A Note on Extractable Witness Encryption in Generic Groups

Matteo Campanelli

Offchain Labs & University of Tartu, Estonia
binarywhalesinternaryseas@gmail.com

October 27, 2026

Abstract. Hair and Sahai (arXiv:2609.18275v1) have recently proposed a construction of witness encryption for NP satisfying semantic security. In this note we observe that this recent witness-encryption construction is unconditionally *extractable* in the generic group model. This yields a construction of extractable witness encryption without assuming subexponential soundness of SNARGs, in contrast to the recent work of Jin (ePrint 2026/2063), which requires this assumption. The result in this note, moreover, complements the concurrent work of Chen and Vaikuntanathan (ePrint 2026/2346), who show the same construction to be public-coin extractable under a knowledge-style assumption; our argument is unconditional in the GGM.

1 Introduction

Witness encryption (WE) is a powerful cryptographic primitive [GGSW13]. Given an NP language L, a witness encryption scheme permits to encrypt a message with respect to a statement x. Anyone holding a witness certifying that $x \in L$ can decrypt the message. Its notion of semantic security informally states that, if $x \notin L$ then encryptions of any two messages with respect to x are computationally indistinguishable. Applications of WE include identity-based encryption [Sha84, BF01] and attribute-based encryption [SW05, GPSW06] for circuits [GVW13, GGH$^+$13b] and round-collapsing in MPC [BL20].

Hair and Sahai [HS26] have recently proposed the first construction of witness encryption (WE) for the NP-complete problem SAT that is "direct"—it does not go through indistinguishability obfuscation [SW14], a primitive that implies WE—and does not require unproven conjectures. Hair and Sahai prove their construction to be semantically secure in the *generic-group model* (GGM) [Nec94, Sho97, Mau05], where access to the group elements is replaced by "handles" and a *generic* adversary can perform group operations only through a stateful oracle.

This work: extractable WE from generic groups. In addition to the "baseline" security notion of witness encryption (stated above), WE schemes may guarantee an even stronger property, *extractability* [GKP$^+$13, GGHW14]. Intuitively, an extractable WE is one where the ability to distinguish encrypted messages implies

knowledge of a corresponding witness. This notion can be *naturally* employed[1] in several applications [CDK+22, CFK24], has been shown to be related to laconic private set-intersection [BGJP25] and forms of registered encryption [GHK+25], and implies extractability obfuscation [BCP14].

We show that the construction in [HS26] is an extractable witness encryption scheme. Our result holds unconditionally in the GGM.

1.1 Comparison with recent work

The recent work of Jin [Jin26] gives a construction of extractable WE for NP in the GGM but requires SNARGs for NP with subexponential soundness and additional efficiency properties. Another recent paper, the concurrent and independent work of Chen and Vaikuntanathan [CV26], shows that Hair and Sahai's construction is public-coin extractable under a knowledge-style assumption, namely the Knowledge of Orthogonality Assumption (KOALA). In contrast to Jin's work, our result does not require additional assumptions on SNARG security. Our result complements that of Chen and Vaikuntanathan, who require a specific assumption but can dispense with generic groups. We note that their model differs slightly from ours. Finally, their assumption guarantees the existence of a potentially *nonuniform* extractor, whereas we provide a single explicit, uniform extraction procedure that runs the adversary and observes its generic-group queries, without requiring its code.

1.2 Other related work

Garg, Gentry, Sahai and Waters introduced witness encryption and proposed a first instantiation from multilinear maps [GGSW13]. Gentry, Lewko and Waters later obtained witness encryption from instance-independent assumptions on multilinear maps [GLW14]. Subsequent cryptanalysis has broken candidate multilinear maps and constructions built on them [CHL+15, MSZ16]; their security remains poorly understood.

Other avenues for building witness encryption have used lattices [CVW18, Tsa22, VWW22] and affine determinant programs [BIJ+20, SRG+26]. Some of the candidate assumptions and constructions proposed in these works have been shown to be broken [BÜW24, DJM+25, GRSS26, Jo26]. Witness encryption from standard assumptions is currently known only via indistinguishability obfuscation [GGH+13a, JLS21, JLS22, RVV24].

The most notable precursor for the techniques in the work of Hair and Sahai [HS26] is the work of Barta, Ishai, Ostrovsky and Wu [BIOW20]. They introduce a construction of WE from generic groups (under an unproven conjecture) and tie the question of building witness encryption to hardness-of-approximation problems.

[1] The standard notion of semantic security for witness encryption, in contrast, is quite bespoke: it holds only for false statements and thus cannot easily be used for, e.g., trivial languages (where every string has a corresponding witness).

Extractable witness encryption was introduced in [GKP+13]. Subsequent work ([GGHW14, BCP14]) has shown the implausibility of the strongest forms of this primitive—where security is guaranteed for any auxiliary information—and its connection to extractability obfuscation, a notion strongly related to differing-input obfuscation [GGHW14, CFO23]. Extractable witness encryption has also been shown to be equivalent to specific forms of cryptographic arguments (*predictable arguments*) [FNV17, Kho22].

Recent works have also studied weaker forms of witness encryption[2] and related primitives, aiming for efficient constructions that can be instantiated from standard assumptions [BL20, CDK+22, CGMN23, CFK24, FHAS24, GHK+25]. With the exception of the work in [CDK+22], all these works present constructions with some form of extractability (albeit sometimes quite restricted as in [CFK24]).

2 Technical Overview

2.1 The WE construction of Hair and Sahai

Reducing SAT to Boolean-factor MinRank. Hair and Sahai reduce a SAT instance φ of size N to an instance of the promise problem *Boolean-factor MinRank*. The latter consists of a basis of matrices $M_1, \ldots, M_k$ whose span is $S = \left\{ \sum_{j=1}^{k} a_j M_j : \boldsymbol{a} \in \mathbb{F}_p^k \right\}$ and whose properties depend on whether φ is satisfiable or not:

- **YES instances:** if φ is satisfiable then S contains a nonzero matrix $\boldsymbol{u}\boldsymbol{v}^\top$, with $\boldsymbol{v}$ Boolean and $\boldsymbol{u} \in \mathbb{F}_p^m$. Moreover, given a satisfying assignment w of φ, it is possible to efficiently find $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{a}$ such that $\boldsymbol{u}\boldsymbol{v}^\top = \sum_j a_j M_j \in S$.
- **NO instances:** if φ is unsatisfiable then every nonzero matrix in S has rank at least $1 + \lfloor \log N \rfloor$.

The WE scheme. We work in a generic group $\mathbb{G}$ (see Section 3) of prime order p. We denote its generator by $[1]$ and use additive notation for group operations. Given $x \in \mathbb{F}_p$, we write $[x] = x[1]$. For a vector $\boldsymbol{x} = (x_1, \ldots, x_\ell)$, we define $[\boldsymbol{x}] = ([x_1], \ldots, [x_\ell])$.

