing key is $\mathsf{hk} = \boldsymbol{\alpha} = (\alpha_1, \alpha_2)^\mathsf{T} \xleftarrow{\$} \mathbb{Z}_p^2$ and the projection key is $\mathsf{hp} = \boldsymbol{\gamma}(C) = \Gamma(C) \odot \boldsymbol{\alpha} = g_1^{\alpha_1} g_2^{\alpha_2} \in \mathbb{G}$. We then have

$$\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) = \Theta_{\mathsf{aux}}(C) \odot \boldsymbol{\alpha} = (u_1, u_2) \odot (\alpha_1, \alpha_2)^\mathsf{T} = u_1^{\alpha_1} u_2^{\alpha_2},$$

$$\mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, w = r) = \boldsymbol{\lambda} \odot \boldsymbol{\gamma}(C) = r \odot (g_1^{\alpha_1} g_2^{\alpha_2}) = (g_1^{\alpha_1} g_2^{\alpha_2})^r.$$

This is exactly the original SPHF of Cramer and Shoup for the DDH language in [12].

Generic Construction of Malleable SPHFs. With the above definitions, we present a generic framework for constructing malleable SPHF based on graded rings.

- $\mathsf{SPHFSetup}(1^\ell)$. Output pp which defines the set $\mathcal{X}$ and the language $\mathcal{L}$ with the positive integers m and n, and functions Γ and Θ_{aux}.
- $\mathsf{HashKG}(\mathsf{pp})$. Sample $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^n$ and output $\mathsf{hk} = \boldsymbol{\alpha}$.
- $\mathsf{ProjKG}(\mathsf{pp}, \mathsf{hk}, C)$. Output $\mathsf{hp} = \boldsymbol{\gamma}(C) = \Gamma(C) \odot \boldsymbol{\alpha} \in \mathbb{G}^k$.
- $\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C)$. Output $\mathsf{hv} = \Theta_{\mathsf{aux}}(C) \odot \boldsymbol{\alpha}$.
- $\mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, w)$. Output $\mathsf{hv} = \boldsymbol{\lambda} \odot \boldsymbol{\gamma}(C)$ where $\boldsymbol{\lambda}$ is derived from w.
- $\mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}, \widetilde{\boldsymbol{r}})$. To re-randomize a projection key $\mathsf{hp} = \boldsymbol{\gamma}(C)$ using the randomness $\widetilde{r}$, compute and output $\widetilde{\mathsf{hp}}$ as:

$$\Delta\mathsf{hp} = \Gamma(C) \odot \widetilde{\boldsymbol{r}}, \qquad \widetilde{\mathsf{hp}} = \boldsymbol{\gamma}(C) \oplus \Delta\mathsf{hp}.$$

- $\mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, \widetilde{\boldsymbol{r}}, C)$. Output $\widetilde{\mathsf{hv}} = \Theta_{\mathsf{aux}}(C) \odot \widetilde{\boldsymbol{r}}$.
- $\mathsf{ReranE}(\mathsf{pp}, C, \widetilde{\boldsymbol{w}})$. To re-randomize an element C using the random witness $\widetilde{\boldsymbol{w}}$, derive $\widetilde{\boldsymbol{\lambda}}$ from $\widetilde{\boldsymbol{w}}$, compute and output $\widetilde{C}$ as:

$$\Delta C = \widetilde{\boldsymbol{\lambda}} \odot \Gamma(C), \qquad \widetilde{C} = \Theta_{\mathsf{aux}}(C) \oplus \Delta C.$$

- $\mathsf{ReranH}(\mathsf{PP}, \mathsf{hp}, C, \widetilde{\boldsymbol{w}})$. Derive $\widetilde{\boldsymbol{\lambda}}$ from $\widetilde{\boldsymbol{w}}$ and output $\widetilde{\mathsf{hv}} = \widetilde{\boldsymbol{\lambda}} \odot \boldsymbol{\gamma}(C)$.

For the above construction, we have the following theorem.

Theorem 1. *The above generic construction is a malleable smooth projective hash function if the following conditions hold:*

a. $\Theta : \mathcal{X} \longmapsto \mathbb{G}^{1\times n}$ *is an identity function; (Diverse Group [12])*
b. $\Gamma : \mathcal{X} \longmapsto \mathbb{G}^{k\times n}$ *is a constant function; (KV-SPHF type)*
c. The subset membership problem between $\mathcal{L}$ *and* $\mathcal{X}$ *is hard.*

Proof. It should be clear that the construction is an SPHF as it is exactly the graded ring-based SPHF framework proposed in [9]. Below we show that it is *projection key-malleable* and *element-rerandomizable.*

Projection Key Malleability. For any $\widetilde{\boldsymbol{r}} = (r_1, ..., r_n)^\mathsf{T} \xleftarrow{\$} \mathsf{SampR}(\mathsf{pp})$, any element $C \in \mathcal{X}$, we have that

$$\begin{aligned}\mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}, \widetilde{\boldsymbol{r}}) &= \boldsymbol{\gamma}(C) \oplus (\Gamma(C) \odot \widetilde{\boldsymbol{r}}) \\ &= \Gamma(C) \odot \boldsymbol{\alpha} \oplus (\Gamma(C) \odot \widetilde{\boldsymbol{r}}) \\ &= \Gamma(C) \odot (\boldsymbol{\alpha} \oplus \widetilde{\boldsymbol{r}}) = \widetilde{\mathsf{hp}}.\end{aligned}$$

One can easily notice that the new projection key $\widetilde{\mathsf{hp}}$ is independent of hp, as the randomness $\widetilde{\boldsymbol{r}}$ is uniformly chosen and Γ is a constant function. Therefore, for any PPT adversary $\mathcal{A}$, we have that $\mathsf{Adv}^{\mathsf{Key\text{-}Ind}}_{\mathcal{A},\mathsf{MSPHF}}(\ell)$ is negligible. Moreover, the associated hashing key of $\widetilde{\mathsf{hp}}$ is $\widetilde{\mathsf{hk}} = \widetilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} \oplus \widetilde{\boldsymbol{r}} = (\alpha_1 + r_1, ..., \alpha_n + r_n)^{\mathsf{T}} \in \mathbb{Z}_p^n$. Therefore, we have

$$\begin{aligned}\mathsf{Hash}(\mathsf{pp}, \widetilde{hk}, C) &= \Theta_{\mathsf{aux}}(C) \odot \widetilde{\boldsymbol{\alpha}} = \Theta_{\mathsf{aux}}(C) \odot (\boldsymbol{\alpha} \oplus \widetilde{\boldsymbol{r}}) \\ &= \Theta_{\mathsf{aux}}(C) \odot \boldsymbol{\alpha} \oplus \Theta_{\mathsf{aux}}(C) \odot \widetilde{\boldsymbol{r}} \\ &= \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) \oplus \mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, \widetilde{\boldsymbol{r}}, C).\end{aligned}$$

This shows the projection consistency and thus the projection key is malleable.

Element Re-randomizability. For any randomness $\widetilde{\boldsymbol{w}}$, and any element $C \in \mathcal{X}$, we have that, $\mathsf{ReranE}(\mathsf{pp}, C, \widetilde{\boldsymbol{w}}) = \Theta_{\mathsf{aux}}(C) \oplus (\widetilde{\boldsymbol{\lambda}} \odot \Gamma(C)) = \widetilde{C}$. Due to the uniformly chosen randomness $\widetilde{\boldsymbol{w}}$ (which derives $\widetilde{\boldsymbol{\lambda}}$) and the hard subset membership problem, we have that $\widetilde{C}$ is computationally independent of C. Particularly, $\widetilde{\boldsymbol{\lambda}} \odot \Gamma(C)$ could be viewed as a random chosen element from $\mathcal{L}$ as Γ is a constant function (i.e., $\Gamma(C) = \Gamma(\widetilde{C})$). Therefore, for any PPT adversary $\mathcal{A}$, if $\mathsf{Adv}^{\mathsf{Element\text{-}Ind}}_{\mathcal{A},\mathsf{MSPHF}}(\ell)$ is non-negligible, we could use $\mathcal{A}$ to break the hard subset membership problem, which is a contradiction. Noting that here we require Θ to be an identity function, i.e., $\Theta_{\mathsf{aux}}(\widetilde{C}) = \widetilde{C}$, we have

$$\begin{aligned}\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, \widetilde{C}) &= \Theta_{\mathsf{aux}}(\widetilde{C}) \odot \boldsymbol{\alpha} = \widetilde{C} \odot \boldsymbol{\alpha} \\ &= (\Theta_{\mathsf{aux}}(C) \oplus \widetilde{\boldsymbol{\lambda}} \odot \Gamma(C))) \odot \boldsymbol{\alpha} \\ &= \Theta_{\mathsf{aux}}(C) \odot \boldsymbol{\alpha} \oplus \widetilde{\boldsymbol{\lambda}} \odot \Gamma(C) \odot \boldsymbol{\alpha} \\ &= \Theta_{\mathsf{aux}}(C) \odot \boldsymbol{\alpha} \oplus \widetilde{\boldsymbol{\lambda}} \odot \boldsymbol{\gamma}(C) \\ &= \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) \oplus \mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{\boldsymbol{w}}).\end{aligned}$$

The above illustrates the *rerandomization consistency*. Below we show that the element rerandomization is also *membership-preserving*. Given any element $C \in \mathcal{L}$ with the witness $C = \boldsymbol{\lambda}$, for any randomness $\widetilde{\boldsymbol{w}}$ that derives $\widetilde{\boldsymbol{\lambda}}$, we have that,

$$\begin{aligned}\mathsf{ReranE}(\mathsf{pp}, C, \widetilde{\boldsymbol{w}}) &= \Theta_{\mathsf{aux}}(C) \oplus (\widetilde{\boldsymbol{\lambda}} \odot \Gamma(C)) \\ &= \boldsymbol{\lambda} \odot \Gamma(C) \oplus (\widetilde{\boldsymbol{\lambda}} \odot \Gamma(C)) \\ &= (\boldsymbol{\lambda} \oplus \widetilde{\boldsymbol{\lambda}}) \odot \Gamma(C) \\ &= \boldsymbol{\lambda}' \odot \Gamma(\widetilde{C}) = \Theta_{\mathsf{aux}}(\widetilde{C}) = \widetilde{C}.\end{aligned}$$

The above holds due to the fact that Θ is an identity function, i.e., $\Theta_{\mathsf{aux}}(\widetilde{C}) = \widetilde{C}$ and Γ is a constant function, i.e., $\Gamma(C) = \Gamma(\widetilde{C})$. The witness to the fact $\widetilde{C} \in \mathcal{L}$ is $\boldsymbol{\lambda}' = \boldsymbol{\lambda} \oplus \widetilde{\boldsymbol{\lambda}}$. For any element $C \in \mathcal{X} \backslash \mathcal{L}$, the vector $\Theta_{\mathsf{aux}}(C)$ is not in the linear span of $\Gamma(C)$. Therefore, for any $\widetilde{\boldsymbol{w}}$, let $\widetilde{C} = \mathsf{ReranE}(\mathsf{pp}, C, \widetilde{\boldsymbol{w}}) = \Theta_{\mathsf{aux}}(C) \oplus (\widetilde{\boldsymbol{\lambda}} \odot \Gamma(C))$, we trivially have that $\Theta_{\mathsf{aux}}(\widetilde{C}) = \widetilde{C}$ is not in the linear span of $\Gamma(C)$ and thus $\widetilde{C} \in \mathcal{X} \backslash \mathcal{L}$.

