Theories of Homomorphic Encryption, Unification, and the Finite Variant Property *

Fan Yang
University of Illinois at Urbana-Champaign, USA
fanyang6@illinois.edu

Santiago Escobar
DSIC-ELP, Universitat Politècnica de València, Spain
sescobar@dsic.upv.es

Catherine Meadows
Naval Research Laboratory, Washington DC, USA
meadows@itd.nrl.navy.mil

José Meseguer
University of Illinois at Urbana-Champaign, USA
meseguer@illinois.edu

Paliath Narendran
University at Albany-SUNY, Albany, NY, USA
dran@cs.albany.edu

Abstract

Recent advances in the automated analysis of cryptographic protocols have aroused new interest in the practical application of unification modulo theories, especially theories that describe the algebraic properties of cryptosystems. However, this application requires unification algorithms that can be easily implemented and easily extended to combinations of different theories of interest. In practice this has meant that most tools use a version of a technique known as *variant unification*. This requires, among other things, that the theory be decomposable into a set of axioms B and a set of rewrite rules R such that R has the *finite variant property* with respect to B. Most theories that arise in cryptographic protocols have decompositions suitable for variant unification, but there is one major exception: the theory that describes encryption that is homomorphic over an Abelian group.

In this paper we address this problem by studying various approximations of homomorphic encryption over an Abelian group. We construct a hierarchy of increasingly richer theories, taking advantage of new results that allow us to automatically verify that their decompositions have the finite variant property. This new verification procedure also allows us to construct a rough metric of the complexity of a theory with respect to variant unification, or *variant complexity*. We specify different versions of protocols using the different theories, and analyze them in the Maude-NPA cryptographic protocol analysis tool to assess their behavior. This gives us greater understanding of how the theories behave in actual application, and suggests possible techniques for improving performance.

Categories and Subject Descriptors C.2.2 [*Computer communication Networks*]: Network Protocols; D.2.4 [*Software Engineering*]: Software/Program Verification; D.3.2 [*Programming Languages*]: Language Classifications; D.4.6 [*Operating Systems*]: Security and Protection; F.3.1 [*Logics and Meanings of Programs*]: Specifying and Verifying and Reasoning about Programs

Keywords cryptographic protocol analysis, homomorphic encryption, finite variant property, unification

1. Introduction

Recent advances in the automated analysis of cryptographic protocols [3, 7, 19, 25, 26, 39] have demonstrated that state exploration via unification modulo theories is both feasible and can make a substantial difference to the expressiveness of the system model. In systems of this type, protocol execution paths are computed by unifying messages received with messages sent. Since equational properties are usually involved, the unification must be *modulo* the equational theory describing those properties. This requires unification algorithms that accommodate well to combinations of theories and can be integrated with state space exploration techniques. The technique of *variant unification*, first formalized as a general approach in [20], although used for specific theories much earlier than that (e.g. in [7, 8, 18]), satisfies all these properties. Thus, it is used by many cryptographic protocol analysis tools in one form or another, including ProVerif [8], OFMC [7], Maude-NPA [19] and Tamarin [39] (see Section 3 of this paper for a comparison).

In order for variant unification to work, the equational theory of interest must be decomposable into (Σ, B, R), where: (1) B is regular and has a finitary unification algorithm, (2) R is confluent, terminating, and coherent modulo B (for simplicity, we will simply say from now on that R is *convergent* modulo B) and (3) R has the *finite variant property* (FVP) [15] with respect to B; i.e., for each term t there is a finite set of most general pairs $\{(\sigma_1, t_1), \ldots, (\sigma_n, t_n)\}$ (called *variants*) of $\rightarrow_{R,B}$-normalized substitutions and terms such that $t\sigma_i$ normalizes to t_i modulo B. Fortunately, many cryptographic theories of interest can be decomposed in this way, with $B = \emptyset$ or $B = AC$ (associativity and commutativity).

*Jose Meseguer and Fan Yang have been partially supported by NSF Grant CNS 13-10109. Santiago Escobar has been partially supported by the EU (FEDER) and the Spanish MINECO under grants TIN 2010-21062-C02-02 and TIN 2013-45732-C4-1-P, and by Generalitat Valenciana PROMETEO2011/052.

PPDP '14, September 08–10, 2014, Canterbury, United Kingdom.

http://dx.doi.org/10.1145/2643135.2643154

However, there is one important family of theories that fails to have a decomposition that satisfies our requirements: H for a homomorphic property of the form $e(X * Y, K) = e(X, K) * e(Y, K)$ where $*$ may or may not have other properties. AGH (the case where $*$ is an Abelian group) is a property belonging to a number of different cryptographic algorithms, starting with RSA in the late 70's [38], and from early on it was realized to have a number of potential applications, including anonymous payment systems [13], computation on encrypted data [37], and voting [12].

The fact that the theory AGH by itself does not satisfy our needs does not mean that all is lost, however. But it does mean that we must investigate approximations that give us the ability to model a broad class of protocols while still satisfying the conditions necessary for variant based unification. First, we could use a dedicated unification algorithm for AGH, as we did in [17] for H; however, this is not satisfactory, see Section 3.2 for the reasons. Second, we can use an under-approximation of AGH, as we did in [17] with only one homomorphic encryption operator. Third, we can embed AGH or a subtheory in a richer theory; this may introduce false paths when verifying a protocol, but these can be discarded upon inspection. Fourth, we can use a combination of under and over-approximations in the same theory.

Such a strategy of course requires a fair amount of experimentation, both in checking for FVP and in using the theories in protocol analysis. This has until recently been hampered by the fact that although necessary and sufficient conditions for FVP have been known for some time [15, 21], checks for these conditions are not straightforward to implement.

Indeed, this turns out to be an undecidable problem [10]. However, a new semi-decision procedure for FVP has been developed which works well in practice: it has recently [14] been shown that, in order to prove FVP for a theory (Σ, B, R), where R is convergent modulo B, it is enough to check, for each function symbol f, whether or not each pattern of the form $f(X_1, \ldots, X_n)$ has a finite number of variants, where the X_i are distinct variables of the appropriate kind and n is the arity of f. This can be done by attempting to generate all the variants of $f(X_1, \ldots, X_n)$, a feature supported by the forthcoming Maude 2.7. The other properties can be checked via the use of the Maude Formal Environment [1]. Thus it has become straightforward for us to verify that an equational theory has the properties we need.

This ability to generate all the variants of each $f(X_1, \ldots, X_n)$ appearing in a theory also gives us a measure of the overhead introduced by using this theory in a variant-based cryptographic protocol analysis tool: the larger the number of variants produced, the larger the number of variants of a state generated by a unification-based protocol analysis tool, and thus the larger the number of states generated during a search, and thus the longer the time required to generate them. We refer to this measure as the *variant complexity* of the theory.

All these capabilities taken together greatly facilitate the generation and verification of theories with the finite variant property, and allow us to refine our strategy for generating theories with FVP decompositions. We start with a non-FVP theory such as homomorphic encryption, or a theory with unacceptably large variant complexity. First, we add equations that we conjecture achieve a finite number of variants. Then we check for convergence modulo B using Maude tools. If convergence is not satisfied we add additional equations (possibly suggested by Maude Church-Rosser checker's output) and try again. Once we are satisfied that the theory is convergent modulo B we then use Maude to check for FVP and compute the variant complexity. We can then test the theory using Maude-NPA. If the performance is unacceptable we go through the process again, introducing under or over approximations to achieve lower variant complexity. If the performance is acceptable we may again try to tweak the theory, but in this case to increase its faithfulness to the theory of interest rather than reducing variant complexity. We note that use of under approximation mean that Maude-NPA may miss attacks (that is, it affects completeness), while over approximation means that Maude-NPA may find spurious attacks (that is, it affects soundness). Thus, ultimately we would expect to use several different theories, some under and some over approximations to approximate a theory of interest.

1.1 Our contributions

The contributions of this paper are thus fourfold. First, we develop a *general strategy* for *approximating* theories without FVP or with high variant complexity by using a combination of under-approximation (eliminating equations from a theory or substituting them by weaker ones) and over-approximation (adding additional function symbols and equations), and then computing the variant complexity. The use of Maude to verify convergence modulo B and to compute the variant complexity greatly facilitates the experimentation needed to carry this strategy out.