Encryption. In order to encrypt a bit $\beta \in \{0, 1\}$, we:

- sample a uniform vector $\boldsymbol{s} \in \mathbb{F}_p^m$, where m is the side length of each basis matrix M_j;
- independently sample $\boldsymbol{r} = (r_1, \ldots, r_m)$ with each r_j uniform in $\{0, \ldots, m-1\}$;

[2] Specifically, "deterministic" forms of WE, where the witness (or a related cryptographic proof) used for decryption appears in a commitment known at encryption time.

– produce a ciphertext $\mathsf{ct} = (\boldsymbol{X}, \boldsymbol{Y})$ with $\boldsymbol{X} = [\boldsymbol{s}]$. If $\beta = 1$, the entries of $\boldsymbol{Y}$ are independent uniform elements of $\mathbb{G}$, also independent of $\boldsymbol{s}$ and $\boldsymbol{r}$; otherwise $\boldsymbol{Y} = ([\boldsymbol{s}^\top M_1 \boldsymbol{r}], \ldots, [\boldsymbol{s}^\top M_k \boldsymbol{r}])$, one group element per basis matrix.

Decryption. The decryption algorithm recognizes whether the ciphertext $\mathsf{ct} = (\boldsymbol{X}, \boldsymbol{Y})$ contains a random $\boldsymbol{Y}$ (encrypting 1) or a "structured" $\boldsymbol{Y}$ (encrypting 0). It tests for the latter case by building two group elements, H from $\boldsymbol{X}$ and Z from $\boldsymbol{Y}$, and searching for a relation between them. If such a relation holds, decryption returns 0; otherwise it returns 1.

Recall that, given a satisfying assignment w of φ, it is possible to efficiently find $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{a}$ such that $\boldsymbol{u}\boldsymbol{v}^\top = \sum_j a_j M_j \in S$. We compute $Z = \sum_i a_i Y_i$. By the property above and by inspecting what we did during encryption, whenever the encrypted message is the bit 0, this holds:

$$Z = \sum_i a_i Y_i = [\, \underbrace{(\boldsymbol{s}^\top \boldsymbol{u})}_{f} \underbrace{(\boldsymbol{v}^\top \boldsymbol{r})}_{f'} \,],$$

that is, Z is the group encoding of the product of two factors, f and f', defined as above.

What the decryptor now does is to use $\boldsymbol{X}$ to check whether this is actually the shape of the Z it computed. It proceeds in two steps, *i)* first by building the group encoding of the first factor f and then *ii)* by searching for the second factor f':

i) it computes $H = \sum_j u_j X_j$. Clearly $H = [\boldsymbol{s}^\top \boldsymbol{u}] = [f]$.

ii) it checks whether there exists $\hat{f}' \in \{0, 1, \ldots, B\}$, where $B = (m-1)\,\mathrm{wt}(\boldsymbol{v})$, such that $Z = \hat{f}' H$. If so, decryption returns 0; otherwise it returns 1.

We can search efficiently for $f' = \boldsymbol{v}^\top \boldsymbol{r}$ because it is small: as an integer, it lies in $\{0, \ldots, B\}$ with $B = O(m^2)$, since $\boldsymbol{v}$ is Boolean and the entries of $\boldsymbol{r}$ lie in $\{0, \ldots, m-1\}$.

On correctness error. Notice that this test can fail, though only in one direction. When the encrypted bit is 0 the element Z has exactly the shape above and the search always succeeds; when it is 1, however, Z is a uniformly random group element, and nothing prevents it from accidentally coinciding with $\hat{f}' H$ for one of the $B + 1$ candidates. By a union bound, this happens with probability at most $(B+1)/p$, which is negligible.

2.2 On extractability

Extractability guarantees that, given the description of a distinguishing adversary, we can efficiently find a satisfying assignment with probability at least the adversary's advantage, up to negligible error. In the GGM, the adversary

manipulates ciphertext group elements through an oracle (we adopt Maurer's GGM [Mau05]). It sees only placeholders for these elements ("handles") and can build new handles from old ones. Every handle it builds represents an affine expression

$$c + \boldsymbol{\alpha}^\top \mathbb{X} + \boldsymbol{\gamma}^\top \mathbb{Y}$$

where $\mathbb{X}$ and $\mathbb{Y}$ are formal variables representing the exponents of the ciphertext elements $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. The coefficients can be tracked from the adversary's queries.

One key intuition, common in GGM-based extractors, is that the extractor can act as the group oracle and observe the adversary's queries. Creating new handles gives the adversary no information about the encrypted bit. Distinguishing requires a successful zero test on a nonzero affine form: the evaluated exponent is zero even though its coefficients are not all zero.

Our extraction approach proceeds as follows: first, the extractor runs the adversary on an encryption of 0 and records the coefficients of its queries. This is the informative case for a good distinguisher, since only encryptions of 0 contain "structured" group elements; see Lemma 4. With probability at least the adversary's advantage minus negligible error, the first successful zero test on a nonzero affine form yields a nonzero matrix $M_{\boldsymbol{\gamma}} = \sum_j \gamma_j M_j$ of rank at most $R = \lfloor \log N \rfloor$. From such a matrix, we can efficiently construct a polynomial-size list containing a satisfying assignment and test its candidates. Hair and Sahai use a similar observation in contrapositive form to establish semantic security for false statements; here it provides the witness we need for extraction.

3 Preliminaries

All logarithms have base two. For a prime p, let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$. Vectors are columns, and all matrix ranks and linear spans are over $\mathbb{F}_p$. For $\boldsymbol{v} \in \{0,1\}^m$, let $\mathrm{wt}(\boldsymbol{v})$ denote its Hamming weight. A function $\nu\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if, for every positive polynomial q, one has $\nu(\lambda) < 1/q(\lambda)$ for all sufficiently large λ. An algorithm is PPT if it is probabilistic polynomial time.

Let $\mathcal{R}$ be an NP witness relation, that is, one that is decidable in polynomial time and satisfies $|w| \leq \mathrm{poly}(|x|)$ whenever $(x, w) \in \mathcal{R}$, and let

$$L_{\mathcal{R}} = \{x : \exists w,\ (x, w) \in \mathcal{R}\}.$$

It suffices to give witness encryption for Boolean circuit satisfiability. A verifier for $\mathcal{R}$ compiles in polynomial time into a circuit φ_x, preserving witnesses both ways, so a scheme for circuit satisfiability yields one for $\mathcal{R}$, and an extractor that outputs a satisfying assignment of φ_x outputs a witness for x after the same compilation. From here on the statement is a Boolean circuit φ and a *satisfying assignment* of φ is an input w with $\varphi(w) = 1$.

3.1 Generic Cyclic Groups

We work in Maurer's formulation [Mau05] of the generic group model. For the prime p, the oracle keeps a table T of field elements indexed by *handles*, initially

empty, together with a counter, initially 0. A handle holding $x \in \mathbb{F}_p$ stands for the group element with discrete logarithm x, written $[x]$, so that $[1]$ is the generator and $[0]$ the identity, and we use additive notation throughout. Every party interacts with T through three queries.