Instantiation from the k-Linear Assumption. We instantiate the above framework based on the k-Linear (k-Lin) assumption. Let $\mathbb{G}$ be a group with prime order p and g a generator. The k-Lin assumption asserts that $g_{k+1}^{r_1+\cdots+r_k}$ is pseudo-random given $g_1, \cdots, g_{k+1}, g_1^{r_1}, \cdots, g_k^{r_k}$ where $g_1, \cdots, g_{k+1} \xleftarrow{R} \mathbb{G}, r_1, \cdots, r_k \xleftarrow{R} \mathbb{Z}_p$. Note that the DDH assumption is equivalent to the 1-Lin assumption.

We show how to construct a malleable SPHF from k-Lin assumption. The language is defined as,

$$\mathcal{L} = \{(c_1, \cdots, c_k) | \exists (r_1, \cdots, r_k) \in \mathbb{Z}_p^k, \text{s.t.}, c_1 = g_1^{r_1}, \cdots, c_k = g_k^{r_k}, c_{k+1} = g_{k+1}^{\sum_{i=1}^{k} r_i})\}.$$

For any $C = (c_1, \cdots, c_{k+1})$, we have $\Theta_{\mathsf{aux}}(C) = C$ and

$$\Gamma(C) = \begin{pmatrix} g_1 & 1 & \cdots & 1 & g_{k+1} \\ 1 & g_2 & \cdots & 1 & g_{k+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & g_k & g_{k+1} \end{pmatrix} \in \mathbb{G}^{k \times (k+1)}.$$

For any $C \in \mathcal{L}$ with witness $\boldsymbol{\lambda} = \boldsymbol{w} = (r_1, \cdots, r_k)$, we have, $\Theta_{\mathsf{aux}}(C) = (g_1^{r_1}, \cdots, g_k^{r_k}, g_{k+1}^{\sum_{i=1}^{k} r_i}) = \boldsymbol{\lambda} \odot \Gamma(C)$. Let $\mathsf{pp} = (\mathbb{G}, p, g_1, \cdots, g_{k+1})$, $\widetilde{\boldsymbol{r}} = (\beta_1, \cdots, \beta_{k+1})^{\mathsf{T}}$ and $\widetilde{\boldsymbol{\lambda}} = \widetilde{\boldsymbol{w}} = (\eta_1, \cdots, \eta_k)$. The instantiation is as follows:

- $\mathsf{HashKG}(\mathsf{pp}) : \mathsf{hk} = \boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_{k+1})^{\mathsf{T}} \xleftarrow{\$} \mathbb{Z}_p^k$;
- $\mathsf{ProjKG}(\mathsf{pp}, \mathsf{hk}, C) : \mathsf{hp} = \boldsymbol{\gamma}(C) = \Gamma(C) \odot \boldsymbol{\alpha} = (g_1^{\alpha_1} g_{k+1}^{\alpha_{k+1}}, \cdots, g_k^{\alpha_k} g_{k+1}^{\alpha_{k+1}})^{\mathsf{T}}$;
- $\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) : \mathsf{hv} = (c_1, \cdots, c_{k+1}) \odot (\alpha_1, \cdots, \alpha_{k+1})^{\mathsf{T}} = \prod_{i=1}^{k} c_i^{\alpha_i}$;
- $\mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, \boldsymbol{w}) : \mathsf{hv} = \boldsymbol{\lambda} \odot \boldsymbol{\gamma}(C) = \prod_{i=1}^{k} (g_i^{\alpha_i} g_{k+1}^{\alpha_{k+1}})^{r_i}$;
- $\mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}, \widetilde{\boldsymbol{r}}) : \widetilde{\mathsf{hp}} = \boldsymbol{\gamma}(C) \oplus (\Gamma(C) \odot \widetilde{\boldsymbol{r}}) = (g_1^{\alpha_1} g_{k+1}^{\alpha_{k+1}}, \cdots, g_k^{\alpha_k} g_{k+1}^{\alpha_{k+1}})^{\mathsf{T}} \oplus (g_1^{\beta_1} g_{k+1}^{\beta_{k+1}}, \cdots, g_k^{\beta_k} g_{k+1}^{\beta_{k+1}})^{\mathsf{T}} = (g_1^{\alpha_1+\beta_1} g_{k+1}^{\alpha_{k+1}+\beta_{k+1}}, \cdots, g_k^{\alpha_k+\beta_k} g_{k+1}^{\alpha_{k+1}+\beta_{k+1}})^{\mathsf{T}}$;
- $\mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, \widetilde{\boldsymbol{r}}, C) : \widetilde{\mathsf{hv}} = \Theta_{\mathsf{aux}}(C) \odot \widetilde{\boldsymbol{r}} = (c_1, \cdots, c_{k+1}) \odot (\beta_1, \cdots, \beta_{k+1})^{\mathsf{T}} = c_1^{\beta_1} \cdot c_2^{\beta_2} \cdots c_{k+1}^{\beta_{k+1}} = \prod_{i=1}^{k+1} c_i^{\beta_i}$;
- $\mathsf{ReranE}(\mathsf{pp}, C, \widetilde{\boldsymbol{w}}) : \widetilde{C} = \Theta_{\mathsf{aux}}(C) \oplus (\widetilde{\boldsymbol{\lambda}} \odot \Gamma(C)) = (c_1 g_1^{\eta_1}, \cdots, c_k g_k^{\eta_k}, c_{k+1} g_{k+1}^{\sum_{i=1}^{k} \eta_i})$;
- $\mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{\boldsymbol{w}}) : \widetilde{\mathsf{hv}} = \widetilde{\boldsymbol{\lambda}} \odot \boldsymbol{\gamma}(C) = (\eta_1, \cdots, \eta_d) \odot (g_1^{\alpha_1} g_{k+1}^{\alpha_{k+1}}, \cdots, g_k^{\alpha_k} g_{k+1}^{\alpha_{k+1}})^{\mathsf{T}} = \prod_{i=1}^{k} (g_i^{\alpha_i} g_{k+1}^{\alpha_{k+1}})^{\eta_i}$.

It is easy to verify that the above instantiation is a malleable SPHF as it satisfies all the conditions of Theorem 1.

Remark. Note that the function Θ_{aux} is required to be an identity function in our framework. That is, the above generic construction is on diverse groups [12]. However, we remark that such a requirement is not necessary. We will show later (Sect. 4.2) a concrete malleable SPHF which demonstrates that instantiating malleable SPHF from graded rings can be done in different ways.

4 Generic Construction of CRFs via Malleable SPHFs

4.1 Warm-Up: Message-Transmission Protocol with CRFs

A message transmission protocol (MTP) enables one party, Alice, to securely communicate a message to another party, Bob. Here we focus on the unkeyed setting for message transmission. That is, both Alice and Bob have neither a shared secret key nor each other's public key. Specifically, the protocol does not assume a public-key infrastructure. It simply lets Bob send a randomly chosen public key as the first message and thereafter Alice sends an encryption of her message under Bob's public key as the second message. Since neither the sender nor the receiver can be authenticated in this setting, the strongest security guarantee is semantic security against passive adversaries. That is, the adversary should not be able to distinguish the protocol transcripts for transferring two different plaintexts which are chosen by the adversary. We remark that our framework can be seen as a generic construction of semantically secure public-key encryption that is both key malleable and re-randomizable defined in [14], and hence provides a more intuitive way to build two-round message-transmission protocols with CRFs. We show a two-round MTP constructed using SPHF in Fig. 2.

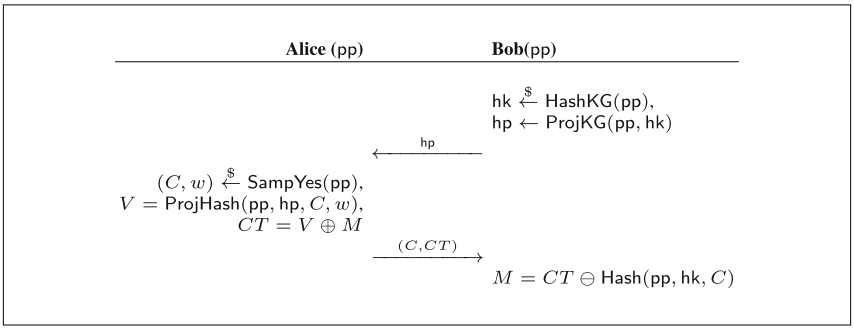

Fig. 2. Generic construction of two-round MTP from SPHF

Theorem 2. *The construction of MTP in Fig. 2 is correct and semantically secure.*

It should be clear that the protocol functionality is ensured by the correctness of the SPHF and the security is guaranteed by the pseudo-randomness of the SPHF, which is implied by the smoothness and the hardness of the subset membership problem.

CRF for the Receiver. In reality, a tampered implementation of Bob (the receiver) might choose an insecure public key so that an eavesdropper will be able to read Alice's plaintext. The key could also act as a channel to leak some

secrets to Alice or an eavesdropper. Even assuming that the protocol is semantically secure, without the CRF, the compromised implementation of Bob can still leak some secret information to the outside. It is thus desirable for the CRF to resist exfiltration. Figure 3 shows the reverse firewall for Bob. The idea is that the CRF re-randomizes the public key chosen by Bob before it is sent to the outside world. To maintain the protocol functionality, it also intercepts Bob's incoming messages and converts Alice's ciphertext under the re-randomized key to that under Bob's original public key. The CRF should also preserve the semantic security of the protocol regardless of how Bob behaves. A computationally bounded adversary learns nothing about Alice's input plaintext from the transcript between Alice and Bob's CRF, even when the original public key chosen by Bob is insecure.

