# Contents

1 Introduction  
2 Basic Preliminaries  
   2.1 Notations  
   2.2 Cryptographic Assumptions  
      2.2.1 Pairing groups and Matrix Diffie-Hellman Assumption  
   2.3 Digital Signatures  
   2.4 (Hierarchical) Identity-based Key Encapsulation  
3 Simplified Security Proofs for Signatures from Identification Schemes  
   3.1 Identification and Fiat-Shamir Transform  
      3.1.1 Canonical Identification Schemes  
      3.1.2 Signatures from Identification Schemes  
   3.2 Security Implications  
      3.2.1 Multi-Instance Reset Lemma  
      3.2.2 Proof of the Main Theorems  
   3.3 Instantiations  
      3.3.1 Schnorr Identification/Signature Scheme  
      3.3.2 Generalized CP Identification/ KW Signature Scheme  
      3.3.3 Guillou-Quisquater Identification/Signature Scheme  
4 Identity-Based Encryption from Affine MACs  
   4.1 Message Authentication Codes  
      4.1.1 Affine MACs  
      4.1.2 An Affine MAC from the Naor-Reingold PRF  
      4.1.3 An Affine MAC from Hash Proof System  
   4.2 Identity-based Encryption from Affine MACs  
   4.3 Hierarchical IBE from Delegatable Affine MACs  
      4.3.1 The Transformation  
      4.3.2 Anonymity-preserving Transformation  
   4.4 Identity-based Hash Proof System  
      4.4.1 Definitions  
      4.4.2 Construction  
      4.4.3 Security  
   4.5 Concrete Instantiation from SXDH  
5 More Efficient Structure-Preserving Signatures  
   5.1 Motivating Example: One-time Secure SPS  
   5.2 Main Construction: Fully UF-CMA-Secure SPS  
      5.2.1 Computational Core Lemma  
      5.2.2 Our Scheme  
   5.3 Security against Random Message Attacks  
      5.3.1 One-Time Secure SPS  
      5.3.2 Unbounded Secure SPS  
   5.4 Structure-Preserving Signatures for Bilateral Message Spaces  
      5.4.1 Two-Tier Signatures  
      5.4.2 Instantiation  
   5.5 Conclusions
Public-key cryptography, introduced by Diffie and Hellman \cite{DH76} in 1976, has been proved to be a useful tool for providing secure electronic communication without any pre-shared initial secret. In particular, it assures the confidentiality, authenticity and non-repudiability of communications and data storage. Public-key encryption (PKE), digital signature and identity-based encryption (IBE) schemes are important primitives in the field of public-key cryptography and they drew lots of attention in the past few years \cite{CS02,HK07,CHKW16,Cor02,AABN02,Wat05,HW09,BF01,BB04,ABB10,HMM15}.

Analysis of the security of the existing schemes is a non-easy but important task. The security flaw of a cryptographic scheme is usually not easy to identify, since a security flaw does not usually have any impact on the functionality of the scheme, but sensitive information can be still leaked. The leakage of sensitive information can have catastrophic consequences: for example, the leakage of voters’ identities in an e-voting system, or exfiltration of passwords in a widely used online banking system.

Thus, modern cryptographic research requires a sound and fully fledged security proof of a cryptographic scheme.

**Provable Security.** Most commonly, a security proof of a public-key cryptographic scheme denotes a security reduction showing that any successful adversary \( A \) attacking the scheme can be efficiently turned into another adversary \( B \) breaking the underlying well-established intractability problem. That means if there exists such an efficient adversary \( A \) then we can efficiently solve the intractability problem, which contradicts the well-established fact that the problem is computationally hard to be solved. Formally, the relation between \( A \)'s success probability \( \varepsilon_A \) and \( B \)'s success probability \( \varepsilon_B \) is as follows:

\[
\varepsilon_A \leq \ell \cdot \varepsilon_B + \negl(\lambda),
\]

where \( \ell \) is the security loss of the scheme and \( \negl(\cdot) \) is a negligible function, and the running times of \( A \) and \( B \) are roughly the same, \( t_A \approx t_B \).

Such security proof technique is known as *provable security* technique. This technique dates back to 1984 \cite{GM84} and is a fundamental security proof technique of modern cryptography \cite{Gol01,Gol04}, in particular for public-key cryptography.

This work focuses on provably secure digital signature and identity-based encryption (IBE) schemes. In particular, we improve the existing security proofs of the well-known and important signature schemes, and construct more efficient IBE and signature schemes with advanced properties. The first important step of provable security analysis is to give a rigorous security definition. Before discussing more technical backgrounds and motivations, we provide a brief overview of signature and IBE schemes as well as their security definitions.

**Digital Signature, IBE, and Their Security.** Digital signature schemes are one of the most fundamental public-key cryptographic primitives, which provide authenticity for digital messages or documents. They are building blocks of numerous advanced cryptographic protocols. Their standard security definition is unforgeability against chosen message attacks (UF-CMA) \cite{GMR88} in the single-user setting, where an adversary obtains one public-key and it is said to break the security of the scheme if he can output a forgery after seeing \( Q_a \) many signatures on messages of his choice. A valid forgery is a fresh message-signature pair that gets verified on the given public-key. In real world applications, the adversary is usually confronted with many public-keys and he can break the security if he produces a valid forgery under any of the given public-keys. This scenario is captured in the multi-user setting for signature schemes. Concretely, in multi-user unforgeability against chosen message attacks (MU-UF-CMA) the
attacker obtains $N_u$ independent public-keys, and is said to break the scheme’s security if he can produce (after obtaining $Q_s$ many signatures under public-keys of his choice) a valid forgery that verifies under any of the public-keys.

The identity-based encryption (IBE) scheme [Sha84] is a good example for the multi-user setting. An IBE scheme enables a user to encrypt a message $m$ under a recipient’s identity $id$ (e.g., an e-mail address or a phone number), and decryption is carried by using a user’s secret key for $id$, obtained from a trusted authority. Since the encryption only uses the recipient’s identities, the cost of public-key management can be avoided, which becomes practical in the setting where a large amount of users are involved. The first instantiations of an IBE scheme were given in 2001 [Coc01,BF01,SOK00]. One of the main application of IBE schemes is to encrypt e-mails. Moreover, IBE schemes are one of the important primitives in the (theoretical) public-key cryptography, since they can be transferred to digital signature schemes via Naor’s transformation [BF01] and to CCA-secure public-key encryption schemes via the CHK transformation [CHK04,BK05]. Standard security definition for an IBE scheme is indistinguishability against chosen plaintext attacks (IND-CPA) [BF01], where, given a ciphertext encrypted with the challenge identity $id^*$, an adversary cannot learn any information about the corresponding plaintext. The challenge identity $id^*$ is chosen by the adversary after seeing the master public-key of the IBE scheme and $Q_{usk}$ many user secret keys for identities of his choice.

The concept of IBE generalizes naturally to hierarchical IBE (HIBE). In an $L$-level HIBE, hierarchical identities are vectors of identities of maximal length $L$ and user secret keys for a hierarchical identity can be delegated. An IBE is simply an $L$-level HIBE with $L = 1$.

**Tight Security.** We say that a scheme has (almost) tight security (or tight reduction) if $\ell$ in Equation (1.1) only linearly depends on the security parameter $\lambda$; otherwise, we say that the scheme has only loose security. Most of the existing schemes have only loose security. For example, the security loss of most of the classical signature schemes (such as Schnorr [Sch91], DSA and Hohenberger-Waters [HW09]) depends on $Q_s$ and $N_u$ (in the multi-user setting), and the security loss of most of the classical IBE schemes depends on $Q_{usk}$.

Many recent results show that designing public-key cryptographic schemes with tight reduction is a desirable goal with practical impacts [KK12,AFLT12,HJ12,ADK13,CW13,BKKP15,BHJ15,GHKW16]. In practice, it is more desirable to have schemes with tight security. The main reason is that a tightly-secure scheme is more efficient and requires shorter security parameters, since $\lambda$ is much smaller than $Q_s$ or $Q_{usk}$ or $N_u$. Consider a signature scheme with security loss $N_u \cdot Q_s$. It is realistic to assume $N_u = 2^{30}$ and $Q_s = 2^{30}$. (For example, there are on average 1.09 billion ($\approx 2^{30}$) daily active users on Facebook.com for March 2016 [FBS] and imagine that each login for each active user requires at least one signature. Then the adversary already sees $2^{30}$ many signatures). Imagine we require 80-bit security for the signature scheme: for a loose scheme, we require the underlying assumption with 140-bit hardness guarantee, since $\epsilon = 2^{30} \cdot 2^{30} \cdot \epsilon_g \leq 2^{-80} \iff \epsilon_g \leq 2^{-140}$; however, a tightly-secure scheme only requires the underlying assumption with 80-bit hardness. That implies that a tightly-secure scheme has shorter security parameters and is therefore more efficient.

Thus, we need to provide a concrete security analysis of the classical schemes so that we can determine precise security parameters in practice. Here we highlight the well-known Schnorr signatures [Sch91], which are arguably the most popular signature scheme used in practice. Moreover, we need to design new and efficient public-key primitives with tight security reductions based on standard hardness assumptions. This will provide concrete improvement on numerous advance cryptographic protocols that rely on those primitives.

**Efficient Modular Constructions.** Another research challenge in public-key cryptography is to give a generic construction (or transformation) of a public-key cryptographic scheme from a simple and low-level primitive. The advantage of modular design method is that, after having the generic transformation, we can just focus on the construction of low-level primitives, which are usually simple and easy to construct. As discussed before, we want schemes with tight security and better efficiency. Thus, we expect that our generic constructions to be tightness-preserving and efficient.

Examples are CCA-secure PKE from hash proof systems [CS02], as well as the digital signature scheme from message authentication codes (MACs) and non-interactive zero-knowledge (NIZK) proof systems [BG90]. However, an exception to this design paradigm are IBE schemes. To the best of our knowledge, all known constructions of IBE (e.g., [Wat05,CLL13,ABB10]) are specific to certain cryptographic assumptions, (e.g., DBDH, SXDH, k-LIN or LWE). A recent work from Chen and Wee [CW13] makes use of a specific pseudorandom function (PRF) at the core of their scheme in a non-

\[\text{Here we ignore the negligible function.}\]
modular fashion. Thus, it is an interesting open problem to construct a tightness-preserving generic construction forIBE schemes from any low-level primitive.

An important example of modular design paradigm is structure-preserving signature (SPS) schemes [AFG+10], which are pairing-based digital signature schemes which sign group elements using only group operations. In particular, all the messages, signatures and public-keys of SPS are group elements and verification can only use pairing product equations. In combination with the Groth-Sahai NIZK proofs [GS08], SPS schemes serve as useful building blocks in modular design of cryptographic protocols, such as blind signatures [FV10], group signatures [Gro06,Gro07,LPY15], delegable anonymous credentials [Fuc11,BCC+09], compact verifiable shuffles [CKLM12], network encoding [ALP13], oblivious transfer [GRH08] and e-cash [BCF+11]. Unfortunately, existing SPS schemes based on standard assumptions (e.g. SXDH and k-LIN) are inefficient. For instance, to sign a single group element, the best construction under the SXDH (1-LIN) assumption contains 11 resp. 21 group elements in signatures resp. public-keys [ACD+12]. Thus, constructing more efficient SPS schemes will give concrete improvements on the aforementioned advanced cryptographic protocols.

**OUR CONTRIBUTIONS.** To address the challenges mentioned above, in this thesis we give a simplified security proof for digital signatures from identification schemes, propose a generic construction of IBE from affine message authentication codes, and construct improved and more efficient SPS schemes. In the following, we give a brief overview of our contributions and the detailed motivation.

- In [Chapter 3](#) we propose a simplified security proof of digital signature scheme from canonical identification schemes via the Fiat-Shamir transformation. Schnorr, GQ [GQ90], KW [KW03] signature schemes are well-known examples of this class of signature schemes. In particular, the Schnorr signature is widely used in practice. In the random oracle model, we show that if an identification scheme is key-recoverability (KR) secure and random self-reducible, then the resulting signature scheme is unforgeable against chosen-message attacks in the multi-user setting (aka. MU-UF-CMA-secure). Key-recoverability is the weakest security notion of an identification scheme. Our security reduction only loses a factor of $Q_h$, the number of random oracle queries, and without an additional multiplicative loss of $N_u$, the number of users in the system, compared with the previous reductions [GM02]. For GQ and KW signature schemes, we can even obtain tight security proofs (without going through KR security) directly based on the underlying assumption in the multi-user setting.

  Our analysis contains four main implications and each implication has a relatively simple and easily understandable proof. Moreover, [KMP16a](#) shows that our reductions are optimal in terms of model assumption (namely, the programmable random oracle model is required) and tightness (namely, the security reduction has to loose a factor of $Q_h$). That raises a new open problem to design a tightly-secure scheme without random oracles, since the security proof in the random oracle model is only heuristic [CGH98]. This problem will be solved by the follow-up chapter. This chapter originally appeared as parts of an extended abstract at CRYPTO 2016 [KMP16b](#).

- In [Chapter 4](#) we propose a new notion of message authentication codes (MACs), called affine MACs. We construct a generic transformation to IBE from any affine MAC in prime-order groups. We show that if an affine MAC is pseudorandom against chosen message attacks and, for example, the $k$-LIN assumption holds, then the IBE scheme is IND-ID-CPA-secure. Our transformation is tightness-preserving. We also show a tightly-secure affine MAC based on standard assumptions and this gives us a tightly-secure IBE without random oracles. Compared with the CW13 tightly-secure IBE [CW13](#), our scheme has shorter master public-keys, user secret-keys and ciphertexts. By Naor’s transformation, we obtain a tightly-secure digital signature scheme in the standard model, which answers the open problem raised in [Chapter 3](#).

  Furthermore, the transformation also extends to HIBE schemes and identity-based hash proof systems (IDHPS). This, among other things, provides the first tightly-secure IDHPS without “Q-type” assumptions, which implies a tightly CCA-secure IBE. Recently, our generic transformation (from an affine MAC to an IBE) has been extended to construct quasi-adaptive non-interactive zero-knowledge (QANIZK) proof systems for linear subspaces [KW15](#) and (linearly homomorphic) structure-preserving signature schemes [KW15,KPW15](#). This chapter originally appeared as an extended abstract in CRYPTO 2014 [BKPT14](#).

---

2It has been pointed out that their tight reduction has a flaw [Ber15](#).
In Chapter 5, we propose more efficient SPS schemes from the $k$-LIN assumption via a conceptually novel approach. Our approach contains three conceptually simple steps: we first construct an information-theoretically secure one-time secure MAC; motivated by the generic transformation in Chapter 4, we compile the one-time secure MAC to a one-time secure SPS; using the affine MAC in Chapter 4, we transfer the one-time secure SPS to a multiple-time secure SPS. Similar ideas extend to improve the efficiency of secure SPS against random message attacks and bilateral SPS.

The resulting scheme shortens the size of signatures from 11 group elements $[ACD+12]$ to 7 and the size of public-keys from $20 + n$ group elements to $6 + n$, where $n$ is the maximal number of group elements in a message. This chapter originally appeared as an extended abstract in CRYPTO 2015 [KPW15].
CHAPTER 2

BASIC PRELIMINARIES

2.1 Notations

For an integer \( p \), define \([p] := \{1, \ldots, p\} \) and \( \mathbb{Z}_p \) as the residual ring \( \mathbb{Z}/p\mathbb{Z} \). If \( B \) is a set, then \( x \overset{\$}{\in} B \) denotes the process of sampling an element \( x \) from set \( B \) uniformly at random. All our algorithms are probabilistic polynomial time unless stated otherwise. If \( A \) is an algorithm, then \( a \overset{\$}{\in} A(b) \) denotes the random variable which is defined as the output of \( A \) on input \( b \). To make the randomness explicit, we use the notation \( a := A(b; \rho) \) meaning that the algorithm is executed on input \( b \) and randomness \( \rho \). Note that \( A \)'s execution is now deterministic.

If \( x \in B^n \), then \(|x|\) denotes the length \( n \) of the vector. If \( A \in \mathbb{Z}_q^{n_1 \times n_2} \) is a matrix with \( n_1 > k \) and \( n_2 \geq 1 \), then \( \overrightarrow{A} \in \mathbb{Z}_q^{k \times n_2} \) denotes the upper matrix of \( A \) and then \( \overrightarrow{A} \in \mathbb{Z}_p^{(n_1-k) \times n_2} \) denotes the remaining \( n_1 - k \) rows of \( A \). We use \( \text{span}(\cdot) \) to denote the column span of a matrix.

**Code-based Game-Playing Proofs.** In this thesis, we use the code-based game-playing framework \cite{BR06} to present our security definitions and proofs, unless stated explicitly. In this framework, a game \( G \) is defined by procedures \textIT{Initialize} and \textIT{Finalize}, plus some optional procedures \( P_1, \ldots, P_n \). All procedures are given using pseudo-code, where initially all variables are undefined. An adversary \( A \) is executed in game \( G \) if it first calls \textIT{Initialize}, obtaining its output. Next, it may make arbitrary queries to \( P_i \) (according to their specification), again obtaining their output. Finally, it makes one single call to \textIT{Finalize}(\cdot) and stops. We define \( G^A \) as the output of \( A \)'s call to \textIT{Finalize}.

In most of the proofs and definitions, parts of the code are boxed and boxed code is only executed in the games marked in the same box style at the top right of every procedure. Non-boxed code is always run.

2.2 Cryptographic Assumptions

2.2.1 Pairing groups and Matrix Diffie-Hellman Assumption

Let \( \textIT{GGen} \) be a probabilistic polynomial time (PPT) algorithm that on input \( 1^A \) returns a description \( G := (G_1, G_2, G_T, p, P_1, P_2, e) \) of asymmetric pairing groups where \( G_1, G_2, G_T \) are cyclic groups of order \( p \) for a \( \lambda \)-bit prime \( p \), \( P_1 \) and \( P_2 \) are generators of \( G_1 \) and \( G_2 \), respectively, and \( e : G_1 \times G_2 \) is an efficiently computable (non-degenerated) bilinear map. Define \( P_T := e(P_1, P_2) \), which is a generator in \( G_T \).

We use implicit representation of group elements as introduced in \cite{EHK+13}. For \( s \in \{1, 2, T\} \) and \( a \in \mathbb{Z}_p \) define \([a]_s = aP_s \in G_s\) as the implicit representation of \( a \) in \( G_s \). More generally, for a matrix \( A = (a_{ij}) \in \mathbb{Z}_p^{n \times m} \) we define \([A]_s \) as the implicit representation of \( A \) in \( G_s \):

\[
[A]_s := \begin{pmatrix}
(a_{11}P_s & \ldots & a_{1m}P_s) \\
(a_{n1}P_s & \ldots & a_{nm}P_s)
\end{pmatrix} \in G_s^{n \times m}
\]

We will always use this implicit notation of elements in \( G_s \), i.e., we let \([a]_s \in G_s \) be an element in \( G_s \). Note that from \([a]_s \in G_s \) it is generally hard to compute the value \( a \) (discrete logarithm problem in \( G_s \)). Further, from \([b]_T \in G_T \) it is hard to compute the value \([b]_1 \in G_1 \) and \([b]_2 \in G_2 \) (pairing inversion problem). Obviously, given \([a]_s \in G_s \) and a scalar \( x \in \mathbb{Z}_p \), one can efficiently compute \([ax]_s \in G_s \).
Further, given \([a],[b]\) one can efficiently compute \([ab]_T\) using the pairing \(e\). For two matrices \(A\) and \(B\) with matching dimensions define \(e([A],[B]) := [AB]_T \in G_T\).

We recall the definition of the matrix Diffie-Hellman (MDDH) assumption \([\text{EHK}^{+}13]\).

**Definition 2.2.1** (Matrix Distribution). Let \(k \in \mathbb{N}\). We call \(\mathcal{D}_k\) a matrix distribution if it outputs matrices in \(\mathbb{Z}_p^{(k+1) \times k}\) of full rank \(k\) in polynomial time.

Without loss of generality, we assume the first \(k\) rows of \(A \leftarrow \mathcal{D}_k\) form a full-rank and invertible matrix. The \(\mathcal{D}_k\)-Matrix Diffie-Hellman problem is to distinguish the two distributions \(([[A],[A]w])\) and \(([[A],[u]]\), where the probability is taken over \(G \leftarrow \mathcal{D}_k, w \leftarrow \mathbb{Z}_p^k\) and \(u \leftarrow \mathbb{Z}_p^{k+1}\).

**Definition 2.2.2** (\(\mathcal{D}_k\)-Matrix Diffie-Hellman Assumption \(\mathcal{D}_k\)-MDDH). Let \(\mathcal{D}_k\) be a matrix distribution and \(s \in \{1,2,T\}\). We say that the \(\mathcal{D}_k\)-Matrix Diffie-Hellman \((\mathcal{D}_k\)-MDDH) problem is \((t,\varepsilon)\)-hard relative to \(\mathcal{G}\) in group \(\mathbb{G}_s\) if for all PPT adversaries \(\mathcal{D}\) with running time \(t\) and

\[
\left| \Pr[G(A_s,[A]s)_s = 1] - \Pr[G(A_s,[Aw]_s) = 1] \right| \leq \varepsilon,
\]

where the probability is taken over \(G \leftarrow \mathcal{D}_k, A \leftarrow [A], w \leftarrow \mathbb{Z}_p^k, u \leftarrow \mathbb{Z}_p^{k+1}\).

**Definition 2.2.3** (\(m\)-fold \(\mathcal{D}_k\)-Matrix Diffie-Hellman Assumption \(\mathcal{D}_k\)-MDDH). Let \(m \geq 1\) and \(\mathcal{D}_k\) be a matrix distribution and \(s \in \{1,2,T\}\). We say that the \(m\)-fold \(\mathcal{D}_k\)-Matrix Diffie-Hellman problem is \((t,\varepsilon)\)-hard relative to \(\mathcal{G}\) in group \(\mathbb{G}_s\) if for all PPT adversaries \(\mathcal{D}\) with running time \(t\) and

\[
\left| \Pr[G(A_s,[A]s)_s = 1] - \Pr[G(A_s,[Aw]_s) = 1] \right| \leq \varepsilon,
\]

where the probability is taken over \(G \leftarrow \mathcal{D}_k, A \leftarrow [A], W \leftarrow \mathbb{Z}_p^{k \times m}, U \leftarrow \mathbb{Z}_p^{(k+1) \times m}\).

The \(m\)-fold \(\mathcal{D}_k\)-MDDH problem contains \(m\) independent instances of the \(\mathcal{D}_k\)-MDDH problem (with the same \(A\) but different \(w_s\)). By a hybrid argument one can show that the two problems are equivalent, where the reduction loses a factor \(m\). The following lemma gives a tight reduction.

**Lemma 2.2.4** (Random self-reducibility \([\text{EHK}^{+}13]\)). For any matrix distribution \(\mathcal{D}_k\), \(\mathcal{D}_k\)-MDDH is random self-reducible. In particular, for any \(m \geq 1\), if the \(\mathcal{D}_k\)-MDDH problem is \((t,\varepsilon)\)-hard in \(\mathbb{G}_s\) \((s \in \{1,2,T\})\), then the \(\mathcal{D}_k\)-MDDH problem is \((t',\varepsilon')\)-hard in \(\mathbb{G}_s\) where

\[
\varepsilon' \leq \varepsilon + \frac{1}{p-1}, \quad t \approx t'.
\]

The Kernel-Diffie-Hellman assumption \(\mathcal{D}_k\)-KerMDH \([\text{MRV}15]\) is a natural computational analogue of the \(\mathcal{D}_k\)-MDDH Assumption.

**Definition 2.2.5** (\(\mathcal{D}_k\)-Kernel Diffie-Hellman Assumption \(\mathcal{D}_k\)-KerMDH). Let \(\mathcal{D}_k\) be a matrix distribution and \(s \in \{1,2\}\). We say that the \(\mathcal{D}_k\)-Kernel Diffie-Hellman \((\mathcal{D}_k\)-KerMDH) problem is \((t,\varepsilon)\)-hard relative to \(\mathcal{G}\) in group \(\mathbb{G}_s\) if for all PPT adversaries \(\mathcal{D}\),

\[
\Pr[cA = 0 \land c \neq 0 \mid c] = \varepsilon, \quad A(A_s,[A]s) \leq \varepsilon,
\]

where the probability is taken over \(G \leftarrow \mathcal{D}_k, A \leftarrow [A]\).

Note that we can use a non-zero vector in the kernel of \(A\) to test membership in the column space of \(A\). This means that the \(\mathcal{D}_k\)-KerMDH assumption is a relaxation of the \(\mathcal{D}_k\)-MDDH assumption, as captured in the following lemma from \([\text{MRV}15]\).

**Lemma 2.2.6**. For any matrix distribution \(\mathcal{D}_k\), \(\mathcal{D}_k\)-MDDH \(\Rightarrow\) \(\mathcal{D}_k\)-KerMDH.

For each \(k \geq 1\), \([\text{EHK}^{+}13]\) specifies distributions \(L_k, C_k, SC_k, IL_k\) such that the corresponding \(\mathcal{D}_k\)-MDDH assumption is the \(k\)-Linear assumption, the \(k\)-Cascade, the \(k\)-Symmetric Cascade, and the Incremental \(k\)-Linear Assumption, respectively. All assumptions are generically secure in bilinear groups and form a hierarchy of increasingly weaker assumptions. The distributions are exemplified for \(k = 2\), where \(a_1,\ldots,a_6 \in \mathbb{Z}_p^6\).

\[
C_2 : A = \begin{pmatrix} a_1 & 0 \\ 1 & a_2 \\ 0 & 1 \end{pmatrix}, \quad SC_2 : A = \begin{pmatrix} a_1 & 0 \\ 1 & a_1 \\ 0 & 1 \end{pmatrix}, \quad L_2 : A = \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \\ 1 & 1 \end{pmatrix}, \quad IL_2 : A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \\ a_5 & a_6 \end{pmatrix}.
\]
It was also shown in [EHK+13] that $\mathcal{U}_k$-MDDH is implied by all other $\mathcal{D}_k$-MDDH assumptions. Then $\text{RE} (\mathcal{S}_k) = 1$, $\text{RE} (\mathcal{L}_k) = k$ and $\text{RE} (\mathcal{U}_k) = k(k + 1)$. As shown in [EHK+13], $\mathcal{S}_k$-MDDH offers the same security guarantees as $\mathcal{L}_k$-MDDH (k-Linear Assumption of [HK07]), while having the advantage of a more compact representation. We define $k$-Lin := $\mathcal{L}_k$-MDDH and $k$-KerLin := $\mathcal{L}_k$-KerMDH. Note that $2$-KerLin = SDP (Simultaneous Double Pairing Assumption of [CLY09]). The relations between the different assumptions for $\mathcal{D}_k = \mathcal{L}_k$ are as follows:

\[
\begin{align*}
\text{DDH} & \quad \rightarrow \quad 2\text{-LIN} \quad \rightarrow \quad 3\text{-Lin} \quad \rightarrow \ldots \\
1\text{-KerLin} & \rightarrow \quad 2\text{-KerLin} = \text{SDP} \rightarrow \quad 3\text{-KerLin} \rightarrow \ldots \rightarrow \text{CDH}
\end{align*}
\]

### 2.3 Digital Signatures

We now define syntax and security of a digital signature scheme. Let $\text{par}$ be common system parameters shared among all participants.

**Definition 2.3.1** (Digital Signature). A digital signature scheme $\text{SIG}$ is defined as a triple of algorithms $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Ver})$.

- The key generation algorithm $\text{Gen}(\text{par})$ returns the public and secret keys $(pk \in \mathcal{P}, sk)$.
- The signing algorithm $\text{Sign}(sk, m \in \mathcal{M})$ returns a signature $\sigma \in \Sigma$.
- The deterministic verification algorithm $\text{Ver}(pk, m, \sigma)$ returns 1 (accept) or 0 (reject).

We require that for all $(pk, sk) \in \text{Gen}(\text{par})$, all messages $m \in \{0, 1\}^*$, we have $\text{Ver}(pk, m, \text{Sign}(sk, m)) = 1$.

\[
\begin{align*}
\text{Procedure INITIALIZE:} \\
&\text{For } i = 1, \ldots, N: \ (pk_i, sk_i) \leftarrow \text{Gen}(\text{par}) \\
&\text{Return } (pk_1, \ldots, pk_N)
\end{align*}
\]

\[
\begin{align*}
\text{Procedure SIGN}(i, m): \\
&\sigma \leftarrow \text{Sign}(sk_i, m); \ Q_M := Q_M \cup \{(i, m, \sigma)\} \\
&\text{Return } \sigma
\end{align*}
\]

\[
\begin{align*}
\text{Procedure FINALIZE}(m^*, \sigma^*): \\
&\text{Return } (\text{Ver}(pk_1, m^*, \sigma^*) = 1 \wedge (i^*, m^*, \sigma^*) \notin \{(i_j, m_j, \sigma_j) \mid j \in [Q_s]\})
\end{align*}
\]

**Figure 2.1:** Game Security MU-UFCMA for defining MU-UFCMA-security.

**Definition 2.3.2** (Multi-user Security). A signature scheme $\text{SIG}$ is said to be $(t, \varepsilon, N, Q_s)$-MU-SUF-CMA secure (multi-user strongly unforgeable against chosen message attacks) if for all adversaries $A$ running in time at most $t$ and making at most $Q_s$ queries to the signing oracle, $Pr[\text{MU-SUF-CMA}^A \Rightarrow 1] \leq \varepsilon$, where game MU-SUF-CMA is defined as in Figure 2.4.

We stress that an adversary in particular breaks multi-user security if he asks for a signature on message $m$ under $pk_1$ and submits a valid forgery on the same message $m$ under $pk_2$.