- *Assignment.* On $x \in \mathbb{F}_p$, the oracle stores x at the handle given by the counter, returns that handle, and increments the counter.
- *Group operation.* On $(i_1, i_2, a_1, a_2) \in \mathbb{Z}^2 \times \mathbb{F}_p^2$, if handles i_1 and i_2 hold x_1 and x_2 the oracle stores $a_1x_1 + a_2x_2$ at the handle given by the counter, returns that handle, and increments the counter; if either of i_1, i_2 is unassigned it returns $\perp$ and the counter does not move.
- *Zero test.* On $i \in \mathbb{Z}$, if handle i holds x the party receives 1 when $x = 0$ and 0 otherwise; if handle i is unassigned it receives $\perp$.

We write Q for the total number of assignment and group-operation queries a party makes. Zero tests do not count toward Q. Group elements are never handed to a party: they live in T, and a party names one by its handle. A single group-operation query already forms an arbitrary linear combination of two handles, so scalar multiples and negations each require one group-operation query.

Handles are thus issued in the order of the assignment and group-operation queries, and a handle, once assigned, keeps its value. By convention the encryptor queries first, so that the generator receives handle 0 and the ciphertext handles $1, \dots, m + k$, and that layout is public. Advice may depend on the statement, but not on the encryption randomness.

3.2 Boolean-Factor Matrix Reduction

Let φ be a Boolean circuit. Fix its Boolean quadratic encoding

$$q_1(z) = \cdots = q_J(z) = 0, \qquad z \in \{0,1\}^{t_\varphi},$$

where the equations impose gate consistency and acceptance. An AND gate gives $z_k - z_i z_j = 0$, a NOT gate gives $z_k + z_i - 1 = 0$, and acceptance gives $z_{\text{out}} - 1 = 0$. A satisfying assignment of φ extends efficiently to a solution z of these equations, and every solution yields one.

Theorem 1 (Boolean-factor matrix reduction [HS26]). *Let N be an integer with $N \geq \max\{2, |\varphi|, t_\varphi\}$, and define*

$$R = \lfloor \log N \rfloor, \qquad m = (N+1)(2NR+1)\binom{2R}{R}.$$

For every prime $p > \max\{2^N, 2NR\}$, a deterministic algorithm

$$(M_1, \dots, M_k) \leftarrow \mathsf{Reduce}(\varphi, 1^N, p)$$

computes an ordered basis of a space $\mathcal{S}_{\varphi,N,p} \subseteq \mathbb{F}_p^{m \times m}$ in time $\mathrm{poly}(N, \log p)$. The following properties hold.

(i) *For every satisfying assignment w of φ, a deterministic polynomial-time algorithm*

$$(\boldsymbol{a}, \boldsymbol{u}, \boldsymbol{v}) \leftarrow \mathsf{Lift}(\varphi, 1^N, p, w)$$

outputs $\boldsymbol{a} \in \mathbb{F}_p^k$ and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}_p^m$ such that

$$\sum_{j=1}^{k} a_j M_j = \boldsymbol{u}\boldsymbol{v}^\top \neq 0, \qquad \boldsymbol{v} \in \{0,1\}^m, \qquad \mathrm{wt}(\boldsymbol{v}) \leq N+1.$$

(ii) *If φ is unsatisfiable, every nonzero $M \in \mathcal{S}_{\varphi,N,p}$ has $\mathrm{rank}(M) \geq R+1$.*

In particular, $m = O(N^4 \log N)$, $k \leq m^2$, and $2^R \leq N$.

4 The Witness Encryption Construction in [HS26]

The construction is that of Hair and Sahai [HS26, Section 3]. For an input $(1^\lambda, \varphi)$, define

$$n = \max\{2, \lambda, |\varphi|\}, \qquad N = \max\{n, t_\varphi\},$$
$$R = \lfloor \log N \rfloor, \qquad m = (N+1)(2NR+1)\binom{2R}{R}.$$

Pad the quadratic encoding with unused variables to obtain N variables. The prime p is a parameter of the scheme and is required to satisfy $2^m \leq p < 2^{m+1}$.

Encryption $\mathsf{Enc}(1^\lambda, \varphi, b)$. On input a message $b \in \{0,1\}$, execute the following steps.

1. For the given prime p, compute

$$(M_1, \ldots, M_k) \leftarrow \mathsf{Reduce}(\varphi, 1^N, p).$$

2. Independently sample

$$\boldsymbol{s} \leftarrow \mathbb{F}_p^m, \qquad \boldsymbol{r} \leftarrow \{0, \ldots, m-1\}^m, \qquad \boldsymbol{\eta} \leftarrow \mathbb{F}_p^k,$$

   drawing all three for either message, including the one the branch below does not use.
3. By assignment queries, in this order, assign 1, then s_i for $i \in \{1, \ldots, m\}$, and then, for $j \in \{1, \ldots, k\}$, the value

$$\begin{cases} \boldsymbol{s}^\top M_j \boldsymbol{r}, & b = 0, \\ \eta_j, & b = 1, \end{cases}$$

   receiving handles $0, 1, \ldots, m+k$ in return. The ciphertext ct is the resulting state of T; we write X_i for the group element at handle i and Y_j for the one at handle $m+j$.

Decryption $\mathsf{Dec}(1^\lambda, \varphi, w, \mathsf{ct})$. Execute the following steps.

1. If w is not a satisfying assignment of φ, return $\perp$. If ct is malformed, return $\perp$.
2. Use the handles of $X_1, \dots, X_m, Y_1, \dots, Y_k$, namely $1, \dots, m+k$, and compute

$$(M_1, \dots, M_k) \leftarrow \mathsf{Reduce}(\varphi, 1^N, p), \qquad (\boldsymbol{a}, \boldsymbol{u}, \boldsymbol{v}) \leftarrow \mathsf{Lift}(\varphi, 1^N, p, w).$$

3. Compute

$$H = \sum_{i=1}^{m} u_i X_i, \qquad Z = \sum_{j=1}^{k} a_j Y_j, \qquad B = (m-1)\,\mathrm{wt}(\boldsymbol{v}).$$

4. For $t \in \{0, 1, \dots, B\}$ form $Z - tH$ and zero-test it. Return 0 if any of these tests succeeds, and 1 otherwise.

Theorem 2 (Correctness and false-statement security [HS26, Proposition 3.3 and Theorem 3.5]). *For every satisfying assignment w of φ and every $b \in \{0,1\}$,*

$$\Pr\big[\mathsf{Dec}(1^\lambda, \varphi, w, \mathsf{Enc}(1^\lambda, \varphi, b)) \neq b\big] \leq \big(m(N+1)+1\big)2^{-m} = 2^{-\Omega(m)}.$$

For every unsatisfiable φ and every generic adversary $\mathcal{A}$ making at most Q assignment and group-operation queries in total,

$$\begin{aligned}\big|\Pr\big[\mathcal{A}(1^\lambda, \varphi, \mathsf{Enc}(1^\lambda, \varphi, 0)) = 1\big] - \Pr\big[\mathcal{A}(1^\lambda, \varphi, \mathsf{Enc}(1^\lambda, \varphi, 1)) = 1\big]\big| \\ \leq O\Big((Q+m+k)^2\left(m^{-(R+1)} + 2^{-m}\right)\Big).\end{aligned}$$

The probabilities include the encryption randomness and the coins of the algorithms.