Secondly, we apply the strategy to develop a *hierarchy* of theories approximating homomorphic encryption that are verified to have the finite variant property.

Thirdly, as a result of proving the finite variant property for the theories in the hierarchy, we automatically obtain unification algorithms for these theories, which to the best of our knowledge are new unification results except for the already known H and AGH cases.

Fourthly, we have performed a careful experimental evaluation of the performance tradeoff between the faithfulness with which the theory models cryptographic operations and the number of variants, giving us a better understanding of how variant complexity affects performance of automated protocol analysis tools. We used the most promising theories from the point of view of variant complexity and expressiveness to specify and analyze different versions of a protocol using the unification-based cryptographic protocol analysis tool Maude-NPA [19]. In this protocol two principals A and B want to learn the result of performing the operation $*$ on their respective secret information D_A and D_B without revealing the information to each other, or $D_A * D_B$ to anyone else. Each one encrypts its information using a homomorphic encryption algorithm and sends it to a server, who performs the operation $*$ on the encrypted data and sends it to A and B who decrypt the result to obtain $D_A * D_B$. In that paper * was a free operator, so the theory E had a trivial decomposition, in which $R = \emptyset$ and $B = E$. In [19] it was shown that the protocol was subject to an authentication attack if a principal was unable to tell whether it had received valid data or nonsense. In the case that $*$ is a group operator it is also subject to a secrecy attack: A can simply compute $D_A^{-1} * (D_A * D_B)$ to obtain D_B. To our knowledge, this is the first successful use of a cryptographic protocol analysis tool to analyze homomophic encryption over theories obeying nontrivial equational theories.

The theories for which we have found unification algorithms using our strategy are summarized in Figure 1 below. We explain the notation in Figure 1 as follows. A full arrow denotes theory inclusion, a full arrow with two heads denotes theory quotients, and a dotted arrow denotes a generalization. All theories are described in Section 4 and involve the function $e(X, K)$ where e is an encryption operator, X is a term of sort Message, and K is of sort Key or Keys depending on the theory, which are always subsorts of Message. H denotes the homomorphic equation $e(X * Y, K) = e(X, K) * e(Y, K)$. kH denotes the *bounded homomorphism theory*, in which sorts are used to restrict the number of variants of $h(x)$ that can be computed. The symbol & represents the addition of an AC binary function symbol & on terms of a new sort Keys; it can be thought of as a multiset union operator. This

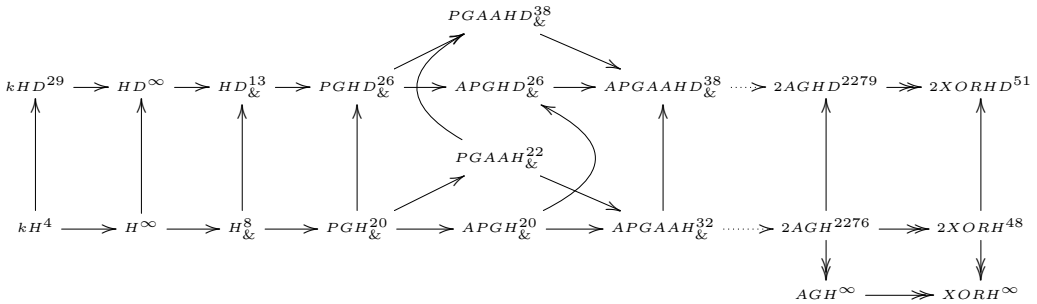

Figure 1. Relations between the theories discussed in this paper

Theories	Under-Approximation	Over-Approximation
kH	Under approximation of H via bound on message length. No group axioms.	-
$H_{\&}$	No group axioms.	Over approximation of H via use of multiset of keys.
$PGH_{\&}, PGAAH_{\&}, APGH_{\&}, APGAAH_{\&}$	Under approximation of associativity axiom.	Over approximation of H via use of multiset of keys.
$2AGH$	Under approximation of H via use of two groups instead of one.* Under approximation of encryption via bound on number of keys.	-
$2XORH$	Under approximation of H via use of two groups instead of one.* Under approximation of encryption via bound on number of keys.	Over approximation of Group Axioms.

* Note that this is not an over-approximation when the encryption function being modeled is indeed a homomorphism from one group to another.

Table 1. Summary of Under and Over approximations w.r.t. AGH

introduces an over approximation if the order of applications of encryption to a message matters. PG adds an inverse $(\_)^{-1}$, a constant 1 and equations making $*$ a *pre-group operator* (unit, inverses, but no associativity, also known as a *loop*, see [36]), while APG makes $*$ an *Abelian pre-group operator* (commutativity, unit, inverses, but no associativity). AA denotes an under approximation of associativity. D denotes the decryption equation $d(e(X,K),K) = X$; this says that the result of decrypting an encrypted message is the original message. 2AGH denotes homomorphic encryption that maps one Abelian group to another: $e(X *_a Y) = e(X) *_b e(Y)$. The operators $*_a$ and $*_b$ are *Abelian group operators* (commutativity, unit, inverses, associativity). Notice that, there is no key explicitly defined in 2AGH, since encryption with a specific key is implicitly captured by the definition of the encryption operator e. 2XORH denotes homomorphic encryption over two Xor operators, which is an over approximation of 2AGH. In all cases the axioms B are either $B = \emptyset$ or the union of all the equations defining C and AC properties. We also note that in many cases we completed the theory to ensure convergence; these are described in detail in Section 4. The superscript number of each theory denotes the "*variant complexity*" and denotes the sum of the number of variants obtained for each function symbol in the theory (excluding constants). If the superscript is ∞, this means that the theory doesn't have FVP. The details of under approximation and/or over approximation w.r.t. AGH of all theories are given in Table 1. The theories with decryption operator are omitted here.

The rest of the paper is organized as follows. In Section 2 we give the background on term rewriting and variant unification necessary for understanding this paper. In Section 3 we give the motivation of FVP in terms of cryptographic protocol analysis. In addition we describe related work in unification and apply it to show that none of the possible decompositions of AGH satisfy the necessary conditions for variant unification. In Section 4 we present the various homomorphic theories we investigated and their properties. In Section 5 we present the results of performing experiments on several representative theories, using Maude-NPA to analyze protocols specified using these theories. In Section 6 we conclude and discuss future work.

2. Background on Term Rewriting

We follow the classical notation and terminology from [41] for term rewriting and from [31, 32] for rewriting logic and order-sorted notions. We assume an *order-sorted signature* Σ with a finite poset of sorts $(\mathsf{S}, \leq)$ and a finite number of function symbols. We furthermore assume that: (i) each connected component in the poset ordering on sorts has a top sort, and for each $\mathsf{s} \in \mathsf{S}$ we denote by $[\mathsf{s}]$ the top sort in the component of s; and (ii) for each operator declaration $f : \mathsf{s}_1 \times \ldots \times \mathsf{s}_n \to \mathsf{s}$ in Σ, there is also a declaration $f : [\mathsf{s}_1] \times \ldots \times [\mathsf{s}_n] \to [\mathsf{s}]$. $\mathcal{T}_\Sigma(\mathcal{X})$ denotes the set of terms for variables $\mathcal{X}$ and $\mathcal{T}_\Sigma$ the set of ground terms. We write $\mathcal{V}ar(t)$ for the set of variables present in a term t. The subterm of t at position p is $t|_p$, and $t[u]_p$ is the result of replacing $t|_p$ by u in t. A *substitution* σ is a sort-preserving mapping from a finite subset of $\mathcal{X}$ to $\mathcal{T}_\Sigma(\mathcal{X})$. The identity substitution is ι. Application of substitution σ to a term t is denoted $t\sigma$.