In [Definition 2.3.2] the condition “$\text{Ver}(pk_1, m^*, \sigma^*) = 1$” is called the correctness condition, the condition “$\{(i^*, m^*, \sigma^*) \notin \{(i_j, m_j, \sigma_j) \mid j \in [Q_s]\}\}$” is called the freshness condition. [Definition 2.3.3] covers strong security in the sense that a new signature on a previously queried message is considered as a fresh forgery. For standard (non-strong) MU-SUF-CMA security (multi-user unforgeability against chosen message attack) we modify the freshness condition in the experiment to $\{i^*, m^* \notin \{(i_j, m_j) \mid j \in [Q_s]\}\}$, i.e., to break the scheme the adversary has to come up with a signature on a message-key pair which has not been queried to the signing oracle. We also define $(t, \varepsilon, N)$-MU-KOA security (multi-user unforgeability against key-only attack) as $(t, \varepsilon, N, 0)$-MU-SUF-CMA security, i.e., $Q_s = 0$, the adversary is not allowed to make any signing query.

**Definition 2.3.3** (Single-user Security). In the single-user setting, i.e. $N = 1$ users, $(t, \varepsilon, Q_s)$-SUFCMA security (strong unforgeability against chosen message attacks) is defined as $(t, \varepsilon, 1, Q_s)$-MU-SUF-CMA security. Similarly, standard (non-strong) $(t, \varepsilon, Q_s)$-SUFCMA security (unforgeability against chosen message attack) is defined as $(t, \varepsilon, 1, Q_s)$-MU-SUF-CMA security. Further, $(t, \varepsilon)$-UF-KOA security (unforgeability against key-only attack) is defined as $(t, \varepsilon, 1, 0)$-MU-SUF-CMA security, i.e., $N = 1$ users and $Q_s = 0$ signing queries.
Security in the Random Oracle Model. The security of identification and signature schemes containing a hash function can be analyzed in the random oracle model [BR93]. In this model hash values can only be accessed by an adversary through queries to an oracle \( H \). On input \( x \) this oracle returns a uniformly random output \( H(x) \) which is consistent with previous queries for input \( x \). Using the random oracle model, the maximal number of queries to \( H \) becomes a parameter in the concrete security notions. For example, for \((t, \varepsilon, N, Q_{\star}, Q_{h})\)-MU-SUF-CMA security we consider all adversaries making at most \( Q_{h} \) queries to the random oracle. We make the convention that each query to the random oracle made during a signing query is counted as the adversary’s random oracle query, meaning \( Q_{h} \geq Q_{\star} \).

2.4 (Hierarchical) Identity-based Key Encapsulation

We now recall syntax and security of identity-based encryption in terms of an ID-based key encapsulation mechanism IBKEM. Every IBKEM can be transformed into an ID-based encryption scheme IBE using a (one-time secure) symmetric cipher.

**Definition 2.4.1** (Identity-based Key Encapsulation Scheme). An identity-based key encapsulation (IBKEM) scheme IBKEM consists of four PPT algorithms IBKEM = \((\text{Gen, USKGen, Enc, Dec})\) with the following properties.

- The probabilistic key generation algorithm \( \text{Gen}(1^\lambda) \) returns the (master) public/secret key \((pk, sk)\).
  - We assume that \( pk \) implicitly defines an identity space \( ID \), a key space \( K \), and ciphertext space \( C \).
- The probabilistic user secret key generation algorithm \( \text{USKGen}(sk, id) \) returns the user secret-key \( usk[id] \) for identity \( id \in ID \).
- The probabilistic encryption algorithm \( \text{Enc}(pk, id) \) returns the symmetric key \( K \in K \) together with a ciphertext \( C \in C \) with respect to identity \( id \).
- The deterministic decapsulation algorithm \( \text{Dec}(usk[id], id, C) \) returns the decapsulated key \( K \in K \) or the reject symbol \( \bot \).

For perfect correctness we require that for all \( \lambda \in \mathbb{N} \), all pairs \((pk, sk)\) generated by \( \text{Gen}(1^\lambda) \), all identities \( id \in ID \), all \( usk[id] \) generated by \( \text{USKGen}(sk, id) \) and all \((K, C)\) output by \( \text{Enc}(pk, id) \):

\[
\Pr\left[\text{Dec}(usk[id], id, C) = K \right] = 1.
\]

The security requirements for an IBKEM we consider here are indistinguishability and anonymity against chosen plaintext and identity attacks (IND-ID-CPA and ANON-ID-CPA). Instead of defining both security notions separately, we define pseudorandom ciphertexts against chosen plaintext and identity attacks (PR-ID-CPA) which means that challenge key and ciphertext are both pseudorandom. Note that PR-ID-CPA trivially implies IND-ID-CPA and ANON-ID-CPA.

We define PR-ID-CPA-security of IBKEM formally via the games given in Figure 2.2.

**Figure 2.2:** Security Games PR-ID-CPA\text{real} and PR-ID-CPA\text{rand} for defining PR-ID-CPA-security.

**Definition 2.4.2** (PR-ID-CPA Security). An identity-based key encapsulation scheme IBKEM is \((t, \varepsilon, Q)\)-PR-ID-CPA-secure if for all adversaries \( A \) with running time \( t \),

\[
| \Pr[\text{PR-ID-CPA}^A_{\text{real}} \Rightarrow 1] - \Pr[\text{PR-ID-CPA}^A_{\text{rand}} \Rightarrow 1] | \leq \varepsilon.
\]

We also recall syntax and security of a hierarchical identity-based key encapsulation mechanism (HIBKEM).

**Definition 2.4.3** (Hierarchical Identity-Based Key Encapsulation Mechanism). A hierarchical identity-based key encapsulation mechanism (HIBE) HIBKEM consists of three PPT algorithms HIBKEM = \((\text{Gen, USKDel}, \text{USKGen, Enc, Dec})\) with the following properties.
### (Hierarchical) Identity-based Key Encapsulation

#### Definition 2.4.4 (IND-HID-CPA and PR-HID-CPA Security)

A hierarchical identity-based key encapsulation scheme $\text{HIBKEM}$ is $(t, \varepsilon, Q)$-IND-HID-CPA-secure if for all $A$ with running time $t$,

$$|\Pr[PR-HID-CPA_{\text{real}}^A \Rightarrow 1] - \Pr[PR-HID-CPA_{\text{rand}}^A]| \leq \varepsilon.$$  

It is $(t, \varepsilon, Q)$-PR-HID-CPA-secure if for all $A$ with running time $t$,

$$|\Pr[PR-HID-CPA_{\text{real}}^A \Rightarrow 1] - \Pr[PR-HID-CPA_{\text{rand}}^A]| \leq \varepsilon.$$  

Note that PR-HID-CPA trivially implies IND-HID-CPA and anonymity of HIBKEM.

---

#### Figure 2.3: Games PR-HID-CPA$_{\text{real}}$, IND-HID-CPA$_{\text{rand}}$, and PR-HID-CPA$_{\text{rand}}$ for defining IND-HID-CPA and PR-HID-CPA-security.

For any identity $id \in B^\ell$, $\text{Prefix}(id)$ denotes the set of all prefixes of $id$ (where $|\text{Prefix}(id)| = O(|B|^\ell)$).

- The probabilistic key generation algorithm $\text{Gen}(1^\lambda)$ returns the (master) public/secret key and delegation key $(pk, sk, dk)$. Note that for some of our constructions $dk$ is empty. We assume that $pk$ implicitly defines a hierarchical identity space $ID = B^{\leq m}$, for some base identity set $B$, and a key space $K$, and ciphertext space $C$.

- The probabilistic user secret key generation algorithm $\text{USKGen}(sk, id)$ returns a secret key $usk[id]$ and a delegation value $udk[id]$ for hierarchical identity $id \in ID$.

- The probabilistic key delegation algorithm $\text{USKDel}(dk, usk[id], udk[id], id \in B^\ell, id_{p+1} \in B)$ returns a user secret key $usk[id_{p+1}]$ for the hierarchical identity $id' = id \mid id_{p+1} \in B^{p+1}$ and the user delegation key $udk[id']$. We require $1 \leq |id| \leq m - 1$.

- The probabilistic encapsulation algorithm $\text{Enc}(pk, id)$ returns a symmetric key $K \in K$ together with a ciphertext $C$ with respect to the hierarchical identity $id \in ID$.

- The deterministic decapsulation algorithm $\text{Dec}(usk[id], id, C)$ returns a decapsulated key $K \in K$ or the reject symbol $\perp$.

For correctness we require that for all $\lambda \in \mathbb{N}$, all pairs $(pk, sk)$ generated by $\text{Gen}(1^\lambda)$, all $id \in ID$, all $usk[id]$ generated by $\text{USKGen}(sk, id)$ and all $(K, c)$ generated by $\text{Enc}(pk, id)$:

$$\Pr[\text{Dec}(usk[id], id, C) = K] = 1.$$  

Moreover, we also require the distribution of $usk[id_{id_{p+1}}]$ from $\text{USKDel}(usk[id], udk[id], id_{p+1})$ is identical to the one from $\text{USKGen}(sk, id_{id_{p+1}})$.

In our HIBKEM definition we make the delegation key $dk$ and the user delegation key $udk[id]$ explicit to make our constructions more readable. We define indistinguishability (IND-HID-CPA) and pseudorandom ciphertexts (PR-HID-CPA) against adaptively chosen identity and plaintext attacks for a HIBKEM via games PR-HID-CPA$_{\text{real}}$, IND-HID-CPA$_{\text{rand}}$, and PR-HID-CPA$_{\text{rand}}$ from Figure 2.3.
3.1 Identification and Fiat-Shamir Transform

3.1.1 Canonical Identification Schemes

A canonical identification scheme \( \text{ID} \) is a three-move protocol of the form depicted in Figure 3.1. The prover’s first message \( R \) is called commitment, the verifier selects a uniform challenge \( h \) from the set \( \text{ChSet} \), and, upon receiving a response \( s \) from the prover, makes a deterministic decision.

**Definition 3.1.1** (Canonical Identification Scheme). A canonical identification scheme \( \text{ID} \) is defined as a tuple of algorithms \( \text{ID} := (\text{IGen}, \text{P}, \text{ChSet}, \text{V}) \).

- The key generation algorithm \( \text{IGen} \) takes system parameters \( \text{par} \) as input and returns public and secret key \( (\text{pk}, \text{sk}) \). We assume that \( \text{pk} \) defines \( \text{ChSet} \), the set of challenges.
- The prover algorithm \( \text{P} = (\text{P}_1, \text{P}_2) \) is split into two algorithms. \( \text{P}_1 \) takes as input the secret key \( \text{sk} \) and returns a commitment \( R \) and a state \( St \); \( \text{P}_2 \) takes as input the secret key \( \text{sk} \), a commitment \( R \), a challenge \( h \), and a state \( St \) and returns a response \( s \).
- The verifier algorithm \( \text{V} \) takes the public key \( \text{pk} \) and the conversation transcript as input and outputs a deterministic decision, 1 (acceptance) or 0 (rejection).

We require that for all \((\text{pk}, \text{sk}) \in \text{IGen} (\text{par}), \ all \ (R, St) \in \text{P}_1 (\text{sk}), \ all \ h \in \text{ChSet} \text{ and all } s \in \text{P}_2 (\text{sk}, R, h, St)\), we have \( \text{V}(\text{pk}, R, h, s) = 1 \).

We make a couple of useful definitions. An identification scheme \( \text{ID} \) is called unique if for all \((\text{pk}, \text{sk}) \in \text{IGen} (\text{par}), \ (R, St) \in \text{P}_1 (\text{sk}), \ h \in \text{ChSet} \), there exists at most one response \( s \in \{0, 1\}^* \) such that \( \text{V}(\text{pk}, R, h, s) = 1 \). A transcript is a three-tuple \((R, h, s)\). It is called valid (with respect to the public-key \( \text{pk} \)) if \( \text{V}(\text{pk}, R, h, s) = 1 \). Furthermore, it is called real, if it is the output of a real interaction between prover and verifier as depicted in Figure 3.1. A canonical identification schemes \( \text{ID} \) has \( \alpha \) bits of min-entropy, if for all \((\text{pk}, \text{sk}) \in \text{IGen} (\text{par})\), the commitment generated by the prover algorithm is chosen from a distribution with at least \( \alpha \) bits of min-entropy. That is, for all strings \( R' \) we have \( \Pr[R = R'] \leq 2^{-\alpha} \), if \((R, St) \in \text{P}_1 (\text{sk}) \) was honestly generated by the prover.

Consider the games defined in Figure 3.2. We now define (parallel) impersonation and key-recovery against key-only attack (KOA), passive attack (PA), and active attack (AA).

**Definition 3.1.2** ((Parallel) Impersonation). Let \( YYY \in \{\text{KOA, PA, AA}\} \). A canonical identification \( \text{ID} \) is said to be \((t, \varepsilon, Q_{Cn}, Q_{O})\)-\text{PIMP}--YYY secure (parallel impersonation against \( YYY \) attacks) if for all adversaries \( A \) running in time at most \( t \),

\[
\Pr[\text{PIMP-YYY}^A \Rightarrow 1] \leq \varepsilon.
\]

If \( YYY = \text{KOA} \), then the parameter \( Q_{O}(=0) \) is not used and we simply speak of \((t, \varepsilon, Q_{Cn})\)-\text{PIMP}--\text{KOA}.

Moreover, \((t, \varepsilon, Q_{O})\)-IMP--YYY (impersonation against \( YYY \) attack) security is defined as \((t, \varepsilon, 1, Q_{O})\)-\text{PIMP}--YYY security, i.e., the adversary is only allowed \( Q_{Cn} = 1 \) query to the \( \text{Ch} \) oracle.
Figure 3.1: A canonical identification scheme and its transcript \((R, h, s)\).

<table>
<thead>
<tr>
<th>INITIALIZE: ((\text{pk}, \text{sk}) \leftarrow \text{IGen}(\text{par}))</th>
<th>(/\text{ Games XXX-YYY })</th>
<th>\text{Ch}(R_i): (/\text{ Games PIMP-YYY }), at most (Q_{\text{C}}) queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return (\text{pk})</td>
<td>(/\text{ at most })</td>
<td>Return (h_i \notin \text{ChSet})</td>
</tr>
</tbody>
</table>

PROVER\((j)\): \(/\text{ Games XXX-AA }\) \(j := j + 1\) \(/\text{ at most }Q_{\text{O}}\) queries |

\((R'_j, \text{St}'_j) \leftarrow \text{P}_1(\text{sk})\) |

Return \(R'_j\) \(/\text{ at most }Q_{\text{O}}\) queries |

\(\text{PROVER}(j, h'_i)\): \(/\text{ Games XXX-AA }\) \(\text{If } R'_j = \bot \text{ return } \bot\) \(/\text{ at most }Q_{\text{O}}\) queries |

Else return \(s'_j \leftarrow \text{P}_2(\text{sk}, R'_j, h'_i, \text{St}'_j)\) |

\(\text{PROVER}(j, h'_i)\): \(/\text{ Games XXX-AA }\) \(\text{If } R'_j = \bot \text{ return } \bot\) \(/\text{ at most }Q_{\text{O}}\) queries |

Else return \(s'_j \leftarrow \text{P}_2(\text{sk}, R'_j, h'_i, \text{St}'_j)\) |

\(\text{PROVER}(j, h'_i)\): \(/\text{ Games XXX-AA }\) \(\text{If } R'_j = \bot \text{ return } \bot\) \(/\text{ at most }Q_{\text{O}}\) queries |

Else return \(s'_j \leftarrow \text{P}_2(\text{sk}, R'_j, h'_i, \text{St}'_j)\) |

Figure 3.2: Security Games XXX-YYY \(\text{XXX} \in \{\text{KR}, \text{PIMP}\}\) and YYY \(\in \{\text{KOA}, \text{PA}, \text{AA}\}\) for Definition 3.1.2 and Definition 3.1.3. On two queries \(\text{Ch}(R_i)\) and \(\text{Ch}(R_i')\) with the same input \(R_i = R_i'\), the oracle returns two independent random challenges \(h_i \notin \text{ChSet}\) and \(h_{i'} \notin \text{ChSet}\).

Definition 3.1.3 (Key-recovery). Let YYY \(\in \{\text{KOA}, \text{PA}, \text{AA}\}\). A canonical identification ID is said to be \((t, \varepsilon, Q_{\text{O}})\)-KR-YYY secure (key recovery under YYY attack) if for all adversaries \(A\) running in time \(t\),

\[\Pr[\text{KR-YYY}^A \Rightarrow 1] \leq \varepsilon.\]

The winning condition \((\text{pk}, \text{sk}^*) \in \text{IGen}(\text{par})\) means that the tuple \((\text{pk}, \text{sk}^*)\) is in the support of \text{IGen}(\text{par}), i.e., that \(A\) outputs a valid secret-key \text{sk}^* with respect to \text{pk}.

Definition 3.1.4 (Special Soundness). A canonical identification ID is said to be SS (special sound) if there exists an extractor algorithm Ext such that, for all \((\text{pk}, \text{sk}) \in \text{IGen}(\text{par})\), given any two accepting transcripts \((R, h, s)\) and \((R, h', s')\) (where \(h \neq h'\)), we have \(\Pr[(\text{sk}^*, \text{pk}) \in \text{IGen}(\text{par}) | \text{sk}^* \leftarrow \text{Ext}(\text{pk}, R, h, s, h', s')] = 1\).

Definition 3.1.5 (Random Self-reducibility). A canonical identification ID is said to be RSR (random self-reducible) if there is an algorithm \text{Rerand} and two deterministic algorithms \text{Tran} and \text{Derand} such that, for all \((\text{pk}, \text{sk}) \in \text{IGen}(\text{par})\):

- \(\text{pk}'\) and \(\text{pk}''\) have the same distribution, where \((\text{pk}', \tau') \leftarrow \text{Rerand}(\text{pk})\) is the rerandomized key-pair and \((\text{pk}'', \text{sk}'') \leftarrow \text{IGen}(\text{par})\) is a freshly generated key-pair.
- For all \((\text{pk}', \tau') \in \text{Rerand}(\text{pk})\), all \((\text{pk}', \text{sk}') \in \text{IGen}(\text{par})\), and \(\text{sk}^* = \text{Derand}(\text{pk}, \text{pk}', \text{sk}', \tau')\), we have \((\text{pk}, \text{sk}^*) \in \text{IGen}(\text{par})\), i.e., \text{Derand} returns a valid secret-key \text{sk}^* with respect to \text{pk}, given any valid \text{sk}' for \text{pk}'.
- For all \((\text{pk}', \tau') \in \text{Rerand}(\text{pk})\), all transcripts \((R', h', s')\) that are valid with respect to \text{pk}', the transcript \((R', h', s' := \text{Tran}(\text{pk}, \text{pk}', \tau', (R', h', s'))\) is valid with respect to \text{pk}.

Definition 3.1.6 (Honest-verifier Zero-knowledge). A canonical identification ID is said to be (perfect) HVZK (honest-verifier zero-knowledge) if there exists an algorithm \text{Sim} that, given a public key \text{pk}, outputs \((R, h, s)\) such that \((R, h, s)\) is a real (i.e., properly distributed) transcript with respect to \text{pk}.
3.1.2 Signatures from Identification Schemes

Let ID := (Gen, P, ChSet, V) be a canonical identification scheme. The generalized Fiat-Shamir transformation [BP02] defines the signature scheme SIG[ID] := (Gen, Sign, Ver) from ID as follows. par contains the system parameters of ID and a hash function \( H : \{0,1\}^* \rightarrow \text{ChSet} \).

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen(par)</td>
<td>(pk, sk) ( \xrightarrow{$} ) IGen(par)</td>
</tr>
<tr>
<td>Return (pk, sk)</td>
<td></td>
</tr>
<tr>
<td>Sign(sk, m)</td>
<td>(R, St) ( \xleftarrow{$} ) P_1(sk)</td>
</tr>
<tr>
<td>h = H(R, m)</td>
<td></td>
</tr>
<tr>
<td>s = P_2(sk, R, h, St)</td>
<td></td>
</tr>
<tr>
<td>Return ( \sigma = (R, s) )</td>
<td></td>
</tr>
<tr>
<td>Ver(pk, m, \sigma)</td>
<td>Parse ( \sigma = (R, s) )</td>
</tr>
<tr>
<td>( h = H(R, m) )</td>
<td></td>
</tr>
<tr>
<td>Return ( V(pk, R, h, s) )</td>
<td></td>
</tr>
</tbody>
</table>

In some variants of the Fiat-Shamir transform the hash additionally inputs some public parameters, for example \( h = H(pk, R, m) \).

**Alternative Fiat-Shamir Transform.** We call ID partially commitment-recoverable if the commitment \( R \) can be partitioned into \( R = (R_L, R_R) \), a left part \( R_L \) and a right part \( R_R \) and \( V(pk, R, h, s) \) is such that it first recomputes \( R'_R = V'(pk, h, s, R_L) \) and then outputs 1 if \( R'_R = R_R \). It is (fully) commitment-recoverable if \( R = R_R \) and \( R_L \) is the empty string. For partially commitment-recoverable ID, we can define an alternative Fiat-Shamir transformation \( \text{SIG}'[ID] := (\text{Gen}, \text{Sign}', \text{Ver}') \), where Gen is as in SIG[ID]. Algorithm \( \text{Sign}'(sk, m) \) is defined as \( \text{Sign}(sk, m) \) with the modified output \( \sigma' = (R_L, h, s) \). Algorithm \( \text{Ver}'(pk, m, \sigma') \) first parses \( \sigma' = (R_L, h, s) \), then recomputes the right part of the commitment as \( R'_R := V'(pk, R_L, h, s) \), and finally returns 1 iff \( H(R', m) = h \).

Since \( \sigma = (R, s) \) can be publicly transformed into \( \sigma' = (R_L, h, s) \) and vice-versa, SIG[ID] and SIG'[ID] are equivalent in terms of security. On the one hand, the alternative Fiat-Shamir transform yields shorter signatures if \( h \in \text{ChSet} \) has a smaller representation size than the response \( s \). On the other hand, signatures of the Fiat-Shamir transform maintain their algebraic structure, which in some cases enables useful properties such as batch verification.

### 3.2 Security Implications

In this section we will prove the following two main results.

**Theorem 3.2.1** (Main Theorem 1). Suppose ID is SS, HVZK, RSR and has \( \alpha \)-bit min-entropy. If ID is \( (t, \varepsilon) \)-KR-KOA secure then SIG[ID] is \( (t', \varepsilon', Q_h, Q_s, Q_{Ch}) \)-UF-CMA-secure and \( (t'', \varepsilon'', N, Q_s, Q_h) \)-MU-UF-CMA-secure in the programmable random oracle model, where

\[
\frac{\varepsilon'}{t'} \leq 6(Q_h + 1) \cdot \frac{\varepsilon}{t} + \frac{Q_s}{2^{2\alpha}} + \frac{1}{|\text{ChSet}|},
\]

\[
\frac{\varepsilon''}{t''} \leq 24(Q_h + 1) \cdot \frac{\varepsilon}{t} + \frac{Q_s}{2^{2\alpha}} + \frac{1}{|\text{ChSet}|}.
\]

The proof of Theorem 3.2.1 is obtained by combining Lemmas 3.2.5 and 3.2.10 below and using \( Q_h \leq t - 1 \).

**Theorem 3.2.2** (Main Theorem 2). Suppose SIG[ID] is HVZK, RSR and has \( \alpha \)-bit min-entropy. If SIG[ID] is \( (t, \varepsilon, Q_h, Q_s) \)-UF-KOA secure then SIG[ID] is \( (t', \varepsilon', N, Q_s, Q_h) \)-MU-UF-CMA-secure in the programmable random oracle model, where

\[
\varepsilon' \leq 4\varepsilon + \frac{Q_h Q_s}{2^{2\alpha}}, \quad t' \approx t
\]

and \( Q_s, Q_h \) are upper bounds on the number of signing and hash queries in the MU-UF-CMA experiment, respectively.

The proof of Theorem 3.2.2 is obtained by combining Lemmas 3.2.9 and 3.2.10 below.
### 3.2.1 Multi-Instance Reset Lemma

We first state a new reset lemma from [KMP16a] that we will later use in the proof of Theorem 3.2.1. It is presented in the style of Bellare and Neven’s General Forking Lemma [BN06] and does not talk about signatures or identification protocols. It is a generalization to many parallel instances of the Reset Lemma [BP02], which is obtained by setting \( N = 1 \).

**Lemma 3.2.3 (Multi-Instance Reset Lemma).** Fix an integer \( N \geq 1 \) and a non-empty set \( H \). Let \( C \) be a randomized algorithm that on input \((I, h)\) returns a pair \((b, \sigma)\), where \( b \) is a bit and \( \sigma \) is called the side output. Let \( IG \) be a randomized algorithm that we call the input generator. The accepting probability of \( C \) is defined as

\[
\text{acc} := \Pr[b = 1 \mid I \mathcal{IG} h \mathcal{H}; (b, \sigma) \mathcal{C}(I, h)]
\]

The (multi-instance) reset algorithm \( R_C \) associated to \( C \) is the randomized algorithm that takes input \( I_1, \ldots, I_N \) and proceeds as follows.

<table>
<thead>
<tr>
<th>Algorithm ( R_C(I_1, \ldots, I_N) ):</th>
</tr>
</thead>
<tbody>
<tr>
<td>For ( i \in [N] ):</td>
</tr>
<tr>
<td>Pick random coins ( \rho_i )</td>
</tr>
<tr>
<td>( h_i \mathcal{H} )</td>
</tr>
<tr>
<td>((b_i, \sigma_i) \mathcal{C}(I_i, h_i; \rho_i))</td>
</tr>
<tr>
<td>If ( b_1 = \ldots = b_N = 0 ) then return ((0, \epsilon)) // Abort in Phase 1</td>
</tr>
<tr>
<td>Fix ( i^* \in [N] ) such that ( b_i = 1 )</td>
</tr>
<tr>
<td>For ( j \in [N] ):</td>
</tr>
<tr>
<td>( h_j^i \mathcal{H} )</td>
</tr>
<tr>
<td>((b_j^i, \sigma_j^i) \mathcal{C}(I_j, h_j^i; \rho_j^i))</td>
</tr>
<tr>
<td>If ( \exists j^* \in [N] ) : ( b_j^i \neq b_j^i^* ), and ( b_j^i^* = 1 ) then return ((i^<em>, \sigma_i^</em>, \sigma_j^i^*))</td>
</tr>
<tr>
<td>Else return ((0, \epsilon, \epsilon)) // Abort in Phase 2</td>
</tr>
</tbody>
</table>

Let

\[
\text{res} := \Pr[i^* \geq 1 \mid I_1, \ldots, I_N \mathcal{IG} (i^*, \sigma, \sigma') \mathcal{R}_C(I_1, \ldots, I_N)].
\]

Then

\[
\text{res} \geq \left( 1 - \left( 1 - \text{acc} + \frac{1}{|H|} \right)^N \right)^2.
\]

### 3.2.2 Proof of the Main Theorems

**Lemma 3.2.4 (XXX-KOA \(\Rightarrow\) XXX-PA).** Let \( XXX \in \{KR, IMP, PIMP\} \). If \( ID \) is \((t, \epsilon, Q_{Cn})\)-XXX-KOA secure and HVZK, then \( ID \) is \((\approx t, \epsilon, \epsilon, Q_{Cn}, Q_O)\)-XXX-PA secure.

**Proof.** Let \( A \) be an adversary against the \((t, \epsilon, Q_{Cn}, Q_O)\)-XXX-PA-security of \( ID \). We now build an adversary \( B \) in Figure 3.3 against the \((t, \epsilon, Q_{Cn})\)-XXX-KOA security of \( ID \), with \((t, \epsilon)\) as claimed. Essentially, \( B \) only has to simulate the \( \text{TRAN} \) oracle of the passive attack PA using the HVZK simulator, \( \text{Sim} \). Finally, \( B \) outputs whatever \( A \) outputs. Thus,

\[
\epsilon = \Pr[XXX-KOA_B \Rightarrow 1] = \Pr[XXX-PA_A \Rightarrow 1].
\]

The running time of \( B \) is that of \( A \) plus roughly \( Q_O \) executions of \( \text{Sim} \) to simulate the \( \text{PROVER} \) oracle, which we ignore for simplicity. \( \square \)

**Lemma 3.2.5** below proving that KR-KOA tightly implies IMP-KOA uses the Multi-Instance Reset Lemma and that takes advantage of \( ID \)'s random self-reducibility (RSR).

**Lemma 3.2.5 (KR-KOA \(\Rightarrow\) IMP-KOA).** If \( ID \) is \((t, \epsilon)\)-KR-KOA secure, SS and RSR, then \( ID \) is \((t', \epsilon')\)-IMP-KOA secure, where for any \( N \geq 1 \),

\[
\epsilon \geq \left( 1 - (1 - \epsilon' + \frac{1}{|\text{ChSet}|})^N \right)^2, \quad t \approx 2Nt'.
\]

In particular, relation between the two success ratios is

\[
\frac{\epsilon'}{t'} - \frac{1}{t'|\text{ChSet}|} \leq 6 \cdot \frac{\epsilon}{t}.
\]
We note that this is the dominating running time of (PIMP Lemma 3.2.6 (Derand algorithms of The running time and returns $\epsilon$ Dividing $\epsilon$ by $t$ yields $\epsilon' = \frac{\epsilon}{t}$, where $t \approx 2t'$.

Proof. We first show how to derive (3.2) from (3.1). If $\epsilon' > 1/|\text{ChSet}|$, then (3.2) holds trivially. Assuming $\epsilon' < 1/|\text{ChSet}|$, we set $N := (\epsilon' - 1/|\text{ChSet}|)^{-1}$ to obtain $t \approx 2t'/\epsilon' \approx 1/\epsilon'$. Dividing $\epsilon$ by $t$ yields $\epsilon'$. To prove (3.1), let $A$ be an adversary against the $(t', \epsilon')$-IMP-KOA-security of ID. We now build an adversary $B$ against the $(t, \epsilon)$-KR-KOA security of ID, with $(t, \epsilon)$ as claimed in (3.1).