5 Extractable Witness Encryption: Definition

A witness-encryption scheme for Boolean circuit satisfiability is a pair of uniform probabilistic polynomial-time algorithms

$$\mathsf{ct} \leftarrow \mathsf{Enc}(1^\lambda, \varphi, b), \qquad b' \leftarrow \mathsf{Dec}(1^\lambda, \varphi, w, \mathsf{ct}),$$

where $b \in \{0,1\}$ and $b' \in \{0,1,\perp\}$. We fix a polynomial ℓ and consider circuits with $|\varphi| \leq \ell(\lambda)$. An *adversary* is a non-uniform probabilistic polynomial-time algorithm, that is, a family of circuits of size polynomial in λ; its advice, its total number of assignment and group-operation queries, and its input length are all bounded by that size. Write

$$\mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) = \big|\Pr[\mathcal{A}(1^\lambda, \varphi, \mathsf{Enc}(1^\lambda, \varphi, 0)) = 1] - \Pr[\mathcal{A}(1^\lambda, \varphi, \mathsf{Enc}(1^\lambda, \varphi, 1)) = 1]\big|.$$

Definition 1 (Correctness). *The scheme is* correct *if there is a negligible function κ such that for every λ, every circuit φ with $|\varphi| \le \ell(\lambda)$, every satisfying assignment w of φ and every $b \in \{0,1\}$,*

$$\Pr\big[\mathsf{Dec}(1^\lambda, \varphi, w, \mathsf{Enc}(1^\lambda, \varphi, b)) = b\big] \ge 1 - \kappa(\lambda).$$

Definition 2 (Extractability). *The scheme is* extractable *if for every efficient (possibly non-uniform) adversary $\mathcal{A}$ there are an efficient extractor E and a negligible function $\nu_{\mathcal{A}}$ such that*

$$\Pr\big[\varphi(w) = 1 \;\big|\; w \leftarrow E(1^\lambda, \varphi)\big] \ge \mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) - \nu_{\mathcal{A}}(\lambda) \tag{1}$$

for every $\lambda \in \mathbb{N}$ and every circuit φ with $|\varphi| \le \ell(\lambda)$.

Remark 1. In the generic-group variant, the probabilities in the extractability definition also range over the fresh encryption randomness, and E answers the table queries of the adversary it runs from a table of its own.

Remark 2 (Uniformity of the extractor). Although Definition 2 asks only for $\forall\mathcal{A}\,\exists E$, we give a uniform polynomial-time procedure that maps a description of $\mathcal{A}_\lambda$ to a description of the corresponding extractor E_λ. The extractor uses the adversary's code and advice; the procedure constructing it is the same for every adversary.

6 Unconditional Extractability of the WE in [HS26] in the GGM

Theorem 3 (Extractability). *The construction of Section 4 is correct in the sense of Definition 1, with error at most $2^{-\Omega(m)}$, and extractable in the sense of Definition 2, with Figure 1 as the extractor E. If $\mathcal{A}$ makes at most Q assignment and group-operation queries in total, then for each circuit φ the extraction error is bounded by*

$$\nu_{\mathcal{A}}(\lambda, \varphi) = O\Big((Q + m + k)\left(m^{-(R+1)} + 2^{-m}\right)\Big). \tag{2}$$

Proof. Correctness follows from Theorem 2: its error bound is negligible uniformly over $|\varphi| \le \ell(\lambda)$, since $m \ge 2N^2 \ge 2\lambda^2$. We prove extractability. The argument revolves around two main facts:

1. on any $M_{\boldsymbol{\gamma}}$ (see Fig. 1) such that $\boldsymbol{\gamma} \neq 0$ and $\mathrm{rank}(M_{\boldsymbol{\gamma}}) \le R$, the call $\mathsf{Recover}(\varphi, 1^N, p, M_{\boldsymbol{\gamma}})$ outputs a satisfying assignment of φ;
2. by recording the coefficients of the adversary's group oracle queries, E finds such a $\boldsymbol{\gamma}$ with probability at least $\mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) - \nu_{\mathcal{A}}(\lambda, \varphi)$.

The first fact follows from Lemma 1 applied to the span of the basis, an observation also used in [HS26], albeit in contrapositive form, to prove standard semantic security. We use the same observation but with the explicit goal of

Extractor $E(1^\lambda, \varphi)$, with $\mathcal{A}_\lambda$ hardwired

1. Compute N, R, m as in Section 4.
2. $(M_1, \ldots, M_k) \leftarrow \mathsf{Reduce}(\varphi, 1^N, p)$ for the given prime p.
3. Sample $\boldsymbol{s} \leftarrow \mathbb{F}_p^m$ and $\boldsymbol{r} \leftarrow \{0, \ldots, m-1\}^m$ and assign an encryption of 0: handle 0 gets 1, handle i gets s_i, and handle $m+j$ gets $\boldsymbol{s}^\top M_j \boldsymbol{r}$.
4. Run $\mathcal{A}_\lambda$ on $(1^\lambda, \varphi)$, answering its queries as the oracle would, and record alongside each handle the coefficient vector of the affine form it holds, handles $0, \ldots, m+k$ starting from $1, \mathbb{X}_1, \ldots, \mathbb{X}_m, \mathbb{Y}_1, \ldots, \mathbb{Y}_k$.
5. For every zero test that returns 1 on a handle whose form $F = c + \boldsymbol{\alpha}^\top \mathbb{X} + \boldsymbol{\gamma}^\top \mathbb{Y}$ is nonzero:
   (a) set $M_{\boldsymbol{\gamma}} = \sum_{j=1}^k \gamma_j M_j$;
   (b) if $\boldsymbol{\gamma} = 0$ or $\mathrm{rank}(M_{\boldsymbol{\gamma}}) \geq R+1$, continue;
   (c) $w \leftarrow \mathsf{Recover}(\varphi, 1^N, p, M_{\boldsymbol{\gamma}})$ via Lemma 1;
   (d) if $\varphi(w) = 1$, return w.
6. Return $\perp$.

Fig. 1. The extractor of Theorem 3. Affine forms are defined in its proof. The recovery algorithm $\mathsf{Recover}$ is provided by Lemma 1.

building an extractor. The second fact relies on the generic group model and basic linear algebra: a distinguishing adversary must make some zero test succeed on a handle it built as a combination whose coefficients are not all zero, and by Lemma 2 those coefficients implicitly describe a matrix $M_{\boldsymbol{\gamma}}$ of rank at most R, except with negligible probability. The extractor records the coefficients of each handle as it answers the queries, so it can read off the corresponding $\boldsymbol{\gamma}$ when such a zero test succeeds.