A Σ-*equation* is an unoriented pair $t = t'$. Given a set B of Σ-equations, order-sorted equational logic induces a congruence relation $=_B$ on terms $t, t' \in \mathcal{T}_\Sigma(\mathcal{X})$; see [32]. A set B of Σ-equations is *regular* if for each $t = t'$ in B, $\mathcal{V}ar(t) = \mathcal{V}ar(t')$. A set B of Σ-equations is *sort-preserving* if for each $t = t'$ in B and for each substitution σ, $t\sigma$ has sort s iff $t'\sigma$ has sort s. A set B of Σ-equations uses *top-sort variables* if for each $t = t'$ in B, the sort of each variable in $\mathcal{V}ar(t) \cup \mathcal{V}ar(t')$ is a top sort. For a set B of Σ-equations, a B-*unifier* for a Σ-equation $t = t'$ is a substitution σ s.t. $\sigma(t) =_B \sigma(t')$. A *complete* set of B-unifiers of an equation $t = t'$ is written $CSU_B(t = t')$. We say $CSU_B(t = t')$ is *finitary* if it contains a finite number of B-unifiers.

A *rewrite rule* is an oriented pair $l \to r$, where $l \notin \mathcal{X}$, $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$, and $l, r \in \mathcal{T}_\Sigma(\mathcal{X})_\mathsf{s}$ for some sort $\mathsf{s} \in \mathsf{S}$. An *(unconditional) order-sorted rewrite theory* is a triple (Σ, B, R) with Σ an order-sorted signature, B a set of Σ-equations, and R a set of rewrite rules. A set R of rules is *sort-decreasing* if for each $t \to t'$ in R, each sort s, and each substitution σ, $t'\sigma$ has sort s implies $t\sigma$ has sort s too. The relation $\to_{R,B}$ on $\mathcal{T}_\Sigma(\mathcal{X})$ is defined as: $t \xrightarrow{p}_{R,B} t'$ (or $\to_{R,B}$) if p is a non-variable position of t, $l \to r \in R$, $t|_p =_B \sigma(l)$, and $t' = t[\sigma(r)]_p$ for some σ.

A *decomposition* (Σ, B, R) of an equational theory E is a rewrite theory that satisfies the following properties: (i) B is regular, sort-preserving and uses top-sort variables, (ii) B has a finitary unification algorithm, and (iii) the rules R are *convergent* modulo B, i.e., sort-decreasing, confluent, terminating, and coherent modulo B.

Given a decomposition $E = (\Sigma, B, R)$, a variant of a term t is a pair (t', θ) such that t' is a $\rightarrow_{R,B}$-canonical form of the substitution instance $t\theta$, i.e., there is a term t'' such that $t\theta \rightarrow^*_{R,B} t''$, t'' is a $\rightarrow_{R,B}$-normal form, and $t' =_B t''$. A decomposition (Σ, B, R) has the *finite variant property* (FVP) if there is a complete and finite set of variants for each term (see [15, 21] for details). If a decomposition (Σ, B, R) of an equational theory E has the *finite variant property*, there is an algorithm to compute a finite complete set $CSU_E(t = t')$ of E-unifiers [21].

3. Motivation and Related Work

In this section we discuss the related work that precedes and motivates the work in this paper. This is divided into two parts. The first motivates our interest in FVP in terms of its application to cryptographic protocol analysis. The second gives a brief history of work on unification modulo one-sided distributivity that applies to homomorphic encryption and uses these results to show that no decomposition of AGH satisfies all the conditions necessary for the finite variant property, and thus demonstrates the need for other solutions such as theory approximations.

3.1 Motivation

Unification-based cryptographic protocol analysis tools are used to analyze cryptographic protocols in which an attacker interacting with the protocol may cause security properties to be violated. Actions of principals are modeled symbolically using logical variables, and paths through protocols are computed by unifying messages expected by a principal with messages sent by a principal, often modulo some equational theory that describes the properties of the crypto algorithms used.

Any unification technique used in cryptographic protocol analysis must satisfy two properties. First of all, it must behave well with respect to composition, especially of disjoint theories, since cryptographic protocols often combine different algorithms described by different theories. Although methods for combining unification algorithms of disjoint theories are well-known [6, 40], the solution in the general case is highly nondeterministic and inefficient, so more efficient special algorithms are desirable.

The second property that must be satisfied is a little more subtle, and has to do with the fact that many of the state space reduction techniques used require that terms in the state be in some kind of normal form with respect to the theory E used. Generally this is expressed by writing E as a decomposition (R, B) where B is regular and has a finitary unification algorithm, and R is convergent modulo B. This ensures enough stability in normal form representations of terms so that syntactic state space reduction techniques can be applied.

Both the first and second desiderata of unification-based cryptographic can be achieved, if the decomposition (R, B) has the finite variant property, via variant unification as described in Section 2. Since the first step of variant unification requires the computation of all the irreducible variants of each side of the unification problem, and a solution is discarded if a solution makes either side reducible, variant unification guarantees the irreducibility constraint required by state space reduction techniques. Moreover, variant unification behaves well under composition, at least in the area of cryptographic protocol analysis. First of all, the axioms B are relatively few and well-understood. Moreover, if the combination of the two theories also has a finite variant decomposition, then the same finite variant algorithm can be applied as well.

Not surprisingly many tools have followed approaches similar, if not identical, to variant unification. Both Maude-NPA [19] and Tamarin [39] use variant-based unification explicitly. Indeed support for variant-based unification is being built into Maude 2.7. Moreover, other tools have used approaches that have many features in common with variant-based unification. For example, ProVerif [8] (see [9, Sec. 5]) and OFMC [7] (see [33, Sec. 10]) both compute the variants of protocol rules, modulo the free theory for ProVerif, and modulo the free theory or AC for OFMC. This has the effect of computing the variants of both sides of the unification problem. More recently, variants have been applied to expanding the capacity of ProVerif to deal with AC theories. Thus, in [26] Küsters and Truderung implement a special case of the exclusive-or theory in the ProVerif tool by expressing it as a rewrite theory with the finite variant property with respect to the free theory ($E = \emptyset$) and computing variants that are unified syntactically. This requires some restrictions on the syntax of the protocol, however. Similar approaches have been applied by Küsters and Truderung for modular exponentiation [25], and Arapinis et al. [3] for commuting encryption and AC theories.

Variant-based unification does have some drawbacks, however. First of all, it can be inefficient for theories of high variant complexity. This can be mitigated by the use of *asymmetric unification* [16], in which only the variants of the right-hand side of a unification problems are computed, and irreducibility constraints are also enforced only on the right-hand sides. This requires the use of specialized asymmetric unification algorithms, so combination is no longer as straightforward, but it is still possible to apply the state space reduction techniques, and efficiency gains, as shown in [16], can be dramatic.

A more serious problem arises when a theory of interest fails to have an FVP decomposition at all. Fortunately, most theories of interest to cryptographic protocol analysis are FVP with respect to a decomposition in which B is either the empty theory or AC. However, there is one notable exception, the theory of encryption homomorphic over another operator, that is $e(X * Y, K) = e(X, K) * e(Y, K)$ where $*$ is an operator that may have some other equational properties, shown not to satisfy FVP when the homomorphic equation is in R by Comon and Delaune in [15]. Comon and Delaune consider the case in which e has only one argument and $*$ is exclusive-or, but their case can be considered as corresponding to a degenerate case of e with two arguments, in which only one key is used. Moreover, their counterexamples apply to any sub theory of the theory they use for $*$, including the Abelian group theory, AC, and the free theory.

Comon and Delaune prove their result by producing counterexamples to a property that they show to be equivalent to FVP. However, it is also easy to produce direct counter examples. Assuming that the distributive equation is oriented as the rewrite rule $e(X * Y) \rightarrow e(X) * e(Y)$, then the irreducible variants of $e(Z)$ are $[(e(Z), \iota), (e(Z_1) * e(Z_2), \{Z \mapsto Z_1 * Z_2\}), \ldots]$. If the equation is oriented as $e(X) * e(Y) \rightarrow e(X * Y)$, then the variants of $e(Z * W)$ are $[(e(Z * W), \iota), (e(e(Z_1 * W_1)), \{Z \mapsto e(Z_1), W \mapsto e(W_1)\}), \ldots]$.

3.2 Related Work in Equational Unification and its Application to Decompositions of AGH

Equational theories that include a homomorphic property appear in many applications, and thus there is a long history of research on unification in this area. As in Comon and Delaune, the homomorphic operator under consideration generally has only one argument. However, as shown before, this theory can be considered as apply-

ing to a degenerate case of homomorphic encryption in which only one key is used and so is still relevant.