We use the Multi-Instance Reset Lemma (Lemma 3.2.3), where $H := \text{ChSet}$ and $\text{IG}$ runs $(pk, sk) \in \text{IGen}$ and returns $pk$ as instance $I$. We first define an adversary $C(pk, h; \rho)$ that executes $A(pk; \rho)$, answers $A$'s single query $R$ with $h$, and finally receives $s$ from $A$. If the transcript $(R, h, s)$ is valid with respect to $pk$ (i.e., $V(pk, R, h, s) = 1$), $C$ returns $(b = 1, \sigma = (R, h, s))$; otherwise, it returns $(b = 0, \epsilon)$. By construction, $C$ returns $b = 1$ iff $A$ is successful:

$$\epsilon = \epsilon'. $$

Adversary $B$ is defined as follows. For each $i \in [N]$, it uses the RSR property of ID to generate a fresh public key/trapdoor pair $(pk_i, \tau_i) \in \text{Rerand}(pk)$. Next, it runs $(\sigma, \sigma') \in \text{RC}(pk_1, \ldots, pk_N)$, with $C$ defined above. If $i^* \geq 1$, then both transcripts $\sigma = (R, h, s)$ and $\sigma' = (R, h, s')$ are valid with respect to $pk_{i^*}$ and $h \neq h'$. B uses the SS property of ID and computes $sk_i := \text{Ext}(pk_i \ldots, R, h, s, h', s')$. Finally, using the RSR property of ID, it returns $sk = \text{Derand}(pk_i, sk_{i^*}, \tau_{i^*})$ and terminates. By construction, $B$ is successful iff $\text{RC}$ is. By Lemma 3.2.3 we can bound $B$'s success probability as

$$\epsilon = \epsilon' = \epsilon'. $$

The running time $t$ of $B$ is that of $\text{RC}$, meaning $2Nt'$ plus the $N$ times the time to run the $\text{Rerand}$ and $\text{Derand}$ algorithms of RSR plus the time to run the $\text{Ext}$ algorithm of SS. We write $t \approx 2Nt'$ to indicate that this is the dominating running time of $B$.

Lemma 3.2.6 (IMP-KOA $\text{loss}_Q \text{PIMP-KOA}$). If ID is $(t, \epsilon)$-IMP-KOA secure, then ID is $(t', \epsilon', Q_{Cn})$-PIMP-KOA secure, where

$$\epsilon' \leq Q_{Cn} \cdot \epsilon, t' \approx t.$$

Proof. Let $A$ be an adversary against the $(t', \epsilon', Q_{Cn})$-PIMP-KOA-security of ID. We now build an adversary $B$ in Figure 3.3 against the $(t, \epsilon)$-IMP-KOA security of ID, with $(t, \epsilon)$ as claimed.

Note that if $j = i^*$ then $B$ wins if $A$ wins. Moreover, since $j$ is chosen uniformly in $\{1, \ldots, Q_{Cn}\}$, $\Pr[\text{Bad} = 0] = \Pr[j = i^*] = 1/Q_{Cn}$. Thus, $\epsilon = \Pr[\text{IMP-KOA}^B \Rightarrow 1] \geq \frac{1}{Q_{Cn}} \Pr[\text{PIMP-KOA}^A \Rightarrow 1] = \frac{1}{Q_{Cn}} \epsilon'$. We note $t \approx t'$.

Lemma 3.2.7 (PIMP-KOA $\text{PR}_Q \text{UF-KOA}$). If ID is $(t, \epsilon, Q_{Cn})$-PIMP-KOA secure, then $\text{SIG[ID]}$ is $(t', \epsilon', Q_{h})$-UF-KOA secure in the programmable random oracle model, where

$$\epsilon' = \epsilon, t' \approx t, Q_{h} = Q_{Cn} - 1.$$


**Proof.** Let $A$ be an adversary against the $(t', \varepsilon', Q_h)$-UF-KOA-security of ID. We now build an adversary $B$ in Figure 3.5 against the $(t, \varepsilon, Q_{Ch})$-PIMP-KOA security of ID, with $(t, \varepsilon, Q_{Ch})$ as claimed. Here we need to program the random oracle \text{Hash} queries.

According to the construction of $B$, we have $(R^*, m^*) \in \{(R_j, m_j) | j \in \{1, \ldots, Q_h + 1\}\}$. Let $i \in \{1, \ldots, Q_h + 1\}$ be the index such that $(R_i, m_i) = (R^*, m^*)$ and $H(R_i, m_i) = H(R^*, m^*) = h^*$. Note that $(R_i, h^*, s^*)$ is a valid transcript which breaks PIMP-KOA security if and only if $A$'s forgery is valid, establishing $\varepsilon = \varepsilon'$. The running time of $B$ is roughly the same as that of $A$, hence $t' \approx t$. \hfill $\square$

---

**Figure 3.4:** Description of $B$ with access to the oracles \text{Initialize}_{\text{IMP}}, \text{Finalize}_{\text{IMP}} and \text{Ch}_{\text{IMP}}$ of the IMP-KOA games of Figure 3.2 for the proof of Lemma 3.2.6

---

**Figure 3.5:** Description of $B$ with access to oracles \text{Initialize}_{\text{PIMP}}, \text{Finalize}_{\text{PIMP}} and \text{Ch}$ of the PIMP-KOA games of Figure 3.2 for the proof of Lemma 3.2.7

---

The following lemma is a special case of Lemma 3.2.10 (with a slightly improved bound).

**Lemma 3.2.8** (UF-KOA $\xrightarrow{\text{PRG}}$ UF-CMA). Suppose $ID$ is HVZK and has $\alpha$-bit min-entropy. If $\text{Sig}[ID]$ is $(t, \varepsilon, Q_h)$-UF-KOA secure, then $\text{Sig}[ID]$ is $(t', \varepsilon', Q_s, Q_h)$-UF-CMA secure in the programmable random oracle model, where

$$
\varepsilon' \leq \varepsilon + \frac{Q_h Q_s}{2^n}, \quad t' \approx t,
$$

and $Q_s, Q_h$ are upper bounds on the number of signing and hash queries in the UF-CMA experiment, respectively.

**Lemma 3.2.9** (UF-KOA $\xrightarrow{\text{RSR}}$ MU-UF-KOA). Suppose $ID$ is RSR. If $\text{Sig}[ID]$ is $(t, \varepsilon)$-UF-KOA secure, then $\text{Sig}[ID]$ is $(t', \varepsilon', N)$-MU-UF-KOA secure, where

$$
\varepsilon' = \varepsilon, \quad t' \approx t.
$$

Note that without the RSR property one can use the generic bounds from GMS02 to obtain a non-tight bound with a loss of $N$.

**Proof.** Let $A$ be an algorithm that breaks $(t', \varepsilon', N)$-MU-UF-KOA security of $\text{Sig}[ID]$. In Figure 3.6 we will describe an adversary $B$ which invokes $A$ and breaks $(t, \varepsilon)$-UF-KOA security of $\text{Sig}[ID]$ with $(t, \varepsilon)$ as stated in the lemma.

By the RSR property of $ID$ (cf. Definition 3.1.5), all the $pk_i$ are well-distributed. We note that if $\sigma^*$ is a valid signature on message $m^*$ under $pk_i$, then $(R^*, s)$ is also a valid signature on $m^*$ under $pk$. Thus, we have $\varepsilon = \varepsilon'$. The running time $t$ of $B$ is $t'$ plus the $N$ times to run the Rerand and Tran algorithms of RSR. We again write $t' \approx t$. \hfill $\square$
3.2 Security Implications

**Lemma 3.2.10** (MU-UF-KOA $\xrightarrow{\text{PRO}}$ MU-UF-CMA). Suppose ID is HVZK and has $\alpha$-bit min-entropy. If $\text{SIG[ID]}$ is $(t, \varepsilon, N, Q_h)$-MU-UF-KOA secure, then $\text{SIG[ID]}$ is $(t', \varepsilon', N, Q_s, Q_h)$-MU-UF-CMA secure in the programmable random oracle model, where

$$\varepsilon' \leq 4\varepsilon + \frac{Q_h Q_s}{2^\alpha}, \quad t' \approx t,$$

and $N$ is the number of users and $Q_s$ and $Q_h$ are upper bounds on the number of signing and hash queries in the MU-CMA experiment, respectively.

**Proof.** Let $A$ be an algorithm that breaks $(t', \varepsilon', N, Q_s, Q_h)$-MU-UF-CMA security of $\text{SIG[ID]}$. In Figure 3.7 we will describe an adversary $B$ that breaks $(t, \varepsilon, N, Q_h)$-MU-UF-KOA security of $\text{SIG[ID]}$ with $(t, \varepsilon)$ as stated in the lemma.

Note that this proof is in the programmable random oracle model and $B$ has access to a random oracle $\text{HASH}'$ from the MU-UF-KOA game and need to answer random oracle queries $\text{HASH}'$ from A.

**Figure 3.6:** Description of $B$ with access to oracles $\text{INITIALIZE}_{\text{UF-KOA}}$ and $\text{FINALIZE}_{\text{UF-KOA}}$ of the UF-KOA game in Definition 2.3.3 for the proof of Lemma 3.2.9

**Figure 3.7:** Description of $B$ with access to oracles $\text{INITIALIZE}_{\text{KOA}}$ and $\text{FINALIZE}_{\text{KOA}}$ of the MU-KOA game in Definition 2.3.2 and the random oracle $\text{HASH}_{\text{mu}}$ for the proof of Lemma 3.2.10

**Analysis of $B$.** By the HVZK property, if $\text{Bad}_1 := \bigvee_{j=1}^{Q_h} \text{Bad}_{1,j} \neq 1$ then the simulation of $\text{SIGN}$ is identical to that of the original MU-UF-KOA game. Since $R_j$ has min-entropy $\alpha$, for each $\text{SIGN}$ query, $\text{Bad}_{1,j} = 1$ with probability at most $Q_h/2^\alpha$. As the maximal number of $\text{SIGN}$ queries is bounded by $Q_s$, $\text{Bad}_1 = 1$ overall with probability at most $Q_h Q_s/2^\alpha$.

Eventually, $A$ will submit its forgery by querying $\text{FINALIZE}(i^*, m^*, \sigma^* := (R^*, s^*))$. We assume that it is a valid forgery in the MU-CMA experiment, i.e., for $h^* = H'(R^*, m^*)$ we have $V(pk'_j, R^*, h^*, s^*) = 1$. Furthermore, it satisfies the freshness condition, i.e.,

$$(i^*, m^*) \notin \{(i_j, m_j) : j \in [Q_s]\}, \quad (3.3)$$
otherwise \texttt{Finalize} aborts.

It remains to show that if \texttt{Finalize} does not abort then B computes a valid forgery for the \textsc{MU-UF-KOA} experiment (namely, \texttt{Finalize}_{\text{KOA}} outputs 1), if A’s forgery is valid. Consider the following case distinction, which has been captured by \texttt{Finalize} in Figure 3.7.

- **Case 1:** There exists a \( j \in [Q_s] \) such that \((R^*, m^*) = (R_j, m_j)\). (If there is more than one \( j \), fix any of them.) We have \( h^* = h_j \) and furthermore \( i^* \neq i_j \) by the freshness condition (3.3).
  - **Case 1a:** \((b_i = 1)\) and \((b_j = 0)\). Then the hash value \( h^* = H'(R^*, m^*) \) was not programmed by B (i.e. \( H'(R^*, m^*) = \text{Hash}(R^*, m^*) \)). Thus, \((i^*, m^*, (R^*, s'))\) is a valid forgery for its \textsc{MU-UF-KOA} experiment.

Since \( b_i \) is chosen uniformly and \( b_j \) is hidden from A, \( \Pr[\text{Bad}_2] = 3/4 \) and \texttt{Finalize}(\( i^*, m^*, \sigma^* \)) outputs \texttt{Finalize}_{\text{KOA}}(\( i^*, m^*, \sigma^* \)) with probability 1/4 in case 1.

- **Case 2:** For all \( j \in [Q_s] \) we have: \((m^*, R^*) \neq (m_j, R_j)\).
  - **Case 2a:** \( b_\ast = 1 \). Then the hash value \( h^* = H'(R^*, m^*) \) was not programmed by B (namely, \( h^* = H'(R^*, m^*) = \text{Hash}(R^*, m^*) \)). \((i^*, m^*, (R^*, s'))\) is a valid forgery to its \textsc{MU-UF-KOA} experiment.

Note that in case 2, \( \Pr[\text{Bad}_3 = 1] = 1/2 \) and \texttt{Finalize}(\( i^*, m^*, \sigma^* \)) outputs \texttt{Finalize}_{\text{KOA}}(\( i^*, m^*, \sigma^* \)) with probability 1/2.

Overall, B returns a valid forgery of \textsc{MU-UF-KOA} experiment with probability

\[
\varepsilon \geq \min \left\{ \frac{1}{4}, \frac{1}{2} \right\} \cdot \left( \varepsilon' - \frac{Q_b Q_s}{2^a} \right) = \frac{1}{4} \left( \varepsilon' - \frac{Q_b Q_s}{2^a} \right).
\]

The running time of B is that of A plus the \( Q_s \) executions of \text{Sim}. We write \( t' \approx t \). This completes the proof. \( \square \)

If \( s \) in ID is uniquely defined by \((pk, R, h)\) (e.g., as in the Schnorr identification scheme), then one can show the above proof even implies \textsc{MU-SUF-CMA} security of \text{SIG[ID]}. The simulation of hash and signing queries is the same as in the above proof. Let \((i^*, m^*, R^*, s^*)\) be A’s forgery. The freshness condition of the \textsc{MU-SUF-CMA} experiment says that \((i^*, m^*, R^*, s^*) \notin \{(i_j, m_j, R_j, s_j) : j \in [Q_s]\}\). Together with the uniqueness of ID, this implies \((i^*, m^*, R^*) \notin \{(i_j, m_j, R_j) : j \in [Q_s]\}\). If \((i^*, m^*) \notin \{(i_j, m_j) : j \in [Q_s]\}\), then B can break \textsc{MU-UF-KOA} security by the same case distinction as in the proof above. Otherwise, we have \( R^* \notin \{R_j : j \in [Q_s]\}\), in which case we can argue as in case 2.

### 3.3 Instantiations

In this section we consider three important identification schemes, namely the ones by Schnorr [Sch91], by Katz-Wang [KW03,CP98] and by Guillou-Quisquater [GQ90]. Here we generalize the Katz-Wang scheme with the MDDH assumption. We use our framework to derive tight security bounds and concrete parameters for the corresponding Schnorr/Katz-Wang/Guillou-Quisquater signature schemes.

### 3.3.1 Schnorr Identification/Signature Scheme

#### 3.3.1.1 Schnorr’s Identification Scheme

The well-known Schnorr’s identification scheme is one of the most important instantiations of our framework. For completeness we show that Schnorr’s identification has large min-entropy, special soundness (SS), honest-verifier zero-knowledge (HVZK), random-self reducibility (RSR) and key-recoversy security (KR-KOA) based on the discrete logarithm problem (DLOG). Moreover, based on the one-more discrete logarithm problem (OMDL), Schnorr’s identification is actively secure (IMP-AA) and weakly secure against man-in-the-middle attack (WIMP-MIM).

Let \( \text{par} := (p, \mathcal{P}, G) \) be a set of system parameters, where \( G \) is a cyclic group of prime order \( p \) with a discrete logarithm problem \( \mathcal{P} \) and \( R \) is a generator of \( G \). Examples of groups \( G \) include appropriate subgroups of certain elliptic curve groups, or subgroups of \( \mathbb{Z}_p^* \). Recall the implicit representation in Section 2.2.1 and the Schnorr identification scheme \( IDS := (\text{IGen}, P, \text{ChSet}, V) \) is defined as follows.
We recall the DLOG assumption.

**Definition 3.3.1 (Discrete Logarithm Assumption)**. The discrete logarithm problem DLOG is (t, ε)-hard in par = (p, P, G) if for all adversaries A running in time at most t,

\[
\Pr \left[ X = x \mid X \xLeftRightArrow{G} x \xLeftRightArrow{A(X)} \right] \leq \varepsilon.
\]

**Lemma 3.3.2.** ID5 is a canonical identification with α = log p bit min-entropy and it is unique, has special soundness (SS), honest-verifier zero-knowledge (HVZK) and is random-self reducible (RSR). Moreover, if DLOG is (t, ε)-hard in par = (p, g, G) then ID5 is (t, ε)-KR-KOA secure.

**Proof.** The correctness of ID5 is straightforward to verify. We note that R in (R, St) is uniformly random over G. Hence, ID5 has log |G| = log p bit min-entropy. We show the other properties as follows.

**Uniqueness.** For all (X, x) ∈ IGen(par), (R := gR, St := r) ∈ P1(sk) and h ∈ {0, 1}n, the value s ∈ Zp satisfying \( g^s = X^R R \Leftrightarrow s = x h + r \) is uniquely defined.

**Special Soundness (SS).** Given two accepting transcripts (R, h, s) and (R, h′, s′) with h ≠ h′, we define an extractor algorithm Ext(X, h, s) := x := (s − s′)/(h − h′) such that, for all (X := gR, x) ∈ IGen(par), we have Pr[x∗ = X] = 1, since we have R = [s] − hX = [s] − h′X and then X = [(s − s′)/(h − h′)].

**Honest-Verifier Zero-Knowledge (HVZK).** Given public key X, we let Sim(X) first sample h ∈ R Zp and then output (R := [s] − hX, h, s). Clearly, (R, h, s) is a real transcript, since s is uniformly random over Zp and R is the unique value satisfying R = [s] − hX.

**Random-self Reducibility (RSR).** Algorithm Rerand and two deterministic algorithm Derand and Tran are defined as follows:

- **Rerand(X)** chooses τ ′ ∈ Zp and outputs (X′ := X + [τ ′], τ ′). We have that, for all (X, x) ∈ IGen(par), X′ is uniform and has the same distribution as X ′ ′, where (X ′ ′, x ′ ′) ∈ IGen(par).

- **Derand(X′, x′, τ ′)** outputs x∗ = x′ − τ ′. We have for all (X′, τ ′) ∈ IGen(par), X′ = x′ and x′ = x + τ ′ and thus x∗ = x.

- **Tran(X, x′, τ ′, (R′, h′, s))** outputs s = s′ − τ ′ · h′. We have for all (X′, τ ′) ∈ IGen(par), (R′, h′, s) is valid with respect to X′ := x + τ ′ then s = s′ − τ ′ · h′ = (x + τ ′)h′ + r − τ ′ · h′ = xh′ + r and (R′, h′, s) is valid with respect to X.

**Key-recovery against Key-only Attack (KR-KOA).** KR-KOA-security for ID is exactly the DLOG assumption.

We recall the OMDL assumption.

**Definition 3.3.3 (One-more Discrete Logarithm Assumption [BNPS03]).** We say that OMDL is (t, ε, Q)-hard in par = (p, P, G) if for all adversaries A running in time at most t and adaptively making at most Q queries to the discrete logarithm oracle DL,

\[
\Pr \left[ \text{for } i \in \{1, \ldots, Q + 1\} : X_i = [x_i] \mid X_1, \ldots, X_{Q+1} \xLeftRightArrow{G} (x_1, \ldots, x_{Q+1}) \xLeftRightArrow{A^{DL}(X_1, \ldots, X_{Q+1})} \right] \leq \varepsilon,
\]

where on input arbitrary group element Y the discrete logarithm oracle DL returns y ∈ Zp such that g^y = Y.

**Lemma 3.3.4 (Theorem 5.1 in [BP02]).** If the OMDL problem is (t, ε, Q)-hard then ID5 is (t′, ε′, QO)-IMP-AA secure, where ε′ ≤ \( \sqrt{2} + \frac{1}{t^2} \), t ≈ 2t′, and QO = Q.
We now define the $Q$-interactive discrete-logarithm problem which precisely models PIMP-KOA-security for $\text{ID}_S$, where $Q = Q_0$ is the number of parallel impersonation rounds.

**Definition 3.3.5** ($Q$-IDLOG). The interactive discrete-logarithm assumption $Q$-IDLOG is said to be $(t, \epsilon)$-hard in $\text{par} = (p, g, \mathbb{G})$ if for all adversaries $A$ running in time at most $t$ and making at most $Q$ queries to the challenge oracle $\mathcal{C}$,

$$\Pr \left[ s \in \{xh_i + r_i \mid i \in \{1, \ldots, Q\} \} \mid x \xleftarrow{} \mathbb{Z}_p; X := [x] \right] \leq \epsilon,$$

where on the $i$-th query $\mathcal{C}(g^i)$, the challenge oracle returns $h_i \xleftarrow{} \mathbb{Z}_p$ to $A$.

Lemma B.1 in [KMP16a] proves that in the generic group model, the $Q$-IDLOG problem is tightly equivalent to the $\text{IDLOG}$ problem. Note that the bound is independent of $Q$.

### 3.3.1.2 Schnorr’s Signature Scheme

Let $H : \{0,1\}^n \rightarrow \{0,1\}^n$ be a hash function with $n < \log_2(p)$. As ID$_S$ is commitment-recoverable we can use the alternative Fiat-Shamir transformation to obtain the Schnorr signature scheme $\text{Schnorr} := (\text{Gen}, \text{Sign}, \text{Ver})$.

<table>
<thead>
<tr>
<th>Gen(par):</th>
<th>Sign(sm, k):</th>
<th>Ver(sm, m, s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{sk} := x \xleftarrow{} \mathbb{Z}_p$</td>
<td>$r \xleftarrow{} \mathbb{Z}_p$; $R := [r]$</td>
<td>Parse $\sigma = (h, s) \in {0,1}^n \times \mathbb{Z}_p$</td>
</tr>
<tr>
<td>$\text{pk} := X = [x]$</td>
<td>$h = H(R, m)$</td>
<td>$R = [s] - hX$</td>
</tr>
<tr>
<td>Return $(\text{pk}, \text{sk})$</td>
<td>$s = x \cdot h + r \mod p$</td>
<td>If $h = H(R, m)$ then return 1</td>
</tr>
<tr>
<td></td>
<td>$\sigma = (h, s) \in {0,1}^n \times \mathbb{Z}_p$</td>
<td>Else return 0.</td>
</tr>
</tbody>
</table>

The DLOG problem is tightly equivalent to the 1-IDLOG problem by Lemma 3.2.5. Assuming the OMDL problem is hard, $\text{Schnorr}$ is wMP-MIM-secure and by Corollary 4.4 in [KMP16a] there cannot exist a tight implication $1$-IDLOG $\rightarrow$ Q-IDLOG meaning the bound from Lemma 3.2.6 is optimal. By Lemmas 3.2.7 and 3.2.8 the Q-IDLOG problem is tightly equivalent to SUF-CMA-security of $\text{Schnorr}$ in the programmable ROM. The latter is only tightly equivalent to MU-SUF-CMA-security in the programmable ROM (via Lemmas 3.2.9 and 3.2.10). Lemma B.1 in [KMP16a] improves this by proving that SUF-CMA security is tightly equivalent to MU-SUF-CMA-security in the standard model. Figure 3.8 summarizes the modular security implications for $\text{Schnorr}$.

![Figure 3.8: Security relations for the Schnorr signature scheme. All implications except the red one are tight.](image)

We derive the following concrete security implications.

**Lemma 3.3.6.** If DLOG is $(t, \epsilon)$-hard in $\text{par} = (p, P, \mathbb{G})$ then $\text{Schnorr}$ is $(t', \epsilon', Q_h, Q_h)$-SUF-CMA secure and $(t'', \epsilon'', N, Q_s, Q_h)$-MU-SUF-CMA secure in the programmable random oracle model, where

$$\epsilon' \leq 6(Q_h + 1) \cdot \frac{\epsilon}{t} + \frac{Q_s}{p} + \frac{1}{2^n},$$

$$\epsilon'' \leq 12(Q_h + 1) \cdot \frac{\epsilon}{t} + \frac{Q_s}{p} + \frac{1}{2^n}.$$
3.3.2 Generalized CP Identification/ KW Signature Scheme

In [KMP16b], under the DDH assumption, the Chaum-Pedersen identification [CP93] scheme is proved to be tightly PIMP-KOA secure, which implies the Katz-Wang signature [KW03] is tightly MU-UF-CMA secure in the random oracle model. This section generalizes this result by using the MDDH assumption (cf. Definition 2.2.2). Here we do not require any pairing.

3.3.2.1 Generalized Chaum-Pedersen Identification Scheme

Let \( \text{par} := ([A] \in \mathbb{G}^{k+1}) \) be a set of system parameters, where \( A \in D_k \) and \( \mathbb{G} \) is a cyclic group of prime order \( p \). (Lemma 3.3.8.)

\[ \text{ID}_{\text{GCP}} := ( \text{IGen}, \text{P}, \text{ChSet}, \text{V}) \]

\[ \text{IGen(par):} \]
- \( \text{sk} := x \in \mathbb{Z}_p^k \)
- \( \text{pk} := [X] := [Ax] \in \mathbb{G}^{k+1} \)
- \( \text{ChSet} := \mathbb{Z}_p \)
- \( \text{Return} (\text{pk}, \text{sk}) \)

\[ \text{V(pk, [R], h, s):} \]
- If \( [R] = [Ax] - h[X] \) then return 1
- Else return 0

Clearly, all security results of Schnorr can carry over to the generalized Chaum-Pedersen identification scheme, i.e., \( \text{ID}_{\text{GCP}} \) is at least as secure as \( \text{ID}_S \). That also means that we cannot hope for tight PIMP-KOA security from the DLOG assumption. Instead, for the generalized Chaum-Pedersen identification scheme, we give a direct tight proof of PIMP-KOA security under the MDDH assumption which we generalize the results from [KMP16b][KW03].

**Lemma 3.3.8.** \( \text{ID}_{\text{GCP}} \) is a canonical identification scheme with \( \alpha = k \log p \) bit min-entropy and it is unique, has honest-verifier zero-knowledge (HVZK) and is random-self reducible (RSR).

**Proof.** The correctness of \( \text{ID}_{\text{GCP}} \) is straightforward to verify. We note that \( R := [Ar] \in (R, St) \in \text{P}_1(\text{sk}) \) is computed by chosen a uniform vector \( r \in \mathbb{Z}_p^k \). Hence, \( \text{ID}_{\text{GCP}} \) has \( k \log |\mathbb{Z}_p| = k \log p \) bits min-entropy.

The proof of uniqueness, the proof of HVZK, and RSR is similar to that in \( \text{ID}_S \):

**UNIQUENESS.** For all \( ([X], x) \in \text{IGen(par)} \), \( (R := [R] := [Ar], St := r) \in \text{P}_1(\text{sk}) \) and \( h \in \mathbb{Z}_p \), the value \( s \in \mathbb{Z}_p^k \) satisfying \( \text{As} = hX + R \) and \( s = hX + r \) is uniquely defined, since \( A \) has rank \( k \).

**HONEST-VERIFIER ZERO-KNOWLEDGE (HVZK).** Given public key \([X] \), we let \( \text{Sim}([X]) \) first sample \( h \in \mathbb{Z}_p^k \) and \( s \in \mathbb{Z}_p^k \) and then output \( (R := [s] - h[X], h, s) \). Clearly, \( (R, h, s) \) is a real transcript, since \( s \) is uniformly random over \( \mathbb{Z}_p^k \) and \( R \) is the unique value satisfying \( R = [s] - h[X] \).

**RANDOM-SELF REDUCIBILITY (RSR).** Algorithm \( \text{Rand} \) and two deterministic algorithm \( \text{Derand} \) and \( \text{Tran} \) are defined as follows:

- **Rand** \((X)\): \( \text{choices} \ t' \in \mathbb{Z}_p^k \) and outputs \([X'] := [X] + [Ar'], t'\). We have that, for all \([X], x) \in \text{IGen(par)} \), \([X'] \) has the same distribution as \([X''], \) where \( ([X''], x'') \in \text{IGen(par)} \).

- **Derand** \((X), [X'], x', t')\): \( \text{outputs} \ x^* = x' - t' \). We have, for all \((X'), x', t' \in \text{Derand}[X] = [Ax]\) and \([X], x) \in \text{IGen(par)}, \), \([X'] = [Ax]\) and \( x = x + t' \) and thus \( Ax^* = Ax \).

- **Tran** \((X), [X]', t', ([R]', h', s')\) outputs \( s = s' - t' \cdot h' \). We have, for all \((X'), x', t' \in \text{Rand}(X) = [Ax]\), if \([R'], h', s')\) is valid with respect to \([X'] := [A(x + t')] \) then \( s = s' - t' \cdot h' = (x + t')h' + r - t' \cdot h' = xh' + r \) and \([R'], h', s')\) is valid with respect to \([X] = [Ax] \).

**Lemma 3.3.9.** If \( D_k \)-MDDH is \((t, \varepsilon)\)-hard relative to \( \text{GGen} \) in \( \mathbb{G} \) then \( \text{ID}_{\text{GCP}} \) is \((t', \varepsilon', Q_{\text{Cn}})\)-PIMP-KOA secure, where \( t \approx t' \) and \( \varepsilon \geq \varepsilon' - Q_{\text{Cn}}/p - 1/p \).

**Proof.** To prove PIMP-KOA security under MDDH, let \( A \) be an adversary that \((t', \varepsilon', Q_{\text{Cn}})\)-breaks PIMP-KOA security. We define a sequence of games in Figure 3.9. Note that \( G_0 = \text{PIMP-KOA} \). We have

**Lemma 3.3.10.** \( \text{Pr}[\text{PIMP-KOA} \Rightarrow 1] = \text{Pr}[G_0 \Rightarrow 1] \).
Proof. We apply an information-theoretical argument to show in Lemma 3.3.11. If Lemma 3.3.13. the first bound matches [KW03, Theorem 1].

The Katz-Wang signature is the instantiation of our generalized scheme with Fiat-Shamir transformation to obtain a signature scheme which is known as the Katz-Wang signature scheme.

3.3.2 Generalized Katz-Wang Signature scheme

Let $H : \{0, 1 \}^* \rightarrow \mathbb{Z}_p$ be a hash function. As $\text{ID}_{GCD}$ is commitment-recoverable we can use the alternative Fiat-Shamir transformation to obtain a signature scheme which is known as the Katz-Wang signature scheme $\text{GKW} := (\text{Gen}, \text{Sign}, \text{Ver})$.