Fix a circuit φ and the parameters N, R, m, k of Section 4. Let p be the given prime, $(M_1, \ldots, M_k) \leftarrow \mathsf{Reduce}(\varphi, 1^N, p)$ the basis, and $\boldsymbol{s}, \boldsymbol{r}$ the encryption randomness used by the $b = 0$ branch. Let $h = m + k + Q + 2$.

Let $\mathbb{X} = (\mathbb{X}_1, \ldots, \mathbb{X}_m)$ and $\mathbb{Y} = (\mathbb{Y}_1, \ldots, \mathbb{Y}_k)$ be formal variables for the exponents of $X_1, \ldots, X_m$ and $Y_1, \ldots, Y_k$. An *affine form* is a vector $F = (c, \boldsymbol{\alpha}, \boldsymbol{\gamma}) \in \mathbb{F}_p \times \mathbb{F}_p^m \times \mathbb{F}_p^k$, written $F = c + \boldsymbol{\alpha}^\top \mathbb{X} + \boldsymbol{\gamma}^\top \mathbb{Y}$. It records how, through group oracle queries, a handle was built out of handle 0 and the ciphertext, and the party that built the handle knows its coefficients. Writing $M_{\boldsymbol{\gamma}} = \sum_{j=1}^k \gamma_j M_j$, the exponent such a handle carries when the encrypted message is 0 is

$$\mathrm{ev}_0(F) = c + \boldsymbol{s}^\top \boldsymbol{\alpha} + \boldsymbol{s}^\top M_{\boldsymbol{\gamma}} \boldsymbol{r}.$$

Let $\mathcal{T} = (F_0, F_1, \ldots)$ be the list of affine forms of Lemma 3. Up to the first successful zero test on a nonzero affine form, F_i records how handle i was built in the real execution.

Fact 1: we can obtain a witness from a low-rank $M_{\boldsymbol{\gamma}}$. Let $\boldsymbol{\gamma} \in \mathbb{F}_p^k$ satisfy $\boldsymbol{\gamma} \neq 0$ and $\mathrm{rank}(M_{\boldsymbol{\gamma}}) \leq R$. The basis $M_1, \ldots, M_k$ is linearly independent, so $M_{\boldsymbol{\gamma}} \neq 0$, and it lies in $\mathcal{S}_{\varphi,N,p}$ by definition of that span. It is therefore a nonzero element

of $\mathcal{S}_{\varphi,N,p}$ of rank at most R. We can thus apply Lemma 1, and use the algorithm $\mathsf{Recover}(\varphi, 1^N, p, M_{\boldsymbol{\gamma}})$ to obtain a satisfying assignment of φ. So whenever step 5 of Fig. 1 reaches such a $\boldsymbol{\gamma}$, the test in step 5(d) passes and E returns a satisfying assignment.

Fact 2: E finds a suitable $\boldsymbol{\gamma}$ whenever $\mathcal{A}$ is a good distinguisher. Write Succ for the event in which $E(1^\lambda, \varphi)$ successfully returns a w with $\varphi(w) = 1$. We prove

$$\Pr[\mathsf{Succ}] \geq \mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) - \nu_{\mathcal{A}}(\lambda, \varphi), \qquad \nu_{\mathcal{A}}(\lambda, \varphi) = h\big(m^{-(R+1)} + 2p^{-1}\big). \quad (3)$$

We first show that a successful zero test with a suitable $\boldsymbol{\gamma}$ occurs with the required probability, then check that E records this $\boldsymbol{\gamma}$ and passes it to step 5. We define two main events:

– a "vanishing" event (under encryption of 0), one where the adversary is effectively obtaining some new information through a successful zero test:

$$\mathsf{Van}_0: \quad \mathcal{A} \text{ zero-tests a handle } i \text{ with } F_i \neq 0 \text{ and } \mathrm{ev}_0(F_i) = 0.$$

  We call the form tested at the first such successful query the *witnessing form*. This is the event of Lemma 3, so the witnessing form lies in $\mathcal{T}$ and the union bound below may range over $\mathcal{T}$.
– an "accident", one where the vanishing happens for a reason that carries no low-rank matrix with it:

$$\mathsf{Acc}: \quad \mathsf{Van}_0 \wedge \big(\boldsymbol{\gamma} = 0 \text{ or } \mathrm{rank}(M_{\boldsymbol{\gamma}}) \geq R + 1\big).$$

  The case $\boldsymbol{\gamma} = 0$ is a form that involves no $\mathbb{Y}$ coordinate, so $M_{\boldsymbol{\gamma}} = 0$; the case $\mathrm{rank}(M_{\boldsymbol{\gamma}}) \geq R+1$ is one whose matrix sits above the rank threshold. Neither is of use to Fact 1.

What we want is a vanishing that is not an accident. The target is therefore $\mathsf{Van}_0 \wedge \neg\mathsf{Acc}$, on which the witnessing form has a $\boldsymbol{\gamma}$ meeting the hypothesis of Fact 1.

We obtain the above from two bounds. The first is Lemma 4, which says that distinguishing forces a success in the $b = 0$ world: $\Pr[\mathsf{Van}_0] \geq \mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) - hp^{-1}$.

The second bound is $\Pr[\mathsf{Acc}] \leq h(m^{-(R+1)} + p^{-1})$. To prove it, we need to apply Lemma 2 to fixed forms. The forms queried in the real execution are chosen adaptively: the adversary's choices depend on oracle answers, which in turn depend on $(\boldsymbol{s}, \boldsymbol{r})$. Instead, we use the symbolic list $\mathcal{T}$ from Lemma 3. This list is independent of the encryption randomness and contains the form witnessing Van_0. Thus, if Acc occurs, some nonzero form in $\mathcal{T}$ with $\boldsymbol{\gamma} = 0$ or $\mathrm{rank}(M_{\boldsymbol{\gamma}}) \geq R + 1$ must vanish.

Condition on $\mathcal{T} = T$. Every form in T is now fixed, while $(\boldsymbol{s}, \boldsymbol{r})$ retain their original independent distributions. For each nonzero $F = (c, \boldsymbol{\alpha}, \boldsymbol{\gamma}) \in T$ of the above kind, Lemma 2 gives

$$\Pr[\mathrm{ev}_0(F) = 0 \mid \mathcal{T} = T] \leq \begin{cases} p^{-1}, & \boldsymbol{\gamma} = 0, \\ m^{-(R+1)} + p^{-1}, & \mathrm{rank}(M_{\boldsymbol{\gamma}}) \geq R + 1. \end{cases}$$

Indeed, in the first case $M_{\boldsymbol{\gamma}} = 0$ and $(c, \boldsymbol{\alpha}) \neq (0, 0)$, so the second clause of the lemma applies. In the second case its first clause gives $m^{-\operatorname{rank}(M_{\boldsymbol{\gamma}})} + p^{-1} \leq m^{-(R+1)} + p^{-1}$. A union bound over the at most h forms in T therefore yields

$$\Pr[\mathsf{Acc} \mid \mathcal{T} = T] \leq h\big(m^{-(R+1)} + p^{-1}\big).$$

This bound is the same for every T, so averaging over $\mathcal{T}$ gives the claimed unconditional bound.