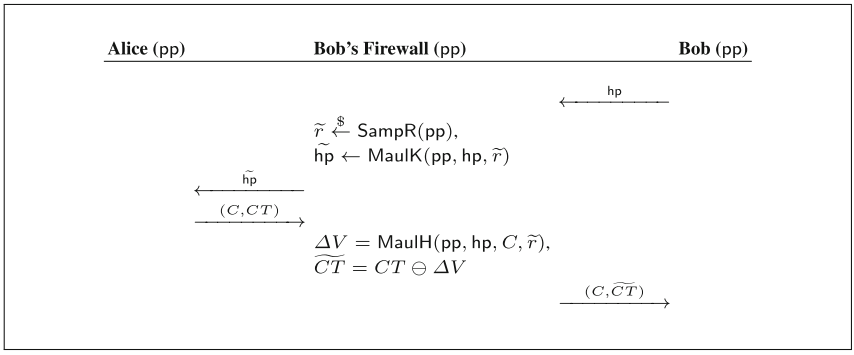

Fig. 3. Bob's CRF for the protocol shown in Fig. 2

Theorem 3. *The* CRF *for Bob shown in Fig. 3 maintains functionality and strongly preserves security for Bob, and strongly resists exfiltration against Alice, provided that the underlying* SPHF *is projection key-malleable.*

Proof. We verify that our construction satisfies the following properties.

Functionality Maintaining. For any ciphertext (C, CT),

$$\begin{aligned}\widetilde{CT} &= CT \ominus \Delta V = CT \ominus \mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{r}) \\ &= M \oplus \mathsf{ProjHash}(\mathsf{pp}, \widetilde{\mathsf{hp}}, C, w) \ominus \mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{r}) \\ &= M \oplus \mathsf{Hash}(\mathsf{pp}, \widetilde{\mathsf{hk}}, C) \ominus \mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{r}) \\ &= M \oplus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C).\end{aligned}$$

The above holds due to the *projection consistency* of the *projection key malleability* in the underlying SPHF. Therefore, Bob is able to recover Alice's plaintext by computing $M = \widetilde{CT} \ominus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C)$.

Strong Security Preservation and Strong Exfiltration Resistance. It suffices to show that the CRF strongly resists exifiltration. Suppose there exists an adversary who has non-negligible advantage $\mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$ in the game LEAK. We then show how to build an adversary $\mathcal{B}$ to break the *key indistinguishability* captured by the *projection key malleability* of the underlying SPHF by running $\mathcal{A}$. Recall that in the game LEAK, $\mathcal{A}$ would provide two parties $(\overline{\mathsf{P}}_1, \overline{\mathsf{P}}_2)$ which represent its chosen tampered implementations of Bob and Alice. $\mathcal{B}$ first runs the protocol between the honest party Bob and $\overline{\mathsf{P}}_2$, and obtains the output of Bob as hp_0. $\mathcal{B}$ then runs again the protocol between $\overline{\mathsf{P}}_1$ and $\overline{\mathsf{P}}_2$, and obtains the output of $\overline{\mathsf{P}}_1$ as hp_1. It then sends $(\mathsf{hp}_0, \mathsf{hp}_1)$ as the challenge projection keys for the key indistinguishability game, and receives the challenge re-randomized projection key $\widetilde{\mathsf{hp}}$. Finally, it forwards $\widetilde{\mathsf{hp}}$ to $\mathcal{A}$ as part of the challenge transcript $\mathcal{T}^*$ of the game LEAK and outputs the guess b' of $\mathcal{A}$ as its guess. It is easy to see that the above behaviours of $\mathcal{B}$ are computationally indistinguishable from the real game LEAK from the view of $\mathcal{A}$. Therefore, we have that $\mathsf{Adv}^{\mathsf{Key\text{-}Ind}}_{\mathcal{B},\mathsf{MSPHF}}(\ell) \geq \mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$, which contradicts the projection key malleability of the underling SPHF. This also trivially implies the strong security preservation of the CRF. □

CRF for the Sender. It is obvious that a CRF cannot prevent an arbitrarily tampered implementation of Alice from sending Bob some secret besides the message to be sent. That is, no CRF for Alice can achieve strong exfiltration resistance against Bob. Therefore, the "best possible" security is against the corrupted implementations of Alice that maintain the functionality. One should note that the MTP functionality requires Bob to recover the plaintext message of Alice. In other words, a functionality-maintaining corruption of Alice can only send the given input but no other message. Formally, we have the following theorem for the CRF depicted in Fig. 4.

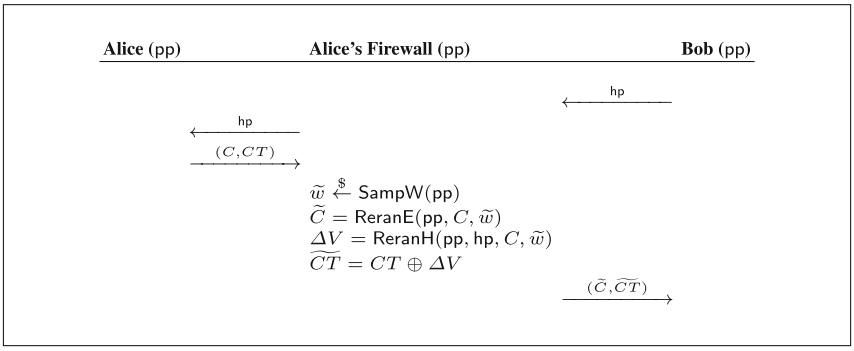

Fig. 4. Alice's CRF for the protocol shown in Fig. 2

Theorem 4. *The* CRF *for Alice shown in Fig. 4 maintains functionality and strongly preserves security for Alice, and weakly resists exfiltration against Bob, provided that the* SPHF *is element-rerandomizable.*

Proof. We verify that our construction satisfies the following properties.

Functionality Maintaining. One could easily have,

$$\begin{aligned}\widetilde{CT} &= CT \oplus \Delta V = CT \oplus \mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{w}) \\ &= M \oplus \mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, w) \oplus \mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{w}) \\ &= M \oplus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) \oplus \mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{w}) \\ &= M \oplus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, \widetilde{C}).\end{aligned}$$

The above holds by the *rerandomization consistency* as the underlying SPHF is element re-randomizable. Bob is thus able to recover Alice's plaintext by computing $M = \widetilde{CT} \ominus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, \widetilde{C})$.

Strong Security Preservation and Weak Exfiltration Resistance. For any tampered implementation of Alice that maintains functionality, suppose there exists an adversary who has non-negligible advantage $\mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$ in the game LEAK. We then show how to build an adversary $\mathcal{B}$ to break the *element indistinguishability* captured by the *element re-randomizability* of the underlying SPHF by running $\mathcal{A}$. Recall that in the game LEAK, $\mathcal{A}$ would provide two parties $(\overline{\mathsf{P}}_1, \overline{\mathsf{P}}_2)$ which represent its chosen tampered implementations of Alice and Bob. Note that the tampered implementation of Alice is functionality-maintaining. $\mathcal{B}$ first runs the protocol between honest party Alice and $\overline{\mathsf{P}}_2$, and obtains the output of Alice as (C_0, CT_0). $\mathcal{B}$ then runs again the protocol between $\overline{\mathsf{P}}_1$ and $\overline{\mathsf{P}}_2$, and obtains the output of $\overline{\mathsf{P}}_1$ as (C_1, CT_1). It then sends (C_0, C_1) as the challenge elements for the element indistinguishability game, and receives the challenge re-randomized element $\widetilde{C}$. It computes $\widetilde{CT} = M \oplus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, \widetilde{C})$ and then forwards $(\widetilde{C}, \widetilde{CT})$ to $\mathcal{A}$ as part of the challenge transcript $\mathcal{T}^*$ of the game LEAK and outputs the guess b' of $\mathcal{A}$ as its guess in the element indistinguishability game. It is easy to see that the above behaviours of $\mathcal{B}$ are computationally indistinguishable from the real game LEAK from the view of $\mathcal{A}$. Therefore, we have that $\mathsf{Adv}^{\mathsf{Element\text{-}Ind}}_{\mathcal{B},\mathsf{MSPHF}}(\ell) \geq \mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$, which contradicts the *element re-randomizability* of the underling SPHF. Therefore, the CRF weakly resists exfiltration against Bob and of course against any eavesdropper. This also trivially implies the security preservation of the firewall. □

4.2 Oblivious Signature-Based Envelope with CRFs

In this section, we introduce the CRF constructions for the oblivious signature-based envelope protocol with an instantiation from the language of encryption of signature. Formally, an OSBE protocol involves: a sender, holding a string P, and a receiver holding a credential. The protocol *functionality* requires that at the end of protocol, the receiver could receive P if and only if he/she possesses a certificate/signature on a predefined message M. The *security notion* asserts that the sender cannot determine whether the receiver owns the valid credential (*obliviousness*) and no other party learns anything about P (*semantic security*).