![Figure 3.9: Games $G_0, G_1$ for the proof of Lemma 3.3.9](image)

**Lemma 3.3.11.** There exists an adversary $B$ that $(t, \varepsilon)$-breaks the MDDH assumption in $\text{par}$ with $t \approx t'$ and $| \Pr[G_A^0 \Rightarrow 1] - \Pr[G_A^0 \Rightarrow 1] | \leq \varepsilon$.

**Proof.** Games $G_0$ and $G_1$ only differ in the distribution of $pk := [X]$ returned by INITIALIZE, namely, $X \in \text{span}(A)$ (in $G_0$) or uniform (in $G_1$). From that, we obtain a straightforward reduction to break the $D_k$-MDDH problem with $(t, \varepsilon)$ as stated in the lemma.

**Lemma 3.3.12.** $\Pr[G_A^i \Rightarrow 1] \leq 1/p + Q_{\text{CMA}}/p$.

**Proof.** We apply an information-theoretical argument to show in $G_1$ even an computationally unbounded adversary $A$ can only win with probability $Q_{\text{CMA}}/p$. For each index $i \in \{1, \ldots, Q_{\text{CMA}} \}$, $A$ first commits to $[R_i]$ (for arbitrary $R_i \in Z_p^{k+1}$) and then gets the response $h_i \in \{0, 1 \}^n$. $A$ can only win if there exists an $s_i \in Z_p^k$ such that

$$R_i = As_i - h_iX$$  \hspace{1cm} (3.4)

Except with probability $1/p$, $X \notin \text{span}(A)$ and matrix $(A|X) \in Z_p^{(k+1) \times (k+1)}$ has full rank. Thus, for each $R_i$, there exists only one $(s_i, -h_i)$ satisfies Equation (3.4). Since $h_i$ is chosen uniformly at random from $\{0, 1 \}^n$ by the simulator, $A$ can win in $G_1$ with probability at most $1/|Z_p| = 1/p$. By the union bound we obtain the bound $Q_{\text{CMA}}/p$ as claimed.

Combining Lemmas 3.3.10 to 3.3.12 we have $\varepsilon' \leq \varepsilon + (Q_{\text{CMA}} + 1)/p$ as claimed in Lemma 3.3.9.

3.3.2.2 Generalized Katz-Wang Signature scheme

By our results [Lemmas 3.2.7, 3.2.8, and 3.2.10], we obtain the following concrete security statements. The Katz-Wang signature is the instantiation of our generalized scheme with DDH ($k = 1$). For $k = 1$, the first bound matches [KW03, Theorem 1].

**Lemma 3.3.13.** If $D_{F_k}$-MDDH is $(t, \varepsilon)$-hard relative to $G\text{Gen}$ in $G$ then $KW$ is $(t', \varepsilon', Q_s, Q_h)$-SU-F-CMA secure and $(t'', \varepsilon'', N, Q_s, Q_h)$-MU-SU-F-CMA secure in the programmable random oracle model, where

$$\frac{\varepsilon'}{t'} \leq \frac{\varepsilon}{t} + \frac{Q_s}{p^k} + \frac{1}{p},$$

$$\frac{\varepsilon''}{t''} \leq 4 \cdot \frac{\varepsilon}{t} + \frac{Q_s}{p^k} + \frac{1}{p}.$$
3.3.3 Guillou-Quisquater Identification/Signature Scheme

3.3.3.1 Background on RSA and Notations

For \( n \in \mathbb{N} \) we denote by \( \mathbb{P}_{n/2} \) the set of all \( n/2 \)-bit primes and \( \text{RSA}_n := \{(N = pq, p, q) \mid p, q \in \mathbb{P}_{n/2}, p \neq q\} \). Let \( \phi(N) := (p-1)(q-1) \) be Euler’s totient function for \( (N, p, q) \in \text{RSA}_n \). Let \( \mathcal{R} \) be a relation on \( p \) and \( q \). By \( \text{RSA}_n[\mathcal{R}] \) we denote the subset of \( \text{RSA}_n \) for that the relation \( \mathcal{R} \) holds on \( p \) and \( q \).

Let \( n \in \mathbb{N} \) and \( 0 < c < \frac{1}{2} \) be a constant. We define the following two distributions.

\[
\mathcal{I}_{n,c} := \{(N, e) \mid e \in \mathbb{P}_{cn}; \langle N, p, q \rangle \in \text{RSA}_n[\langle \gcd(e, \phi(N) = 1 \rangle]\}
\]

\[
\mathcal{L}_{n,c} := \{(N, e) \mid e \in \mathbb{P}_{cn}; \langle N, p, q \rangle \in \text{RSA}_n[p = 1 \mod e, p \neq 1 \mod e^2, q \neq 1 \mod e]\}
\]

Using the above notation, we recall the Phi-hiding assumption [CMS99, KOS10, KK12].

**Definition 3.3.14 (Phi-hiding Assumption).** The Phi-hiding problem \( \phi-H_{n,e} \) is \((t, \varepsilon)\)-hard if for all adversaries \( A \) running in time at most \( t \),

\[
\left| \Pr[1 \not\in A(N, e) \mid \langle N, e \rangle \in \mathcal{I}_{n,c}] - \Pr[1 \not\in A(N, e) \mid \langle N, e \rangle \in \mathcal{L}_{n,c}] \right| \leq \varepsilon.
\]

3.3.3.2 Guillou-Quisquater Identification Scheme

Let \( \text{par} = (N, e) \in \mathcal{I}_{n,c} \) be system parameters. The Guillou-Quisquater identification scheme \( \text{ID}_{GQ} := (\text{IGen}, P, \text{ChSet}, V) \) is defined as follows, where \( \mathbb{Z}_N^* := \{y \in \mathbb{Z}_N \mid \gcd(y, N) = 1\} \).

<table>
<thead>
<tr>
<th>lGen(par):</th>
<th>P1(sk):</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{sk} := x \not\in \mathbb{Z}_N^* )</td>
<td>( r \not\in \mathbb{Z}_N; R = r^e \mod N )</td>
</tr>
<tr>
<td>( \text{pk} := X := x^e \mod N )</td>
<td>( \text{St} := r )</td>
</tr>
<tr>
<td>( \text{ChSet} := \mathbb{Z}_n )</td>
<td>( \text{Return } (R, \text{St}) )</td>
</tr>
<tr>
<td>( \text{Return } (\text{pk}, \text{sk}) )</td>
<td>( \text{Parse } \text{St} = r )</td>
</tr>
</tbody>
</table>

\( \text{V(pk, R, h, s)}: \)

- If \( R = s^e \cdot X^{-h} \mod N \) and \( \langle (R, s) \rangle \in \mathbb{Z}_N^* \times \mathbb{Z}_N^* \), then return 1
- Else return 0

It is easy to prove IMP-KOA security of \( \text{ID}_{GQ} \) under the standard RSA assumption. Using our framework this implies MU-UF-CMA security of the implies GQ signature scheme, with an unavoidable security loss of \( O_\eta \). Under the \( \phi-H_{n,e} \) assumption we can, however, give a direct tight proof of \( \text{PIMP-KOA} \) security, which is similar to [ABP13].

**Lemma 3.3.15.** \( \text{ID}_{GQ} \) is a canonical identification scheme with \( \alpha = \log(\phi(N)) \) bit min-entropy and it is unique, has special sound (SS), honest-verifier zero-knowledge (HVZK) and random-self reducible (RSR). Moreover, if \( \phi-H_{n,e} \) is \((t, \varepsilon)\)-hard then \( \text{ID}_{GQ} \) is \((t', e', Q_{Cn})\)-\( \text{PIMP-KOA} \) secure, where \( t \approx t' \) and \( \varepsilon \geq \varepsilon' = (Q_{Cn} + 1)/e \geq \varepsilon' - (Q_{Cn} + 1)/2^{cn} \).

**Proof.** The correctness of \( \text{ID}_{GQ} \) is straightforward to verify. We note that \( R \not\in \mathbb{P}_1(\text{sk}) \) is uniformly random over \( \mathbb{Z}_N^* \). Hence, \( \text{ID}_{GQ} \) has \( \log |\mathbb{Z}_N^*| = \log(\phi(N)) \) bit min-entropy. We show the other properties as follows.

**Uniqueness.** For all \((X, x) \in \text{IGen(par)}\), \( R := r^e \in \mathbb{P}_1(\text{sk}) \) and \( h \in \mathbb{Z}_e \), the value \( s = x^h \cdot r \mod N \) is uniquely defined, since \( \gcd(e, \phi(N)) = 1 \).

**Special Soundness (SS).** Given two accepting transcripts \((R, h, s)\) and \((R, h', s')\) with \( h \neq h' \) (wlog. let \( h > h' \)), we have \( s^e X^h = R = s'^e X^{h'} \mod N \) and \( (s/s')^e = X^{h-h'} \mod N \). Since \( h, h' \in \mathbb{Z}_e \), \( \gcd(e, h-h') = 1 \). Applying the extended Euclidean algorithm we can compute \( A, B \in \mathbb{Z}_N^* \) such that

\[
Ae + B(h-h') = \gcd(e, h-h') = 1.
\]

Then we define an extractor algorithm \( \text{Ext}(X, R, h, s, h', s') := x^+ := X^A (s/s')^B \) such that, for all \((X := x^e \mod N, x) \in \text{IGen(par)}\), we have \( \Pr[(x)^e = X \mod N] = 1 \), since we have \((x)^e = (X^A (s/s')^B)^e = X^{Ae} (s/s')^{Be} = X^{Ae} X^{B(h-h')} = X\).
HONEST-VERIFIER ZERO-KNOWLEDGE (HVZK). Given a public key \( pk = X \), we let \( \text{Sim}(pk) \) first sample \( h \overset{\$}{\leftarrow} \mathbb{Z}_e \) and \( s \overset{\$}{\leftarrow} \mathbb{Z}_N^* \) and then output \( (R := s^X \cdot h^m \mod N, h, s) \). Clearly, \( (R, h, s) \) is a real transcript, since \( (h, s) \) is uniformly random over \( \mathbb{Z}_e \times \mathbb{Z}_N^* \) and \( R \) is the unique value satisfying \( R = s^X \cdot h^m \mod N \).

RANDOM-SELF REDUCIBILITY (RSR). Algorithm \( \text{Rerand} \) and two deterministic algorithm \( \text{Derand} \) and \( \text{Tran} \) are defined as follows:

- **Derand**\((X_1, X_2, x, \tau_2)\) chooses \( \tau_2 \overset{\$}{\leftarrow} \mathbb{Z}_N^* \), computes \( X_2 := x \cdot \tau_2^e \mod N \) and returns \((X_2, \tau_2)\). We have that, for all \((X_1, x_1) \in \text{Gen}(\text{par})\), \( X_2 \) is uniform and has the same distribution as \( X_3 \), where \((X_2, \tau_2) \overset{\$}{\leftarrow} \text{Derand}(X_1) \) and \((X_3, x_3) \overset{\$}{\leftarrow} \text{Gen}(\text{par})\).

- **Rerand**\((X_1, X_2, x, \tau_2)\) outputs \( x^* = x \cdot (\tau_2 / \tau_2^e \mod N) \). We have, for all \((X_2, \tau_2) \overset{\$}{\leftarrow} \text{Rerand}(X_1 := x_1^e \mod N) \) with \((X_2, x_2) \overset{\$}{\leftarrow} \text{Gen}(\text{par})\), \( X_2 = x^e \mod N \) and \( x_2 = x_1 \cdot \tau_2 \mod N \) and thus \( x^* = x_1 \mod N \).

- **Tran**\((X_1, X_2, \tau_2, (R_2 := r^{\tau_2^e} \mod N, h_2, s_2))\) outputs \( s_1 = s_2 / r^{h_2} \mod N \). We have, for all \((X_2, \tau_2) \overset{\$}{\leftarrow} \text{Rerand}(X_1 := x_1^e \mod N) \), \((R_2, h_2, s_2)\) is valid with respect to \( X_2 := x_1 \cdot \tau_2^e \mod N \) then \( s_1 = s_2 / \tau_2^e = (x_1 \tau_2^e)^h_2 \cdot r^{h_2} / \tau_2^e \mod N \) and \((R_2, h_2, s_1)\) is valid with respect to \( X_1 \).

PIMP-KOA SECURITY. Let \( A \) be an adversary that \((t', e', Q_{\text{Ch}})\)-breaks PIMP-KOA-security. We build an adversary \( B \) against the \((t, e)\)-hardness of \( \psi \)-H\( \text{H}_n \) as follows. Adversary \( B \) inputs \((N, e)\), chooses \( X \overset{\$}{\leftarrow} \mathbb{Z}_N^* \) and defines \( pk := X \). On the \( i \)-th challenge query \( \text{Ch}(R_i) \), it returns \( h_i \overset{\$}{\leftarrow} \mathbb{Z}_e \). Eventually, \( A \) returns \( t^* \in [Q_{\text{Ch}}] \) and \( s_i \) and terminates. Finally, \( B \) outputs \( d := \text{V}(pk, R_{t^*}, h_{t^*}, s_{t^*}) \).

ANALYSIS OF B. If \((N, e) \overset{\$}{\leftarrow} \mathbb{L}_{n, e}\), then \( B \) perfectly simulates the PIMP-KOA game and hence \( \Pr[d = 1 \mid (N, e) \overset{\$}{\leftarrow} \mathbb{L}_{n, e}] = e' \). If \((N, e) \overset{\$}{\leftarrow} \mathbb{L}_{n, e} \), then we claim that even a computationally unbounded \( A \) can only win with probability \((Q_{\text{Ch}} + 1) / e \), i.e., \( \Pr[d = 1 \mid (N, e) \overset{\$}{\leftarrow} \mathbb{L}_{n, e}] \leq (Q_{\text{Ch}} + 1) / e \).

It remains to prove the claim. Let \( R_c := \{X \mid \exists x \in \mathbb{Z}_N^* : X = x^e \mod N\} \) be the set of all \( e \)-th residues in \( \mathbb{Z}_N^* \). For \( X, R \in \mathbb{Z}_N^* \) we first analyze

\[
p(X, R) := \Pr_{h \overset{\$}{\leftarrow} \mathbb{Z}_e} [\exists s \in \mathbb{Z}_N^* : s^e = X^h \cdot R \mod N].
\]

- **Case 1:** \( X \in R_c \). Then \( p(X, R) \leq 1 \) by choosing \( s := (X^{1/e})^h \cdot R^{1/e} \) if \( R \in R_c \).

- **Case 2:** \( X \notin R_c \). Then \( p(X, R) \leq 1/e \) because

\[
p(X, R)(p(X, R) - 1/e) = \Pr_{h \neq h} [\exists s, \tilde{s} : s^e = X^h \cdot R \mod N \wedge \tilde{s}^e = \tilde{X}^h \cdot R \mod N] = \Pr_{h \neq h} [\exists s, \tilde{s} : (s/\tilde{s})^e = X^{h-h} \mod N] = 0.
\]

The last equality holds, since \( \gcd(e, h - h) = 1 \) which implies that \((s/\tilde{s})^e/(h-h)\) is an \( e \)-th residue and cannot equal to \( X \notin R_c \).

Using the bounds on \( p(X, R) \) we obtain

\[
\Pr[A \text{ wins}] = \Pr[A \text{ wins} \mid X \in R_c] \Pr[X \in R_c] + \Pr[A \text{ wins} \mid X \notin R_c] \cdot \Pr[X \notin R_c] \\
\leq \frac{1}{e} + (1 - \frac{1}{e}) \Pr_{h_1, \ldots, h_{Q_{\text{Ch}}}} [\exists s_i : s_i^e = X^{h_i} \cdot R \mod N \mid X \notin R_c] \\
\leq \frac{1}{e} + (1 - \frac{1}{e}) \frac{Q_{\text{Ch}}}{e} \\
\leq \frac{Q_{\text{Ch}} + 1}{e}
\]

This completes the proof of the claim.

### 3.3.3.3 Guillou-Quisquater Signature Scheme

Let \( H : \{0, 1\}^* \rightarrow \mathbb{Z}_e \) be a hash function. As \( \text{IDGQ} \) is commitment-recoverable we can use the alternative Fiat-Shamir transformation to obtain the Guillou-Quisquater signature scheme \( \text{GQ} := (\text{Gen}, \text{Sign}, \text{Ver}) \).

<table>
<thead>
<tr>
<th>\text{Gen(par)}</th>
<th>\text{Sign(sk, m)}</th>
<th>\text{Ver(sk, m, σ)}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{sk} := x \overset{$}{\leftarrow} \mathbb{Z}_N^* \quad X := x^e \mod N \quad pk := X \quad \text{Return (pk, sk)}</td>
<td>\text{r} \overset{$}{\leftarrow} \mathbb{Z}_N^<em>/R = r^e \mod N \quad h = H(R, m) \quad s = h^{-1} \cdot r \mod N \quad \text{σ} = (h, s) \in \mathbb{Z}_e \times \mathbb{Z}_N^</em> \quad \text{Return σ}</td>
<td>\text{σ} = (h, s) \in \mathbb{Z}_e \times \mathbb{Z}_N^* \quad \text{Parse σ} = (h, s) \in \mathbb{Z}_e \times \mathbb{Z}_N^* \quad R = s^X \cdot h^m \mod N \quad \text{If} h = H(R, m) \text{ and} R \in \mathbb{Z}_N^* \text{ then return 1} \quad \text{Else return 0}</td>
</tr>
</tbody>
</table>
By our results we obtain the following concrete security statements.

**Lemma 3.3.16.** If $\phi$-H$_{n,c}$ is $(t, \varepsilon)$-hard then GQ is $(t', \varepsilon', Q_s, Q_h)$-SUF-CMA secure and $(t'', \varepsilon'', N, Q_s, Q_h)$-MU-SUF-CMA secure in the programmable random oracle model, where

\[
\begin{align*}
\frac{\varepsilon'}{p'} &\leq \frac{\varepsilon}{t} + \frac{Q_s}{2^{n-2}} + \frac{3}{2^{2n}} , \\
\frac{\varepsilon''}{p''} &\leq 4 \cdot \frac{\varepsilon}{t} + \frac{Q_s}{2^{n-2}} + \frac{3}{2^{2n}}.
\end{align*}
\]
4.1 Message Authentication Codes

We use the standard definition of a (randomized) message authentication code $\text{MAC} = (\text{Gen}_\text{MAC}, \text{Tag}, \text{Ver}_\text{MAC})$, where $\text{sk}_\text{MAC} \leftarrow \text{Gen}_\text{MAC}(\text{par})$ returns a secret key, $\tau \leftarrow \text{Tag}(\text{sk}_\text{MAC}, m)$ returns a tag $\tau$ on message $m$ from some message space $\mathcal{M}$, and $\text{Ver}_\text{MAC}(\text{sk}_\text{MAC}, m, \tau) \in \{0, 1\}$ returns a verification bit.

4.1.1 Affine MACs

Affine MACs over $\mathbb{Z}_p^n$ are group-based MACs with a specific algebraic structure.

**Definition 4.1.1** (Affine MAC). Let $\text{par}$ be system parameters containing a group $\mathcal{G} = (\mathbb{G}_2, q, g_2)$ of prime-order $q$ and let $n \in \mathbb{N}$. We say that $\text{MAC} = (\text{Gen}_\text{MAC}, \text{Tag}, \text{Ver}_\text{MAC})$ is affine over $\mathbb{Z}_p^n$ if the following conditions hold:

1. $\text{Gen}_\text{MAC}(\text{par})$ returns $\text{sk}_\text{MAC}$ containing $(B, x_0, \ldots, x_t, x'_0, \ldots, x'_t)$, where $B \in \mathbb{Z}_p^{n \times n'}$, $x_i \in \mathbb{Z}_p^n$, $x'_j \in \mathbb{Z}_p$, for some $n', t, t' \in \mathbb{N}$. We assume $B$ has rank at least one.
2. $\text{Tag}(\text{sk}_\text{MAC}, m \in B^t)$ returns a tag $\tau = ([t]_2, [u]_2) \in \mathbb{G}_2^n \times \mathbb{G}_2$, computed as
   \[
   t = Bs \in \mathbb{Z}_p^n \quad \text{for } s \in \mathbb{Z}_p^{n'}
   \]
   \[
   u = \sum_{i=0}^t f_i(m)x_i^T t + \sum_{i=0}^{t'} f'_i(m)x'_i \in \mathbb{Z}_p
   \]
   for some public defining functions $f_i : \mathcal{M} \rightarrow \mathbb{Z}_p$ and $f'_i : \mathcal{M} \rightarrow \mathbb{Z}_p$. Vector $s$ can be generated either pseudorandomly (cf. $\text{MAC}_{\text{PR}}[\mathcal{D}_b]$) or randomly (cf. $\text{MAC}_{\text{HPS}}[\mathcal{D}_b]$) and $u$ is the (deterministic) message-depending part.
3. $\text{Ver}_\text{MAC}(\text{sk}_\text{MAC}, m, \tau = ([t]_2, [u]_2))$ verifies if equation (4.2) holds.

The standard security notion for probabilistic MACs is unforgeability against chosen-message attacks UF-CMA \cite{DKPW12}. In this work we require pseudorandomness against chosen-message attacks (PR-CMA), which is stronger than UF-CMA. Essentially, we require that the values used for one single verification equation (4.2) on message $m^*$ are pseudorandom over $\mathbb{G}_1$ and $\mathbb{G}_T$. Moreover, we need additional security requirements, such as delegationability (Definition 4.1.3) for hierarchical identity-based encryption (HIBE) and anonymity (Definition 4.1.4) for anonymous HIBE.

Let $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, q, g_1, g_2, e)$ be an asymmetric pairing group such that $(\mathcal{G}_2, g_2, q)$ is contained in $\text{par}$. We define the PR-CMA security via games PR-CMA$_{\text{real}}$ and PR-CMA$_{\text{rand}}$ from Figure 4.1. Note that the output $([h]_1, [h_0]_1, [h_1]_T)$ of $\text{CHAL}(m^*)$ in game PR-CMA$_{\text{real}}$ can be viewed as a “token” for message $m^*$ to check verification equation (4.2) for arbitrary tags $([t]_2, [u]_2)$ via equation $e([h]_1, [u]_2) = e([t]_1, [h_0]_1) + [h_1]_T$. Intuitively, the pseudorandomness of $[h_1]_T$ is responsible for indistinguishability and of $[h_0]_1$ to prove anonymity of the IBE scheme.

**Definition 4.1.2.** An affine MAC over $\mathbb{Z}_p^n$ is $(t, \varepsilon, Q)$-PR-CMA-secure if for all PPT $A$ with running time $t$, $\Pr[\text{PR-CMA}^A_{\text{real}}] \Rightarrow 1] - \Pr[\text{PR-CMA}^A_{\text{rand}} \Rightarrow 1] \leq \varepsilon$ is negligible, where the experiments are defined in Figure 4.1.
We also define syntax and security requirements of *delegatable* affine MACs, which will be used to construct hierarchical identity-based encryption in Section 4.3.

**Definition 4.1.3** (Delegatable Affine MAC). An affine MAC over $\mathbb{Z}_p^n$ \footnote{Definition 4.1.1} is delegatable, if the message space is $\mathcal{M} = B^{\leq m}$ for some finite base set $B$, $\ell' = 0$ with $f_0'(m) = 1$, and there exists a public function $l : \mathcal{M} \to \{0, \ldots, \ell\}$ such that for all $m' \in \mathcal{M}$ with $m' = (m_1, \ldots, m_{p+1}) \in B^{p+1}$ and length $p$ prefix $m = (m_1, \ldots, m_p)$ of $m$, we have $l(m) \leq l(m')$ and

\[
f_i(m') = \begin{cases} f_i(m) & 0 \leq i \leq l(m) \\ f_i'(m') & l(m') < i \leq \ell \end{cases}.
\]

We note that for a delegatable MAC, equation (4.2) simplifies to

\[
u = \left(\sum_{i=0}^{l(m)} f_i(m)x_i^T + \sum_{i=l(m)+1}^{l(m')} f_i(m')x_i^T\right)t + f_0'(m)x_0'.
\]

Intuitively, this property will be used for HIBE user secret key delegation.

**Security requirements.** Let MAC be a delegatable affine MAC over $\mathbb{Z}_p^n$ with message space $\mathcal{M} = B^{\leq m} := \bigcup_{\ell=1}^{m} B^{\ell}$. To build a HIBE, we require a new notion denoted as $\text{HPR}_0$-CMA security. It differs from PR-CMA security in two ways. Firstly, additional values needed for HIBE delegation are provided to the adversary through the call to initialize and eval. Secondly, CHAL always returns a real $h_0$ which is the reason why our HIBE is not anonymous. (In fact, the additional values actually allow the adversary to distinguish real from random $h_0$.)

Consider the games from Figure 4.2.

**Definition 4.1.4.** A delegatable affine MAC over $\mathbb{Z}_p^n$ is $(t, \varepsilon, Q)$-HPR$_0$-CMA-secure if for all PPT adversaries $A$, $\Pr[\text{HPR-CMA}_{\text{real}}^A \Rightarrow 1] - \Pr[\text{HPR}_0$-CMA$_{\text{rand}}^A \Rightarrow 1] \leq \varepsilon$. 

---

**Figure 4.1:** Games PR-CMA$_{\text{real}}$ and PR-CMA$_{\text{rand}}$ for defining PR-CMA security. The boxed statement redefining $(h_0, h_1)$ are only executed in game PR-CMA$_{\text{rand}}$. 

**Figure 4.2:** Games HPR-CMA$_{\text{real}}$ and HPR$_0$-CMA$_{\text{rand}}$ for defining HPR$_0$-CMA security.
In order to construct an anonymous HIBE, we define the notion of APR-CMA-security (anonymity-preserving pseudorandomness against chosen-message attacks) for a delegatable affine MAC over $\mathbb{Z}_p^n$ with message space $M = B^{\leq m} := \bigcup_{i=1}^{m} B^i$. It differs from HPR-CMA-security in the sense that $\text{Eval}(m)$ will output the terms required for usk rerandomization, not INITIALIZE.

Consider the games from Figure 4.3, where the (publicly known) $\mu$ is defined as the rank of matrix $B$ output by $\text{GenMac}(\text{par})$.

**Definition 4.1.5.** An affine MAC over $\mathbb{Z}_p^n$ is $(t, \varepsilon, Q)$-APR-CMA-secure if for PPT $A$ with running time $t$ and $\text{Pr}[\text{APR-CMA}_{\text{real}}^A = 1] - \text{Pr}[\text{APR-CMA}_{\text{rand}}^A = 1] \leq \varepsilon$.

### 4.1.2 An Affine MAC from the Naor-Reingold PRF

Unfortunately, the (deterministic) Naor-Reingold pseudorandom function is not affine. Let $\text{PRF} : M \rightarrow \mathbb{Z}_p^n$ be a pseudorandom function and we use the following randomized version $\text{MAC}_{\text{NR}}[D_k] = (\text{GenMac}, \text{Tag}, \text{VerMac})$ of it based on any matrix assumption $D_k$. For the special case $D_k = \mathcal{L}_k$, it was implicitly given in [CW13]. Recall that, for any matrix $A \in \mathbb{Z}_p^{(k+1) \times k}$ we denote the upper $k$ rows by $\overline{A} \in \mathbb{Z}_p^{k \times k}$ and the last row by $A \in \mathbb{Z}_p^{1 \times k}$.

Note that $\text{MAC}_{\text{NR}}[D_k]$ is $n$-affine over $\mathbb{Z}_p^n$ with message space $M = \{0, 1\}^m$. Writing $x_{i,b} = x_{2i+b}$ we have $n = n' = k$, $\ell' = 0$, $\ell = 2m + 1$ and functions $f_0(m) = f_1(m) = 0$, $f_0(m) = 1$, and $f_{2i+b}(m) = (m_i = b)$ for $1 \leq i \leq m$. (To perfectly fit our definition, $x_{i,b}$ should be renamed to $x_{2i+b}$, but we conserve the other notations for better readability.)

We show $\text{MAC}_{\text{NR}}[D_k]$ satisfies a weaker version of PR-CMA security, called PR-CMA$_0$ security, where an adversary can query $\text{Eval}$ on $m$ at most once. This weaker security can be converted to PR-CMA security by generating $s$ with a pseudo-random function $\text{PRF}$ on $m$ such that $\text{Tag}$ always returns the same answer for the same $m$. In the following of this subsection, we assume an adversary only queries $\text{Eval}$ at most once for every $m$.

**Theorem 4.1.6.** If the $D_k$-MDDH problem is $(t, \varepsilon)$-hard in $G_2$ then $\text{MAC}_{\text{NR}}[D_k]$ is $(t', \varepsilon', Q)$-PR-CMA$_0$-secure, where $\varepsilon' \leq 4m(\varepsilon + 1/(p-1))$ and $t \approx t'$.

Note that the security bound is (almost) tight, as $m$ is the bit-length of message space $M$. The proof follows the ideas from [CW13, NR97]. We use $m$ hybrids, where in hybrid $i$ all the (maximal $Q$) values $x_{i-1, m} \cdot t$ in the response to an $\text{Eval}$ query are replaced by uniform randomness. Here $m^*$ is the message from the challenge query. We use the $Q$-fold $D_k$-MDDH assumption to interpolate between the hybrids,
Lemma 4.1.9. There exists an adversary \( t \) tightly implied by the standard Lemma 4.1.8. \( \Pr[\text{bit}] = \frac{1}{2} \) and \( \epsilon \hat{\text{B}}(G) \), the proof follows.

Proof. We prove Theorem 4.1.6 by defining a sequence of intermediate games as in Figure 4.4 and 4.6.