Combining the two bounds, we obtain

$$\Pr[\mathsf{Van}_0 \wedge \neg\mathsf{Acc}] \geq \Pr[\mathsf{Van}_0] - \Pr[\mathsf{Acc}] \geq \mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) - \nu_{\mathcal{A}}(\lambda, \varphi),$$

where $\nu_{\mathcal{A}}(\lambda, \varphi) = h(m^{-(R+1)} + 2p^{-1})$: one hp^{-1} term comes from Lemma 4, and the remaining terms come from the accident bound. On this event, $\mathcal{A}$ queries a form whose $\boldsymbol{\gamma}$ satisfies the hypothesis of Fact 1.

Since $h = O(Q + m + k)$ and $p \geq 2^m$, this error has the form stated in (2). To see that it is negligible, recall that h is polynomial in λ. Also, $m \geq 2N^2$ and $R = \lfloor \log N \rfloor$ imply

$$m^{-(R+1)} \leq N^{-2(R+1)} \leq 2^{-2\log^2 N}.$$

Because $N \geq \lambda$, both this term and $p^{-1} \leq 2^{-m}$ are negligible in λ, even after multiplication by h. For parameter values where $\nu_{\mathcal{A}}(\lambda, \varphi)$ exceeds one, the probability lower bound is vacuous.

Steps 3 and 4 of Fig. 1 run the real encryption-of-0 experiment. By Lemma 3, the coefficients E records agree with $\mathcal{T}$ up to the query witnessing Van_0. Thus E can read $\boldsymbol{\gamma}$ directly from the coefficient vector $F_i = (c, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ of the tested handle. On $\mathsf{Van}_0 \wedge \neg\mathsf{Acc}$, step 5 processes this vector and Fact 1 gives a satisfying assignment. Hence $\Pr[\mathsf{Succ}] \geq \Pr[\mathsf{Van}_0 \wedge \neg\mathsf{Acc}]$, which proves (3).

The error bound is negligible uniformly over $|\varphi| \leq \ell(\lambda)$. Taking its maximum over these circuits and all admissible primes gives a negligible function $\nu_{\mathcal{A}}(\lambda)$ independent of φ, as required by Definition 2.

The extractor E runs $\mathcal{A}_\lambda$ once and performs polynomial work per query. Each call to $\mathsf{Recover}$ takes $2^R \operatorname{poly}(N, \log p) \leq N \operatorname{poly}(N, \log p)$ time by Lemma 1. Thus E runs in polynomial time.

□

6.1 Supporting Lemmas

Fix φ, N, R, m, p and the basis $M_1, \ldots, M_k$ from Theorem 1. Affine forms, $M_{\boldsymbol{\gamma}}$ and ev_0 are as in the proof of Theorem 3, and

$$\mathrm{ev}_1(F) = c + \boldsymbol{\alpha}^\top \boldsymbol{s} + \boldsymbol{\gamma}^\top \boldsymbol{\eta}$$

is the exponent a handle carries when the encrypted message is 1. The properties below hold for arbitrary φ, including satisfiable instances.

Lemma 1 (Recovery from a low-rank matrix). *There is a deterministic algorithm* Recover *such that, for every circuit φ and every nonzero $M \in \mathcal{S}_{\varphi,N,p}$ of rank $\rho \le R$,*

$$w \leftarrow \mathsf{Recover}(\varphi, 1^N, p, M)$$

is a satisfying assignment of φ, in time $2^\rho \operatorname{poly}(N, \log p)$, which is $\operatorname{poly}(N, \log p)$ since $2^\rho \le 2^R \le N$.

Proof. This fact is provided by the explicit candidate-list conclusion at the end of [HS26, Section 4.5, p. 18]. The approach there is as follows. The reduction of Theorem 1 builds matrices with $N+1$ columns and appends $m-(N+1)$ zero columns to make them square, which leaves the rank unchanged [HS26, Section 4.4]. Discard those columns of M and let $\boldsymbol{c}_0, \ldots, \boldsymbol{c}_N$ be the ones that remain, in the ordering the reduction fixes. Choose a column basis $\boldsymbol{c}_{i_1}, \ldots, \boldsymbol{c}_{i_\rho}$ and write

$$\boldsymbol{c}_i = \sum_{j=1}^{\rho} \gamma_{ij} \boldsymbol{c}_{i_j}, \qquad L_i(\boldsymbol{y}) = \sum_{j=1}^{\rho} \gamma_{ij} y_j \quad (0 \le i \le N),$$

both obtained from a single Gaussian elimination, which also yields ρ. The algorithm $\mathsf{List}(\varphi, 1^N, p, M)$ outputs

$$\mathcal{L}(M) = \big\{(L_1(\boldsymbol{e}), \ldots, L_N(\boldsymbol{e})) : \boldsymbol{e} \in \{0,1\}^\rho\big\} \cap \{0,1\}^N,$$

a list of at most 2^ρ Boolean vectors, and [HS26] shows that it contains a solution of the padded quadratic encoding of φ. Testing the list against that encoding and keeping the circuit-input coordinates of the first vector that passes yields a satisfying assignment of φ, which is the output of Recover. The enumeration dominates the cost and runs for $2^\rho \le 2^R \le N$ iterations. □

Lemma 2 (Rank-sensitive collision bound). *Let $M \in \mathbb{F}_p^{m \times m}$ have rank ρ, and fix $c \in \mathbb{F}_p$ and $\boldsymbol{\alpha} \in \mathbb{F}_p^m$. For independent uniform $\boldsymbol{s} \leftarrow \mathbb{F}_p^m$ and $\boldsymbol{r} \leftarrow \{0, \ldots, m-1\}^m$, where $m < p$,*

$$\Pr\big[c + \boldsymbol{s}^\top \boldsymbol{\alpha} + \boldsymbol{s}^\top M \boldsymbol{r} = 0\big] \le m^{-\rho} + p^{-1}.$$

If $M = 0$ and $(c, \boldsymbol{\alpha}) \ne (0, 0)$, the probability is at most p^{-1}.

Proof. This is [HS26, Lemma 3.4, p. 8]. Its proof picks ρ independent columns of M and fixes the coordinates of $\boldsymbol{r}$ outside them; at most one choice of the remaining coordinates gives $M\boldsymbol{r} = -\boldsymbol{\alpha}$, and their grid values are distinct field elements because $m < p$, so that event has probability at most $m^{-\rho}$. Off it the expression is a nonconstant affine form in uniform $\boldsymbol{s}$ and vanishes with probability p^{-1}. □

The following lemma lets us analyze the adversary using a symbolic execution that does not depend on the encryption randomness. This execution agrees with the real one until a nonzero affine form passes a zero test (so any distinguishing advantage must come from such a vanishing event).