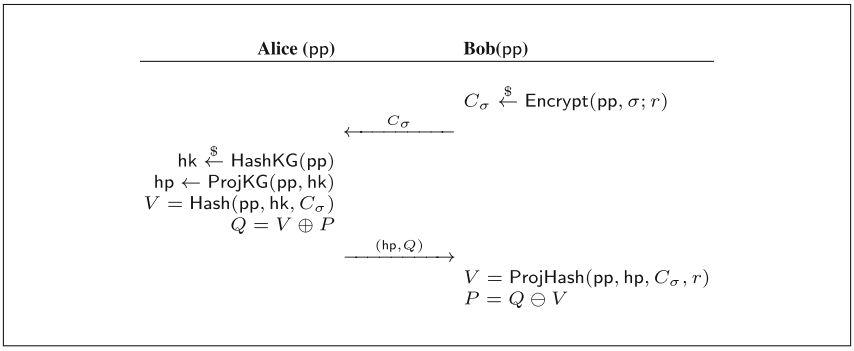

Fig. 5. Blazy-Pointcheval-Vergnaud OSBE framework [11]

Blazy-Pointcheval-Vergnaud OSBE Framework [11]**.** Noting that the original OSBE requires a secure channel during the execution to protect against eavesdroppers, Blazy, Pointcheval and Vergaud [11] clarified and enhanced the security models of OSBE by considering the security for both the sender and the receiver against the authority. Their new notion, namely *semantic security w.r.t. the authority*, requires that the authority who plays as the eavesdropper on the protocol, learns nothing about the private message of the sender. They showed how to generically build a 2-round OSBE scheme that can achieve the defined strong security in the standard model with a Common Reference String (CRS). We first recall a slightly modified version of their general framework, which is illustrated in Fig. 5. In particular, without loss of generality, we assume that the string P is in the hash value space of the underlying SPHF. The main idea of the BPV-OSBE framework relies on the SPHF from the language defined by the encryption of valid signatures. Let $\mathsf{pp} = (\mathsf{PP}, \mathsf{ek}, \mathsf{vk}, \mathsf{M})$ where PP is the collection of global parameters for the signature scheme, the encryption scheme and the SPHF system, ek is the public key of the encryption scheme, vk is the verification key of the signature scheme and M is the predefined message. Suppose $\mathsf{Encrypt}$ is the encryption algorithm of the encryption scheme and Ver is the verification algorithm of the signature scheme. The language of the underlying SPHF is then defined as $\mathcal{L} = \{C_\sigma \mid \exists r, \sigma, \text{s.t.}, C_\sigma = \mathsf{Encrypt}(\mathsf{pp}, \sigma; r) \wedge \mathsf{Ver}(\mathsf{pp}, \sigma, M) = 1\}$. We then have that the subset membership problem is hard due to the security of the encryption scheme. Readers are referred to [11] for the detailed analysis of protocol correctness and security.

CRF **for the Receiver.** An tampered implementation of the receiver might produce a ciphertext C_σ that either enables an eavesdropper to read Alice's message P, or acts as a channel to leak some secrets to the outsider (Alice or an eavesdropper). A CRF for Bob (denoted by $\mathcal{W}_B$) should be able to re-randomize the ciphertext C_σ while still preserves the protocol functionality. It is also a requirement for $\mathcal{W}_B$ to preserve the protocol security, i.e., obliviousness, semantic security and semantic security w.r.t the authority. Regarding exfiltration, $\mathcal{W}_B$

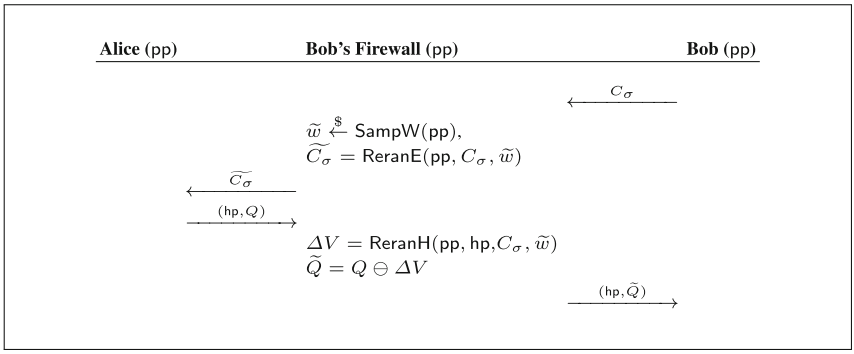

Fig. 6. Bob's CRF for the OSBE protocol shown in Fig. 5

should prevent the compromised Bob from using C_σ as a channel to leak secrets. Figure 6 depicts the firewall $\mathcal{W}_B$ in the OSBE protocol.

Theorem 5. *The* CRF *for Bob shown in Fig. 6 maintains functionality and strongly preserves security for Bob, and strongly resists exfiltration against Alice, provided that the underlying* SPHF *is element-rerandomizable.*

Proof. We verify that our construction satisfies the following properties.
Functionality Maintaining. Due to the *rerandomization consistency* of the *element re-randomizability*, we have

$$\begin{aligned}\widetilde{Q} &= Q \ominus \Delta V \\ &= Q \ominus \mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C_\sigma, \widetilde{w}) \\ &= P \oplus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, \widetilde{C_\sigma}) \ominus \mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C_\sigma, \widetilde{w}) \\ &= P \oplus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C_\sigma).\end{aligned}$$

Bob is thus able to recover P by computing $P = \widetilde{Q} \ominus \mathsf{ProjHash}(\mathsf{pp}, \mathsf{hk}, C_\sigma, r)$.

Strong Security Preservation and Strong Exfiltration Resistance. The strong exfiltration resistance follows from the fact that $\widetilde{C_\sigma}$ is independent of the original ciphertext C_σ chosen by Bob who might be arbitrarily compromised. Precisely, suppose there exists an adversary who has non-negligible advantage $\mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$ in the game LEAK. We then show how to build an adversary $\mathcal{B}$ to break the *element indistinguishability* captured by the *element re-randomizability* of the underlying SPHF by running $\mathcal{A}$. Recall that in the game LEAK, $\mathcal{A}$ would provide two parties $(\overline{\mathsf{P}}_1, \overline{\mathsf{P}}_2)$ which represent its chosen tampered implementations of Bob and Alice. $\mathcal{B}$ first runs the protocol between the honest party Bob and $\overline{\mathsf{P}}_2$, and obtains the output of Bob as C_0. $\mathcal{B}$ then runs again the protocol between $\overline{\mathsf{P}}_1$ and $\overline{\mathsf{P}}_2$, and obtains the output of $\overline{\mathsf{P}}_1$ as C_1. It then sends (C_0, C_1) as the challenge elements for the element indistinguishability game, and receives the challenge re-randomized element $\widetilde{C_\sigma}$. Finally, it forwards $\widetilde{C_\sigma}$ to $\mathcal{A}$ as part of the challenge transcript $\mathcal{T}^*$ of the game LEAK and outputs the guess b' of $\mathcal{A}$ as

its guess in the key indistinguishability game. It is easy to see that the above behaviours of $\mathcal{B}$ are computationally indistinguishable from the real game LEAK from the view of $\mathcal{A}$. Therefore, we have that $\mathsf{Adv}^{\mathsf{Element\text{-}Ind}}_{\mathcal{B},\mathsf{MSPHF}}(\ell) \geq \mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$, which contradicts the element-rerandomizability of the underling SPHF. This trivially implies that the CRF also strongly preserves the protocol security. □

CRF for the Sender. Similar to the message-transmission protocol, it is easy to see that no CRF for Alice can achieve strong exfiltration resistance against Bob. The "best possible" security is thus against the corrupted implementations of Alice that maintain the functionality. We show the CRF for Alice (denoted by $\mathcal{W}_A$) in Fig. 7. Formally, we have the following theorem.

Theorem 6. *The* CRF *for Alice shown in Fig. 7 maintains functionality and strongly preserves security for Alice, and weakly resists exfiltration against Bob, provided that the underlying* SPHF *is projection key-malleable.*

Proof. We verify that our construction satisfies the following properties.

Functionality Maintaining. Due to the *projection consistency* of the *projection key-malleability* of the underlying SPHF, we have

$$\begin{aligned}\widetilde{Q} &= Q \oplus \Delta V = Q \oplus \mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, C_\sigma, \widetilde{r}) \\ &= P \oplus \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C_\sigma) \oplus \mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, C_\sigma, \widetilde{r}) \\ &= P \oplus \mathsf{Hash}(\mathsf{pp}, \widetilde{\mathsf{hk}}, C_\sigma).\end{aligned}$$

In the above, $\widetilde{\mathsf{hk}}$ is the associated key of projection key $\widetilde{\mathsf{hp}} \leftarrow \mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}, \widetilde{r})$. We can see that Bob can recover P by computing $P = \widetilde{Q} \ominus \mathsf{ProjHash}(\mathsf{pp}, \widetilde{\mathsf{hp}}, C_\sigma, r)$.

Strong Security Preservation and Weak Exfiltration Resistance. For any tampered implementation of Alice that maintains functionality, suppose there exists an adversary who has non-negligible advantage $\mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$ in the game LEAK. We then show how to build an adversary $\mathcal{B}$ to break the *key indistinguishability* captured by the *projection key-malleability* of the underlying MSPHF by running $\mathcal{A}$. Recall that in the game LEAK, $\mathcal{A}$ would provide two parties $(\overline{\mathsf{P}}_1, \overline{\mathsf{P}}_2)$ which represent its chosen tampered implementations of Alice and Bob. Note that the tampered implementation of Alice is functionality-maintaining. $\mathcal{B}$ first runs the protocol between honest party Alice and $\overline{\mathsf{P}}_2$, and obtains the output of Alice as (hp_0, Q_0). $\mathcal{B}$ then runs again the protocol between $\overline{\mathsf{P}}_1$ and $\overline{\mathsf{P}}_2$, and obtains the output of $\overline{\mathsf{P}}_1$ as (hp_1, Q_1). It then sends $(\mathsf{hp}_0, \mathsf{hp}_1)$ as the challenge projection key for the key indistinguishability game, and receives the challenge re-randomized projection key $\widetilde{\mathsf{hp}}$. It computes $\widetilde{Q} = P \oplus \mathsf{ProjHash}(\mathsf{pp}, \widetilde{\mathsf{hp}}, C_\sigma, r)$, and then forwards $(\widetilde{\mathsf{hp}}, \widetilde{Q})$ to $\mathcal{A}$ as part of the challenge transcript $\mathcal{T}^*$ of the game LEAK and outputs the guess b' of $\mathcal{A}$ as its guess in the key indistinguishability game. It is easy to see that the above behaviours of $\mathcal{B}$ are computationally indistinguishable from the real game LEAK from the view of $\mathcal{A}$. Therefore, we have that $\mathsf{Adv}^{\mathsf{Key\text{-}Ind}}_{\mathcal{B},\mathsf{MSPHF}}(\ell) \geq \mathsf{Adv}^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(\ell)$, which contradicts the *projection key-malleability* of the underling MSPHF. Therefore, the firewall weakly resists exfiltration against