Let \( Q \) be the maximal number of \( \text{Eval} \) queries made by \( A \). We first build an adversary \( B' \) (\( \hat{i}, \hat{\epsilon} \))-breaks the \( \text{PR-CMA} \)-security of \( \text{MAC}_{\text{NR}}[D_k] \). The construction of \( B' \) is described in Figure 4.5. Note that if \( \text{RF}_{i-1} \) and \( \text{RF}'_{i-1} \) are random functions, then \( \text{RF}_i \) is a random function.

Assume \( B' \) correctly guesses \( b = m \) (which happens with probability \( 1/2 \)). By the definition of \( \text{RF}_i \) and \( \text{RF}'_{i-1} \), we have \( \text{RF}_i(m_i) = \text{RF}_{i-1}(m_{i-1}) \), which implies \( \text{CHAL}(m^*) \) is identically distributed in \( G_2, i \). and \( G_{1, i-1} \).

We now analyze the output distribution of the \( \text{Eval} \) queries. First note that \( t \) is uniformly random over \( Z_p \) in both games \( G_i \) and \( G_{1, i-1} \). As for the distribution of \( u_i \), we only need to consider the case \( m_i = 1 - b \), since \( u_i = m \) for \( m_i = b \) is identically distributed in \( G_i \) and \( G_{1, i-1} \). Assume \( m_i = 1 - b \).
Figure 4.5: Description of $B'(\mathcal{G}, [A], [H])$ interpolating between the Games $G_{1,i}$ and $G_{1,i-1}$, where $H_c$ denotes the $c$-th column of $H$ and $\alpha_i : \{0, 1\} \rightarrow \{1, \ldots, Q\}$ is an injective function.

Write $H_c = AW_c + R_c$ for some $W_c \in \mathbb{Z}_p^k$, where $R_c = 0$ (i.e., $H$ is from the $D_k$-MDDH distribution) or $R_c$ is uniform. Then,

$$
    u_m = \sum_{j \neq i} x_{j,m}^T t_m + r^T A(s_m' + W_c) + r^T R_c + RF_{i-1}(m_{i-1}) \\
    = \sum_{j \neq i} x_{j,m}^T t_m + x_{i+1-b}^T A(s_m' + W_c) + r^T R_c + RF_{i-1}(m_{i-1}) \\
    = \sum_{j=0}^{m} x_{j,m}^T t_m + r^T R_c + RF_{i-1}(m_{i-1}).
$$

If $R_c = 0$, then $u_m$ is distributed as in game $G_{1,i-1}$. If $R_c$ is uniform, then define $RF'(m_{i-1}) := r^T R_c$ and $u$ is distributed as in $G_{1,i}$.

Lemma 4.1.10. $\Pr[\mathcal{G}_{1,m} \Rightarrow 1] = \Pr[\mathcal{G}_2 \Rightarrow 1]$.

Proof. In $G_{1,m}$, the values $u_m$ computed in $\text{Eval}(m)$ are masked by $RF_m(m)$ and are hence uniformly random for distinct $m$.

Finally, we do all the previous steps in reverse order, as shown in Figure 4.6, where the index $i$ of hybrid $H_{1,i}$ starts with $m$ and ends with 0. Clearly, $H_2 = G_2$ and $H_0 = \text{PR-CMA}_{\text{rand}}$. Following the arguments of Lemmas 4.1.8 to 4.1.10 in reverse order, one obtains the following lemma.

<table>
<thead>
<tr>
<th>INITIALIZE:</th>
<th>// Games $H_0$, $H_{1,i}$, $H_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b \leftarrow {0, 1}$</td>
<td></td>
</tr>
</tbody>
</table>
| For $j = 1, \ldots, m$ and $j' = 1, \ldots, 1$:
| If $j \neq i$ or $j' = 1$ then $x_{j,j'} \leftarrow \mathbb{Z}_p^k$
| $r \leftarrow \mathbb{Z}_p^{k+1}$; $x_{1,i-1} \leftarrow r^T A \in \mathbb{Z}_p^k$
| Return $\varepsilon$                   |
| CHAL(m*):
| Abort if $m_i \neq b$,
| $h \leftarrow \mathbb{Z}_p$; $x_0 = RF_{i-1}(m_{i-1})$
| $h_0 = (\sum_{j=1}^m x_{j,m}) h \in \mathbb{Z}_p^k$
| $h_1 = x_0 h \in \mathbb{Z}_p^k$
| Return $[h_1], [h_0], [h_1]_r$      |
| EVAL(m):
| $\mathcal{Q}_M = \mathcal{Q}_M \cup \{m\}$
| $c := \alpha_i(m_i)$
| $s_m' \leftarrow \mathbb{Z}_p^k$;
| $t_m = \mathbf{A}s_m + \mathbf{H}_i$
| $u_m = (\sum_{j=1}^m x_{j,m}) t_m + RF_{i-1}(m_{i-1})$ \quad $m_i = 1$
| $u_m = (\sum_{j \neq i} x_{j,m}) t_m + r^T (A s_m + H_i) + RF_{i-1}(m_{i-1})$ \quad $m_i = 1 - b$
| Return $([t_m], [u_m])$                     |
| FINALIZE(d \in \{0, 1\}):            |
| Return $d \land (m^* \notin \mathcal{Q}_M)$ |

Figure 4.6: Games $H_0$, $H_{1,i}$ ($m \geq i \geq 0$) and $H_2$ for the proof of Lemma 4.1.11
Lemma 4.1.11. If the $D_k$-MDDH assumption is $(t,\varepsilon)$-hard in $G_2$, then

$$2m(\varepsilon + 1/(p-1)) \geq |\Pr[G_2^A] \Rightarrow 1| - \Pr[\text{PR-CMA}_{\text{rand}}^A \Rightarrow 1]|$$

and $t \approx t'$.

This completes the proof of Theorem 4.1.6.

### 4.1.3 An Affine MAC from Hash Proof System

Let $D_k$ be a matrix distribution. We now combine the hash proof system for the subset membership problem induced by the $D_k$-MDDH assumption from [EHK+13] with the generic MAC construction from [DKPW12] and obtain the following MAC$_{\text{HPS}}[D_k]$ for $M = Z_p^\ell$.

<table>
<thead>
<tr>
<th>GenMAC(par):</th>
<th>Tag(sMAC, m):</th>
<th>VenMAC(sMAC, r, m):</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B \leftarrow D_k$</td>
<td>$s \leftarrow Z_p^k$</td>
<td>If $u = (x_0 + \sum_{i=1}^m m_i \cdot x_i^0) t + x_0'$ then return 1</td>
</tr>
<tr>
<td>$x_0, \ldots, x_l \leftarrow Z_p^{k+1}$</td>
<td>$t = Bs \in Z_p^{k+1}$</td>
<td>Else return 0</td>
</tr>
<tr>
<td>$x_0' \leftarrow Z_p$</td>
<td>$u = (x_0^0 + \sum_{i=1}^m m_i \cdot x_i^0) t + x_0'$</td>
<td></td>
</tr>
<tr>
<td>Return $skMAC = (B, x_0, \ldots, x_l, x_0')$</td>
<td>Return $\tau = (t[2], [u_2]) \in G_2^{k+1} \times G_2$</td>
<td></td>
</tr>
</tbody>
</table>

Note that MAC$_{\text{HPS}}[D_k]$ is $n$-affine over $Z_p^n$ with $n = k + 1$, $n' = 0$, and defining functions $f_0(m) = 1$, $f_1(m) = m$, and $f_0'(m) = 1$, where $m$ is the $i$-th component of $m$. For the moment we use $\ell = 1$ which already gives a MAC with exponential message space $M = Z_p$.

Combining [EHK+13] and [DKPW12] we obtain that MAC$_{\text{HPS}}[D_k]$ is UF-CMA under the $D_k$-MDDH assumption. The proof extends to show even PR-CMA security. Compared to MAC$_{\text{NR}}[D_k]$, we lose the tight reduction, but gain much shorter public parameters.

**Theorem 4.1.12.** If the $D_k$-MDDH problem is $(t,\varepsilon)$-hard in $G_2$, then MAC$_{\text{HPS}}[D_k]$ is $(t',\varepsilon', Q)$-PR-CMA secure where $t \approx t'$ and $\varepsilon' \leq 2Q/(1+1/p)$.

\[\text{Proof.}\] We prove Theorem 4.1.12 by defining a sequence of intermediate games as in Figure 4.7. Let $A$ be an adversary against the PR-CMA-security of MAC$_{\text{HPS}}[D_k]$. Game $G_0$ is the real attack game. In games $G_{i,j}$, the first $i - 1$ queries to the $\text{EVAL}$ oracle are answered with uniform values in $G_2^{k+1} \times G_2$ and the remaining are answered as in the real scheme. To interpolate between $G_{i,j}$ and $G_{i,j+1}$, we also define $G'_{i,j}$ which answers the $i$-th query to $\text{EVAL}$ by picking a random $t \leftarrow Z_p^{k+1}$. By definition, we have $G_0 = G_{1,1}$.
Lemma 4.1.13. \( \Pr[\text{PR-CMA}_{\text{real}}^0 \Rightarrow 1] = \Pr[G_0^0 \Rightarrow 1] = \Pr[G_{1,1}^1 \Rightarrow 1] \).

Lemma 4.1.14. There exists an adversary \( B \) that \((t, \varepsilon)\)-breaks the \( D_k \)-MDDH with \( t \approx t' \) and

\[
\varepsilon \geq |\Pr[G_{1,1}^1 \Rightarrow 1] - \Pr[G_{1,1}^1 \Rightarrow 1]|.
\]

Proof. Games \( G_{1,i} \) and \( G_{1,i}^1 \) only differ in the distribution of \( t \) returned by the EVAL oracle for its \( i \)-th query, namely, \( t \in \text{span}(B) \) or uniform. From that, we obtain a straightforward reduction to the \( D_k \)-MDDH Assumption.

Lemma 4.1.15. \( |\Pr[G_{1,i+1}^1 \Rightarrow 1] - \Pr[G_{1,i}^1 \Rightarrow 1]| \leq 1/p \).

Proof. At a high level, these two games are only separated by the 2-universality of the underlying hash proof system. Let \( m \) be the \( i \)-th query to EVAL and let \([t]_2, [u]_2\) be its tag. As \( m \neq m^* \), there exists an index \( i' \) such that \( m_{i'} \neq m_{i'}^* \), where \( m_{i'} \) (resp. \( m_{i'}^* \)) denotes the \( i' \)-th entry of \( m \) (resp. \( m^* \)). We use an information-theoretic argument to show that in \( G_{1,i}^1 \), the value \( u - x_0' \) is uniformly random. For simplicity, we assume \( x_0' \) and \( x_j \) \((j \notin \{0,i'\})\) are known to \( A \). Information-theoretically, adversary \( A \) may also learn \( B^\top x_0 \) and \( B^\top x_{i'} \) from the \( c \)-th query with \( c > i \). Thus, \( A \) information-theoretically obtains the following equations in the unknown variables \((x_0, x_{i'}) \in \mathbb{Z}_p^{2(t+1)} \):

\[
\begin{pmatrix}
B^\top x_0 \\
B^\top x_{i'} \\
h_0 \\
u - x_0'
\end{pmatrix}
= \begin{pmatrix}
B^\top \\
0 \\
h \cdot I_{k+1} \\
t^\top
\end{pmatrix}
\begin{pmatrix}
m_{i'} \\
m_{i'}^* \\
h \cdot I_{k+1} \\
m_{i'}^* t^\top
\end{pmatrix}
\begin{pmatrix}
x_0 \\
x_{i'}
\end{pmatrix}

\text{mod } \mathbb{Z}_p^{(2k+2) \times (2k+2)}
\]

where \( I_{k+1} \) is the \((k+1) \times (k+1)\) identity matrix. To show that \( u - x_0' \) is linearly independent of \( B^\top x_0 \), \( B^\top x_{i'} \) and \( h_0 \), we argue that the last row of \( M \) is linearly independent of all the other rows. Since \( t \notin \text{span}(B) \) (except with probability \( 1/p \)), \( t^\top \) is independent of \( B^\top \); by \( m_{i'} \neq m_{i'}^* \), the last row of \( M \) is linearly independent of rows \( 2k + 1 \) to \( 3k + 1 \). We conclude that \( u \) is uniformly random in \( A \)’s view.

Lemma 4.1.16. \( \Pr[G_{1,q+1}^1 \Rightarrow 1] = \Pr[G_{1,q+1}^1 \Rightarrow 1] \).

Proof. Note that \( A \) can ask at most \( Q \)-many EVAL queries. In both \( G_{1,q+1} \) and \( G_{2} \), all answers of EVAL are uniformly at random and independent of the secret keys \((x_0', x_0, \ldots, x_\ell)\). Hence, the values \( h_0 \) and \( h_1 \) from \( G_{1,q+1} \) are uniform in the view of \( A \).

We now do all the previous steps in the reverse order as in Figure 4.8. Then, by using the above arguments in a reverse order, we have the following lemma.

Lemma 4.1.17. There exists an adversary \( B \) that \((t, \varepsilon)\)-breaks the \( D_k \)-MDDH assumption with \( t \approx t' \) and \( Q(\varepsilon + 1/p) \geq |\Pr[\text{PR-CMA}_{\text{rand}}^A \Rightarrow 1] - \Pr[G_{2}^1 \Rightarrow 1]| \).

Theorem 4.1.12 follows by combining Lemmas 4.1.13-4.1.17.

HPR0-CMA-security. We note that \( \text{MAC}_{\text{HPS}}[D_k] \) with message space \( \mathcal{M} = B^{\leq m} = (\mathbb{Z}_p^*)^{\leq m} \) is delegatable. One should remark the change on \( B \), where we now define \( B = \mathbb{Z}_p^* \) to avoid having a collision between the MAC of \( m \) and the MAC of \( m || 0 \).

Theorem 4.1.18. If the \( D_k \)-MDDH assumption is \((t, \varepsilon)\)-hard in \( G_2 \), then \( \text{MAC}_{\text{HPS}}[D_k] \) is \((t', \varepsilon', Q)\)-HPR0-CMA-secure with \( t \approx t' \) and \( \varepsilon' \leq 4Q\varepsilon + 2Q/p \).

Proof. We prove Theorem 4.1.18 by defining a sequence of intermediate games \( G_0 \rightarrow G_2 \) as in Figure 4.9.

Lemma 4.1.19. \( \Pr[\text{HPR-CMA}_{\text{rand}}^A \Rightarrow 1] = \Pr[G_{0}^1 \Rightarrow 1] \).

Similar to the proof of Theorem 4.1.12, we define the game \( G_1 \). Different to Theorem 4.1.12, we always honestly compute \( t = Bs \) in \( G_{1,1} \), otherwise, too much information about \( x_i \) will be leaked from terms \([d_i]_2\). To interpolate between \( G_{1,1} \) and \( G_{1,1+1} \), we also define \( G_{1,1+1} \) to \( G_{1,3,1} \).

By the same arguments as in Section 4.1.13 we have the following two lemmas:

Lemma 4.1.20. \( \Pr[G_{0}^1 \Rightarrow 1] = \Pr[G_{1,1}^1 \Rightarrow 1] \).
Lemma 4.1.22. Let \( \mathbb{M} \) be the \( c \)-th query (\( 1 \leq c \leq Q \)).

Proof. At a high level, those two games are only separated by the 2-universality of the underlying hash function. Let \( \mathcal{H}_0 \) and \( \mathcal{H}_1 \) be the two games, and \( \mathcal{H}_0 \) is the game where \( \mathcal{H}_0 \) is initialized with a random value. Similar to Lemma 4.1.16, since \( \mathcal{H}_0 \) is independent of the \( c \)-th query, \( \mathcal{H}_0 \) and \( \mathcal{H}_1 \) are independent of \( x_0^* \) in the view of \( \mathcal{H}_1 \).

We have the following two straightforward lemmas.

Lemma 4.1.23. There exists an adversary \( D \) that \( (t, \varepsilon) \)-breaks the \( D_k \)-MDDH assumption in \( \mathbb{G}_2 \) with \( t \approx t' \) and \( \varepsilon \geq |\Pr[\mathcal{G}_1^{A} \Rightarrow 1] - \Pr[\mathcal{G}_1^{B} \Rightarrow 1]| \).

Lemma 4.1.24. \( \Pr[\mathcal{G}_1^{A} \Rightarrow 1] = \Pr[\mathcal{G}_1^{B} \Rightarrow 1] \)

Lemma 4.1.25. \( \Pr[\mathcal{G}_1^{A} \Rightarrow 1] = \Pr[\mathcal{G}_1^{Q+1} \Rightarrow 1] \).

Proof. In \( \mathbb{G}_2 \), we replace \( h_1 \) output by \( \mathcal{H}_0(m^*, c) \) with a random value. Similar to Lemma 4.1.16, all the answers of \( \text{Eval} \) are independent of \( x_0^* \) (in particular, \( u \) is uniformly random), \( h_1 \) is uniformly random in the view of \( \mathcal{A} \).
We apply all the arguments before in a reverse order and then we easily get the following:

Lemma 4.1.26. There exists an adversary \( D \) that \((t, \varepsilon)\)-breaks the \( \mathcal{D}_k \)-MDDH assumption in \( G_2 \) with \( t \approx t' \) and \( |Pr[HPR_0-\text{CMA}^A_{\text{rand}} \Rightarrow 1] - Pr[G^A_0 \Rightarrow 1]| \leq Q(2\varepsilon + 1/p) \).

The proof of the theorem follows by combining Lemmas 4.1.19-4.1.26.

ANONYMITY. Unfortunately, \( \text{MAC}_{\text{HPR}}[\mathcal{D}_k] \) is unlikely to be \( \text{APR-CMA} \)-secure, since in the security proof the value \( u \) is not pseudorandom and, thus, we can not make \( h_0 \) random. The following theorem states that \( \text{MAC}_{\text{HPS}}[\mathcal{D}_k] \) with message space \( \mathcal{M} = \mathbb{B}^\leq m = (\mathbb{Z}_p^*)^\leq m \) is anonymous and delegatable.

Theorem 4.1.27. If the \( \mathcal{D}_k \)-MDDH problem is \((t, \varepsilon)\)-hard in group \( G_2 \), then \( \text{MAC}_{\text{HPS}}[\mathcal{D}_k] \) is \((t', \varepsilon', Q)\)-\( \text{APR-CMA} \)-secure, where \( t \approx t' \) and \( \varepsilon' \leq 4Q\varepsilon + 2Q/p \), where \( Q \) is the maximal number of queries to \( \text{Eval}(\cdot) \).

Proof. We prove Theorem 4.1.27 by defining a sequence of intermediate games as in Figure 4.10. Let \( A \) be an adversary against the \( \text{APR-CMA} \)-security of \( \text{MAC}_{\text{HPS}}[\mathcal{D}_k] \).

\( G_0 \) is the real attack game and we have:

Lemma 4.1.28. \( Pr[\text{APR-CMA}^A_{\text{rand}} \Rightarrow 1] = Pr[G_0^A \Rightarrow 1] \).

In games \( G_{1,i} \), for the first \( i-1 \) queries to the \( \text{Eval} \) oracle, \((u, u)\) is answered with uniformly random values and the rest are answered as in the real scheme. To interpolate between \( G_{1,i} \) and \( G_{1,i+1} \), we also define \( G_{1,i} \) to \( G_{1,3,i} \).

By the same arguments as in Theorem 4.1.18 we have the following two lemmas:

Lemma 4.1.29. \( Pr[G^A_0 \Rightarrow 1] = Pr[G^A_{1,1} \Rightarrow 1] \).

Lemma 4.1.30. There exists an adversary \( D \) that \((t, \varepsilon)\)-breaks the \( \mathcal{D}_k \)-MDDH assumption in \( G_2 \) with \( t \approx t' \) and \( \varepsilon \geq |Pr[G^A_{1,i} \Rightarrow 1] - Pr[G^A_{1,i+1} \Rightarrow 1]| \).

Lemma 4.1.31. \( |Pr[G^A_{2,i} \Rightarrow 1] - Pr[G^A_{1,i} \Rightarrow 1]| \leq 1/p \).

Proof. Similar to Lemma 4.1.22, let \( m \) be the \( i \)-th query to \( \text{Eval} \). As \( m \neq m^* \), let \( i' \) be the smallest index such that \( m_{i'} \neq m^* \), where \( m_{i'} \) (resp. \( m^* \)) denotes the \( i' \)-th entry of \( m \) (resp. \( m^* \)). We apply an information-theoretical argument to show in \( G_{1,i} \) the values \( u \) and \( u \) are uniformly random. For simplicity, we assume \( x_0^* \) and \( x_j \) (\( j \notin \{0, i'\} \)) are learned by \( A \). Similar to the proof of Lemma 4.1.22 we assume \( A \) learns \( B^\top x_0 \) and \( B^\top x_{i'} \). By the definition of \( \text{APR-CMA} \)-security (Definition 4.1.5), \( m \) can not
be a prefix of $m^*$. Thus, either ($i' \leq |m^*|$ and $i' \leq |m|$) (Case 1) or $i' \geq |m^*| + 1$ and $m^*_j = m_j$ for all $j \leq |m^*|$ (Case 2).

**Case 1:** From the execution of $G_{1,1,i}$, A information-theoretically obtains the following equations in the unknown variables $x_0, x_{i'}$:

$$
\begin{pmatrix}
B^\top x_0 \\
B^\top x_{i'} \\
h_0 \\
u - x'_0 \\
u
\end{pmatrix}
= \begin{pmatrix}
B^\top & 0 & B^\top & 0 \\
h \cdot I_{k+1} & m^*_j \cdot h \cdot I_{k+1} \\
t^\top & m^*_j t^\top \\
t^\top & m^*_j t^\top
\end{pmatrix}
\begin{pmatrix}
x_0 \\
x_{i'}
\end{pmatrix},
$$

where $h_0$ is from CHAL($m^*$), $u$ and $u$ are from EVAL(m), and $I_{k+1}$ is the $(k+1) \times (k+1)$ identity matrix.

We note that the terms $|d_{i'}|_2$ and $|D_{j'}|_2$ from the $c$-th EVAL queries ($c \neq i$) only leak the information about $x_{i'}$ on the span of $B$. Since $(t, T)$ is chosen uniformly from $Z_{2^{(k+1) \times (k+1)}}$ in $G_{1,1,i}$, $t$ and $T$ are not in the span of $B$ and $(t, T)$ has rank $k + 1$ (except with probability at most $1/p$), which implies $u - x'_0$ and $u$ are independent from $B^\top x_0$ and $B^\top x_{i'}$ and $u - x'_0$ and $u$ are independent from each other. By $m_{i'} \neq m_j$, $u - x'_0$ and $u$ are also independent of $h_0$.

**Case 2:** Similarly, from the execution of $G_{1,1,i}$, A information-theoretically obtains the following equations in the unknown variables $x_0, x_{i'}$:

$$
\begin{pmatrix}
B^\top x_0 \\
B^\top x_{i'} \\
h_0 \\
u - x'_0 \\
u
\end{pmatrix}
= \begin{pmatrix}
B^\top & 0 \\
h \cdot I_{k+1} & 0 \\
t^\top & m^*_j t^\top \\
t^\top & m^*_j t^\top
\end{pmatrix}
\begin{pmatrix}
x_0 \\
x_{i'}
\end{pmatrix},
$$

As in Case 1, $u - x'_0$ and $u$ are linearly independent of $B^\top x_0$ and $B^\top x_{i'}$. It is easy to see that $u - x'_0$ and $u$ are linearly independent of $h_0$. $u - x'_0$ and $u$ are independent from each other, since $(t, T)$ is a
full rank matrix in $\mathbb{Z}_p^{(k+1) \times (k+1)}$

We conclude $u$ and $u$ are distributed uniformly at random in $G_{1,1,\ldots}$.

We have the following two straightforward lemmas.

**Lemma 4.1.32.** There exists an adversary $D$ that $(t, \varepsilon)$-breaks the $D_k$-MDDH assumption in $G_2$ with $t \approx t'$ and $\varepsilon \geq |Pr[\mathcal{G}^A_{1,1,\ldots}] - Pr[\mathcal{G}^A_{1,2,\ldots}]|$.

**Lemma 4.1.33.** $Pr[\mathcal{G}^A_{1,3,\ldots}] = 1 = Pr[\mathcal{G}^A_{1,4,\ldots}]$

**Lemma 4.1.34.** $Pr[\mathcal{G}^A_{1,5,\ldots}] = 1 = Pr[\mathcal{G}^A_{1,6,\ldots}]$

**Proof.** Note that $A$ can ask at most $Q$-many EVAL queries. In both $G_{1,7,\ldots}$ and $G_2$, all answers of EVAL are uniformly random and independent of the secret keys $(x_0, x_0, \ldots, x_L)$ where $L = \max\{|m_1|, \ldots, |m_Q|\}$. Hence, the values $h_0$ and $h_1$ from $G_{1,7,\ldots}$ are uniform in the view of $A$.

We apply the above arguments in a reverse order, we have the following lemma.

**Lemma 4.1.35.** There exists an adversary $D$ that $(t, \varepsilon)$-breaks the $D_k$-MDDH assumption in $G_2$ with $t \approx t'$ and $|Pr[\mathcal{APR-CMA}_A] - Pr[\mathcal{G}^A_{1,3,\ldots}]| \leq Q(2\varepsilon + 1/p)$.

### 4.2 Identity-based Encryption from Affine MACs

In this section, we will present our transformation $\text{IBE[MAC, }D_k\text{]}$ from affine MACs to IBE based on the $D_k$-MDDH assumption.

Let $D_k$ be a matrix distribution that outputs matrices $A \in \mathbb{Z}_p^{(k+1) \times k}$. Let MAC be an affine MAC over $\mathbb{Z}_p^n$ with message space $\mathcal{M}$. Our IBKEM $\text{IBKEM[MAC, }D_k\text{]} = (\text{Gen, USKGen, Enc, Dec})$ for key-space $K = G_T$ and identity space $\mathcal{I}$ is defined as follows:

<table>
<thead>
<tr>
<th>Gen(par):</th>
<th>Enc(pk.id):</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \triangleq D_k$</td>
<td>$r \triangleq \mathbb{Z}_p^k$</td>
</tr>
<tr>
<td>$\text{sk}_{\text{MAC}} = (B, x_0, \ldots, x_r, x'_0, \ldots, x'_r) \triangleq \text{GenMAC(par)}$</td>
<td>$c_0 = Ar \in \mathbb{Z}_p^{k+1}$</td>
</tr>
<tr>
<td>For $i = 0, \ldots, \ell$ : $Y_i \triangleq \mathbb{Z}_p^{k \times n}, Z_i = (Y_i^\top</td>
<td>x_i) \cdot A \in \mathbb{Z}_p^{n \times k}$</td>
</tr>
<tr>
<td>For $i = 0, \ldots, \ell'$ : $y'_i \triangleq \mathbb{Z}_p^k, x'_i = (y'_i^\top</td>
<td>x'_i) \cdot A \in \mathbb{Z}_p^{n \times k}$</td>
</tr>
<tr>
<td>$\text{sk} := (\text{sk}<em>{\text{MAC}}, (Y_i)</em>{0 \leq i \leq \ell}, (y'<em>i)</em>{0 \leq i \leq \ell'})$</td>
<td>$\text{Return } K = [K_T]r$ and $\mathcal{C} = ([c_0], [c_1])_1 \in G_1^{n+k+1}$</td>
</tr>
<tr>
<td>Return $(pk, sk)$</td>
<td>$\text{Dec(usk{id, id, C})}$:</td>
</tr>
<tr>
<td>$(t)[2], [n][2] \triangleq \text{Tag(usk_{MAC}, id)}$</td>
<td>$\text{Parse usk{id} = ([t][2], [n][2], [v][2])}$</td>
</tr>
<tr>
<td>$v = \sum_{i=0}^\ell f_i(id)Y_i^\top t + \sum_{i=0}^{\ell'} f_i(id)y'_i \in \mathbb{Z}_p^n$</td>
<td>$\text{Parse C = ([c_0], [c_1])}$</td>
</tr>
<tr>
<td>$\text{Return } \text{usk{id}} := ([t][2], [n][2], [v][2]) \in G_2^{n+k+1}$</td>
<td>$K = e([c_0][1], [v][2]) - e([c_1][1], [t][2])$</td>
</tr>
<tr>
<td>$\text{Return } K \in G_T$</td>
<td>$\text{Return } K = G_T$</td>
</tr>
</tbody>
</table>

The intuition behind our construction is that the values $[Z_i][1], [x'_i][1]$ from $pk$ can be viewed as perfectly hiding commitments to the secrets keys $\text{sk}_{\text{MAC}} = (x_1, \ldots, x_r, x'_1, \ldots, x'_r)$ of MAC. User secret key generation computes the MAC tag $\tau = ([t][2], [n][2]) \triangleq \text{Tag(usk_{MAC})}$ plus a “non-interactive zero-knowledge proof” $[v][2]$ proving that $\tau$ was computed correctly with respect to the commitments. As the MAC is affine, the NIZK proof has a very simple structure. The encryption algorithm is derived from a randomized version of the NIZK verification equation. Here we again make use of the affine structure of MAC.

To show correctness of $\text{IBKEM[MAC, }D_k\text{]}$, let $(K, \mathcal{C})$ be the output of $\text{Enc(pk, id)}$ and let $\text{usk\{id\}}$ be the output of $\text{USKGen(sk, id)}$. By Equation (4.2) in Section 4.1, we have

$$e([c_0][1], [v][2]) = (Ar)^\top (\sum_{i=0}^\ell f_i(id)Y_i^\top t + \sum_{i=0}^{\ell'} f_i(id)y'_i)$$

$$e([c_1][1], [t][2]) = (Ar)^\top (\sum_{i=0}^{\ell'} f_i(id)x'_i) \cdot t$$

and the quotient of the two elements yields $K = (\sum_{i=0}^{\ell'} f_i(id)x'_i) \cdot r$. 
Figure 4.11: Games $G_0$-$G_4$ for the proof of Theorem 4.2.1.