The earliest work on homomorphic theories is by Tiden and Arnborg [42] who gave a unification algorithm modulo the theory of one-sided distributivity:[1] $x * (y + z) = (x * y) + (x * z)$. The complexity of their algorithm is exponential. A polynomial-time algorithm for unifiability (existence of a unifier) modulo this theory was developed by Marshall [29, 30]. An alternative unification algorithm using a very different approach was developed by Hai Lin in his dissertation [2, 27]. This algorithm for the equational property $e(X * Y, K) = e(X, K) * e(Y, K)$ in which $*$ is a free operator was implemented in Maude-NPA and several protocols were tested [17].

Theories of homomorphisms of the form $e(x*y) = e(x)*e(y)$, where the $*$ operator has additional properties, were first considered by Nutt [35] and also by Baader [4, 5]. The unification problem is decidable when $*$ is an Abelian group [4] and undecidable when $*$ only has the associative and commutative properties [34]. The decidability results were extended to one-sided distributivity in [23, 24]. Liu in [28] gives a dedicated algorithm for the case in which $*$ is exclusive-or.

We make use of all these previous results to study the decompositions of AGH and show that none of them meet our needs. We give the equations for AGH below (ignoring equations needed to complete the theory for coherence):

$$\begin{aligned}
(x * y) * z &= x * (y * z) && (1)\\
x * y &= y * x && (2)\\
x * 1 &= x && (3)\\
x * (x)^{-1} &= 1 && (4)\\
e(1) &= 1 && (5)\\
e(x * y) &= e(x) * e(y) && (6)
\end{aligned}$$

We note that although unification modulo AGH itself is finitary, this does not help us for variant-based unification, since AGH is not regular. Indeed, because of the need for regularity of B (without it a convergent decomposition becomes practically impossible), Eq. 4 must be in R for any decomposition (Σ, B, R). Moreover, because commutativity can not be written as a rewrite rule, and because unification modulo associativity without commutativity is not finitary, Eqs. 1 and 2 must be in B.

We also know that if Eq. 6 is in R, then the decomposition will fail to have the finite variant property, as we have noted earlier. Thus the only choices left for B are $B_1 = \{1, 2, 3, 5, 6\}$, $B_2 = \{1, 2, 6\}$, $B_3 = \{1, 2, 5, 6\}$, and $B_4 = \{1, 2, 3, 6\}$. Unification for B_1 and B_2 is known to be undecidable [34]. The theories B_3 and B_4 have not been as well studied, but we note that the problems $e(x) =^? x$ and $x * x =^? x$ have nonfinitary solutions for B_3 and B_4 respectively. For the former, the set of mgu's is $\{x \to 1, x \to 1 * 1, \ldots\}$; for the latter it is $\{x \to e(1), x \to e(e(1)), \ldots\}$.

4. FVP Theories of Homomorphic Encryption

In this section we present all the theories mentioned in Figure 1. Theories kH and kHD are presented in Section 4.1. Theories $H_{\&}$ and $HD_{\&}$ are presented in Section 4.2.1. Theories $PGH_{\&}$, $PGHD_{\&}$, $PGAAH_{\&}$ and $PGAAHD_{\&}$ are presented in Section 4.2.2. Theories $APGH_{\&}$, $APGHD_{\&}$, $APGAAH_{\&}$ and $APGAAHD_{\&}$ are presented in Section 4.2.3. Theories 2AGH and 2AGHD are presented in Section 4.3.1, and 2XORH in Section 4.3.2 . Since these theories **all have FVP**, they could often be combined with other theories, which is another great benefit of our approach. All the theories are based on the following homomorphic theory.

Definition 1. The homomorphic theory is defined as $T_H = (\Sigma_H, \emptyset, R_H)$. The signature Σ_H is defined by sorts $\{\mathsf{Key}, \mathsf{Msg}\}$ with the subsort relation $\mathsf{Key} < \mathsf{Msg}$. The set R_H of rules contains only the following rule, which is a variation of Equation (6), where X, Y are of sort Msg and K is of sort Key:

$$e(X * Y, K) \quad \to \quad e(X, K) * e(Y, K) \qquad (7)$$

We also note that, given an order of the rules R of a decomposition (Σ, B, R), there is a fixed minimal set of variants that can be determined. We use this to introduce the notion of *variant complexity*, which can be used as a rough metric of the complexity inherent in using a particular theory.

Definition 2. For $E = (\Sigma, B, R)$, and a total order $<$ on the rules in R, the *variant complexity*, written $vc(E, <)$, provides the number of variants generated for all the terms, i.e., $vc(E) = \sum_{f \in \Sigma} v(f)$, where $v(f)$ is the cardinality of the complete and finite set of variants of $f(v_1, \ldots, v_n)$ where $v_1, \ldots, v_n$ are distinct variables.

We will often refer to $vc(E)$ instead of $vc(E, <)$ when using variant complexity as a metric.

Example 1. Consider the theory with sorts $\{Msg, Key\}$, operators $e : Msg \times Key \to Msg$, $d : Msg \times Key \to Msg$ and rule $d(e(X, K), K) = X$, where X is of sort Msg, and K is of sort Key. The variant complexity of this theory is 3, since the term $e(X, K)$ has 1 variant: $(e(X, K), \iota)$ and the term $d(X, K)$ has 2 variants: $(d(X, K), \iota)$ and $(X_1, \{X \to e(X_1, K_1), K \to K_1\})$, with X, X_1 of sort Msg and K, K_1 of sort Key.

4.1 Theory of Bounded Homomorphism

First, we define the equational theory for bounded homomorphism by using a new sort $\mathsf{SingleMsg}$, which is a subsort of Msg. The theory T_{kH} obtained is convergent and have the FVP. Indeed, $vc(T_{kH}) = 4$.

Definition 3. The bounded homomorphic theory is defined as $T_{kH} = (\Sigma_{kH}, \emptyset, \widehat{R_H}^k)$ for k the bound. The signature Σ_{kH} is defined by adding sort $\mathsf{SingleMsg}$ to the previous sorts Msg and Key with the subsort relation $\mathsf{Key} < \mathsf{SingleMsg} < \mathsf{Msg}$. The signature Σ_{kH} contains an overloaded definition of the encryption operator, i.e., $e : \mathsf{SingleMsg} \times \mathsf{Key} \to \mathsf{SingleMsg}$ and $e : \mathsf{Msg} \times \mathsf{Key} \to \mathsf{Msg}$.

The equation (7) of R_H is replaced in $\widehat{R_H}^k$ by the following rules: for all $1 < j \leq k$, add $e(S_1 * \cdots * S_j, K) \to e(S_1, K) * \cdots * e(S_j, K)$ to $\widehat{R_H}^k$, where K is a variable of sort Key, and $S_1, \ldots, S_k$ are variables of sort $\mathsf{SingleMsg}$. For $k = 3$, the set of equations in $\widehat{R_H}^3$ is:

$$e(S_1 * S_2, K) \to e(S_1, K) * e(S_2, K)$$
$$e(S_1 * S_2 * S_3, K) \to e(S_1, K) * e(S_2, K) * e(S_3, K)$$

Example 2. Consider the following unification problem:

$$X{:}\mathsf{Msg} * Y{:}\mathsf{Msg} =^?$$
$$e(data(A{:}\mathsf{Name}, r'{:}\mathsf{Fresh}) * data(B{:}\mathsf{Name}, r'{:}\mathsf{Fresh}), K{:}\mathsf{Key})$$

in the theory T_{3H}, where $data : \mathsf{Name} \times \mathsf{Fresh} \to \mathsf{Data}$ is an additional operator that generates principle's secrets, and we assume the subsort relation $\mathsf{Data} < \mathsf{SingleMsg}$. We get the following unifier by variant based unification:

[1] Note: in this theory the operation $+$ corresponds to our use of $*$ and $*$ corresponds to our use of e.

$$\{X \mapsto e(data(A1{:}\mathsf{Name}, r1{:}\mathsf{Fresh}), K1{:}\mathsf{Key}),$$
$$Y \mapsto e(data(B1{:}\mathsf{Name}, r1{:}\mathsf{Fresh}), K1{:}\mathsf{Key}),$$
$$A{:}\mathsf{Name} \mapsto A1{:}\mathsf{Name},$$
$$r'{:}\mathsf{Fresh} \mapsto r1{:}\mathsf{Fresh},$$
$$B{:}\mathsf{Name} \mapsto B1{:}\mathsf{Name},$$
$$K : \mathsf{Key} \mapsto K1{:}\mathsf{Key}\}$$

We can extend the theory T_{kH} by adding a decryption operator and the equations capturing the encryption/decryption cancellation properties.