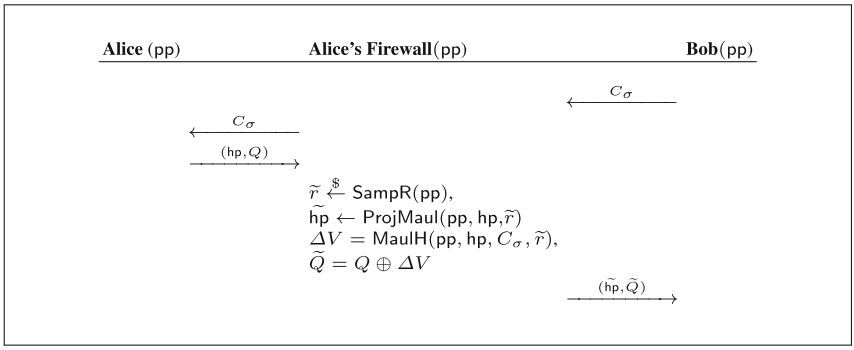

Fig. 7. Alice's CRF for the OSBE protocol shown in Fig. 5

Bob and of course against any eavesdropper. This also trivially implies the security preservation of the CRF. □

Instantiation from the Linear Encryption of Valid Signatures. In the work [11], an efficient OSBE protocol is proposed by combining the linear encryption scheme, the Waters signature [25] and an SPHF on the language of linear ciphertexts. Here we show how to extend the instantiated SPHF to be malleable for the CRF constructions. It is worth noting that the introduced malleable SPHF here could also be represented by graded ring but does not follow the generic framework proposed in Sect. 3.2 (i.e., Θ_{aux} is not an identity function). We first recall the SPHF proposed in the work [11]. Let $\mathbb{G}, \mathbb{G}_T$ be two multiplicative groups with the same prime order p. Let g be the generator of $\mathbb{G}$ and I be the identity element of $\mathbb{G}_T$. A symmetric bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that $e(u^a, v^b) = e(u, v)^{ab}$ for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$. It is worth noting that e can be efficiently computed and $e(g, g) \neq 1_{\mathbb{G}_T}$.

Linear Encryption of Waters Signatures. Let $h \xleftarrow{\$} \mathbb{G}$ and $\mathbf{u} = (u_0, ..., u_k) \xleftarrow{\$} \mathbb{G}^{k+1}$ which defines the Waters hash of a message $M = (M_1, ..., M_k) \in \{0,1\}^k$ as $\mathcal{F}(M) = u_0 \prod_{i=1}^{k} u_i^{M_i}$. The verification key is $\mathsf{vk} = g^z$ and the associated signing key is $\mathsf{sk} = h^z$ where $z \xleftarrow{\$} \mathbb{Z}_p$. The signature on a message M is $\sigma = (\sigma_1 = \mathsf{sk} \cdot \mathcal{F}(M)^s, \sigma_2 = g^s)$ for some random $s \xleftarrow{\$} \mathbb{Z}_p$. It can be verified by checking $e(g, \sigma_1) = e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), \sigma_2)$. The linear encryption public key is $\mathsf{ek} = (Y_1 = g^{y_1}, Y_2 = g^{y_2})$ and the secret key is $\mathsf{dk} = (y_1, y_2) \xleftarrow{\$} \mathbb{Z}_p^2$. The ciphertext of a Waters signature $\sigma = (\sigma_1, \sigma_2)$ is $C_\sigma = (c_1 = Y_1^{r_1}, c_2 = Y_2^{r_2}, c_3 = g^{r_1+r_2} \cdot \sigma_1, c_4 = \sigma_2)$, where $(r_1, r_2) \xleftarrow{\$} \mathbb{Z}_p^2$.

The Instantiated Malleable SPHF. We first interpret the underlying SPHF using the graded ring. The language is defined as,

$$\mathcal{L} = \Big\{(c_1, c_2, c_3, c_4) | \exists (r_1, r_2) \in \mathbb{Z}_p^2, (\sigma_1, \sigma_2) \in \mathbb{G}_1^2, \text{s.t.}, \big(c_1 = Y_1^{r_1}, c_2 = Y_2^{r_2}, \\ c_3 = g^{r_1+r_2} \cdot \sigma_1, c_4 = \sigma_2\big) \bigwedge \big(e(g, \sigma_1) = e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), \sigma_2)\big)\Big\}.$$

For any $C_\sigma = (c_1, c_2, c_3, c_4)$, we have

$$\Theta_{\mathsf{aux}}(C_\sigma) = \Big(c_1' = e(c_1, g), c_2' = e(c_2, g), c_3' = e(c_3, g)/\big(e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), c_4)\big)\Big),$$

and $\Gamma(C_\sigma) = \begin{pmatrix} Y_1 & 1 & g \\ 1 & Y_2 & g \end{pmatrix} \in \mathbb{G}^{2\times 3}$. We can see that if $C_\sigma \in \mathcal{L}$ with witness $w = (r_1, r_2)$, let $\boldsymbol{\lambda} = (g^{r_1}, g^{r_2})$, we have,

$$\begin{aligned} \Theta_{\mathsf{aux}}(C_\sigma) &= \Big(e(c_1, g), e(c_2, g), e(c_3, g)/\big(e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), c_4)\big)\Big) \\ &= \big(e(Y_1^{r_1}, g), e(Y_2^{r_2}, g), e(g^{r_1+r_2}, g)\big) \\ &= \boldsymbol{\lambda} \odot \Gamma(C_\sigma). \end{aligned}$$

Let $\mathsf{pp} = (\mathbb{G}, p, g, Y_1, Y_2, \mathbf{u})$, $\widetilde{\boldsymbol{r}} = (\beta_1, \beta_2, \beta_3)^\mathsf{T}$ and $\widetilde{\boldsymbol{\lambda}} = \widetilde{\boldsymbol{w}} = (\eta_1, \eta_2, \eta_3)$. The instantiation is as follows:

- $\mathsf{HashKG}(\mathsf{pp}) : \mathsf{hk} = \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)^\mathsf{T} \xleftarrow{\$} \mathbb{Z}_p^3$;
- $\mathsf{ProjKG}(\mathsf{pp}, \mathsf{hk}, C) : \mathsf{hp} = \boldsymbol{\gamma}(C_\sigma) = \Gamma(C_\sigma) \odot \boldsymbol{\alpha} = (Y_1^{\alpha_1} g^{\alpha_3}, Y_2^{\alpha_2} g^{\alpha_3})^\mathsf{T}$;
- $\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C) : \mathsf{hv} = \Theta_{\mathsf{aux}}(C_\sigma) \odot \boldsymbol{\alpha} = (c_1', c_2', c_3') \odot (\alpha_1, \alpha_2, \alpha_3)^\mathsf{T} = e(c_1, g)^{\alpha_1} \cdot e(c_2, g)^{\alpha_2} \cdot \Big(e(c_3, g)/\big(e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), c_4)\big)\Big)^{\alpha_3}$;
- $\mathsf{ProjHash}(\mathsf{pp}, \mathsf{hp}, C, \boldsymbol{w}) : \mathsf{hv} = \boldsymbol{\lambda} \odot \boldsymbol{\gamma}(C_\sigma) = (g^{r_1}, g^{r_2}) \odot (Y_1^{\alpha_1} g^{\alpha_3}, Y_2^{\alpha_2} g^{\alpha_3})^\mathsf{T} = e\Big((Y_1^{\alpha_1} g^{\alpha_3})^{r_1} \cdot (Y_2^{\alpha_2} g^{\alpha_3})^{r_2}, g\Big)$;
- $\mathsf{MaulK}(\mathsf{pp}, \mathsf{hp}, \widetilde{\boldsymbol{r}}) : \widetilde{\mathsf{hp}} = \boldsymbol{\gamma}(C) \oplus (\Gamma(C) \odot \widetilde{\boldsymbol{r}}) = \boldsymbol{\gamma}(C_\sigma) \oplus (\Gamma(C_\sigma) \odot \boldsymbol{\Delta r}) = (Y_1^{\alpha_1} g^{\alpha_3}, Y_2^{\alpha_2} g^{\alpha_3}) \oplus (Y_1^{\beta_1} g^{\beta_3}, Y_2^{\beta_2} g^{\beta_3}) = ((Y_1^{\alpha_1+\beta_1} g^{\alpha_3+\beta_3}, Y_2^{\alpha_2+\beta_2} g^{\alpha_3+\beta_3}))^\mathsf{T}$;
- $\mathsf{MaulH}(\mathsf{pp}, \mathsf{hp}, \widetilde{\boldsymbol{r}}, C) : \widetilde{\mathsf{hv}} = \Theta_{\mathsf{aux}}(C) \odot \widetilde{\boldsymbol{r}} = (c_1', c_2', c_3') \odot (\beta_1, \beta_2, \beta_3)^\mathsf{T} = e(c_1, g)^{\beta_1} \cdot e(c_2, g)^{\beta_2} \cdot \Big(e(c_3, g)/\big(e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), c_4)\big)\Big)^{\beta_3}$;
- $\mathsf{ReranE}(\mathsf{pp}, C, \widetilde{\boldsymbol{w}}) : \widetilde{C} = C_\sigma \oplus (Y_1^{\eta_1}, Y_2^{\eta_2}, g^{\eta_1+\eta_2} \mathcal{F}(M)^{\eta_3}, g^{\eta_3}) = (c_1 \cdot Y_1^{\eta_1}, c_2 \cdot Y_2^{\eta_2}, c_3 \cdot g^{\eta_1+\eta_2} \mathcal{F}(M)^{\eta_3}, c_4 \cdot g^{\eta_3})$;
- $\mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C, \widetilde{\boldsymbol{w}}) : \widetilde{\mathsf{hv}} = (g^{\eta_1}, g^{\eta_2}) \odot \Gamma(C_\sigma) = (g^{\eta_1}, g^{\eta_2}) \odot (Y_1^{\alpha_1} g^{\alpha_3}, Y_2^{\alpha_2} g^{\alpha_3}) = e\Big((Y_1^{\alpha_1} g^{\alpha_3})^{\eta_1} \cdot (Y_2^{\alpha_2} g^{\alpha_3})^{\eta_2}, g\Big)$.