Theorem 4.2.1. If the $\mathcal{D}_k$-MDDH assumption is $(t, \varepsilon)$-hard in $G_1$, and MAC is $(t', \varepsilon', Q)$-PR-CMA-secure, then IBKEM[$\text{MAC, D}_k$] is a $(t', \varepsilon', Q)$-PR-ID-CPA-secure IBKEM, where $t \approx t' \approx t''$ and $\varepsilon'' \leq \varepsilon + \varepsilon'$.

Proof. We prove Theorem 4.2.1 by defining a sequence of games $G_0$-$G_4$ as in Figure 4.11. Let $A$ be an adversary against the PR-ID-CPA security of IBKEM[$\text{MAC, D}_k$].

Lemma 4.2.2. $\Pr[\text{PR-ID-CPA}^A_{\text{real}} \Rightarrow 1] = \Pr[\text{G}_A^\dagger \Rightarrow 1] = \Pr[\text{G}_0^\dagger \Rightarrow 1]$.

Proof. $G_0$ is the real attack game. In game $G_1$, we change the simulation of $c_1^*$ and $K^*$ in $\text{Enc(id)}^*$ by substituting $Z_i$ and $z_i'$ with their respective definitions:

$$c_1^* = \sum_{i=0}^{\ell} f_i(id^*) Z_i r = \sum_{i=0}^{\ell} f_i(id^*) (Y_i^T | x_i) A r = \sum_{i=0}^{\ell} f_i(id^*) (Y_i^T | x_i) c_0^*$$

and, similarly $K^* = \sum_{i=0}^{\ell} f_i(id^*) (y_i^T | x_i) A r = \sum_{i=0}^{\ell} f_i(id^*) (y_i^T | x_i) c_0^*$. Thus, $G_1$ is identical to $G_0$.

Lemma 4.2.3. There exists an adversary $B_1$ that $(t, \varepsilon)$-breaks the $\mathcal{D}_k$-MDDH assumption with $t \approx t''$ and $\varepsilon \geq | \Pr[\text{G}_A^\dagger \Rightarrow 1] - \Pr[\text{G}_0^\dagger \Rightarrow 1] |$.

Proof. The only difference between $G_2$ and $G_1$ is that $c_0^*$ is chosen uniformly at random over $Z_{p}$ k+1. It is easy to see that the joint distribution of $(G, A_1, [c_0^*])$ in $G_1$ is identical to the real $\mathcal{D}_k$-MDDH distribution and $(G, A_1, [c_0^*])$ in $G_2$ is identical to the random $\mathcal{D}_k$-MDDH distribution.

More formally, we build a distinguisher $B_1$. $B_1$ takes as input $(G, A_1, [b])$ and it has to distinguish if $b = Aw$ for some random vector $w \in Z_{p}$ k+1 or $b$ is uniformly random. $B_1$ simulates USKGen and finalize the same way as in $G_2$ and $G_1$. We only describe the simulation of INITIALIZE and ENC in Figure 4.12. Note that $B_1$ knows the secrets $x_i, x_i', y_i, y_i'$ explicitly over $Z_{p}$. Hence, $B_1$ can compute $(Z_i, [b], [a], [c_0^*])$ from $pk$ and $(c_0^*, [K^*], T)$ from the encryption query from $A_1$, and $[b]$. If $b = Aw$ for $w \in Z_{p}$ k+1, then the simulation is distributed as in $G_1$. If $b$ is uniformly random, then the simulation is distributed as in $G_2$.

Following the intuition of the construction, in Game $G_3$, we simulate the values $v$ computed in the USKGen algorithm using a “perfect zero-knowledge” simulator, and ENC is simulated without using $(Y_i)_{0 \leq i \leq \ell}$ and $(y_i')_{0 \leq i \leq \ell'}$, which is ready to conclude the proof by using the PR-CMA security of MAC.

Lemma 4.2.4. $\Pr[\text{G}_A^\dagger \Rightarrow 1] = \Pr[\text{G}_0^\dagger \Rightarrow 1]$.
can be carried out as in both 
\( \text{Enc} \) and 
\( \text{USKGen} \).

In this section, we will describe our transformation

4.3 Hierarchical IBE from Delegatable Affine MACs

\[ \text{Lemma 4.2.5.} \]

Description of Figure 4.12: \( B_1(G, [A], [b]) \) for the proof of Lemma 4.2.3

Proof. \( G_3 \) does not use \( (Y_i)_0 \leq i \leq t \) and \((y'_i)_{0 \leq i \leq t'}\) any more. We now show that the changes are purely conceptual. By \( Y_i = (Y_i^T \mid x_i)A \), we have \( Y_i^T = (Z_i - x_i \cdot A) \cdot (A)_{-1} \), and similarly we have \( y'_i = (z'_i - x'_i \cdot A) \cdot (A)_{-1} \).

For USKGen(id), by substituting \( Y_i^T \) and \( y'_i \), we obtain
\[
\begin{align*}
\mathbf{v}^T &= \left( t^T \sum f_i(\text{id})(Z_i - x_i \cdot A) + \sum f_i'(\text{id})(z'_i - x'_i \cdot A) \right) (A)^{-1} \\
&= \left( t^T \sum f_i(\text{id})Z_i + \sum f_i'(\text{id})z'_i - (t^T \sum f_i(\text{id})x_i + \sum f_i'(\text{id})x'_i) \cdot A \right) (A)^{-1}.
\end{align*}
\]

Note that we can compute \([v]_2 \in G_2\), since \( A, z' \) and \( Z_i \) are known explicitly over \( Z_p \) and \([t]_2 \) and \([u]_2 \) are known.

As for the distribution of \( \text{Enc}(\text{id}^*) \), it is easy to see that \( c_0^* \) is uniformly random, as in \( G_2 \). By \( h = c_0^* - A \cdot (A)^{-1} c_0^* \), we have
\[
\begin{align*}
c_i^* &= \sum f_i(\text{id}^*)(Z_i - (A)^{-1} c_0^* + x_i \cdot (c_0^* - A \cdot (A)^{-1} c_0^*)) \\
&= \sum f_i(\text{id}^*)(Y_i^T A + x_i A) \cdot (A)^{-1} c_0^* + x_i \cdot (c_0^* - A \cdot (A)^{-1} c_0^*)) \\
&= \sum f_i(\text{id}')(Y_i^T \mid x_i)c_0^*
\end{align*}
\]
and \( c_i^* \) is distributed as in \( G_2 \). The distribution of \( K^* \) can be analyzed with a similar argument. \( \square \)

\[ \text{Lemma 4.2.5.} \] There exists an adversary \( B_2 \) that \((t', z', Q)\)-breaks PR-CMA-security of MAC with \( t' \approx t'' \) and \( z' \geq |Pr[G^A_\text{A} \Rightarrow 1] - Pr[G^A_3 \Rightarrow 1]| \).

Proof. In \( G_4 \), we answer the \( \text{Enc}(\text{id}^*) \) query by choosing random \( K^* \) and \( C^* \). We construct an adversary \( B_2 \) in Figure 4.13 to show the differences between \( G_4 \) and \( G_3 \) can be bounded by the advantage of breaking PR-CMA-security of MAC. Intuitively, the reduction to PR-CMA-security of the symmetric primitive MAC can be carried out as in both \( G_3 \) and \( G_4 \), \( \text{skMAC} \) (i.e., \( x_i \) and \( x'_i \)) is perfectly hidden until \( B_2 \)'s call to \( \text{Enc}(\text{id}^*) \).

If \((h_0, h_1)\) is uniform (i.e., \( B_2 \) is in Game PR-CMA\_rand) then the view of \( A \) is the same as in \( G_4 \). If \((h_0, h_1)\) is real (i.e., \( B_2 \) is in Game PR-CMA\_real) then the view of \( A \) is the same as in \( G_3 \). \( \square \)

The proof of Theorem 1.2.1 follows by Lemmas 4.2.2, 4.2.5, and observing that \( G_4 \) = PR-ID-CPA\_rand.

4.3 Hierarchical IBE from Delegatable Affine MACs

In this section, we will describe our transformation \( \text{HIBE}[\text{MAC}, D_k] \) from delegatable affine MACs to HIBE based on any \( D_k \)-MDDH assumption. Moreover, we will also give an anonymity-preserving transformation \( \text{AHIBKEM}[\text{MAC}, D_k] \).

4.3.1 The Transformation

Let \( D_k \) be a matrix distribution that outputs matrices \( A \in Z_{2^m}^{(k+1) \times k} \). Let \( \text{MAC} \) be a delegatable affine MAC over \( Z_{2^m} \) with message space \( M = B^{\leq m} \). Our \( \text{HIBKEM}[\text{MAC}, D_k] = (\text{Gen}, \text{USKGen}, \text{USKDel}, \text{Enc}, \text{Mac} \)
Figure 4.13: Description of \( B_2 \) (having access to the oracles \textsc{InitializeMac}, \textsc{Eval}, \textsc{Chal}, \textsc{FinalizeMac} of the PR-\textsc{CMA\_real}/PR-\textsc{CMA\_rand} games of Figure 4.1) for the proof of Lemma 4.2.5.
In this section, we give an alternative (but less efficient) transformation, which is anonymity-preserving. Let $D_k$ be a matrix distribution that outputs matrices $A \in \mathbb{Z}_p^{(k+1) \times k}$. Let MAC be a delegable affine MAC over $\mathbb{Z}_p^m$ with message space $\mathcal{M} = B^m$. Our AHIBKEM[$\mathcal{MAC}, D_k$] = (Gen, USKGen, USKDel, Enc, Dec) for key-space $K = G_p$ and hierarchical identity space $\mathcal{T}D = M = B^{\leq m}$ is defined as in Figure 4.17. Compared to the HIBE construction from Section 4.3.1, the new construction uses a different usk rerandomization method: USKGen outputs a random basis $T$ for vector $t$ which allows
Theorem 4.3.6. If the $D_k$-MDDH assumption is $(t, \varepsilon)$-hard in $G_1$ and MAC is $(t', \varepsilon', Q)$-APR-CMA-secure, then AHIBKEM[$\text{MAC}, D_k$] is a $(t'', \varepsilon'', Q)$-PR-HID-CPA-secure IBKEM, where $t'' \approx t' \approx t''$ and $\varepsilon'' \leq \varepsilon + \varepsilon'$.

Proof. The proof of Theorem 4.3.6 is similar to that of Theorem 4.3.1. We define the sequence of games $G_0$-$G_4$ in Figure 4.15. Let $A$ be an adversary against the PR-HID-CPA security of AHIBKEM[$\text{MAC}, D_k$]. $G_0$ is the real attack game (PR-HID-CPA$_{\text{real}}$) and $Pr[G_0 \Rightarrow 1] = Pr[\text{PR-HID-CPA$_{\text{real}}$} \Rightarrow 1]$.

Analogously to Lemmas 4.3.2 and 4.3.3, we have

Lemma 4.3.7. $Pr[G_1 \Rightarrow 1] = Pr[G_0 \Rightarrow 1]$.

Lemma 4.3.8. There exists an adversary $B_1$ $(t, \varepsilon)$-breaks the $D_k$-MDDH assumption with $t \approx t''$ and $\varepsilon \geq |Pr[G_1 \Rightarrow 1] - Pr[G_0 \Rightarrow 1]|$.

Lemma 4.3.9. $Pr[G_0 \Rightarrow 1] = Pr[G_0 \Rightarrow 1]$.

Proof. $G_3$ is defined without using $y_0^\varepsilon$ and $(Y_i)_{0 \leq i \leq \ell}$. By Lemma 4.3.4, we have values $[v]_2, [e_i]_2, [E_i]_2, K^*$. 

Figure 4.15: Games $G_0$-$G_4$ for the proof of IND-CPA security (Theorem 4.3.1)}
The image contains a page from a document discussing cryptographic protocols and algorithms. The text is dense and formal, likely from a section on Hierarchical IBE from Delegatable Affine MACs. The text appears to be discussing specific equations and procedures for initializing, encrypting, and generating keys in a cryptographic context. The page includes mathematical notation and algorithms, typical of a technical or academic document. A specific section on the initialization of keys and the encryption process is highlighted, along with references to related protocols and security conditions.

The page also contains a figure (Figure 4.16) labeled as follows:

**Figure 4.16:** Description of \( B_2 \) (having access to the oracles \( \text{INITIALIZEMAC, \text{EVAL, \text{CHAL, FINALIZEMAC of the HPR-CMA}_{\text{id}}, HPR-CMA}_{\text{and}} } \) games of Figure 4.12) for the proof of Lemma 4.3.5.

The page further includes another figure (Figure 4.17), labeled as follows:

**Figure 4.17:** Definition of the transformation AHIBKEM[MAC, \( D_b \)]. \( \mu \) denotes the rank of \( B \).
and $C^*$ are identical in both $G_3$ and $G_2$. By $Y_i^T = (Z_i - x_i \mathbf{A})^{-1}$, we have

$$V = \sum f_i(id) (A^{-1})^T (Z_i^T - A^{-T} x_i^T) = (A^{-1})^T \left( \sum f_i(id) Z_i^T - A^{-T} \sum f_i(id) x_i^T \right)$$

$$E_i = (A^{-1})^T (Z_i^T - A^{-T} x_i^T) = (A^{-1})^T (Z_i^T - A^{-T} x_i^T).$$

Thus, $G_3$ is identical to $G_2$.

\[\square\]

**Lemma 4.3.10.** There exists an adversary $B_2$ $(t', \varepsilon', Q)$-breaks APR-CMA-security of MAC with $t' \approx t''$ and $\varepsilon' \approx |\Pr[\mathsf{G}_4] = 0] - \Pr[\mathsf{G}_3] = 1]$.}

**Proof.** In $G_4$, we answer the $\mathsf{ENC}(id^*)$ query by choosing random $K^*$ and $C^*$. We construct algorithm $B_2$ in Figure 4.19 to show the differences between $G_4$ and $G_3$ is bounded by the advantage of breaking APR-CMA security of MAC.

We note that, in both $G_3$ and $G_4$, the values $x_i$ and $x_0'$ are hidden until the call to $\mathsf{ENC}(id^*)$. It is easy to see $c_0^*$ is uniform, since $h$ and $r_0^*$ are chosen uniformly at random. If $(h_0, h_1)$ is uniform (i.e. $B_2$ is APR-CMA$_\mathsf{rand}$) then the view of $A$ is the same as in $G_4$. If $(h_0, h_1)$ is real (i.e. $B_2$ is in Game APR-CMA$_\mathsf{real}$) then the view of $A$ is the same as in $G_3$.

The proof follows by combining Lemmas 4.3.7, 4.3.10 and observing that $G_4 = \mathsf{PR-HID-CPA}_{\mathsf{rand}}$.

\[\square\]
4.4 Identity-based Hash Proof System

In this section, we will show that our IBE construction from Section 4.2 gives a secure identity-based hash proof system (ID-HPS) which implies IND-CCA secure and leakage-resilient IBE. Moreover, one of our constructions is the first tightly secure ID-HPS without a “Q-type” assumption in prime-order groups.

4.4.1 Definitions

We recall syntax and security of ID-HPS from [ADN+10].

**Definition 4.4.1** (Identity-based Hash Proof System). An identity-based hash proof system (ID-HPS) consists of five PPT algorithms $\text{IDHPS} = (\text{Setup}, \text{USKGen}, \text{Encap}, \text{Decap}^*, \text{Decap})$ with the following properties.

- The probabilistic key generation algorithm $\text{Setup}(\text{par})$ returns the (master) public/secret key $(pk, sk)$.
- We assume that $pk$ implicitly defines an identity space $\mathcal{ID}$, an encapsulated-key set $\mathcal{K}$.
- The probabilistic user secret key generation algorithm $\text{USKGen}(sk, id)$ returns the secret key $\text{usk}[id]$ for an identity id $\in \mathcal{ID}$.
- The probabilistic valid encryption algorithm $\text{Encap}(pk, id)$ returns a pair $(c, K)$ where $c$ is a valid ciphertext, and $K \in \mathcal{K}$ is the encapsulated-key with respect to identity id.
- The probabilistic invalid encryption algorithm $\text{Encap}^*(pk, id)$ samples an invalid ciphertext $c$.
- The deterministic decapsulation algorithm $\text{Decap}(\text{usk}[id], c)$ returns a decapsulated key $K$.

For perfect correctness we require that for all $\lambda \in \mathbb{N}$, all pairs $(pk, sk)$ generated by $\text{Setup}(1^\lambda)$, all identities $id \in \mathcal{ID}$, all $\text{usk}[id]$ generated by $\text{USKGen}(sk, id)$ and all $(c, K)$ output by $\text{Encap}(pk, id)$:

$$\Pr[\text{Decap}(\text{usk}[id], c) = K] = 1.$$  

The security requirements for an ID-HPS are valid/invalid ciphertext indistinguishability (VI-IND) and smoothness. VI-IND security is defined via the games $\text{VI-IND}_{\text{real}}$ and $\text{VI-IND}_{\text{rand}}$ in Figure 4.20. Note that VI-IND security game, the adversary is allowed to ask for $\text{usk}[id^*]$ (where $id^*$ is the challenge identity) and that $\text{USKGen}(id)$ returns the same answer when queried twice on the same identity.

**Definition 4.4.2** (VI-IND Security). An identity-based hash proof system IDHPS is $(t, \varepsilon)$-VI-IND-secure if for all PPT $A$ with running time $t$, $|\Pr[\text{VI-IND}_{\text{real}}^A] - 1| - |\Pr[\text{VI-IND}_{\text{rand}}^A] - 1| \leq \varepsilon$.

Smoothness is a statistical property saying that the decapsulated key $K$ for an invalid ciphertext is distributed statistically close to the uniform distribution.

**Definition 4.4.3** (Statistical Distance). The statistical distance between two random variables $X$ and $Y$ over a finite domain $\Omega$ is defined as:


**Figure 4.20:** Games VI-IND\textsubscript{adv} and VI-IND\textsubscript{adv} for defining valid/invalid ciphertext indistinguishability. We note that in game VI-IND\textsubscript{adv} the adversary is allowed to query USKGen with the challenge id\textsuperscript{*}.

\[
\text{SD}(X, Y) := \frac{1}{2} \sum_{w \in \Omega} |\Pr[X = w] - \Pr[Y = w]|.
\]

**Definition 4.4.4** (Smooth ID-HPS). An identity-based hash proof system IDHPS is smooth if for any fixed (pk, sk) produced by Setup(par), any id \( \in ID \),

\[
\text{SD}((C, K), (C, K')) \leq \text{negl}(\lambda),
\]

where \( C \overset{\$}{=} \text{Encap}\textsuperscript{*}(id), K \leftarrow \text{Decap}(C, \text{usk}[id]), K' \overset{\$}{=} K \) and \( \text{usk}[id] \overset{\$}{=} \text{USKGen}(\text{id}) \).

### 4.4.2 Construction

Let \( D_\lambda \) be a matrix distribution that outputs matrices \( A \in \mathbb{Z}_{p}^{(k+1) \times k} \). Let MAC be an affine MAC over \( \mathbb{Z}_{p}^{n} \) with message space \( ID \). In Figure 4.21, we describe the transformation IDHPS[MAC, \( D_\lambda \)]. We note that Setup, USKGen, Encap and Decap are the same as in IBKEM[MAC, \( D_\lambda \)] from Section 4.2 and Encap\textsuperscript{*} returns a random ciphertext \( C \) from the ciphertext space \( C = \mathbb{G}_{1}^{n+k+1} \). Correctness of IDHPS[MAC, \( D_\lambda \)] follows from the correctness of IBE[MAC, \( D_\lambda \)].

**Figure 4.21:** Definition of the ID-based HPS IDHPS[MAC, \( D_\lambda \)].

### 4.4.3 Security

We show that our ID-HPS is both smooth and VI-IND secure.
Table 4.4.5. An affine MAC over $\mathbb{Z}_p^n$ is $(t, e, Q)$-SPR-CMA-secure if for all PPT $A$ with running time $t$ and $\Pr[\text{SPR-CMA}_A \Rightarrow 1] - \Pr[\text{SPR-CMA}_A^\text{rand} \Rightarrow 1] \leq \varepsilon$.

We note that both MAC$_{\text{MP}}[D_K]$ and MAC$_{\text{CPS}}[D_K]$ are SPR-CMA secure. The security proof from Section 4.4 can be easily adapted to show both schemes are SPR-CMA secure. Here we just outline the ideas. For MAC$_{\text{MP}}[D_K]$, we first use the $Q$-fold $D_K$-MDDH assumption to make the answers all EVAL queries random; next, we store a list of $(h_{0[i]}, h_{1[i]})$ values to make the output of CHAL$(m^*)$ random and consistent with EVAL$(m^*)$. One can also adapt the proof of MAC$_{\text{CPS}}[D_K]$ in a similar way.

The following theorem shows VI-IND security of IDPSPH[MAC, $D_K$].

Theorem 4.4.6. If the $D_K$-MDDH assumption is $(t, e)$-hard in $G_1$ and MAC is $(t', e', Q)$-SPR-CMA-secure, then IDPSPH[MAC, $D_K$] is $(t'', e'', Q)$-VI-IND-secure IDPSPH, where $t'' = t'$ and $e'' = \varepsilon + e'$.

Proof. The proof is similar to the one of Theorems 4.2.1 except that we need to simulate the user secret key for the challenge identity id*. The games are defined as in Figure 4.23.

The following three lemmas and their proofs are exactly the same as Lemmas 4.2.2 to 4.2.4.

Figure 4.22: Games SPR-CMA$_{\text{real}}$ and SPR-CMA$_{\text{rand}}$ for defining SPR-CMA security.

Figure 4.23: Games $G_0$-$G_4$ for the proof of the VI-IND security.
Lemma 4.4.7. \( \Pr[\text{VI-IND}_{\text{real}}^A \Rightarrow 1] = \Pr[G^A \Rightarrow 1] = \Pr[G^A \Rightarrow 1]. \)

Lemma 4.4.8. There exists an adversary \( B_1(t, \varepsilon) \)-breaks the \( D_k \)-MDDH assumption with \( t \approx t' \) and \( \varepsilon \geq |\Pr[G^A \Rightarrow 1] - \Pr[G^A \Rightarrow 1]|. \)

Lemma 4.4.9. \( \Pr[G^A \Rightarrow 1] = \Pr[G^A \Rightarrow 1]. \)

Lemma 4.4.10. There exists an adversary \( B_2 \) that \( (t', \varepsilon', Q) \)-breaks \( \text{SPR-CMA}_{\text{real}} \) with \( t' \approx t'' \) and \( \varepsilon' \geq |\Pr[G^A \Rightarrow 1] - \Pr[G^A \Rightarrow 1]|. \)

Proof. We construct an adversary \( B_2 \) in Figure 4.23 to show that the difference between \( G_4 \) and \( G_3 \) is bounded by the advantage of breaking \( \text{SPR-CMA}_{\text{real}} \) security of MAC.

By the definition of \( \text{SPR-CMA} \), if \( B_2 \) is in Game \( \text{SPR-CMA}_{\text{real}} \) then the view of \( A \) is the same as in \( G_4 \); and if \( B_2 \) is in Game \( \text{SPR-CMA}_{\text{rand}} \) then the view of \( A \) is the same as in \( G_3 \). We observe that in Game \( G_4 \) \( c_1 \) is masked by the value \( h_0[\text{id}'] \), which is uniformly random. The reason is that \( h_0[\text{id}'] \) is hidden from \( \text{USKGen}(\text{id}') \) query, since it is masked by a random \( h_1[\text{id}'] \). Thus, \( G_4 = \text{VI-IND}_{\text{rand}}. \)

Theorem 4.4.11. \( \text{IDHPS}[\text{MAC}, D_k] \) is smooth.

Proof. We show that for almost all \((c_0, c_1) \in \mathbb{Z}_q^{n+k+1},
K = c_0 \left( \frac{v}{u} \right) - c_1 T \) is uniformly random, where \((t_2, u_2, v_2) \notin \text{USKGen}(\text{id}) \). Similar to game \( G_3 \) of VI-IND security proof, one can rewrite:

\[
K = c_0 \left( \frac{v}{u} \right) - c_1 T = c_0 \left( (t^T \sum_{i=0}^{n+k+1} f_i(id)Z_i + \sum_{i=0}^{n+k+1} f_i(id)z_i - u \cdot A) \cdot \overline{A}^{-1} \right) - c_1 T
\]

\[
= c_0 \left( \overline{A}^{-1} \right)^T \sum_{i=0}^{n+k+1} f_i(id)z_i \left( \overline{A}^{-1} \right)^T
+ \sum_{i=0}^{n+k+1} f_i(id)z_i \left( \overline{A}^{-1} \right)^T - c_1 T
\]

Since \((c_0, c_1) \) from \( \text{Encap}^* \) is chosen uniformly at random, \( c_1 \neq \left( \overline{A}^{-1} \right)^T \sum_{i=0}^{n+k+1} f_i(id)z_i \)

with probability \( 1 - 1/q^n \). Conditioned on that and since \( \text{rank}(B) \geq 1 \) and \( t = Bs \) for \( s \in \mathbb{Z}_q^n \), \( E \) is uniformly random. This concludes the theorem. \( \square \)
4.5 Concrete Instantiation from SXDH

We now describe our tightly PR-ID-CPA-secure IBKEM $\text{IBKEM}_{\text{MACNRT}}[\mathcal{D}_k, \mathcal{U}_1]$ for the special case of $k = 1$ and $\mathcal{D}_k = \mathcal{U}_1$, such that $\mathcal{D}_k$-MDDH is the DDH assumption. The identity space is $\mathcal{I} = \{0, 1\}^t$.

Theorems 4.1.6 and 4.2.1 provide a tight security reduction under the DDH assumptions in $\mathbb{G}_1$ and $\mathbb{G}_2$, i.e., under the SXDH assumption.

<table>
<thead>
<tr>
<th>Gen(par):</th>
<th>Enc(pk, id):</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{sk}<em>{\text{MAC}} = (x_0, \ldots, x</em>\ell, x') \in \mathbb{Z}_q^{t+2}$</td>
<td>$r \in \mathbb{Z}_p$</td>
</tr>
<tr>
<td>$z_i \in \mathbb{Z}_p$</td>
<td>$c_0 := (c_{0,0}, c_{0,1}) = (r, a \cdot r)$</td>
</tr>
<tr>
<td>$y_i \in \mathbb{Z}_q$</td>
<td>$c_1 = r(z_0 + \sum_{i=1}^{\ell} \text{id}_i z_i)$</td>
</tr>
<tr>
<td>$y' \in \mathbb{Z}_q$</td>
<td>$K = z' \cdot r$</td>
</tr>
<tr>
<td>$pk := (G, [a], ([z_i]<em>1)</em>{0 \leq i \leq \ell}, [z'_1])$</td>
<td>Return $K = [K]_T$ and $C = ([c_0]_1, [c_1]_1) \in \mathbb{G}_1^2$.</td>
</tr>
<tr>
<td>$sk := (\text{sk}<em>{\text{MAC}}, (y_i)</em>{0 \leq i \leq \ell}, y')$</td>
<td>Dec(usk[id], id, C),</td>
</tr>
<tr>
<td>Return $(pk, sk)$.</td>
<td>Parse $\text{usk}[id] = ([t]_2, [u]_2, [v]_2)$</td>
</tr>
<tr>
<td>Parse sk[id] := $([t]_2, [u]_2, [v]_2) \in \mathbb{G}_2^3$</td>
<td>Parse $C = ([c_0]_1, [c_1]_1)$</td>
</tr>
<tr>
<td>USKGen(sk, id):</td>
<td>$K = e([c_0]_1, [v]_2) \cdot e([c_0,0]_1, [u]_2) / e([c_1]_1, [t]_2)$</td>
</tr>
<tr>
<td>$t \in \mathbb{Z}_q$</td>
<td>Return $K \in \mathbb{G}_1$.</td>
</tr>
<tr>
<td>$u = x' + t(x_0 + \sum_{i=1}^{\ell} \text{id}_i \cdot x_i)$</td>
<td></td>
</tr>
<tr>
<td>$v = y' + t(y_0 + \sum_{i=1}^{\ell} \text{id}_i \cdot y_i)$</td>
<td></td>
</tr>
<tr>
<td>Return usk[id] := $([t]_2, [u]_2, [v]_2) \in \mathbb{G}_2^3$.</td>
<td></td>
</tr>
</tbody>
</table>

$\text{IBKEM}_{\text{MACNRT}}[\mathcal{D}_k, \mathcal{U}_1]$ is a "Cramer-Shoup variant" of Waters’ IBKEM $\text{Wat05}$ [Wat05]. Concretely, $\text{Wat05}$ is a projected variant of our scheme and is obtained by setting $\text{usk}[id] := ([t]_2, [u + av]_2) \in \mathbb{G}_2^3$ and $C = ([c_0 = c_{0,0} + ac_{0,1}, [c_1]) \in \mathbb{G}_2^2$. $\text{Wat05}$ is IND-ID-CPA-secure under the DBDH assumption, with a non-tight security proofs. See [BR09] for a discussion on the impact of the non-tight assumption. Our IBKEM is tightly IND-ID-CPA-secure and anonymous under the SXDH assumption.
CHAPTER 5

MORE EFFICIENT
STRUCTURE-PRESERVING SIGNATURES

Structural-preserving signatures are signature schemes (cf. Definition 2.3.1) which sign group elements using only group operations:

**Definition 5.0.1** (Structure-preserving signature). A digital signature scheme $SPS$ is structure-preserving if the public key space $\mathcal{PK} := \mathbb{G}^n$, message space $\mathcal{M} := \mathbb{G}^n$, signature space $\mathcal{\Sigma} := \mathbb{G}^n$ and the verification algorithm only consists of pairing product equations and returns 1 (accept) or 0 (reject).