Definition 4. The bounded homomorphic theory with decryption is defined as $T_{kHD} = (\Sigma_H \cup \Sigma_{kH} \cup \Sigma_{Dec}, \emptyset, \widehat{R_H}^k \cup R_{Dec} \cup R_{kH\text{-}Dec})$. Σ_{Dec} contains the overloaded decryption operator $d : \mathsf{Msg} \times \mathsf{Key} \to \mathsf{Msg}$, $d : \mathsf{SingleMsg} \times \mathsf{Key} \to \mathsf{SingleMsg}$. The encryption/decryption cancellation properties are captured by the equations R_{Dec}, where X is of sort Msg and K is of sort Key:

$$e(d(X, K), K) \to X \qquad d(e(X, K), K) \to X$$

Equations $R_{kH\text{-}Dec}$ capture the homomorphic property of the decryption operation, and are added for the theory to be convergent. The equations for bound $k = 3$ are as follows:

$$d(S_1 * S_2, K) \to d(S_1, K) * d(S_2, K)$$
$$d(S_1 * S_2 * S_3, K) \to d(S_1, K) * d(S_2, K) * d(S_3, K)$$

4.2 Homomorphic Theories: Theory of Homomorphic Encryption with a Multiset of Keys

Since neither AGH nor H have the FVP, we have extended H with a new presentation of $e(M,\ K)$ built on top of a new symbol $\_\&\_$, which is associative and commutative (AC), and keeps all the keys used for homomorphic encryption in a multiset. Since nested encryptions with the same subset of keys can be flattened using AC, this theory admits the FVP. But one side effect is that the order of applications of encryption to a message become immaterial. This is not a standard property of encryption, homomorphic or not, so in most cases, the multiset AC axioms are over approximations that are used to get the finite variant property. The soundness (i.e., any attacks found are real attacks) is lost since the over approximation may introduce spurious attacks, but they can be discarded upon inspection. Also, we note that there are a few cases, such as Distributed ElGamal [11], in which encryption does satisfy this multiset condition, so this may be useful for reasoning about an additional class of crypto-algorithms as well.

In this section, we first introduce the theory $T_{H_\&}$, which is consists of homomorphic encryption over a free operator and using a multiset of keys. This theory is convergent and has the FVP. Indeed, $vc(T_{H_\&}) = 8$.

$T_{H_\&}$ is then enriched with group operators and axioms. Since the FVP is lost again when adding the full group axioms due to the fact that confluence required an infinite number of extra rules, associativity is approximated by a sub-theory. We call a group without associativity a *pre-group*. The theory $T_{PGH_\&}$, which is obtained by extending the $T_{H_\&}$ to homomorphic encryption over a pre-group operator is convergent and has the FVP. Indeed, $vc(T_{PGH_\&}) = 20$.

$T_{H_\&}$ is then further enriched with Abelian group operators and axioms. Again, associativity is approximated. The theory $T_{APGH_\&}$, which is obtained by extending the $T_{H_\&}$ to homomorphic encryption over an Abelian pre-group operator is convergent and has the FVP. Indeed, the variant complexity is $vc(T_{APGH_\&}) = 20$.

4.2.1 Theory of Homomorphic Encryption over a Free Operator

Definition 5. The theory for homomorphic encryption with a multiset of keys is defined as $T_{H_\&} = (\Sigma_{H_\&}, B_{H_\&}, R_{H_\&})$. The signature $\Sigma_{H_\&}$ is defined by sorts $\{\mathsf{Key},\ \mathsf{Keys},\ \mathsf{Msg}\}$ with the subsort relation $\mathsf{Key} < \mathsf{Keys}$, $\mathsf{Keys} < \mathsf{Msg}$, and operators $\_*\_ : \mathsf{Msg} \times \mathsf{Msg} \to \mathsf{Msg}$, $\_\&\_ : \mathsf{Keys} \times \mathsf{Keys} \to \mathsf{Keys}$, and $e : \mathsf{Msg} \times \mathsf{Keys} \to \mathsf{Msg}$, where $\_ * \_$ is a free operator, which can be understood as list concatenation operator, $\_\&\_$ denotes a multiset union operator that is associative and commutative, and e denotes message encryption. The axioms $B_{H_\&}$ are AC for $\_\&\_$. There are five equations defined in $R_{H_\&}$.

$$e(X, U) * e(Y, U) \to e(X * Y, U) \tag{8}$$
$$e(e(X, V), U) \to e(X, U\&V) \tag{9}$$
$$e(X, U\&V) * e(Y, U) \to e(e(X, V) * Y, U) \tag{10}$$
$$e(X, U) * e(Y, U\&V) \to e(X * e(Y, V), U) \tag{11}$$
$$e(X, U\&V) * e(Y, U\&W) \to e(e(X, V) * e(Y, W), U) \tag{12}$$

where X, Y are variables of sort Msg, and U, V, W are variables of sort Keys. The equation (8) is the reversed version of (7). The equation (9) captures the property that a nested encryption is simplified into an encryption with a multiset of keys. The remaining equations describe the homomorphic property of an encryption with respect to a multiset of keys, and are added for confluence of the theory.

The unification problem of Example 2 returns the same unifiers with this new theory $T_{H_\&}$ as with the previous theory T_{kH}.

The extension of $T_{H_\&}$ with a decryption operator is denoted $T_{HD_\&}$. The theory obtained is convergent and has the FVP ($vc(T_{HD_\&}) = 13$).

Definition 6. The theory $T_{HD_\&}$ is defined by extending $T_{H_\&}$ with a decryption operator $d : \mathsf{Msg} \times \mathsf{Keys} \to \mathsf{Msg}$ together with additional equations $R_{H_\&\text{-}Dec}$ which describe encryption/decryption cancellation properties with respect to a multiset of keys:

$$d(e(X, U), U) \to X$$
$$d(e(X, U\&V), U) \to e(X, V)$$
$$d(e(X, U), U\&W) \to d(X, W)$$
$$d(e(X, U\&V), U\&W) \to d(e(X, V), W)$$

with X of sort Msg, and U, V, W of sort Keys.

4.2.2 Theory of Homomorphic Encryption over a Pre-Group

In this section, we extend $T_{H_\&}$ to be an homomorphic encryption over a pre-group. For the reason that we mentioned before, associativity is under approximated by a sub-associativity theory.

Definition 7. The pre-group theory is defined as $T_{PG} = (\Sigma_{PG}, \emptyset, R_{PG})$. There is only one sort Msg, and three group operators $\{\_ * \_, \_^{-1}, 1\}$, where $\_ * \_ : \mathsf{Msg} \times \mathsf{Msg} \to \mathsf{Msg}$ is the group operator, $\_^{-1} : \mathsf{Msg} \to \mathsf{Msg}$ generates the inverse of an element and the constant 1 is the identity. The equations R_{PG} contains all group axioms except associativity, where X is a variable of sort Msg:

$$X * 1 \to X \qquad 1 * X \to X \qquad X * X^{-1} \to 1$$
$$X^{-1} * X \to 1 \qquad (X^{-1})^{-1} \to X \qquad 1^{-1} \to 1$$

Definition 8. The theory for homomorphic encryption with a multiset of keys over pre-group is defined as $T_{PGH_\&} = (\Sigma_{H_\&} \cup \Sigma_{PG}, B_{H_\&}, R_{PGH_\&})$, where $R_{PGH_\&} = R_{PG} \cup R_{H_\&} \cup R_{PGH_\&\text{-}Aux}$. The following set of equations $R_{PGH_\&\text{-}Aux}$ is added to complete the theory:

$$e(1, U) \to 1 \qquad (e(X, U))^{-1} \to e(X^{-1}, U)$$

with X of sort Msg and U of sort Keys.