Theorem 7. *The above construction is a malleable smooth projective hash function.*

Proof. We verify that our construction satisfies the following properties. Note that the constructions of both MaulK and MaulH follow the framework proposed in Sect. 3.2. According to Theorem 1, we have that our constructed SPHF is projection key-malleable. Note that in our construction, $C_\sigma' = C_\sigma \oplus (Y_1^{\eta_1}, Y_2^{\eta_2},$

$g^{\eta_1+\eta_2}\mathcal{F}(M)^{\eta_3}, g^{\eta_3})$, one can easily observe the rerandomization is *element-indistinguishable* due to the 2-Lin assumption. Particularly, we have that $(Y_1^{\eta_1}, Y_2^{\eta_2}, g^{\eta_1+\eta_2})$ is a linear tuple w.r.t (Y_1, Y_2, g). If any adversary can distinguish the rerandomized element, we can use it as a subroutine to break the 2-Lin assumption. We then prove that the element rerandomization is *membership-preserving*. Suppose $C_\sigma = \big(c_1 = Y_1^{r_1}, c_2 = Y_2^{r_2}, c_3 = g^{r_1+r_2} \cdot \sigma_1, c_4 = \sigma_2\big) \in \mathcal{L}$. We have that after it is rerandomized,

$$\begin{aligned}
\widetilde{C_\sigma} &= C_\sigma \oplus (Y_1^{\eta_1}, Y_2^{\eta_2}, g^{\eta_1+\eta_2}\mathcal{F}(M)^{\eta_3}, g^{\eta_3}) \\
&= (c_1 \cdot Y_1^{\eta_1}, c_2 \cdot Y_2^{\eta_2}, c_3 \cdot g^{\eta_1+\eta_2}\mathcal{F}(M)^{\eta_3}, c_4 \cdot g^{\eta_3}) \\
&= \big(Y_1^{r_1+\eta_1},\ Y_2^{r_2+\eta_2},\ g^{r_1+r_2+\eta_1+\eta_2} \cdot \sigma_1 \cdot \mathcal{F}(M)^{\eta_3},\ \sigma_2 \cdot g^{\eta_3}\big) \\
&\stackrel{\mathsf{def}}{=} \big(\widetilde{c}_1, \widetilde{c}_2, \widetilde{c}_3, \widetilde{c}_4\big)
\end{aligned}$$

Since Γ is a constant function, we know that, $\Gamma(\widetilde{C_\sigma}) = \Gamma(C_\sigma) = \begin{pmatrix} Y_1 & 1 & g \\ 1 & Y_2 & g \end{pmatrix}$. Let $\widetilde{\boldsymbol{\lambda}} = (g^{r_1+\eta_1}, g^{r_2+\eta_2})$, we then obtain:

$$\begin{aligned}
\Theta_{\mathsf{aux}}(\widetilde{C_\sigma}) &= \Big(e(\widetilde{c}_1, g), e(\widetilde{c}_2, g), e(\widetilde{c}_3, g)/\big(e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), \widetilde{c}_4)\big)\Big) \\
&= \Big(e(Y_1^{r_1+\eta_1}, g), e(Y_2^{r_2+\eta_2}, g), \frac{e(g^{r_1+r_2+\eta_1+\eta_2} \cdot \sigma_1 \cdot \mathcal{F}(M)^{\eta_3}, g)}{e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), \sigma_2 \cdot g^{\eta_3})}\Big) \\
&= \big(e(Y_1^{r_1+\eta_1}, g), e(Y_2^{r_2+\eta_2}, g), e(g^{r_1+r_2+\eta_1+\eta_2}, g)\big) \\
&= \widetilde{\boldsymbol{\lambda}} \odot \Gamma(\widetilde{C_\sigma}).
\end{aligned}$$

This shows that $\widetilde{C_\sigma} \in \mathcal{L}$. If $C_\sigma \notin \mathcal{L}$, we trivially have that $\widetilde{C_\sigma} \notin \mathcal{L}$.

We then justify the rerandomization consistency. For any hashing key $\mathsf{hk} = \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)^\mathsf{T} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$, we have that,

$$\begin{aligned}
\mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, \widetilde{C_\sigma}) &= \Theta_{\mathsf{aux}}(\widetilde{C_\sigma}) \odot \boldsymbol{\alpha} \\
&= \Big(e(\widetilde{c}_1, g), e(\widetilde{c}_2, g), \frac{e(\widetilde{c}_3, g)}{e(\mathsf{vk}, h) \cdot e(\mathcal{F}(M), \widetilde{c}_4)}\Big) \odot (\alpha_1, \alpha_2, \alpha_3)^\mathsf{T} \\
&= (c_1', c_2', c_3') \odot (\alpha_1, \alpha_2, \alpha_3)^\mathsf{T} \oplus (g^{\eta_1}, g^{\eta_2}) \odot (Y_1^{\alpha_1} g^{\alpha_3}, Y_2^{\alpha_2} g^{\alpha_3}) \\
&= \Theta_{\mathsf{aux}}(C_\sigma) \odot \boldsymbol{\alpha} \oplus e\Big((Y_1^{\alpha_1} g^{\alpha_3})^{\eta_1} \cdot (Y_2^{\alpha_2} g^{\alpha_3})^{\eta_2}, g\Big) \\
&= \mathsf{Hash}(\mathsf{pp}, \mathsf{hk}, C_\sigma) \oplus \mathsf{ReranH}(\mathsf{pp}, \mathsf{hp}, C_\sigma, \widetilde{\boldsymbol{w}}).
\end{aligned}$$

5 Oblivious Transfer with Reverse Firewall

5.1 A New OT Framework from Graded Rings

Oblivious transfer forms a central primitive in modern cryptography. It is a protocol between the sender, holding two message M_0 and M_1, and a receiver holding a choice bit b. The OT *functionality* requires that at the end of the protocol, the receiver can learn the message M_b. The *security requirement* is

$\underline{\mathsf{SampI}(\boldsymbol{\Gamma}, b)}$:

$\boldsymbol{w} \xleftarrow{\$} \mathsf{SampW}(\mathsf{pp})$
$C := \lambda(\boldsymbol{w}) \odot \boldsymbol{\Gamma}$
Parse $\boldsymbol{\Gamma}$ as $(\Gamma_1, ..., \Gamma_n)$
Set $\boldsymbol{e} = (0_{\mathbb{Z}_p}, ..., 0_{\mathbb{Z}_p}, b_{\mathbb{Z}_p})_{1\times m}$
$\Delta C := \boldsymbol{e} \odot (\mathbf{1}_{\mathbb{G}}, ..., \mathbf{1}_{\mathbb{G}}, \Gamma_n)_{1\times n}$
$C_0 := C \oplus \Delta C$
Return $(C_0, \boldsymbol{w})$

$\underline{\mathsf{PairG}(\boldsymbol{\Gamma}, C_0)}$:

Parse $\boldsymbol{\Gamma}$ as $(\Gamma_1, ..., \Gamma_n)$
set $\boldsymbol{\Gamma}' = (\mathbf{1}_{\mathbb{G}}, ..., \mathbf{1}_{\mathbb{G}}, \Gamma_n)_{1\times n}$
set $\boldsymbol{e} = (0_{\mathbb{Z}_p}, ..., 0_{\mathbb{Z}_p}, 1_{\mathbb{Z}_p})_{1\times m}$
$\Delta C := \boldsymbol{e} \odot \boldsymbol{\Gamma}'$
$C_1 := C_0 \ominus \Delta C$
return C_1

Note: $\mathbf{1}_{\mathbb{G}}$ is a $m \times 1$ matrix of $1_{\mathbb{G}}$

Fig. 8. Definitions of algortihms SampI, PairG.

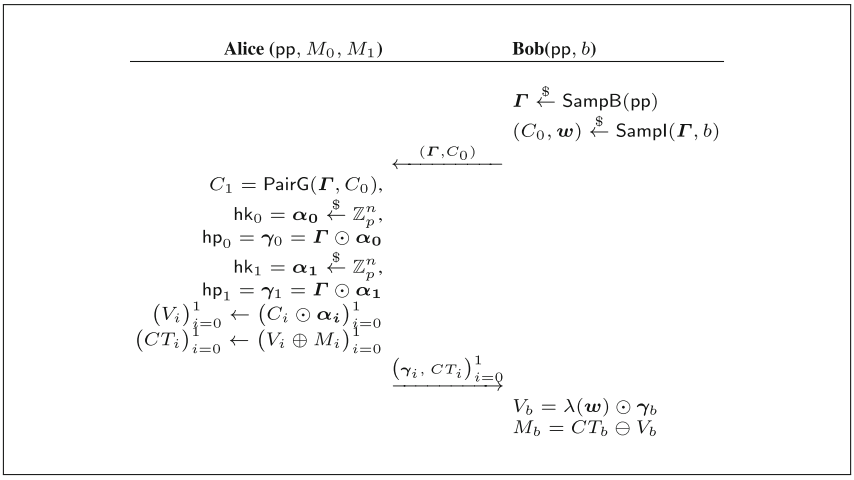

Fig. 9. OT Protocols from graded rings.

that the receiver learns nothing about M_{1-b} (*sender security*), and the sender learns nothing about the receiver's choice b (*receiver security*). We introduce a variant of the HK-OT [16] framework in the context of graded rings. Essentially, we follow the generic framework of (malleable) SPHF from graded rings (shown in Sect. 3.2). The modified semi-generic framework narrows the possible instantiations of the HK-OT framework. However, as we will show later, the CRF construction following our framework not only captures the prior work [21], which is the only known OT-CRF to date, but also yields new constructions under weaker assumptions.

Before introducing our framework, we define two new algorithms SampI, PairG depicted in Fig. 8. For the sake of clarity, we use $\boldsymbol{\lambda} = \lambda(w)$ to represent the derivation of $\boldsymbol{\lambda}$ from the witness w. We require Θ_{aux} to be an identity function and Γ to be a constant function. That is, we only consider the KV type SPHF on diverse groups. As before, the subset membership problem must also be hard. Note that these are exactly the same conditions (Theorem 1) for our malleable SPHF construction presented in Sect. 3.2. Our graded ring-based OT framework is shown in Fig. 9. Suppose the element basis (denoted by $\boldsymbol{\Gamma} = (\Gamma_1, ..., \Gamma_n) \in$

$\mathbb{G}^{m\times n}$) is chosen by the receiver using the algorithm named SampB. It is worth noting that for the sake of simplicity, we assume without loss of generality the receiver (even the tampered implementation) would not trivially choose $\Gamma_i = \mathbf{1}_{\mathbb{G}}$ for any $i \in [1, n]$, since such an attempt can be easily detected in reality. One can note that:

- $b = 0$: $C_0 \in \mathcal{L}$ as $C_0 = \lambda(\boldsymbol{w}) \odot \boldsymbol{\Gamma}$ and $C_1 \notin \mathcal{L}$ as C_1 is not a linear span of $\boldsymbol{\Gamma}$.
- $b = 1$: $C_0 \notin \mathcal{L}$ as C_0 is not a linear span of $\boldsymbol{\Gamma}$ and $C_1 \in \mathcal{L}$ as $C_1 = \lambda(\boldsymbol{w}) \odot \boldsymbol{\Gamma}$.