### 5.1 Motivating Example: One-time Secure SPS

We call a signature scheme $SPS$ one-time secure if $SPS$ is $(t, \varepsilon, 1)$-UF-CMA secure (cf. Definition 2.3.3). As a motivating example, we construct an one-time secure SPS scheme is given in Figure 5.1 and its parameters are:

| $|pk|$ | $(n + 1)k + \text{RE}(D_k)$ |
| $|\sigma|$ | $k + 1$ |

As defined in Section 2.2.1, $\text{RE}(D_k)$ denotes the number of group elements needed to represent $[A]_2$, where $A \leftarrow D_k$. For $k$-Lin, we achieve 2 group elements in the signature for $k = 1$ and 3 group elements for $k = 2$. Moreover, we note that the verification needs $k$ pairing product equations: for $e(\sigma, [A]_2) = e([1, m]^\top, [C]_2)$ we need to pair the vector $\sigma$ with every column of $[A]_2$ and thus this check needs $k$ pairing product equations.

**Figure 5.1:** $(t, \varepsilon, 1)$-UF-CMA-secure structure-preserving signature $SPS_{\text{ot}}$ with message-space $\mathcal{M} = \mathbb{G}_1^n$.

We will exploit the following lemma in the analysis of our scheme. Informally, the lemma says that $m \mapsto (1, m^\top)K$ is a secure information-theoretic one-time MAC even if the adversary first sees $(A, KA)$.

**Lemma 5.1.1** (Core lemma for one-time security). Let $n, k$ be integers. For any $A \in \mathbb{Z}_p^{(k+1)\times k}$ and any (possibly unbounded) adversary $A$, \[
\Pr\left[m^* \neq m, z^T = (1, m^{*T})K, K \leftarrow \mathbb{Z}_p^{(n+1)\times (k+1)} \left( z, m^* \right) \leftarrow \mathcal{O}(\cdot)(KA) \right] \leq \frac{1}{p^2}, \tag{5.1}
\]

where $\mathcal{O}(m \in \mathbb{Z}_p^n)$ returns $(1, m^\top)K$ and $A$ only gets a single call to $\mathcal{O}$.
This lemma can be seen as an adaptive version of a special case of [KW15, Lemma 2] in that we fix \( t = 1 \), \( M \) to be the matrix \((1, m^\top) \in \mathbb{Z}_p^{(n+1) \times 1}\), and we use the fact that \( \sigma = (1, m^\top)K \) is a pair-wise independent hash. In particular, if \( m^* \neq m \), then \((1, m^\top)\notin \text{span}(M)\). In our adaptive version, \( m \) may depend on \( KA \) but the proof is essentially the same as in [KW15].

**Proof.** First, fix any \( A \in \mathbb{Z}_p^{(k+1) \times k} \) and any pair of distinct \( m, m^* \in \mathbb{Z}_p^n \), along with a non-zero vector \( \mathbf{a} \notin \text{span}(A) \). Observe that the following distributions

\[
((1, m^\top)K, KA, (1, m^*\top)K\mathbf{a}) \quad \text{and} \quad ((1, m^\top)K, KA, u)
\]

(5.2)

are the same, where \( K \not\equiv Z_p^{(n+1) \times (k+1)} \), \( u \not\equiv Z_p \). Here, we use the fact that if \( m \neq m^* \), then \((1, m^\top)\) and \((1, m^\top)\) are linearly independent. By a standard argument (e.g. complexity leveraging\(^1\)), this means that the two distributions are the same even if \( m, m^* \) are adaptively chosen, that is, seeing \( KA \) for \( m^* \), after seeing \((KA, (1, m^\top)K)\) for \( m^* \). Therefore, for any adversary \( A \), we have

\[
\Pr[m^* \neq m \land z^\top \mathbf{a} = (1, m^*\top)K\mathbf{a} \mid K \not\equiv Z_p^{(n+1) \times (k+1)}, (z, m^*) \not\equiv A^{(t)}(KA)] \leq \frac{1}{p},
\]

since \((1, m^*\top)K\mathbf{a}\) is uniformly random from the adversary’s viewpoint. The lemma then follows from the fact that \( z^\top = (1, m^\top)K \) implies \( z^\top\mathbf{a} = (1, m^*\top)K\mathbf{a} \).

\(\square\)

**Theorem 5.1.2.** If the \( D_k\)-KerMDH problem is \((t, \varepsilon)\)-hard in group \( G_2 \), then \( SPS_{ot} \) from Figure 5.1 is \((t', \varepsilon', 1)\)-UF-CMA secure, where \( t \approx t' \) and \( \varepsilon' \leq \varepsilon + \frac{1}{p} \).

**Proof.** Perfect correctness and the structure-preserving property are straight-forward. We proceed to establish one-time UF-CMA-security based on the \( D_k\)-KerMDH assumption. We will show that for all adversaries \( A \), there exists an adversary \( B \) with \( t \approx t' \) and

\[
\varepsilon' \leq \varepsilon + \frac{1}{p}.
\]

We construct a simple reduction \( B(PG, [A]_2 \in G_2^{(k+1) \times k}) \) as follows: \( B \) generates \( \text{pk} = ([C]_2, [A]_2) \) as in the real scheme by picking \( K \in Z_p^{(n+1) \times (k+1)} \) and computing \( C := KA \). Next, \( B \) runs \( A \) on \( \text{pk} \), simulates a signature on \([m]_1 \) honestly using \( K \), and obtains \((m^*), (\sigma^*) \) satisfying \( m^* \neq m \) and \( e(\sigma^*, [A]_2) = e(1, m^*)_1 \) with probability \( \varepsilon' \). Finally, \( B \) returns \([s]_1 \) computed as

\[
[s]_1 = \sigma^* - [(1, m^*\top)]_1 K.
\]

Clearly, \( s \cdot A = 0 \) and \( \Pr[s = 0] \leq 1/p \) by Lemma 5.1.1. This proves equation (5.3).

\(\square\)

### 5.2 Main Construction: Fully UF-CMA-Secure SPS

In the section, we use the MACSPS[\( D_k \)] from Section 4.1.3 to transfer the one-time secure SPS, \( SPS_{ot} \), to a fully secure SPS, \( SPS_{full} \) in Figure 5.3.

#### 5.2.1 Computational Core Lemma

We present a variant of the computational core lemma from [KW15, Lemma 3], which shows that \( t^\top (K_0 + rK_1) \) hides \( \mathbf{a}^\top \) even if \(([B]_1, K_0A, K_1A, A) \) is given.

**Lemma 5.2.1 (Computational core lemma).** If the \( D_k\)-MDH problem is \((t, \varepsilon)\)-hard in group \( G_1 \) then there exists an adversary \( A \) can \((t', \varepsilon', Q)\)-distinguish between games \( \text{Core}_{\text{Real}} \) and \( \text{Core}_{\text{Ker}} \) are defined as in Figure 5.2 where

\[
\varepsilon' := \Pr[\text{Core}_{\text{Real}} \Rightarrow 1] - \Pr[\text{Core}_{\text{Ker}} \Rightarrow 1] \leq 2Q \cdot \varepsilon + Q/p, \quad t' \approx t.
\]

Compared to [KW15, Lemma 3], oracle \( \text{CHAL} \) is modified as follows. Instead of getting tag \( \tau^* \) and returning \( K_0 + \tau^*K_1 \) in the clear, both the query and the output are encoded in \( G_2 \). It is straight-forward to check that the proof goes through as in [KW15].

\(^1\)Using complexity leveraging, we can transform any adaptive distinguisher into a non-adaptive one with an exponential loss in the distinguishing advantage. If the optimal non-adaptive distinguishing advantage is 0 as is the case for two identical distributions, then the optimal adaptive distinguishing advantage must also be 0.
\begin{align*}
\text{INITIALIZE:} & \quad A, B \leftarrow D_k \\
& \quad K_0, K_1 \leftarrow Z_p^{(k+1)\times(k+1)} \\
& \quad (P_0, P_1) := (B^T K_0, B^T K_1) \\
& \quad \text{pk} := ([B], [P_0], [P_1], K_0, K_1, A, A) \\
& \quad \text{Return pk} \\
& \quad \text{CHAL}([\tau^*]_2): \quad /\text{one query} \\
& \quad \text{Return} \ [K_0 + \tau^* K_1]_2 \\
\text{EVAL}(\tau): \quad /\text{Q queries} \\
& \quad Q_{\text{tag}} := \{\tau\} \cup Q_{\text{tag}} \\
& \quad \mu \leftarrow Z_p; \ r \leftarrow \mathbb{Z}_p^k; \ t := Br \\
& \quad u := t^T (K_0 + \tau K_1) \\
& \quad \text{Return} \ ([t], [u]) \in G_2^{(k+1)\times(k+1)} \\
\text{FINALIZE}(d \in \{0, 1\})): \quad /\text{one query} \\
& \quad \text{Return} \ d \land (\tau^* \not\in Q_{\text{tag}})
\end{align*}

\textbf{Figure 5.2:} Games Core$_{\text{Real}}$ and Core$_{\text{Ker}}$ where $a^* \in \mathbb{Z}_p^{2\times(k+1)}$ is a non-zero vector in the kernel of $A$ such that $a^* A = 0$

- the simulator knows $K_0, K_1$, and therefore it can compute $[K_0 + \tau^* K_1]_2$ given $[\tau^*]_2$;
- the quantity $[K_0 + \tau^* K_1]_2$ does not reveal any additional information about $K_0, K_1$ beyond $K_0 + \tau^* K_1$.

For completeness, we reproduce the proof of [KW15] Lemma 3 as follows.

\begin{align*}
\text{INITIALIZE:} & \quad A, B \leftarrow D_k \\
& \quad K_0, K_1 \leftarrow Z_p^{(k+1)\times(k+1)} \\
& \quad (P_0, P_1) := (B^T K_0, B^T K_1) \\
& \quad \text{pk} := ([B], [P_0], [P_1], K_0, K_1, A, A) \\
& \quad \text{Return pk} \\
& \quad \text{CHAL}([\tau^*]_2): \quad /\text{one query} \\
& \quad \text{Return} \ [K_0 + \tau^* K_1]_2 \\
& \quad \text{EVAL}(\tau): \quad /\text{Game G}_0 \\
& \quad Q_{\text{tag}} := \{\tau\} \cup Q_{\text{tag}} \\
& \quad \mu \leftarrow Z_p; \ r \leftarrow \mathbb{Z}_p^k; \ t := Br \\
& \quad u := t^T (K_0 + \tau K_1) \\
& \quad \text{Return} \ ([t], [u]) \\
& \quad \text{FINALIZE}(d \in \{0, 1\})): \quad /\text{one query} \\
& \quad \text{Return} \ d \land (\tau^* \not\in Q_{\text{tag}})
\end{align*}

\textbf{Figure 5.3:} Games $G_0$, $(G_{1,i}, G_{1,i+1}, G_{1,2,i}, G_{1,3,i})_{1 \leq i \leq Q}$ and $G_{1,Q+1}$ for the proof of Lemma 5.2.1 where $a^* \in \mathbb{Z}_p^{2\times(k+1)}$ is a non-zero vector in the kernel of $A$ such that $a^* A = 0$

\textbf{Proof.} We prove Lemma 5.2.1 by defining a sequence of intermediate games as in Figure 5.3. Let $A$ be a distinguisher between games Core$_{\text{Real}}$ and Core$_{\text{Ker}}$. Game $G_0 := \text{Core}_{\text{Real}}$ is the real attack game. In games $G_{1,i}$, the first $i - 1$ queries to the EVAL oracle are answered with $u := \mu a^* + t^T (K_0 + \tau K_1)$ and the rest are answered as in the real scheme. To interpolate between $G_{1,i}$ and $G_{1,i+1}$, we also define $G_{1,i+1}$. By definition, we have $G_0 = G_{1,1}$.

\textbf{Lemma 5.2.2.} $\Pr[\text{Core}_{\text{Real}}^A \Rightarrow 1] = \Pr[G_0^A \Rightarrow 1] = \Pr[G_{1,1}^A \Rightarrow 1]$.

\textbf{Lemma 5.2.3.} \textit{There exists an adversary $B_1$ $(t, \varepsilon)$-breaks $\mathcal{D}_k$-MDDH problem in group $G_1$, where

\[ \varepsilon \geq |\Pr[G_{1,i}^A \Rightarrow 1] - \Pr[G_{1,i+1}^A \Rightarrow 1]|, \ t \approx t'. \]

\textbf{Proof.} Games $G_{1,i}$ and $G_{1,i}$ only differ in the distribution of $t$ returned by EVAL for its $i$-th query, namely, $t \in \text{span}(B)$ or uniform. From that, we obtain a straightforward reduction to break the $\mathcal{D}_k$-MDDH problem with $(t, \varepsilon)$ as stated in the lemma.}

\textbf{Lemma 5.2.4.} $|\Pr[G_{1,2,i}^A \Rightarrow 1] - \Pr[G_{1,1,i}^A \Rightarrow 1]| \leq \frac{1}{p}$.
**Proof.** Games $G_{1,2,i}$ and $G_{1,1,i}$ only differ in the distribution of $u$ in the $i$-th EVAL query. We apply an information-theoretic argument to bound this difference. For $\tau \neq \tau^*$, consider the information of $(K_0, K_i)$ leaked from INITIALIZE, the $i$-th EVAL query, $j$-th EVAL queries ($j \neq i$) and CHAL query:

\[
\begin{align*}
(J_{\mathrm{EVAL}})_{G_{1,2,i}} & = (pk, \mu a^T + t^T (K_0 + \tau K_1), B^T K_0, B^T K_1, K_0 + \tau^* K_1) \\
(J_{\mathrm{EVAL}})_{G_{1,1,i}} & = (pk, t^T (K_0 + \tau K_1), B^T K_0, B^T K_1, K_0 + \tau^* K_1)
\end{align*}
\]

where $t \not\in \text{span}(B)$ with probability $1 - 1/p$. We show these two distributions are identical except with probability $1/p$. Equivalently, we eliminate the terms involving $K_0 + \tau^* K_1$ and show the following two distributions are the same:

\[
(\mu a^T + (\tau - \tau^*) t^T K_1, K_1 A, B^T K_1),
\]

where $K_1 \overset{\text{R}}{\leftarrow} \mathbb{Z}_p^{(k+1)\times(k+1)}$. To establish this statement, we define $K_1 := K + \mu' b^L a^T$, where $K' \overset{\text{R}}{\leftarrow} \mathbb{Z}_p^{(k+1)\times(k+1)}$, $\mu' \overset{\text{R}}{\leftarrow} \mathbb{Z}_p$ and $b^T a^T \neq 0$ satisfies $b^T B = 0$. Observe that $(K_1 A, B^T K_1) = (K' A, B^T K')$ and $b^T a^T \neq 0$ with probability $1 - 1/p$. Thus, the following distributions are the same:

\[
(\mu a^T + (\tau - \tau^*) \mu' t^T b^L a^T) and ((\tau - \tau^*) \mu' t^T b^T a^T),
\]

where $\mu' \overset{\text{R}}{\leftarrow} \mathbb{Z}_p$. That implies games $G_{1,2,i}$ and $G_{1,1,i}$ are distributed the same, except with $1/p$.

**Lemma 5.2.5.** There exists an adversary $B_1 (t, \varepsilon)$-breaks $D_k$-MDDH problem in group $G_1$, where

\[
\varepsilon \geq |\Pr[G_{1,1,i}^A \Rightarrow 1] - \Pr[G_{1,2,i}^A \Rightarrow 1]|, \quad t \approx t'.
\]

**Proof.** Similar to **Lemma 5.2.3**, games $G_{1,3,i}$ and $G_{1,2,i}$ only differ in the distribution of $t$ returned by EVAL for its $i$-th query, namely, $t \in \text{span}(B)$ (in $G_{1,3,i}$) or $t \overset{\text{R}}{\leftarrow} \mathbb{Z}_p^{k+1}$ (in $G_{1,2,i}$). From that, we obtain a straightforward reduction to break the $D_k$-MDDH problem with $(t, \varepsilon)$ as stated in the lemma.

By observing that $G_{1,3,i} = G_{1,i+1}$ and $G_{1,Q+1} = \text{Core}_{\text{ker}}$, we have the following two lemmas:

**Lemma 5.2.6.** $\Pr[G_{1,3,i}^A \Rightarrow 1] = \Pr[G_{1,i+1}^A \Rightarrow 1]$.  

**Lemma 5.2.7.** $\Pr[G_{1,3,i}^A \Rightarrow 1] = \Pr[\text{Core}_{\text{ker}}^A \Rightarrow 1]$.

To sum up,

\[
\begin{align*}
\varepsilon' & := |\Pr[\text{Core}_{\text{real}}^A \Rightarrow 1] - \Pr[\text{Core}_{\text{ker}}^A \Rightarrow 1]| \\
& = |\Pr[G_{1,i}^A \Rightarrow 1] - \Pr[G_{1,Q+1}^A \Rightarrow 1]| \quad (\text{by Lemmas 5.2.2 and 5.2.7}) \\
& = |\Pr[G_{1,i}^A \Rightarrow 1] - \Pr[G_{1,Q+1}^A \Rightarrow 1]| \\
& = \sum_{i=1}^Q |\Pr[G_{1,i}^A \Rightarrow 1] - \Pr[G_{1,i}^A \Rightarrow 1]| \\
& \leq Q(2\varepsilon + \frac{1}{p}) \quad (\text{by Lemmas 5.2.3 to 5.2.6}),
\end{align*}
\]

which concludes **Lemma 5.2.1**.

**5.2.2 Our Scheme**

The parameters are:

\[
|pk| = (n + 1)k + 2(k + 1)k + \text{RE}(D_k), \quad |\sigma| = (3(k + 1), 1),
\]

where notation $(x, y)$ means $x$ elements in $G_1$ and $y$ elements in $G_2$. For $k$-Lin, this yields $(n + 6, (6, 1))$ for $k = 1$ and $(2n + 16, (9, 1))$ for $k = 2$. Moreover, we note that the verification needs $2k + 1$ pairing product equations: for $e(\sigma_1, [A]_2) = e(([1, m]_1, [C]_2) + e(\sigma_2, [C]_2) + e(\sigma_3, [C]_2)$ we need to pair the vector $\sigma_1$ with every column of $[A]_2$ and thus this check needs $k$ pairing product equations; and for $e(\sigma_2, [\tau]_2) = e(\sigma_3, [1]_2)$ we need to pair every element from $\sigma_2$ with $[\tau]_2 \in G_2$ and thus this requires $k + 1$ pairing product equations.
Lemma 5.2.9.

Figure 5.4: Structure-preserving signature $SPS_{\text{full}}$ with message-space $M = G_1^\tau$.

Theorem 5.2.8. If the $D_k$-KerMDH problem is $(t, \varepsilon)$-hard in $G_2$ and $D_k$-MDDH problem is $(t', \varepsilon')$-hard in $G_1$, $SPS_{\text{full}}$ from Figure 5.4 is an $(t'', \varepsilon'', Q)$-UF-CMA-secure structure-preserving signature scheme, where

\[ \varepsilon'' \leq \varepsilon + 2Q(Q+1)\varepsilon' + \frac{(Q+1)^2}{p} + \frac{Q^2}{2p}, \quad t \approx t''. \]

Figure 5.5: Games $G_0$-$G_5$ for the proof of Theorem 5.2.8, where $a^+ \in \mathbb{Z}_p^{(k+1)}$ is a non-zero vector in the kernel of $A$ such that $a^+A = 0$. Boxed code is only executed in the games marked in the same box style at the top right of every procedure and non-boxed code is always run.

Proof. Perfect correctness and the structure-preserving property are straightforward to verify. We prove Theorem 5.2.8 by defining a sequence of games as in Figure 5.5. Let $A$ be an adversary against the $(t'', \varepsilon'', Q)$-UF-CMA security of $SPS_{\text{full}}$.

By the definition of UF-CMA security (cf. Definition 2.3.3), we have the following lemma.

Lemma 5.2.9. $\varepsilon'' = Pr[G_0^A \Rightarrow 1]$. 

**Lemma 5.2.10.** There exists an adversary $B_1(t, \epsilon)$-breaks $D_k$-KerMDH problem in group $G_2$, where

$$\epsilon \geq |\Pr[G^A_1 \Rightarrow 1] - \Pr[G^A_0 \Rightarrow 1]|, \quad t \approx t'$$

**Proof.** Suppose $e(\sigma^*_1, [\tau^*]_2) = e(\sigma^*_1, [1]_2)$. We note that

$$e(\sigma^*_1, [A]_2) = e([(1, m^*)], [C]_2) + e(\sigma^*_2, [C_0]_2) + e(\sigma^*_3, [C_1]_2)$$

\[\iff\]

$$e(\sigma^*_1, [A]_2) = e([(1, m^*)], [KA]_2) + e(\sigma^*_2, [K_0A]_2) + e(\sigma^*_3, [K_1A]_2)$$

\[\iff\]

$$e(\sigma^*_1, [1]_2) = e([(1, m^*)], [K_2]_2) + e(\sigma^*_2, [K_0]_2) + e(\sigma^*_3, [K_1]_2)$$

\[\iff\]

$$e(\sigma^*_1, [1]_2) = e([(1, m^*)], [K_2]_2) + e(\sigma^*_2, [K_0 + \tau^*K_1]_2).$$

For any $([m^*_1], \sigma^*)$ accepted by $G_0$ but not by $G_1$, the value

$$\sigma^*_1 - ([1, m^*]K_1] + \sigma^*_2 K_0 + \sigma^*_3 K_1) \in G_1^{1 \times (k+1)}$$

is a non-zero vector in the kernel of $A$, which is hard to be computed under the $D_k$-KerMDH assumption in $G_2$. This means that

$$|\Pr[G^A_1 \Rightarrow 1] - \Pr[G^A_0 \Rightarrow 1]| \leq \epsilon.$$

\[\square\]

**Lemma 5.2.11.** $\Pr[G^A_0 \Rightarrow 1] \geq \Pr[G^A_1 \Rightarrow 1] - \frac{Q^2}{2p}$.

**Proof.** The only difference between $G_2$ and $G_1$ is that $\tau_1, \ldots, \tau_Q$ are all distinct in $G_2$. Since all the $\tau_i$'s are chosen uniformly at random from $Z_p$, by the birthday paradox, we have

$$|\Pr[G^A_0 \Rightarrow 1] - \Pr[G^A_1 \Rightarrow 1]| \leq \frac{Q^2}{2p}.$$

\[\square\]

$G_3$ guesses $\tau^* = \tau_{j^*} (i^* \notin [Q + 1]$ and $\tau_{Q+1} := \tau^*$), otherwise $G_3$ aborts. It is easy to see the following lemma:

**Lemma 5.2.12.** $\Pr[G^A_0 \Rightarrow 1] \geq \frac{1}{Q+1} \Pr[G^A_1 \Rightarrow 1]$.

**Lemma 5.2.13.** There exists an adversary $B_2(t', \epsilon')$-breaks the $D_k$-MDDH problem in group $G_1$, where

$$\epsilon' \geq \frac{1}{2Q} |\Pr[G^A_0 \Rightarrow 1] - \Pr[G^A_1 \Rightarrow 1]| - \frac{Q}{p}, \quad t' \approx t''$$

**Proof.** Instead of giving a direct construction of $B_2$, we construct an adversary $B_2$ in [Figure 5.6] to show that the difference between $G_1$ and $G_3$ is bounded by the advantage of breaking $\text{Lemma 5.2.1}$. which implies an adversary $B_2(t', \epsilon')$-breaks the $D_k$-MDDH problem in $G_1$ with probability and running time as stated. In the construction of $B_2$, if $u = \mu a^+ + t'(K_0 + \tau_i K_1)$ (i.e. $B$ is in Game Core$_{\text{Real}}$) then the view of $A$ is the same as in $G_3$. If $u = \mu a^+ + t'(K_0 + \tau_i K_1)$ (i.e. $B$ is in Game Core$_{\text{Adv}}$) then the view of $A$ is the same as in $G_4$.

\[\square\]

**Lemma 5.2.14.** $\Pr[G^A_1 \Rightarrow 1] = \Pr[G^A_0 \Rightarrow 1] \leq 1/p$.

**Proof.** We first show that games $G_5$ and $G_4$ are identical by observing $K$ is uniformly at random, as $K' \notin 2_p^{(n+1) \times (k+1)}$.

To conclude the proof, we bound the adversarial advantage in $G_5$ via an information-theoretic argument. We consider the information about $v$ leaked from $pk$ and signing queries:

- From $pk$, $C = (K' + va^+)A = K'A$ completely hides $v$;
- From $\text{Stg}(m_i)$ with $i \neq i^*$, $v$ is completely hidden, since $(1, m^*)(K' + va^+) + ma^+$ is identically distributed to $(1, m^*)K' + ma^+$ (namely, $(1, m^*)v$ is masked by $\mu \notin 2_p$);
- From $\text{Stg}(m_i)$ with $i = i^*$ (at most one query), it leaks $(1, m^*)(K' + va^+)$, which is captured by $(1, m^*)v$.
and III setting. For the Type I setting we have one-time UF LIN.

Figure 5.7: Structure-preserving signatures secure against random message attacks for rSPS that from Figure 5.1 and an unbounded attacks

In this section, we consider possible efficiency improvements on the structure-preserving signatures (SPS)

5.3 Security against Random Message Attacks

Figure 5.6: Description of B (having access to the oracles INITIALIZECore, EVAL, CHAL and FINALIZECore) for the proof of Lemma 5.2.13.

To convince FINALIZE([m*]1, σ*) output 1, adversary A must correctly compute

\[(1, m^\top)(K' + v\perp)\]

and thus \((1, m^\top)v \in \mathbb{Z}_p\). Given \((1, m^\top)v\), for any adaptively chosen \(m* \neq m\), \((1, m^\top)v\) is uniformly random over \(\mathbb{Z}_p\) from the adversary’s viewpoint. Therefore, \(Pr[G^A_\perp \Rightarrow 1] \leq 1/p\).

Summing up Lemma 5.2.9 to Lemma 5.2.14 we have \(\epsilon'' \leq \epsilon + Q^2/2p + (Q + 1)(2Q\epsilon' + Q/p + 1/p)\), which concludes our theorem.

5.3 Security against Random Message Attacks

In this section, we consider possible efficiency improvements on the structure-preserving signatures (SPS) from Section 5.1 and Section 5.2 for the weaker security notion of unforgeability against random message attacks (UF-RMA). Precisely, we obtain a one-time UF-RMA-secure SPS with signature size one less than that from Figure 5.1 and an unbounded UF-RMA-secure SPS with signature size \(k + 1\) less than that from Figure 5.4. Figure 5.7 summarizes our results.

Our rSPS_sat is optimal for both the Type I and III settings: in the Type I setting, under the 2-Lin assumption, rSPS_sat requires 2 elements and 2 verification equations, matching the lower bound for one-time UF-RMA-secure SPS from AGOT14: in the Type III setting, under the SXDH assumption, rSPS_sat requires 1 element and 1 verification equation, which is clearly optimal.

![Table showing security assumptions and their corresponding number of elements and equations](image)

Unforgeability against Random Message Attacks. UF-RMA-security states that it is hard for an adversary to forge a signature even if he sees many signatures on randomly chosen messages from the message space. The security is formally defined as follows:
Theorem 5.3.2. If the $\mathcal{D}_k$-KerMDH problem is $(t, \varepsilon)$-hard in $G_2$, then $rSPS_{ot}$ from Figure 5.9 is $(t', \varepsilon', 1)$-UF-RMA secure, where $\varepsilon' \leq \varepsilon + 1/p$ and $t' \approx t$.

Our proof is similar to that in [KW15, Theorem 2]. As we choose $m \sim Z_{p^n}^\ast$ in the security game ourselves, we can compute the kernel basis $M^\perp \in \mathbb{Z}_{p^{n+1}}^{x \times n}$ of $(1, m^\perp)$ such that $(1, m^\perp) \cdot M^\perp = 0$ and then we embed $M^\perp$ in the secret key $K$. This way we do not need to compute the kernel of $[A]^\perp$ when answering the signing query. However, for the forgery $m^* \neq m$, since $(1, m^{*\perp})M^\perp \neq 0$, the adversary has to compute an element from the kernel to break UF-RMA-security, which is infeasible under the $\mathcal{D}_k$-KerMDH Assumption.

Proof. Perfect correctness and the structure-preserving property are straight-forward to verify. We proceed to establish one-time UF-RMA-security based on the $\mathcal{D}_k$-KerMDH assumption. Let $A$ be an adversary against the $(t', \varepsilon', 1)$-UF-RMA security of $rSPS_{ot}$. In Figure 5.10 we construct an adversary $B (t, \varepsilon)$-breaks the $\mathcal{D}_k$-KerMDH assumption, where

$$\varepsilon' \leq \varepsilon + 1/p$$

(5.4)

We first show that the simulated distribution by $B$ is identical to the real distribution:

- We note that $C = (K'[M^\perp])A' = K'[\hat{A}] + M^\perp\hat{A}' = (K' + M^\perp T_{A'}) \cdot \hat{A} = K \cdot \hat{A}$. Since $K$ is uniform over $\mathbb{Z}_{p^{n+1}}^{x \times k}$, $C$ is distributed the same as in the real distribution.
- For the signature on $[m]_1$, $\sigma := [(1, m^\top)K]_1 = [(1, m^\top)(K' + M^\perp T_{A'})]_1 = [(1, m^\top)K' + 0]_1 = [(1, m^\top)K']_1$. Thus, $\sigma$ is valid under the simulated $pk$. 

\[\text{Procedure \textsc{Initialize}:} \]

\[
(pk, sk) \equiv \text{Gen}(\text{par}) \\
\text{Return } pk
\]

\[\text{Procedure \textsc{Sign}():} \]

\[
m \equiv M ; Q_M := Q_M \cup \{m\} \\
\sigma \equiv \text{Sign}(sk, m) \\
\text{Return } (m, \sigma)
\]

\[\text{Procedure \textsc{Finalize}(m^*, \sigma^*):} \]

\[
\text{Return } (\text{Ver}(pk^*, m^*, \sigma^*) = 1 \land m^* \notin Q_M)
\]
Then we can view the random vector $B_r$ Consider the scheme where notation $\text{Theorem } 5.3.3$. If the final parameter is $\text{RE}$, then $\text{CMA}$ means $\text{UF}$ and matrix $G$ from Figure 5.11 is a full problem is $\text{rSPS}$ secure structure-preserving signature scheme, where $\varepsilon'' \leq \varepsilon + \frac{Q^2}{2p}$ for $k = 1$ and $(2n + 16, (6, 1))$ for $k = 2$. Moreover, we note that the verification needs $2k + 1$ pairing product equations. Compared with the $\text{SPS}_{\text{full}}$ from Figure 5.11, $\text{rSPS}_{\text{full}}$ requires $(k + 1)$ elements less in the signature.