Example 3. Recall the unification problem in Example 2, with an instance of theory $T_{PGH_{\&}}$, we get all the unifiers in Example 1 together with the following unifiers:

$$\{X \mapsto e(data(A1, r1) * data(B1, r1), K1),$$
$$Y \mapsto 1, A \mapsto A1, r' \mapsto r1, B \mapsto B1, K \mapsto K1\}$$

$$\{X \mapsto 1, Y \mapsto e(data(A1, r1) * data(B1, r1), K1),$$
$$A \mapsto A1, r' \mapsto r1, B \mapsto B1, K \mapsto K1\}$$

Adding associativity approximation to T_{PG}. Although the theory T_{PG} defines a group without associativity, we can provide several sound approximations of associativity by adding a subtheory of the full associativity theory. This is an under approximation and the completeness (i.e., if there is an attack, an attack will be found) is lost since there may be attacks that can show up with full associativity theory but cannot be found with a sub-associativity theory. Here we introduce one of the possible sub-associativity theories as an example to illustrate this approach. The associativity approximation below captures the property that a term of sort Nonce can be canceled by its inverse when they are in the two separate ends of a sequence of terms of sort Msg.

Definition 9. The theory for a pre-group with associativity approximation T_{PGAA} is defined as $T_{PGAA} = (\Sigma_{PG} \cup \{\mathsf{Nonce}\}, \emptyset, R_{PG} \cup R_{PGAA})$. We assume the subsort relation $\mathsf{Nonce} < \mathsf{Msg}$. The sub-associativity axioms R_{PGAA} are specified as follows, where X is a variable of sort Msg and N is a variable of sort Nonce:

$$(N^{-1} * X) * N \to X \qquad N * (X * N^{-1}) \to X$$
$$(N * X) * N^{-1} \to X \qquad N^{-1} * (X * N) \to X$$

Remark 1. Notice that, in order to get a sound sub-associativity approximation, it is crucial to take advantage of the order-sorted type structure.

We can also add associativity approximation to the theory of homomorphic encryption over a pre-group by combining $T_{PGH_{\&}}$ with T_{PGAA}.

Definition 10. Combining the theories $T_{PGH_{\&}}$ and T_{PGAA}, we obtained the theory $T_{PGAAH_{\&}} = T_{PGH_{\&}} \cup T_{PGAA}$.

We can also extend $T_{PGH_{\&}}$ by adding a decryption operator, and/or associativity approximations, keeping FVP.

Definition 11. The theory $T_{PGHD_{\&}}$ is defined by extending the theory $T_{PGH_{\&}}$ with the decryption operator $d : \mathsf{Msg} \times \mathsf{Keys} \to \mathsf{Msg}$, together with the equation $R_{H_{\&}\text{-}Dec}$ introduced in Definition 5, and the auxiliary equation $d(1, U) = 1$.

Definition 12. Adding the same associativity approximation to $T_{PGHD_{\&}}$, we obtain the theory $T_{PGAAHD_{\&}} = T_{PGHD_{\&}} \cup T_{PGAA}$.

4.2.3 Theory of Homomorphic Encryption over an Abelian Pre-Group

We further approximate the theory AGH by adding a commutative axiom to the binary group operator in the pre-group. This provides a theory for homomorphic encryption over an Abelian pre-group.

Definition 13. The theory of Abelian pre-groups $T_{APG} = (\Sigma_{APG}, B_{APG}, R_{APG})$ is obtained from the theory of pre-group T_{PG} by just adding as axioms B_{APG} the commutativity equation $X * Y = Y * X$. Because of commutativity, the rules $1 * X \to X$ and $X * X^{-1} \to 1$ in T_{PG} become redundant.

We can similarly provide a sound approximation of associativity by adding a subtheory of the full associativity theory. Here again, we introduce an Abelian pre-group with sub-associativity theory as an example to illustrate this approach.

Definition 14. The theory for an Abelian pre-group with associativity approximation T_{APGAA} is defined as $T_{APGAA} = (\Sigma_{APG} \cup \{\mathsf{Nonce}\}, B_{APG}, R_{APG} \cup R_{APGAA})$. We assume the subsort relation $\mathsf{Nonce} < \mathsf{Msg}$. The sub-associativity rules R_{APGAA} are specified as follows:

$$(N^{-1} * X) * N \to X \qquad (N * X) * N^{-1} \to X$$

with X of sort Msg and N of sort Nonce.

Definition 15. The theory of homomorphic encryption with a multiset of keys over an Abelian pre-group is defined as $T_{APGH_{\&}} = (\Sigma_{H_{\&}} \cup \Sigma_{APG}, B_{APGH_{\&}}, R_{PGH_{\&}} \cup R_{PGH_{\&}\text{-}Aux})$. The axioms $B_{APGH_{\&}}$ define the commutativity property of $\_ * \_$ and AC of $\_\&\_$.

Example 4. For the unification problem in Example 2, with an instance of the theory $T_{APGH_{\&}}$, we find the unifiers described in Example 2, together with the following unifier:

$$\{X \mapsto e(data(B1{:}\mathsf{Name}, r1{:}\mathsf{Fresh}), K1{:}\mathsf{Key}),$$
$$Y \mapsto e(data(A1{:}\mathsf{Name}, r1{:}\mathsf{Fresh}), K1{:}\mathsf{Key}),$$
$$A{:}\mathsf{Name} \mapsto A1{:}\mathsf{Name}, r'{:}\mathsf{Fresh} \mapsto r1{:}\mathsf{Fresh},$$
$$B{:}\mathsf{Name} \mapsto B1{:}\mathsf{Name}, K : \mathsf{Key} \mapsto K1{:}\mathsf{Key}\}$$

We can extend $T_{APGH_{\&}}$ by adding a decryption operator, and associativity approximations. Both extensions have the FVP.

Definition 16. $T_{APGHD_{\&}}$ is obtained by adding decryption operator $d : \mathsf{Msg} \times \mathsf{Keys} \to \mathsf{Msg}$ to the signature of $T_{APGH_{\&}}$, together with the set of encryption/decryption cancellation equations $R_{H_{\&}\text{-}Dec}$ in Definition 5, and together with the auxiliary equation $d(1, U) = 1$.

We can similarly add associativity approximation to the theory of homomorphic encryption with a multiset of keys over an Abelian pre-group. By Combining the theories $T_{APGH_{\&}}$ with T_{APGAA}, we get the theory of homomorphic encryption with a multiset of keys over Abelian pre-group with associativity approximation, which is defined as: $T_{APGAAH_{\&}} = T_{APGH_{\&}} \cup T_{APGAA}$.

Adding the same associativity approximation to $T_{APGHD_{\&}}$, we obtain the theory $T_{APGAAHD_{\&}} = T_{APGHD_{\&}} \cup T_{APGAA}$.

4.3 Homomorphic Theories: Theory of Homomorphic Encryption over Two Groups

In this section, we introduce theories of homomorphic encryption over two different groups. Since this is essentially a function mapping from one group to another, it does not have recursive calls, allowing the theories to have the FVP.

We first introduce the theory T_{2AGH}, which defines homomorphic encryption over two Abelian groups. This theory is defined based on the decomposition of an Abelian group by Lankford [22], which was proved to have FVP by [15, 21]. The theory T_{AG} has the FVP. Indeed, $vc(T_{2AGH}) = 2276$.

Notice that the variant complexity of the theory T_{2AGH} is really high. To achieve a lower variant complexity, we introduce the theory T_{2XORH}, which defines a homomorphic encryption over two Exclusive-or(Xor) theories. This is an over approximation of homomorphic encryption over two Abelian groups. The variant complexity of this theory is much lower, indeed, $vc(T_{2XORH}) = 48$.

4.3.1 Theory of Homomorphic Encryption over Two Abelian Groups

Definition 17. The theory of Abelian groups is defined as $T_{AG} = (\Sigma_{AG}, B_{AG}, R_{AG})$. The signature Σ_{AG} is defined by sort AG and the set of Abelian-group operators

$$\_ * \_ : AG \times AG \to AG \qquad \_^{-1} : AG \to AG \qquad 1 :\to AG$$

The axioms B_{AG} are associativity and commutativity axioms for $\_ * \_$.The set of rules R_{AG} define all other Abelian group axioms:

$$X * 1 \to X \qquad X^{-1} * Y^{-1} \to (X * Y)^{-1}$$
$$X * (X^{-1}) \to 1 \qquad (X * Y)^{-1} * Y \to (X)^{-1}$$
$$(X^{-1})^{-1} \to X \qquad (X^{-1} * Y)^{-1} \to X * (Y^{-1})$$
$$1^{-1} \to 1 \qquad X^{-1} * (Y^{-1} * Z) \to (X * Y)^{-1} * Z$$
$$X * (X^{-1} * Y) \to Y \qquad (X * Y)^{-1} * (Y * Z) \to X^{-1} * Z$$

with X, Y, Z of sort AG.