Formally, we have the following result for the above framework.

Theorem 8. *The generic construction of OT shown in Fig. 9 is correct and secure.*

The protocol functionality (correctness) follows from the fact that $C_b \in \mathcal{L}$ and the sender security is guaranteed as $C_{1-b} \notin \mathcal{L}$. The receiver security is due to the hardness of the subset membership problem.

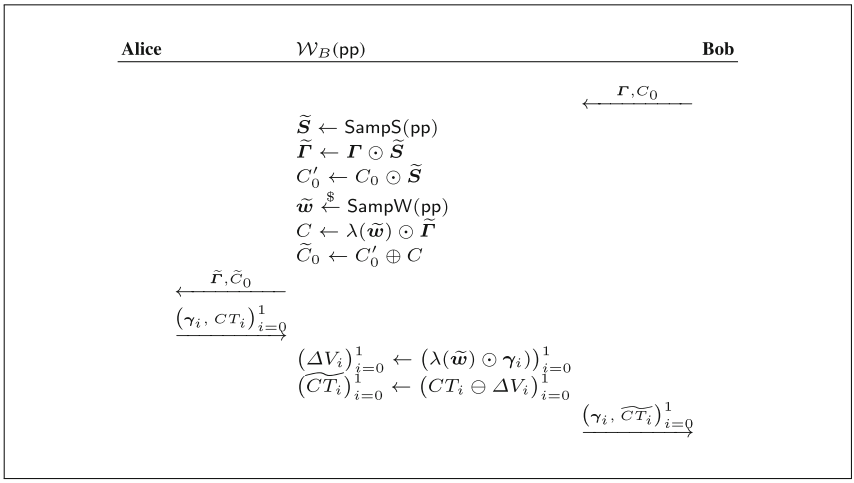

Fig. 10. Bob's CRF for the OT protocol in Fig. 9

5.2 Constructions of CRFs

CRF for the Receiver. The construction of the receiver CRF (denoted by $\mathcal{W}_B$) under our OT framework is shown in Fig. 10. The algorithm SampS outputs a *transformation matrix* (denoted by $\widetilde{\boldsymbol{S}} \in \mathbb{Z}_p^{n\times n}$) for the element basis $\boldsymbol{\Gamma}$. We denote the output of $\mathsf{PairG}(\boldsymbol{\Gamma}, \widetilde{C_0})$ as $\widetilde{C_1}$ and it should be clear that:

- $b = 0$: $\widetilde{C_0} = \lambda(\boldsymbol{w}) \odot \boldsymbol{\Gamma} \odot \widetilde{\boldsymbol{S}} \oplus \widetilde{w} \odot \widetilde{\boldsymbol{\Gamma}} = (\lambda(\boldsymbol{w}) \oplus \lambda(\widetilde{\boldsymbol{w}})) \odot \widetilde{\boldsymbol{\Gamma}}$. $\widetilde{C_1} = (\lambda(\boldsymbol{w}) \odot \boldsymbol{\Gamma} \ominus \Delta C) \odot \widetilde{\boldsymbol{S}} \oplus \widetilde{w} \odot \widetilde{\boldsymbol{\Gamma}} = (\lambda(\boldsymbol{w}) \oplus \lambda(\widetilde{\boldsymbol{w}})) \odot \widetilde{\boldsymbol{\Gamma}} \ominus \Delta C \odot \widetilde{\boldsymbol{S}}$, where $\Delta C = (0_{\mathbb{Z}_p}, ..., 0_{\mathbb{Z}_p}, 1_{\mathbb{Z}_p})_{1\times m} \odot (\mathbf{1}_{\mathbb{G}}, ..., \mathbf{1}_{\mathbb{G}}, \Gamma_n)_{1\times n}$.

– $b = 1$: $\widetilde{C_0} = (\lambda(\boldsymbol{w}) \odot \boldsymbol{\Gamma} \oplus \Delta C) \odot \widetilde{\boldsymbol{S}} \oplus \widetilde{w} \odot \widetilde{\boldsymbol{\Gamma}} = (\lambda(\boldsymbol{w}) \oplus \lambda(\widetilde{\boldsymbol{w}})) \odot \widetilde{\boldsymbol{\Gamma}} \oplus \Delta C \odot \widetilde{\boldsymbol{S}}$, where $\Delta C = (0_{\mathbb{Z}_p}, ..., 0_{\mathbb{Z}_p}, 1_{\mathbb{Z}_p})_{1\times m} \odot (\mathbf{1}_{\mathbb{G}}, ..., \mathbf{1}_{\mathbb{G}}, \Gamma_n)_{1\times n}$. $\widetilde{C_1} = \lambda(\boldsymbol{w}) \odot \boldsymbol{\Gamma} \odot \widetilde{\boldsymbol{S}} \oplus \widetilde{w} \odot \widetilde{\boldsymbol{\Gamma}} = (\lambda(\boldsymbol{w}) \oplus \lambda(\widetilde{\boldsymbol{w}})) \odot \widetilde{\boldsymbol{\Gamma}}$.

That is, $\widetilde{C_b} \in \mathcal{L}$ and thus $\mathcal{W}_B$ maintains the protocol functionality:

$$\begin{aligned}
\widetilde{CT_b} &= CT_b \ominus \Delta V_b \\
&= M_b \oplus \left(\widetilde{C_b} \odot \boldsymbol{\alpha_b}\right) \ominus \left(\lambda(\widetilde{\boldsymbol{w}}) \odot \boldsymbol{\gamma}_b\right) \\
&= M_b \oplus \left(\lambda(\boldsymbol{w}) \oplus \lambda(\widetilde{\boldsymbol{w}})\right) \odot \widetilde{\boldsymbol{\Gamma}} \odot \boldsymbol{\alpha_b}\right) \ominus \left(\lambda(\widetilde{\boldsymbol{w}}) \odot \widetilde{\boldsymbol{\Gamma}} \odot \boldsymbol{\alpha_b}\right) \\
&= M_b \oplus \left(\lambda(\boldsymbol{w}) \odot \widetilde{\boldsymbol{\Gamma}} \odot \boldsymbol{\alpha_b}\right) \\
&= M_b \oplus \left(\lambda(\boldsymbol{w}) \odot \boldsymbol{\gamma}_b\right).
\end{aligned}$$

Discussions on $\widetilde{\boldsymbol{S}}$. It is a trivial observation that $\mathcal{W}_B$ could strongly resist exfiltration if $\widetilde{\boldsymbol{\Gamma}}$ is independent from $\boldsymbol{\Gamma}$ as this also results in a random element $\widetilde{C}$ (by uniformly sampling $\widetilde{\boldsymbol{w}}$). Precisely, let $\boldsymbol{\Gamma} = (\Gamma_1, ..., \Gamma_n)$. An ideal transformation matrix $\widetilde{\boldsymbol{S}}$ should transfer each Γ_i to another random $\widetilde{\Gamma_i}$ and for any $i, j \in [1, n]$ and $i \neq j$, $\widetilde{\Gamma_i}$ is independent from $\widetilde{\Gamma_j}$. To realize such a transformation, one could either *shear and uniformly scale* or *globally and non-uniformly scale* the matrix $\boldsymbol{\Gamma}$ as follows:

– *Shear and uniform scaling.* Choose a column and then independently shear each other column. Then uniformly scale all the columns. The shearing and scaling could be in any order. A corresponding transformation matrix for this type of transformation has the following format (assuming the chosen column is Γ_1):

$$\widetilde{\boldsymbol{S}} = \mathbf{A} \odot \mathbf{B} = \begin{pmatrix} \alpha & 0 & \cdots & 0 \\ 0 & \alpha & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha \end{pmatrix} \odot \begin{pmatrix} 1 & \beta_2 & \cdots & \beta_n \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \in \mathbb{Z}_p^{n\times n},$$

where $(\alpha, \beta_2, ..., \beta_n) \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{A}$ is the a scaling matrix and $\mathbf{B}$ is the shearing matrix.

– *Globally non-uniform scaling.* Independently scale each column. A corresponding transformation matrix for this type of transformation has the following shape:

$$\widetilde{\boldsymbol{S}} = \begin{pmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_n \end{pmatrix} \in \mathbb{Z}_p^{n\times n},$$

where $(\alpha_1, ..., \alpha_n) \xleftarrow{\$} \mathbb{Z}_p^n$.

The first type has been used by Mironov and Stephens-Davidowitz in their OT-CRF construction [21]. One can note the second type of transformation is more efficient and thus can improve the efficiency. We will show the details in Sect. 5.3.

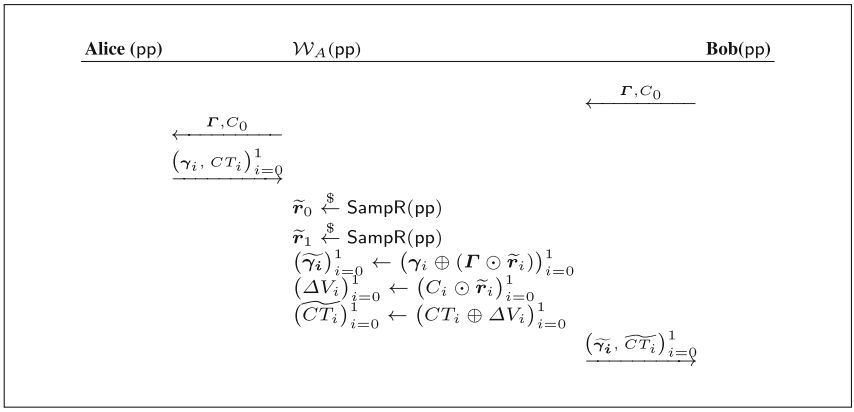

Fig. 11. Alice's CRF for the OT protocol in Fig. 9

CRF for the Sender. Figure 11 depicts the construction of CRF for the sender (denoted by $\mathcal{W}_A$). One may note that the construction is exactly part of the garded ring-based construction of malleable SPHF shown in Sect. 3.2. Therefore, according to Theorem 1, one could easily see that $\mathcal{W}_A$ maintains functionality, weakly resist exfiltration against Bob and strongly resist exfiltration against an eavesdropper. The composed firewall $\mathcal{W}_B \circ \mathcal{W}_A$ also weakly preservers security against Bob.