Lemma 3. *Let $\mathcal{A}$ make at most Q assignment and group-operation queries in total and let $h = m+k+Q+2$. Consider a symbolic execution of $\mathcal{A}$ in which handles store affine forms in $\mathbb{X}, \mathbb{Y}$ instead of their evaluated exponents. The initial handles store $1, \mathbb{X}_1, \ldots, \mathbb{X}_m$, and $\mathbb{Y}_1, \ldots, \mathbb{Y}_k$, in that order; assignment queries store constant forms, group operations combine forms, and a zero test returns 1 exactly when the tested form has all coefficients zero.*

Let $\mathcal{T} = (F_0, F_1, \ldots)$ list the forms in this execution, in handle order. It has at most h entries and, for fixed public inputs, depends only on the coins of $\mathcal{A}$. In particular, $\mathcal{T}$ is independent of $(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{\eta})$.

For each $b \in \{0,1\}$, run $\mathcal{A}$ on a real encryption of b using the same adversary coins as in the symbolic execution. The two executions make the same queries and issue the same handles until the first zero test on a handle i with $F_i \neq 0$ but $\mathrm{ev}_b(F_i) = 0$. Up to and including this query, every real handle i holds $\mathrm{ev}_b(F_i)$. At this query the real zero test returns 1 while the symbolic zero test returns 0. Write

$$\mathsf{Van}_b : \quad \textit{such a zero-test query occurs in the real execution.}$$

If Van_b does not occur, the two executions have identical queries, answers and outputs. Consequently,

$$\mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) \leq \Pr[\mathsf{Van}_0] + \Pr[\mathsf{Van}_1], \tag{4}$$

Proof. First consider the symbolic execution alone. It starts with $m + k + 1$ forms, and each assignment or group-operation query adds at most one form. Thus $|\mathcal{T}| \leq m + k + Q + 1 \leq h$. All symbolic answers are determined by the coefficient vectors and the order in which handles are issued. Once the public inputs and the coins of $\mathcal{A}$ are fixed, these answers fix its subsequent queries, even though $\mathcal{A}$ is adaptive. The entire list $\mathcal{T}$ is therefore fixed without using any encryption randomness, proving its independence of $(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{\eta})$.

Now fix b, the coins of $\mathcal{A}$, and the encryption randomness, and compare the symbolic and real executions. Initially, the real handle corresponding to each form F_i holds $\mathrm{ev}_b(F_i)$ by the definition of encryption. As long as the answers agree, $\mathcal{A}$ makes the same next query in both executions. We check that this agreement and the correspondence of handle values are preserved:

- An assignment of x stores the constant form x in the symbolic table and the exponent $x = \mathrm{ev}_b(x)$ in the real table. Both return the same next handle.
- A group operation with coefficients a_1, a_2 stores $a_1 F_{i_1} + a_2 F_{i_2}$ in the symbolic table. Its evaluation is $a_1 \,\mathrm{ev}_b(F_{i_1}) + a_2 \,\mathrm{ev}_b(F_{i_2})$, exactly the exponent stored in the real table, and both return the same next handle.
- A zero test on an assigned handle returns 1 symbolically if $F_i = 0$, in which case $\mathrm{ev}_b(F_i) = 0$ and the real answer is also 1. If $F_i \neq 0$, both answers are 0 unless $\mathrm{ev}_b(F_i) = 0$. In that case the symbolic answer is 0 and the real answer is 1: this is precisely the query witnessing Van_b.

Queries involving unassigned handles return $\perp$ in both executions, since the same handles have been issued. Induction on the queries proves the claimed agreement

up to the first differing zero-test answer, including the value of the tested handle. If Van_b never occurs, the executions receive identical answers throughout and produce the same output.

Finally, let π be the probability that the symbolic execution outputs 1, and let π_b be that probability in the real encryption-of-b experiment. The symbolic execution does not use b, so the same π applies to both messages. With the shared adversary coins above, the outputs can differ only on Van_b, giving

$$|\pi_b - \pi| \le \Pr[\text{the two executions have different outputs}] \le \Pr[\mathsf{Van}_b].$$

The triangle inequality now yields

$$\begin{aligned}\mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) &= |\pi_0 - \pi_1| \\ &\le |\pi_0 - \pi| + |\pi_1 - \pi| \le \Pr[\mathsf{Van}_0] + \Pr[\mathsf{Van}_1].\end{aligned}$$

A similar comparison with a symbolic execution appears in [HS26, proof of Theorem 3.5, pp. 8–9]. □

Lemma 4 (Distinguishing forces a vanishing event in the $b = 0$ world). *In the setting of Lemma 3,*

$$\mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) \le \Pr[\mathsf{Van}_0] + hp^{-1}.$$

Proof. By Lemma 3,

$$\mathrm{Adv}_{\mathcal{A}}(\lambda, \varphi) \le \Pr[\mathsf{Van}_0] + \Pr[\mathsf{Van}_1].$$

It therefore suffices to prove $\Pr[\mathsf{Van}_1] \le hp^{-1}$.

To handle the adversary's adaptive queries, condition on the symbolic list $\mathcal{T} = T$ of Lemma 3. This fixes all forms in T. Since $\mathcal{T}$ is independent of the encryption randomness, $(\boldsymbol{s}, \boldsymbol{\eta})$ remains uniform on $\mathbb{F}_p^{m+k}$ after conditioning.

For any nonzero $F = (c, \boldsymbol{\alpha}, \boldsymbol{\gamma}) \in T$, we have

$$\mathrm{ev}_1(F) = c + \boldsymbol{\alpha}^\top \boldsymbol{s} + \boldsymbol{\gamma}^\top \boldsymbol{\eta}.$$

If some entry of $\boldsymbol{\alpha}$ or $\boldsymbol{\gamma}$ is nonzero, fix all coordinates of $(\boldsymbol{s}, \boldsymbol{\eta})$ except the corresponding one. Exactly one value of that remaining uniform coordinate makes $\mathrm{ev}_1(F) = 0$, so the probability is p^{-1}. If both vectors are zero, then $c \neq 0$ because $F \neq 0$, and the form never vanishes. In either case,

$$\Pr[\mathrm{ev}_1(F) = 0 \mid \mathcal{T} = T] \le p^{-1}.$$

If Van_1 occurs, Lemma 3 ensures that the form tested at the first differing answer belongs to T. Thus some nonzero form in T must evaluate to zero. A union bound over its at most h forms gives

$$\Pr[\mathsf{Van}_1 \mid \mathcal{T} = T] \le \sum_{\substack{F \in T \\ F \neq 0}} \Pr[\mathrm{ev}_1(F) = 0 \mid \mathcal{T} = T] \le hp^{-1}.$$

Averaging over $\mathcal{T}$ yields $\Pr[\mathsf{Van}_1] \le hp^{-1}$, which proves the claim when substituted into the first inequality. □

Acknowledgments. Thanks to Janno Siim for useful discussions around this topic. This work was supported by the Estonian Research Council grant PRG2531

References

BCP14. Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 52–73. Springer, Berlin, Heidelberg, February 2014.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Berlin, Heidelberg, August 2001.