Definition 18. The theory of homomorphic encryption over two Abelian groups is defined as $T_{2AGH} = (\Sigma_{2AGH}, B_{2AGH}, R_{2AGH})$. The signature Σ_{2AGH} is defined by sorts $\{\mathsf{AG_a}, \mathsf{AG_b}\}$, and the homomorphic encryption operator $e : \mathsf{AG_a} \to \mathsf{AG_b}$, together with the set of group operators for the domain Abelian group $\mathsf{AG_a}$, which are $\{*_a, 1_a, \_^{-1_a}\}$, and the set of group operators for codomain Abelian group $\mathsf{AG_b}$, which are $\{*_b, 1_b, \_^{-1_b}\}$. Notice that there is no key explicitly defined here, since encryption with a specific key is implicitly captured by the definition of the encryption operator e. The axioms B_{2AGH} are AC axioms for the binary Abelian group operators $\_ *_a \_, \_ *_b \_$. The set of equations R_{2AGH} defines the remaining Abelian group axioms of AG_a and AG_b, which are instances of R_{AG} in T_{AG} together with the homomorphic property of e:

$$e(X) *_b e(Y) \to e(X *_a Y)$$
$$e(X) *_b e(Y) *_b Z \to e(X *_a Y) *_b Z$$

The following equations are added to complete the theory.

$$e(1_a) \to 1_b$$
$$e(X)^{-1_a} \to e((X)^{-1_b})$$
$$(e(X) *_b Z)^{-1_b} \to e(X^{-1_a}) *_b (Z^{-1_b})$$

with X, Y of sort $\mathsf{AG_a}$, and Z of sort $\mathsf{AG_b}$

We can also extend T_{2AGH} by adding decryption operator in the following way:

Definition 19. T_{2AGHD} is obtained by adding a decryption operator $d : \mathsf{AG_b} \to \mathsf{AG_a}$ to the signature of T_{2AGH}, together with adding the equations $d(1_b) = 1_a$ and $d(e(X)) = X$ with X of sort $\mathsf{AG_a}$.

4.3.2 Theory of Homomorphic Encryption over Two Xor Operators

Definition 20. The theory for homomorphic encryption over two Xor operators is defined as $T_{2XORH} = (\Sigma_{2XORH}, B_{2XORH}, R_{2XORH})$. The signature Σ_{2XORH} is defined by sorts $\{\mathsf{Xor_a}, \mathsf{Xor_b}\}$, and the homomorphic encryption operator $e : \mathsf{Xor_a} \to \mathsf{Xor_b}$, together with the Xor operator and the corresponding identity of $\mathsf{Xor_a}$, which are $\{*_a, 1_a\}$, and together with the Xor operator and the corresponding identity of $\mathsf{Xor_b}$, which are $\{*_b, 1_b\}$. Notice again that there is no key explicitly defined here. The axioms B_{2XORH} are associativity and commutativity axioms for $\_ *_a \_, \_ *_b \_$. The set of equations R_{2XORH} are defined as following:

$$X *_a X \to 1_a \qquad X *_a X *_a Y \to Y \qquad X *_a 1_a \to X$$
$$P *_b P \to 1_b \qquad P *_b P *_b Q \to Q \qquad P *_b 1_b \to P$$

together with the homomorphic property of e:

$$e(1_a) \to 1_b$$
$$e(X) *_b e(Y) \to e(X *_a Y)$$
$$e(X) *_b e(Y) *_b P \to e(X *_a Y) *_b P$$

with X, Y of sort $\mathsf{Xor_a}$, and P, Q of sort $\mathsf{Xor_b}$

We can also extend T_{2XORH} by adding a decryption operator in the similar way as in T_{2AGHD}

5. Experiments

In this section we describe the experiments [2] we have performed on various of the Multiparty Computation Protocol that we had specified and analyzed in [17]. In that paper we analyzed a number of protocols that were specified using a dedicated unification algorithm for equational theory H. That is, Equation (7) was used and nothing else. This was necessary because the special-purpose unification algorithm was not easily combinable with other theories. However, since in this paper we are using variant unification, we have much more freedom with respect to the equations we can include, as long as the theories satisfy FVP.

The experiments in this section serve several purposes. The first is to determine which of the theories we have generated are suitable for cryptographic protocol analysis. The second is to evaluate the variant complexity metric defined in this paper. How well does lower variant complexity correlate with performance, and if it does, at what point does higher variant complexity begin to make analysis impossible? The third is to use the experimental results to gain insights into how performance can be improved.

The protocol was specified and analyzed using the Maude-NPA cryptographic protocol analysis tool. One uses Maude-NPA by specifying an insecure state, called an *attack state*, from which Maude-NPA searches backwards. If it finds a path to an initial state then it has found an attack on the protocol. If it terminates without reaching an initial state then the attack state has been proven unreachable.

In the Multiparty Computation Protocol an initiator Alice and a responder Bob send messages encrypted with a homomorphic public key encryption operator e to a server, who combines the encrypted data using an operator $*$. As a result of participating in the protocol, both Alice and Bob are supposed to receive a homomorphically encrypted version of $D_A * D_B$, where D_A is Alice's secret data and D_B is Bob's secret data, without either learning the other's secret. However, it is possible for Alice to accept data that did not come from Bob if she is not able to distinguish $D_A * D_B$ from nonsense. If she is able to, no authentication attack is possible. We specified two attack states: one in which Alice cannot reject nonsense, and one in which she can.

The protocol itself proceeds as follows:

1. $A \to B : sign(B; N_A; pk(e(D_A, pkey(A, B)), S), A)$

   A starts by encrypting her data first under the homomorphic public key, then under the server's public key. She then attaches a nonce and B's name, signs it, and sends it to B.

2. $B \to A : sign(N_A; N_B; pk(e(D_B, pkey(A, B)), S), B)$

   B sends a similar message to A, including both his and A's nonce.

3. $A \to S : sign(A; B; N_A; N_B; pk(e(D_A, pkey(A, B)), S);$
   $pk(e(D_B, pkey(A, B)), S), A)$

   A sends a signed message containing both nonces and both encrypted data sets to S.

4. $S \to A, B : sign(A; B; N_A; N_B;$
   $sign(e(D_A, pkey(A, B)) *$
   $e(D_B, pkey(A, B)), S)$

   The server combines both encrypted data sets using $*$ and sends the result to A and B. They can now decrypt it to obtain $D_A * D_B$.

[2] Available at `http://formal.cs.illinois.edu/fanyang/homomorphism/`

The attack runs as follows:

1. $A \rightarrow I(B) : sign(B; N_A; pk(e(D_A, pkey(A, B)), S), A)$
   A initiates the protocol with B.
2. $I \rightarrow B : sign(B; N_A; E, I)$
   I intercept's A's message, and uses it to create a message for B. The message E could or could not be A's encrypted data. This is irrelevant to the attack.
3. $B \rightarrow A : sign(N_A; N_B; pk(e(D_B, pkey(I, B)), S), B)$
   B believes that he is talking to I and sends the corresponding reply message. I forwards it to A.
4. $A \rightarrow S : sign(A; B; N_A; N_B;$
   $pk(e(D_A, pkey(A, B)), S);$
   $pk(e(D_B, pkey(I, B)), S), A)$
   A now forwards both encrypted data sets to the server S, who removes the outer layer of encryption, applies f, and sends the results back to A and B.
5. $S \rightarrow A, B : sign(A; B; N_A; NB;$
   $sign(e(D_A, pkey(A, B)) *$
   $e(D_B, pkey(I, B)), S)$
   If A now attempts to decrypt the result of S's computation with her private key corresponding to $pkey(A, B)$, she will get nonsense, because one of the data sets was encrypted with $pkey(I, B)$. Depending upon whether or not A can recognize that she has received nonsense, this can be used to prevent this attack.