5.3 Instantiations

Due to the space limitation, the hardness assumptions and security analysis are given in the full version.

Capturing the OT-CRF in [21]. Below we show that our framework indeed encompasses the construction in [21]. Precisely, in [21] the basis chosen by the receiver is (g, c) and the chosen element is $C_0 = (d, h)$, where $d = g^y, h = c^y g^b$. We have that.

$$\boldsymbol{\Gamma} = (g, c), \quad \widetilde{\boldsymbol{S}} = \begin{pmatrix} \alpha & \alpha x' \\ 0 & \alpha \end{pmatrix}, \quad \widetilde{\boldsymbol{w}} = y',$$

$$\widetilde{\boldsymbol{\Gamma}} = \boldsymbol{\Gamma} \odot \widetilde{\boldsymbol{S}} = (g^{\alpha}, c^{\alpha} g^{\alpha x'}), \quad C_0' = C_0 \odot \widetilde{\boldsymbol{S}} = (d^{\alpha}, h^{\alpha} d^{\alpha x'}),$$

$$C = \widetilde{\boldsymbol{w}} \odot \widetilde{\boldsymbol{\Gamma}} = (g^{\alpha y'}, c^{\alpha y'} g^{\alpha x' y'}), \quad \widetilde{C_0} = C_0' \oplus C = (d^{\alpha} g^{\alpha y'}, h^{\alpha} d^{\alpha x'} c^{\alpha y'} g^{\alpha x' y'}).$$

One can note that the transformation of $\boldsymbol{\Gamma}$ adopted here is via shearing and uniform scaling as:

$$\widetilde{\boldsymbol{S}} = \begin{pmatrix} \alpha & \alpha x' \\ 0 & \alpha \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha \end{pmatrix} \odot \begin{pmatrix} 1 & x' \\ 0 & 1 \end{pmatrix}.$$

It is clear that other parts of protocol also follow the above framework.

Improving the Efficiency of [21]**.** As mentioned above, we can construct a more efficient $\mathcal{W}_B$ based on the DDH assumption by applying the globally non-uniform scaling of $\boldsymbol{\Gamma}$. Specifically, suppose the element basis provided by the receiver is (g, c) and the chosen element is $C_0 = (d, h)$, where $d = g^y, h = c^y c^b$. We have

$$\boldsymbol{\Gamma} = (g, c) \in \mathbb{G}^{1\times 2}, \quad \widetilde{\boldsymbol{S}} = \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix} \in \mathbb{Z}_p^{2\times 2}, \quad \widetilde{\boldsymbol{w}} = y',$$

$$\widetilde{\boldsymbol{\Gamma}} = \boldsymbol{\Gamma} \odot \widetilde{\boldsymbol{S}} = (g^{s_1}, c^{s_2}), \quad C_0' = C_0 \odot \widetilde{\boldsymbol{S}} = (d^{s_1}, h^{s_2}),$$

$$C = \widetilde{\boldsymbol{w}} \odot \widetilde{\boldsymbol{\Gamma}} = (g^{s_1 y'}, c^{s_2 y'}), \quad \widetilde{C_0} = C_0' \oplus C = (d^{s_1} g^{s_1 y'}, h^{s_2} c^{s_2 y'}).$$

Instantiation from k-Linear Assumption. We now show the construction of CRF for the above protocol. We only show the construction of $\mathcal{W}_B$ since $\mathcal{W}_A$ can be easily obtained from the k-linear assumption based instantiation of malleable SPHF shown in Sect. 3.2. Specifically, we have

$$\boldsymbol{\Gamma} = \begin{pmatrix} g_1 & 1 & \cdots & 1 & g_{k+1} \\ 1 & g_2 & \cdots & 1 & g_{k+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & g_k & g_{k+1} \end{pmatrix}, \quad \widetilde{\boldsymbol{S}} = \begin{pmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{k+1} \end{pmatrix},$$

$$\widetilde{\boldsymbol{\Gamma}} = \boldsymbol{\Gamma} \odot \widetilde{\boldsymbol{S}} = \begin{pmatrix} g_1^{s_1} & 1 & \cdots & 1 & g_{k+1}^{s_{k+1}} \\ 1 & g_2^{s_2} & \cdots & 1 & g_{k+1}^{s_{k+1}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & g_k^{s_k} & g_{k+1}^{s_{k+1}} \end{pmatrix}, \; C_0' = C_0 \odot \widetilde{\boldsymbol{S}} = (c_1^{s_1}, c_2^{s_2}, ..., c_{k+1}^{s_{k+1}}),$$

$$\widetilde{\boldsymbol{w}} = (r_1', r_2', ..., r_{k+1}') \in \mathbb{Z}_p^k, \quad C = \widetilde{\boldsymbol{w}} \odot \widetilde{\boldsymbol{\Gamma}} = (g_1^{s_1 r_1'}, g_2^{s_2 r_2'}, ..., g_{k+1}^{s_{k+1} \sum_{i=1}^{k} r_i'}),$$

$$\widetilde{C_0} = C_0' \oplus C = (c_1^{s_1} g_1^{s_1 r_1'}, c_2^{s_2} g_2^{s_2 r_2'}, \cdots, c_k^{s_k} g_k^{s_k r_k'}, c_{k+1}^{s_{k+1}} g_{k+1}^{s_{k+1} \sum_{i=1}^{k} r_i'}).$$

6 Conclusion

In this work, we presented generic CRF constructions for several widely used cryptographic protocols based on a new notion named *malleable smooth projective hash function*, which is an extension of the SPHF with new properties. We showed how to generically construct CRFs via malleable SPHFs in a modular way. Specifically, we proposed generic constructions of CRFs for the unkeyed message-transmission protocol and the OSBE protocol. We further studied the OT protocol and developed a new OT framework from graded rings and showed how to construct OT-CRFs via a modified version of the malleable SPHF framework.

Acknowledgements. We would like to thank the anonymous reviewers for their invaluable comments on a previous version of this paper. Dr. Guomin Yang is supported by the Australian Research Council Discovery Early Career Researcher Award (Grant No. DE150101116). Dr. Mingwu Zhang is supported by the National Natural Science Foundation of China (Grant No. 61370224 and Grant No. 61672010).

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: Disjunctions for hash proof systems: new constructions and applications. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 69–100. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46803-6_3
2. Abdalla, M., Chevalier, C., Pointcheval, D.: Smooth projective hashing for conditionally extractable commitments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 671–689. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03356-8_39
3. Alwen, J., Shelat, A., Visconti, I.: Collusion-free protocols in the mediated model. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 497–514. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85174-5_28
4. Ateniese, G., Magri, B., Venturi, D.: Subversion-resilient signature schemes. In: ACM CCS, pp. 364–375 (2015)
5. Balfanz, D., Durfee, G., Shankar, N., Smetters, D.K., Staddon, J., Wong, H.: Secret handshakes from pairing-based key agreements. In: S&P, pp. 180–196 (2003)
6. Bellare, M., Hoang, V.T.: Resisting randomness subversion: fast deterministic and hedged public-key encryption in the standard model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 627–656. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46803-6_21
7. Bellare, M., Jaeger, J., Kane, D.: Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks. In: ACM CCS, pp. 1431–1440 (2015)
8. Bellare, M., Paterson, K.G., Rogaway, P.: Security of symmetric encryption against mass surveillance. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 1–19. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44371-2_1
9. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: New techniques for SPHFs and efficient one-round PAKE protocols. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 449–475. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40041-4_25
10. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (1998). doi:10.1007/BFb0054122
11. Blazy, O., Pointcheval, D., Vergnaud, D.: Round-optimal privacy-preserving protocols with smooth projective hash functions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 94–111. Springer, Heidelberg (2012). doi:10.1007/978-3-642-28914-9_6
12. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). doi:10.1007/3-540-46035-7_4
13. Dodis, Y., Ganesh, C., Golovnev, A., Juels, A., Ristenpart, T.: A formal treatment of backdoored pseudorandom generators. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 101–126. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46800-5_5
14. Dodis, Y., Mironov, I., Stephens-Davidowitz, N.: Message transmission with reverse firewalls—secure communication on corrupted machines. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 341–372. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53018-4_13
15. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003). doi:10.1007/3-540-39200-9_33

16. Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. J. Crypt. **25**(1), 158–193 (2012)
17. Katz, J., Vaikuntanathan, V.: Round-optimal password-based authenticated key exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19571-6_18
18. Larson, J., Perlroth, N., Shane, S.: Revealed: The NSAs Secret Campaign to Crack, Undermine Internet Security. Pro-Publica, New York (2013)
19. Lepinski, M., Micali, S., Shelat, A.: Collusion-free protocols. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005, pp. 543–552 (2005)
20. Li, N., Du, W., Boneh, D.: Oblivious signature-based envelope. In: PODC, pp. 182–189 (2003)
21. Mironov, I., Stephens-Davidowitz, N.: Cryptographic reverse firewalls. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 657–686. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46803-6_22
22. Perlroth, N., Larson, J., Shane, S.: NSA Able to Foil Basic Safeguards of Privacy on Web. The New York Times (2013)
23. Rogaway, P.: The moral character of cryptographic work. IACR Crypt. ePrint Arch. **2015**, 1162 (2015)
24. Russell, A., Tang, Q., Yung, M., Zhou, H.: Cliptography: clipping the power of kleptographic attacks. IACR Crypt. ePrint Arch. **2015**, 695 (2015)
25. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). doi:10.1007/11426639_7
26. Wee, H.: KDM-security via homomorphic smooth projective hashing. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016. LNCS, vol. 9615, pp. 159–179. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49387-8_7
27. Yang, R., Xu, Q., Zhou, Y., Zhang, R., Hu, C., Yu, Z.: Updatable hash proof system and its applications. In: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015. LNCS, vol. 9326, pp. 266–285. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24174-6_14
28. Young, A., Yung, M.: The dark side of "Black-Box" cryptography or: should we trust capstone? In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 89–103. Springer, Heidelberg (1996). doi:10.1007/3-540-68697-5_8