BGJP25. James Bartusek, Sanjam Garg, Abhishek Jain, and Guru-Vamsi Policharla. Laconic PSI on authenticated inputs and applications. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part V*, volume 16249 of *LNCS*, pages 262–293. Springer, Singapore, December 2025.

BIJ+20. James Bartusek, Yuval Ishai, Aayush Jain, Fermi Ma, Amit Sahai, and Mark Zhandry. Affine determinant programs: A framework for obfuscation and witness encryption. In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 82:1–82:39. LIPIcs, January 2020.

BIOW20. Ohad Barta, Yuval Ishai, Rafail Ostrovsky, and David J. Wu. On succinct arguments and witness encryption from groups. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 776–806. Springer, Cham, August 2020.

BL20. Fabrice Benhamouda and Huijia Lin. Mr NISC: Multiparty reusable non-interactive secure computation. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part II*, volume 12551 of *LNCS*, pages 349–378. Springer, Cham, November 2020.

BÜW24. Chris Brzuska, Akin Ünal, and Ivy K. Y. Woo. Evasive LWE assumptions: Definitions, classes, and counterexamples. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 418–449. Springer, Singapore, December 2024.

CDK+22. Matteo Campanelli, Bernardo David, Hamidreza Khoshakhlagh, Anders Konring, and Jesper Buus Nielsen. Encryption to the future - A paradigm for sending secret messages to future (anonymous) committees. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part III*, volume 13793 of *LNCS*, pages 151–180. Springer, Cham, December 2022.

CFK24. Matteo Campanelli, Dario Fiore, and Hamidreza Khoshakhlagh. Witness encryption for succinct functional commitments and applications. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part II*, volume 14602 of *LNCS*, pages 132–167. Springer, Cham, April 2024.

CFO23. Matteo Campanelli, Danilo Francati, and Claudio Orlandi. Structure-preserving compilers from new notions of obfuscations. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 663–693. Springer, Cham, May 2023.

CGMN23. Matteo Campanelli, Rosario Gennaro, Kelsey Melissaris, and Luca Nizzardo. Witness-authenticated key exchange, revisited: Extensions to groups, improved models, simpler constructions. In Foteini Baldimtsi and Christian Cachin, editors, *FC 2023, Part I*, volume 13950 of *LNCS*, pages 112–128. Springer, Cham, May 2023.

CHL+15. Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 3–12. Springer, Berlin, Heidelberg, April 2015.

CV26. Liyan Chen and Vinod Vaikuntanathan. Public-coin differing-input obfuscation from KOALA and iO. Cryptology ePrint Archive, Paper 2026/2346, 2026. https://eprint.iacr.org/2026/2346.

CVW18. Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 577–607. Springer, Cham, August 2018.

DJM+25. Nico Döttling, Abhishek Jain, Giulio Malavolta, Surya Mathialagan, and Vinod Vaikuntanathan. Simple and general counterexamples for private-coin evasive LWE. Cryptology ePrint Archive, Report 2025/374, 2025.

FHAS24. Nils Fleischhacker, Mathias Hall-Andersen, and Mark Simkin. Extractable witness encryption for KZG commitments and efficient laconic OT. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part II*, volume 15485 of *LNCS*, pages 423–453. Springer, Singapore, December 2024.

FNV17. Antonio Faonio, Jesper Buus Nielsen, and Daniele Venturi. Predictable arguments of knowledge. In Serge Fehr, editor, *PKC 2017, Part I*, volume 10174 of *LNCS*, pages 121–150. Springer, Berlin, Heidelberg, March 2017.

GGH+13a. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

GGH+13b. Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 479–499. Springer, Berlin, Heidelberg, August 2013.

GGHW14. Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 518–535. Springer, Berlin, Heidelberg, August 2014.

GGSW13. Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

GHK+25. Sanjam Garg, Mohammad Hajiabadi, Dimitris Kolonelos, Abhiram Kothapalli, and Guru-Vamsi Policharla. A framework for witness encryption from linearly verifiable SNARKs and applications. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 504–539. Springer, Cham, August 2025.

GKP+13. Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run Turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. Springer, Berlin, Heidelberg, August 2013.

GLW14. Craig Gentry, Allison B. Lewko, and Brent Waters. Witness encryption from instance independent assumptions. In Juan A. Garay and Rosario

Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 426–443. Springer, Berlin, Heidelberg, August 2014.

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

GRSS26. Muhammad El Gebali, Yaroslav Rebenko, Markus Schofnegger, and Lev Soukhanov. Breaking ADP-based witness encryption. Cryptology ePrint Archive, Paper 2026/1594, 2026. https://eprint.iacr.org/2026/1594.

GVW13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

HS26. Isaac M. Hair and Amit Sahai. Witness encryption via prime-order generic groups. arXiv:2609.18275v1, 2026. https://arxiv.org/abs/2609.18275v1.

Jin26. Zhengzhong Jin. Witness encryption for NP from SNARGs and groups. Cryptology ePrint Archive, Paper 2026/2063, 2026. https://eprint.iacr.org/2026/2063.

JLS21. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021.

JLS22. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 670–699. Springer, Cham, May / June 2022.

Jo26. Sunghyeon Jo. Cryptanalysis of a candidate witness encryption scheme for affine determinant programs. Cryptology ePrint Archive, Paper 2026/1583, 2026. https://eprint.iacr.org/2026/1583.

Kho22. Hamidreza Khoshakhlagh. (Commit-and-prove) predictable arguments with privacy. In Giuseppe Ateniese and Daniele Venturi, editors, *ACNS 2022*, volume 13269 of *LNCS*, pages 542–561. Springer, Cham, June 2022.

Mau05. Ueli Maurer. Abstract models of computation in cryptography. In *Cryptography and Coding, 10th IMA International Conference*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.

MSZ16. Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 629–658. Springer, Berlin, Heidelberg, August 2016.

Nec94. Vassiliy Ilyich Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

RVV24. Seyoon Ragavan, Neekon Vafa, and Vinod Vaikuntanathan. Indistinguishability obfuscation from bilinear maps and LPN variants. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024, Part IV*, volume 15367 of *LNCS*, pages 3–36. Springer, Cham, December 2024.

Sha84. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Berlin, Heidelberg, August 1984.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Berlin, Heidelberg, May 1997.

SRG+26. Lev Soukhanov, Yaroslav Rebenko, Muhammad El Gebali, Mikhail Komarov, Handan Kilinc-Alper, Michel Abdalla, and Patrick Towa. Implementable witness encryption from arithmetic affine determinant programs. Cryptology ePrint Archive, Paper 2026/175, 2026. https://eprint.iacr.org/2026/175.

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Berlin, Heidelberg, May 2005.

SW14. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

Tsa22. Rotem Tsabary. Candidate witness encryption from lattice techniques. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 535–559. Springer, Cham, August 2022.

VWW22. Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 195–221. Springer, Cham, December 2022.