**Theorem 5.3.3.** If the $\mathcal{D}_k$-KerMDH problem is $(t, \varepsilon)$-hard in $G_2$ and $\mathcal{D}_k$-MDDH problem is $(t', \varepsilon')$-hard in $G_1$, then $\text{rSPS}_{\text{full}}$ from Figure 5.11 is a $(t'', \varepsilon'', Q)$-UF-RMA secure structure-preserving signature scheme, where $t \approx t' \approx t''$. 

---

**Figure 5.10:** Description of $\mathcal{B}(\mathcal{PG}, [A]_2 \in G_k^{(k+1) \times k})$ for the proof of Theorem 5.3.2. $\overline{A}$ denotes the first $k$ rows of $A'$ (i.e., $\overline{A} = \overline{A}$) and $\overline{A}'$ denotes the last $n$ rows of $A'$ and $M^\perp$ is a kernel basis of $(1|m^\top)$ such that $(1|m^\top)M^\perp = 0$. 

In the following, we show that if $\text{FINALIZE}$ outputs 1 then $\mathcal{B}$ can compute a solution to the $\mathcal{D}_k$-KerMDH problem as follows:

Let $([m^*], \sigma^* := [z^\top])$ be a valid forgery from $A$ and $y^\top := (1, m^\top)$, i.e., $z^\top \cdot \overline{A} = y^\top \cdot C$. By the definitions of $C$ and $A'$,

$$z^\top \overline{A} = (z^\top | 0)A' = y^\top \cdot C = y^\top (K' | M^\perp) \cdot A'$$

such that $[c]_1$ with

$$c^\top = ((z^\top - y^\top K') | - y^\top M^\perp)$$

satisfies $c^\top A' = 0$. As $m^* \neq m$, $y^\top \notin \text{span}(1, m^\top)$ and thus $y^\top \cdot M^\perp \neq 0$. That implies $c \neq 0$. Finally, $\mathcal{B}$ can extract a solution $[s]_1$ to the $\mathcal{D}_k$-KerMDH problem in $G_2$, from $[c]_1 = [c_1] | [c_2] \in G_1^{(k+1) \times (n-1)}$.

Define $s^\top := c_1^\top + c_2^\top R \in \mathcal{Z}_p^{1 \times (k+1)}$ such that

$$s^\top A = c_1^\top A + c_2^\top RA = (c_1^\top | c_2^\top) \cdot \begin{pmatrix} A \\ R \cdot A \end{pmatrix} = c^\top A' = 0.$$

As $\mathcal{B}$ knows $R, K'$ and $M^\perp$ over $\mathcal{Z}_p$, he can efficiently compute $[s]_1$. It remains to show that $s \neq 0$, with high probability. As $c \neq 0$ and matrix $R$ is only leaked through $A'$ via $RA$, we have

$$\Pr_{R \in \mathcal{Z}_p^{(n-1) \times (k+1)}} [c_1^\top + Rc_2^\top = 0 | RA] \leq \frac{1}{q}.$$ 

The running time of $\mathcal{B}$ is roughly the same as $A'$. This proves equation (5.4). 

---

**5.3.2 Unbounded Secure SPS**

Consider the scheme $\text{SPS}_{\text{full}}$ from Figure 5.4 with the modification that in the signing algorithm, vector $B_r$ is chosen as a random vector as $t \in \mathcal{Z}_p^{2k+1}$. Clearly, under the $\mathcal{D}_k$-MDDH Assumption, this modified scheme is also a UF-CMA-secure SPS. Suppose that the message space is $G_1^n$ with $n = n' + k + 1 \geq k + 1$. Then we can view the random vector $[t]_1 \in G_1^{k+1}$ as part of the message space which reduces the signature size from $3k + 4$ elements to $2k + 3$. The modified scheme is presented in Figure 5.11. Its parameters are:

$$|pk| = (n + 1)k + (2k + 1)k + \text{RE}(\mathcal{D}_k), \quad |\sigma| = (2k + 1, 1),$$

where notation $(x, y)$ means $x$ elements in $G_1$ and $y$ elements in $G_2$. For $k$-Lin, $(|pk|, |\sigma|) = (n + 6, (4, 1))$ for $k = 1$ and $(2n + 16, (6, 1))$ for $k = 2$. Moreover, we note that the verification needs $2k + 1$ pairing product equations. Compared with the $\text{SPS}_{\text{full}}$ from Figure 5.4, $\text{rSPS}_{\text{full}}$ requires $(k + 1)$ elements less in the signature.
Theorem 5.3.3 by defining a sequence of games as in Figure 5.12. Let $\mathbf{A}$ be an adversary against the $(t''', e'', Q)$-UF-RMA security of $\text{rSPS}_{\text{full}}$ and we show that
\[ \epsilon'' \leq \epsilon + \frac{Q^2}{2p} + (Q+1) \left( (2Q+1)\epsilon' + \frac{Q+1}{p} + \frac{1}{p-1} \right). \] (5.5)

By the definition of UF-RMA security (cf. Definition 5.3.1), we have the following lemma.

**Lemma 5.3.4.** \( \epsilon'' = \Pr[G_0^A \Rightarrow 1] \).

**Lemma 5.3.5.** There exists an adversary \( B_1 \) \((t, \epsilon)\)-breaks \( D_k\text{-KerMDH} \) problem in group \( G_2 \), where

\[ \epsilon \geq |\Pr[G_1^A \Rightarrow 1] - \Pr[G_0^A \Rightarrow 1]|, \quad t \approx t''. \]

**Proof.** Suppose \( \epsilon([t^\top], [\sigma^*]) = \epsilon(\sigma_2^*, [1, 2]) \). We note that

\[
\epsilon(\sigma_1^*, [A]) = \epsilon(([1|m^\top]), [C]) + \epsilon(\tau_{k}, [1, 2]) + \epsilon(\tau_{k^*}, [1, 2])
\]

\[ \iff \epsilon(\sigma_1^*, [A]) = \epsilon(([1|m^\top]), [K^a]) + \epsilon(\tau_{k}, [1, 2]) + \epsilon(\tau_{k^*}, [1, 2]) \]

\[ \iff \epsilon(\sigma_1^*, [1, 2]) = \epsilon(([1|m^\top]), [K^a]) + \epsilon(\tau_{k}, [1, 2]) + \epsilon(\tau_{k^*}, [1, 2]) \]

By the same argument in \( G_1 \) of Theorem 5.2.8 for any \(([m^\top]), \sigma^* \) accepted by \( G_0 \) but not by \( G_1 \), then the value

\[ x^\top := \sigma_1^* - ([1|m^\top])K + [\tau^\top]K + \sigma_2^*K \in G_1^{\times(k+1)} \]

is a non-zero vector in the kernel of \( A \), which is hard to be computed under the \( D_k\text{-KerMDH} \) assumption in \( G_2 \). This means that

\[ |\Pr[G_1^A \Rightarrow 1] - \Pr[G_0^A \Rightarrow 1]| \leq \epsilon. \]

By the same arguments as in games \( G_2 \) and \( G_3 \) in Theorem 5.2.8 we have the following two lemmas:

**Lemma 5.3.6.** \( \Pr[G_0^A \Rightarrow 1] \geq \Pr[G_1^A \Rightarrow 1] - \frac{Q^2}{2p} \).

**Lemma 5.3.7.** \( \Pr[G_0^A \Rightarrow 1] \geq \frac{1}{t'\epsilon'} \Pr[G_0^A \Rightarrow 1] \).

**Lemma 5.3.8.** There exists an adversary \( B_1 \) \((t', \epsilon')\)-breaks the \( D_k\text{-MDDH} \) problem in group \( G_1 \), where

\[ \epsilon' \geq |\Pr[G_1^A \Rightarrow 1] - \Pr[G_0^A \Rightarrow 1]| - 1/(p-1) \]

and \( t \approx t' \).

**Proof.** The only difference between \( G_4 \) and \( G_3 \) is that we compute \( t = Br \) instead of picking a random vector \( t \). It is easy to see that the difference is bounded by the \( D_k\text{-MDDH} \) Assumption in \( G_1 \).

Precisely, we construct an adversary \( B_1 \) to break the \( Q \)-fold \( D_k\text{-MDDH} \) Assumption if \( A \) can distinguish \( G_4 \) and \( G_3 \). Let \((B_1, H_1)\) be the \( Q \)-fold \( D_k\text{-MDDH} \) challenge. \( B_1 \) picks \( K, K_0 \) and \( K_1 \) over \( Z_p \) and runs \( \text{Gen(par)} \) honestly. On answering the \( i \)-th \( \text{SIGN} \) query, \( B_1 \) defines \( [t^\top] := [H_1^i] \) and the rest is simulated by using the explicit expressions of \( \tau, K, K_0 \) and \( K_1 \) over \( Z_p \).

One can see that if \([H_1^i] = [BW] \), then the simulation is identical to \( G_4 \); and, otherwise, the simulation is identical to \( G_3 \). By Lemma 2.2.4 we can tightly bound \( G_4 \) and \( G_4 \), \( |\Pr[G_1^A \Rightarrow 1] - \Pr[G_0^A \Rightarrow 1]| \leq \epsilon' + 1/(p-1) \).

**Lemma 5.3.9.** There exists an adversary \( B_2 \) \((t', \epsilon')\)-breaks the \( D_k\text{-MDDH} \) problem in group \( G_1 \), where

\[ |\Pr[G_1^A \Rightarrow 1] - \Pr[G_0^A \Rightarrow 1]| \leq 2Q\epsilon' + Q/p \]

and \( t \approx t' \).

**Proof.** Similar to the proof of Lemma 5.2.13 instead of giving a direct construction of \( B_2 \), we construct an adversary \( B \) in Figure 5.13 to show that the difference between \( G_5 \) and \( G_4 \) is bounded by the advantage of breaking \( D_k\text{-MDDH} \) problem in \( G_1 \) with probability and running time as stated.

In the construction of \( B \), if \( u = t^\top(K_0 + \tau_1 K_1) \) \( \text{(i.e. } B \text{ is in Game Core} \text{)} \), then the view of \( A \) is the same as in \( G_4 \). If \( u = m a^\top + t^\top(K_0 + \tau_1 K_1) \) \( \text{(i.e. } B \text{ is in Game Core} \text{)} \), then the view of \( A \) is the same as in \( G_5 \).

**Lemma 5.3.10.** \( \Pr[G_0^A \Rightarrow 1] = \Pr[G_0^A \Rightarrow 1] \leq 1/p \).
5.4 Structure-Preserving Signatures for Bilateral Message Spaces

Consider the information about \( \tau \) and a method of Abe
and thus
\( \sigma = 0 \) otherwise,
\( \sigma = 0 \)

Initialize
\( \sigma = (\sigma_1, \sigma_2, \sigma_3) \)
\( \sigma = (\sigma_1, \sigma_2) \)
\( \sigma = \tau \)
\( \sigma = \tau \)

Return \( \sigma \)

To conclude the proof, we bound the adversarial advantage in

\( \Pr[\mathbf{G}_0 \Rightarrow 1] \leq 1/p \)

which concludes our theorem.

5.4 Structure-Preserving Signatures for Bilateral Message Spaces

Let \( \mathcal{M} := \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \) be a message space. In Type III pairing groups, \( \mathcal{M} \) is bilateral if both \( n_1 \neq 0 \) and \( n_2 \neq 0 \); otherwise, \( \mathcal{M} \) is unilateral. In this section, we extend the construction from Section 5.2 to sign bilateral message spaces.

The main idea of our construction is to use the Even-Goldreich-Micali (EGM) framework \([\text{EGM96}]\) and a method of Abe et al. \([\text{ACD}^+12]\) for \( m = (m_1 [1], m_2 [2]) \in \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \), and sign \( m_1 [1] \) by using a one-time SPS with a fresh public key \( \text{pk}_{\text{ot}} \) over \( \mathbb{G}_2 \) and then sign message \((m_2 [2], \text{pk}_{\text{ot}})\) using an unbounded UF-CMA-secure SPS; the signature on \( m_1 [1], m_2 [2] \) is \( \text{pk}_{\text{ot}} \) together with the concatenation of both signatures. However, this yields long signatures as \( \text{pk}_{\text{ot}} \) contains \( \mathcal{O}(n_1 k) \) group element for the best known one-time SPS. Next, we observe that our one-time SPS is in fact a so-called “two-tier” signature scheme, i.e. \( \text{pk}_{\text{ot}} \) can be decomposed into a reusable long primary key plus a one-time short secondary key which contains only \( k \) group elements. For the transformation sketched above it is sufficient to put the short secondary key in the signature which leads to short signatures.

Concretely, under the SXDH assumption, our signature on messages in \( \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \) contains \( (7, 3) \) group elements (7 elements in \( \mathbb{G}_1 \) and 3 elements in \( \mathbb{G}_2 \)), 4 pairing product equations for verification and \( (n_1 + n_2 + 8) \) group elements in public keys. A previous SXDH-based construction from \([\text{ACD}^+12]\)
required \((8,6)\) group elements in the signature, 5 pairing product equations, and \((n_1 + n_2 + 22)\) elements in the public key.

We note that our idea gives a generic way to extend message space \(M_1\) to \(M_1 \times M_2\) for signature schemes, where \(M_1\) and \(M_2\) are arbitrary message spaces. In Subsection 5.4.1, we present our transformation for arbitrary (not necessarily structure-preserving) signatures and show that \(\text{SPS}_{\text{ot}}\) from Figure 5.1 satisfies the stronger notion of two-tier signatures. Finally, in Subsection 5.4.2, we instantiate the transformation with the above two-tier \(\text{SPS}\) and the unbounded UF-CMA-secure \(\text{SPS}_{\text{full}}\) from Figure 5.3. By our generic composition theorem the resulting scheme is unbounded UF-CMA secure. Furthermore, it can be verified to be structure-preserving for bilateral message spaces.

### 5.4.1 Two-Tier Signatures

The notion of two-tier signatures was firstly proposed by Bellare and Shoup [BS07] and considered to the structure-preserving setting by Abe et al. [ACD+12] (called partial one-time signatures in [ACD+12]). A two-tier signature scheme is like a standard signature scheme except that the public (secret) key is split into a fixed primary part \(pk\) (sk) and a variable secondary part \(opk\) (osk). We recall the definition of a two-tier signature scheme and its security.

**Definition 5.4.1 (Two-tier signature).** A two-tier signature scheme \(\text{TTS}\) is defined as a tuple of probabilistic polynomial time (PPT) algorithms \(\text{TTS} := (\text{PGen}, \text{SGen}, \text{TTSSign}, \text{TTVer})\):

- The probabilistic primary key generation algorithm \(\text{PGen}(\text{par})\) returns the primary public/secret key \((pk, sk)\). We assume that \(pk\) implicitly defines a message space \(M\) and a secondary public key space \(OPKs\).
- The probabilistic secondary key generation algorithm \(\text{SGen}(pk, sk)\) returns the secondary public/secret key \((opk, osk)\).
- The probabilistic signing algorithm \(\text{TTSSign}(sk, osk, m)\) returns a signature \(\sigma\).
- The deterministic verification algorithm \(\text{TTVer}(pk, opk, m, \sigma)\) returns 1 (accept) or 0 (reject).

(Perfect correctness.) for all \((pk, sk) \not\mathcal{\in} \text{PGen}(\text{par})\), all \((opk, osk) \not\mathcal{\in} \text{SGen}(pk, sk)\), all messages \(m \in M\) and all \(\sigma \not\mathcal{\in} \text{TTSSign}(sk, osk, m)\) we have \(\text{TTVer}(pk, opk, m, \sigma) = 1\).

In the following, we define two-tier CMA security (TT-CMA-security) for \(\text{TTS}\) (which was called OT-NACMA-security in [ACD+12]). It is weaker than the original security notion from [BS07] but sufficient for our application. (We note that our two-tier \(\text{SPS}\) in Figure 5.15 satisfies the stronger security from [BS07].)

<table>
<thead>
<tr>
<th>Procedure</th>
<th>INITIALIZE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>((pk, sk) \mathcal{\in} \text{PGen}(\text{par}))</td>
<td></td>
</tr>
<tr>
<td>Return (pk)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Procedure</th>
<th>TTSign(m):</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i = i + 1); ((opk_i, osk_i) \mathcal{\in} \text{SGen}(pk, sk))</td>
<td></td>
</tr>
<tr>
<td>(\sigma \mathcal{\in} \text{TTSSign}(sk, osk, m))</td>
<td></td>
</tr>
<tr>
<td>(Q_M := Q_M \cup {(i, m, \sigma)})</td>
<td></td>
</tr>
<tr>
<td>Return ((opk_i, \sigma))</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Procedure</th>
<th>FINALIZE((i^<em>, m^</em>, \sigma^*)):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return ({(i^<em>, m, \sigma) \in Q_M \land m^</em> \neq m \land \text{TTVer}(pk, opk, m, \sigma^*) = 1})</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.14: Security Game TT-CMA for Definition 5.4.2

**Definition 5.4.2 (TT-CMA-security).** A two-tier signature scheme \(\text{TTS}\) is said to be \((t, \varepsilon, Q)\)-TT-CMA secure if for all adversaries \(A\) running in time at most \(t\) and making at most \(Q\) queries to the signing oracle, \(\Pr[\text{TT-CMA}^A \Rightarrow 1] \leq \varepsilon\), where game TT-CMA is defined as in Figure 5.14

Our two-tier signature scheme. We now show that \(\text{SPS}_{\text{ot}}\) from Figure 5.1 can be modified to be a two-tier signature scheme with message space \(M = \mathbb{G}_1^n\) in Figure 5.15. We split the secret key of \(\text{SPS}_{\text{ot}}\) (matrix \(K\)) into the first row \(k^1\) and the lower \(n\) rows \(K^t\). Matrix \(K^t\) is the primary secret key and vector \(k\) is the secondary secret key. The reason why we can reuse \(K^t\) is that in each signing query a fresh \(k^i\) is chosen which hides \(m^t K^t\). The only information leaked from signing queries is \((1, m^t)(k^i, K^t)\).
Given that, \((1, m^T)(k^\top_{t'})\) is uniform for \(m^* \neq m\) by the same arguments as in Section 5.1. Lemma 5.4.3 formalizes the above intuition and security of TTSPS\(_{ot}\) is shown in Theorem 5.4.4. We note that TTSPS\(_{ot}\) is a generalization of POSu2 from \([\text{ACD}+12]\).

\begin{figure}[h]
\centering
\begin{tabular}{|c|c|}
\hline
PGen(par): & TTSign(sk, osk, \([m]_1\)):
\hline
\begin{align*}
P \triangleq & \mathbb{D}_\text{par}; K' \triangleq \mathbb{Z}^{n \times (k+1)} \\
C' \triangleq & K' A \in \mathbb{Z}^{n \times k}
\end{align*}
\begin{align*}
K & \triangleq \left(k_{\perp}, k_{\top}\right) \\
\sigma & \triangleq \left([1, m^T]_1 K\right) \\
\text{Return } & \sigma \in \mathbb{G}_1^{1 \times (k+1)}
\end{align*}
\hline
\begin{align*}
\left(i^*, m\right) \in & \mathbb{Q}_M \land m^* \neq m \\
\land (z^T = (1, m^T) \cdot \left(k_{\perp}^\top, k_{\top}^\top\right) \\
& (i^*, m^*, z) \in \mathcal{A}'(1, m^T)(K' A)
\right)
\end{align*}
\begin{align*}
\left(\sigma \neq \frac{1}{p}\right),
\end{align*}
\end{tabular}
\caption{Two-tier signature scheme TTSPS\(_{ot}\) with message-space \(\mathcal{M} = \mathbb{G}_1^n\).}
\end{figure}

The following is the main computational core lemma required for the proof of TTSPS\(_{ot}\).

**Lemma 5.4.3.** Let \(n, k\) be integers. For any \(A \in \mathbb{Z}^{(k+1) \times k}\) and any (possibly unbounded) adversary \(A\),

\[
\Pr \left[ (i^*, m) \in \mathbb{Q}_M \land m^* \neq m \land (z^T = (1, m^T) \cdot \left(k_{\perp}^\top, k_{\top}^\top\right) \mid \left(i^*, m^*, z\right) \in \mathcal{A}'(1, m^T)(K' A) \right] \leq \frac{1}{p},
\]

where:

- \(\mathcal{O}(m): i = i + 1 \text{ (initialized with } 0\text{) picks } k_i \in \mathbb{Z}^{k+1}_p, \text{ adds } (i, m) \text{ to } \mathbb{Q}_M \text{ (initialized with } 0) \text{ and returns } k_{\perp}^\top A \text{ and } (1, m^T) \cdot (k_{\top}^\top).

**Proof.** Fix any \(A \in \mathbb{Z}^{(k+1) \times k}\). Let \(a^+ \in \mathbb{Z}^{1 \times (k+1)}\) be a non-zero vector in the kernel of \(A\) such that \(a^+ \cdot A = 0 \in \mathbb{Z}^{1 \times k}\). We make the following changes to the distribution of the experiment:

- Switch \(K' \in \mathbb{Z}^{n \times (k+1)}\) to \(K'' = K'' + u a^+\), where \(K'' \in \mathbb{Z}^{n \times (k+1)}\) and \(u \in \mathbb{Z}^n\).
- Switch \(k_{\perp} \in \mathbb{Z}^{k+1}_p \in \mathcal{O} \text{ to } k_{\perp}' = (u, a^+)^\top\), where \(k_{\perp}' \in \mathbb{Z}^{k+1}_p\) and \(u \in \mathbb{Z}^n\).

We note that the modified distribution is identical to the real distribution of the experiment, since \(K''\) and \(k_{\perp}'\) are uniformly chosen.

In the following, we consider the information about \((u^*, u)\) leaked from \(K' A\) and the answers of the \(\mathcal{O}\) queries in order to argue that equation (5.6) holds for any (possibly unbounded) adversary \(A\):

- Since \(K' A = (K'' + u a^+) A = K'' A\), the matrix \(K'' A\) leaks nothing about \(u\). By the same argument, the values \(k_{\perp}^\top A\) from the \(\mathcal{O}\) queries leak nothing about the \(u_i\).
- The output of the \(j\)-th query to \(\mathcal{O}\) on \(m_j\) for \(j \neq i^*\) hides \(u_i\). The reason is that \((1, m^T)(k_{\perp}^\top) = k_{\perp}^\top + m_j k_{\top} = k_{\top}^\top u_j + a^+ m_j K'\), which is identically distributed to \(k_j^\top + u_j a^+ m_j K'\), since \(m_j u_j \in \mathbb{Z}_p\) is masked by fresh randomness.
- The output of the \(i^*\)-th \(\mathcal{O}\) query on \(m\) leaks \((1, m^T)(u_{i^*})\), since \((1, m^T)(k_{\perp}^\top) = (1, m^T)(k_{\perp}^\top) + (1, m^T)(u^*)\) holds. To compute \((i^*, m^*, z)\) such that \(z^T = (1, m^T) \cdot \left(k_{\perp}^\top, k_{\top}^\top\right) = (1, m^T)(k_{\perp}^\top) + (1, m^T)(u^*)\), we have that \((1, m^T)(u_{i^*})\) is uniformly random over \(\mathbb{Z}_p\) from the adversary’s view. This shows equation (5.6). \(\square\)

**Theorem 5.4.4.** If the \(D_k\)-KerMDH problem is \((t, \varepsilon)\)-hard in \(G_2\), TTSPS\(_{ot}\) from Figure 5.15 is a \((t', \varepsilon', Q)\)-TT-CMA-secure two-tier signature scheme.

**Proof.** Perfect correctness is straightforward to verify. We proceed to establish TT-CMA-security based on the \(D_k\)-KerMDH assumption. We will show that for all adversaries \(A\), there exists an adversary \(B\) \((t, \varepsilon)\)-breaks the \(D_k\)-MDDH problem with \(t \approx t'\) and \(\varepsilon \geq \varepsilon' - 1/p\).
We construct a simple reduction \( B(\mathcal{P}_{G}, [A]_2 \in \mathbb{G}_2^{(k+1) \times k}) \) as follows: \( B \) generates \( pk = ([C]_2, [A]_2) \) as in the real scheme by picking \( K' \in \mathbb{Z}_p^{n \times (k+1)} \) and computing \( [C]_2 := [K'A]_2 \). Next, \( B \) runs \( A \) on \( pk \) and simulates \( \text{TTS} \text{Sign} \) as in the real scheme:

- \( \text{TTS} \text{Sign}([m]_1) : i = i + 1, \) picks \( k_i \notin \mathbb{Z}_p^{k+1} \), computes \( opk_i = [k_i] A_2 \), computes \( \sigma = ([1, m^T] \cdot (k_i^T))_1 \), adds \((i, [m]_1, \sigma)\) to \( Q_M \) and returns \( \sigma \).

With probability \( \varepsilon' \), \( B \) obtains \((i^*, [m^*_1], \sigma^* \rangle \) such that there exists \((i^*, [m]_1, \sigma) \in Q_M \) and \( m^* \neq m \) and \( \varepsilon(\sigma^*_1, [A]_2) = \varepsilon([1, m^T]_1, [K'A]_2) \), where \( K' := (k_i^T) \). Then \( B \) returns \( s_i \) computed as \( s_i = \sigma^* - [1, m^T]_1 K^* \).

Clearly, \( s \cdot A = 0 \). The information-theoretic argument of Lemma 5.4.3 captures the fact that, for any \( A \in \mathbb{Z}_p^{(k+1) \times k} \) and any adversary \( A \), given \((A, K'A) \) over \( \mathbb{Z}_p \) and \( Q \)-many \((1, m^T_1)(k_i^T)\) for adversarial chosen \( m_1 (K' \notin \mathbb{Z}_p^{n \times (k+1)}), k_i \notin \mathbb{Z}_p^{k+1}\), \( A \) can not come up with \((z, m^*)\) such that \( z - (1, m^T)_1 (k_i^T) = 0 \) \((i^* \in \{1, \ldots, Q\})\). Thus, \( \Pr[s = 0] \leq 1/p \) by Lemma 5.4.3.

**Theorem 5.4.5.** If \( \text{TTS} \) is \((t, \varepsilon, Q)\)-TT-CMA secure and \( \text{SIG} \) is \((t', \varepsilon', Q)\)-UF-CMA secure, then \( \text{TTS} \text{SIG} \) is a \((t'', \varepsilon'', Q)\)-UF-CMA-secure signature scheme, where

\[
\varepsilon'' \leq \varepsilon + \varepsilon', \quad t \approx t' \approx t''.
\]

Perfect correctness is implied by perfect correctness of \( \text{TTS} \) and \( \text{SIG} \). We will show that for any adversary \( A \), its advantage is bounded by \( \varepsilon'' \leq \varepsilon + \varepsilon' \).

Since the proof is similar to that for the EGM framework [EGM96, ACD+12], we only sketch the proof. Let \((m_1^*, m_2^*, \sigma^* = (opk^*, \sigma_1^*, \sigma_2^*) \rangle\) be a forgery from \( A \). \( A \) can make at most \( Q \) signing queries to \( \text{Sign} \) for \( \text{TTS} \text{SIG}, \text{TTS} \) and we denote the \( i \)-th query by \((m_1^i, m_2^i)\) and its answer as \((opk_i, \sigma_1^i, \sigma_2^i)\).

There are two complementary cases:

- There exists an \( i \in \{1, \ldots, Q\} \) such that \((m_2^i, opk^*) = (m_2^i, opk_i)\). As \((m_1^i, m_2^i) \notin Q_M, m_1^i \neq m_1^i \).

- \((m_2^i, opk^*) \neq (m_2^i, opk_i)\) for all \( i \in \{1, \ldots, Q\} \). Clearly, \((m_2^*, opk^*), \sigma_2^*\) is a valid forgery that breaks the unbounded UF-CMA-security of \( \text{SIG} \).

**5.4.2 Instantiation**

Combining \( \text{TTSPS}_{\text{ot}} \) from Figure 5.15 and \( \text{SPS}_{\text{full}} \) from Figure 5.4, we obtain an UF-CMA-secure signature scheme \( \text{BSPS}_{\text{full}} \), see Figure 5.17. One can verify that it is structure preserving with bilateral message space \( M = \mathbb{G}_1^{k+1} \times \mathbb{G}_2^k \) and the following parameters:

\[
|pk| = (n_1 + n_2)k + 3(k + 1)k + 2\text{REL}(D_k), \quad |\sigma| = (k + 2, 4k + 3), \quad \#\text{equations} = 3k + 1.
\]

Notation \((x, y)\) means \( x \) elements in \( \mathbb{G}_1 \) and \( y \) elements in \( \mathbb{G}_2 \). We note that the representation of \( \mathbb{G}_2 \) is longer than that of \( \mathbb{G}_1 \) elements. To simplify the efficiency comparison, one can use \( \text{TTSPS}_{\text{ot}} \).
More Efficient Structure-Preserving Signatures

5.4

Figure 5.17: Structure-preserving signature $\mathcal{B}_{\text{SPS}^\text{full}}$ with bilateral message spaces $\mathcal{M} = \mathbb{G}_1^{k+1} \times \mathbb{G}_2^{n_2}$.

to sign $[m_2]_2$ and $\text{SPS}^\text{full}$ to sign $([m_1]_1, [z]_1)$, which gives us a scheme with $|\sigma| = (4k + 3, k + 2)$. Under the SXDH assumption, our scheme achieves $(|pk|, |\sigma|, \#\text{equations}) = (n_1 + n_2 + 8, (7, 3), 4)$. Compared with $(n_1 + n_2 + 22, (8, 6), 5)$ of $\mathcal{ACD}^\text{12}$, we obtain better efficiency under standard assumptions.
Bibliography