We thus specify two versions of this protocol : one in which A verifies that she has received $e(f(X, Y), pkey(A, B))$ for some X and Y, and one in which she does not.

If, in addition, we assume that $*$ is an Abelian group operator, there are several attacks in which Bob can learn Alice's secret (and vice versa). In the first, Bob simply sends the unit 0 as his data and receives $D_A * 1 = D_A$ in return. In the second, Bob sends his correct data and receives $D_A * D_B$, and multiplies by $(D_B)^{-1}$ to obtain D_A. These attacks, although simple, were of interest to us because they follow from the Abelian group properties of $*$, and so we ran Maude-NPA on an attack state in which Bob learns Alice's secret, using the homormorphic encryption over an Abelian pre-group with associativity approximation. To demonstrate the associativity approximation's influence on performance, we investigated two such theories: $T_{APGAAH1_{\&}}$ is the theory $T_{APGH_{\&}}$ with the approximation equation: $(D_1 * D_2) * (D_1)^{-1} = D_2$. $T_{APGAAH2_{\&}}$ is the theory $T_{APGH_{\&}}$ with the approximation equation: $(D_1 * X) * (D_1)^{-1} = X$ with X of sort Msg, and D_1 and D_2 of sort Data, which is a subsort of Msg. Notice that we start from an associativity approximation that is just expressive enough for this specific protocol and then try a more expressive one.

We tried the authentication attack on the protocol specified with the theories T_{kH} (bounded homomorphism, variant complexity 4), $T_{H_{\&}}$ (multi set of keys with free operator, variant complexity 8), and $T_{APGH_{\&}}$ (Abelian pre-group,variant complexity 20). The results are given in Table 2 for the insecure version of the protocol and Table 3 for the second version of the protocol. We note that all the theories we obtained can be used for analyzing this attack, since the higher the variant complexity, the longer the analysis time, we thus consider only three equational theories. For the secrecy attack we investigated $T_{APGAAH1_{\&}}$ and $T_{APGAAH2_{\&}}$ as well the theories T_{2AGH} (homomorphism between two Abelian groups, variant complexity 2276) and T_{2XOR} (homomomorphism between two exclusive-or theories, variant complexity 48), since Abelian group axioms are necessary for the secrecy attack. The results for $T_{APGAAH1_{\&}}$ and $T_{APGAAH2_{\&}}$ are given in Table 4.

Tables 2 and 3 show the number of states generated by Maude-NPA and the amount of time taken in each step of the backwards reachability analysis. In Table 2, for each theory Maude-NPA found the authentication attack in six steps. However, in each case Maude-NPA failed to terminate and kept generating five states after fifteen steps. Upon investigation, these appeared to be infinite paths that were not discarded by the Maude-NPA state space reduction techniques. As we can also see from the table, as the variant complexity of theories involved in the specification grows, the number of states and the time needed for Maude-NPA to find the attack also grows, as well as the time it takes to complete each step. Furthermore, the time it takes to complete a step increases with the variant complexity of the theory even when the number of states generated at the step is the same for all three theories. We conjecture that this is the result of Maude-NPA generating many states that are then removed by the state space reduction mechanisms. Greater variant complexity means that more failed states are generated as well as successful ones.

For the second attack analysis, when Alice can tell whether she received nonsense or not, we verified that there is no authentication attack between Alice and Bob. Table 3 shows the number of states and attacks generated by Maude-NPA in each step for the attack state with different theories described above. For each theory Maude-NPA terminated at Step 4. We note a similar relationship between between performance and variant complexity as in Table 2.

For the secrecy attack we found that the theories T_{2AGH} and T_{2XOR} gave very discouraging results. For Theory T_{2AGH} Maude-NPA was not even able to complete Step 1, even for a simpler version of the protocol we constructed (see below). For theory T_{2XOR} Maude-NPA did a little better; it was able to complete Steps 1 and 2, but not Step 3. This is not surprising, given the high variant complexity of the theories. We did not investigate these two theories any further.

Even with $T_{APGAAH1_{\&}}$ and $T_{APGAAH2_{\&}}$, Maude-NPA struggled to find the secrecy attack, as we can see in Table 4 for $T_{APGAAH2_{\&}}$. We thus tried Maude-NPA on a simpler version of the protocol to get a better idea of the performance tradeoffs, omitting the checks for authentication and freshness. This simplification is intended to reduce search space while keeping the part of the protocol that is of interest to us. Maude-NPA was able to find the two attacks in five steps for the simplified protocols. The search of the protocol with $T_{APGAAH1_{\&}}$ terminated after 12 steps and 4 possible attack sequences were found, while the one with $T_{APGAAH2_{\&}}$ took a much longer time for each search step and suffered from state explosion. Even so, it was able to produce 5 attack sequences before the explosion.

Since the associativity approximation of $T_{APGAAH1_{\&}}$ is more restrictive than that of $T_{APGAAH2_{\&}}$ (indeed it is a special case of it), less variants are generated for the same term, which reduced the state generation time. This result thus shows the tradeoff needed between a more general theory and performance in practice.

6. Conclusions

The lack of FVP for H and AGH has made unification-based formal protocol analysis difficult to perform by extensible and generic methods such as variant-based unification. In this paper we have addressed this problem by studying a hierarchy of theories for homomorphic encryption that are all FVP. The existence of finitary unification algorithms for these theories seems to be a new result in the area of unification theory. We have also introduced variant complexity as a metric and shown how it affects performance. One important lesson learned from our experiments in using these theories for protocol analysis is that there is a tradeoff between theory accuracy (typically at the cost of higher variant complexity)

Steps	T_{kH}		$T_{H_{\&}}$		$T_{APGH_{\&}}$	
	States	Time(ms)	States	Time(ms)	States	Time(ms)
Step 1	5	15941	8	1114538	11	4885493
Step 2	6	34298	9	1557783	13	4800678
Step 3	1	36514	2	2194422	2	8633163
Step 4	1	3084	1	6174	1	38997
Step 5	3	2409	3	5287	3	28363
Step 6	6	8106	6	29910	6	45339
Step 7	4	19615	4	103886	4	416986
Step ...	...	...	...	...	...	...
Step 15	5	44835	5	187119	5	321147

Table 2. Results for authentication of Bob to Alice

Steps	T_{kH}		$T_{H_{\&}}$		$T_{APGH_{\&}}$	
	States	Time(ms)	States	Time(ms)	States	Time(ms)
Step 1	3	26183	4	1308574	8	3513537
Step 2	2	13251	3	1382475	6	4867452
Step 3	1	8489	1	2203056	1	8218407
Step 4	0	2974	0	4933	0	2775

Table 3. Results for authentication of Bob to a stronger Alice

Steps	$Simplified$				$Original$			
	$T_{APGAAH1_{\&}}$		$T_{APGAAH2_{\&}}$		$T_{APGAAH1_{\&}}$		$T_{APGAAH2_{\&}}$	
	States	Time	States	Time	States	Time	States	Time
Step 1	12	36965	12	66910	8	133571	8	250829
Step 2	14	38855	32	67577	11	7196213	17	8383711
Step 3	11	52049	72	323521	13	13728328	44	25724593
Step 4	7	34262	179	1486682	8	35712864	106	199325826
Step 5	10	21881	482	15140859	8	21233267	(timeout)	
Step 6	8	29695	(timeout)		11	30402427		
Step 7	8	18233			23	40922662		
Step 8	9	22508			35	62267212		
...	...	...			...	...		
Step 12	4	4534			(timeout)			

Table 4. Results for secrecy

and efficiency of the analysis process. Our hierarchy allows users to choose the right balance for their problem within this tradeoff.

The work also points out a number of avenues for future work. In particular, it demonstrates that better state space reduction techniques, while useful, are likely not to be adequate by themselves for addressing performance issues when dealing with theories of high variant complexity. That is because the techniques are only applied *after* a state is generated. This points out the need of techniques that can be applied earlier in the state generation process. We have also discovered that different approximations of equational properties via intruder strands can have widely different effects on the number of variants generated as well. Metrics that can be applied to these approximations are also desirable. We plan to address these issues in future work. We also plan to refine our techniques for generating and identifying theories having FVP, and to apply them to develop new classes of FVP theories, with and without homomorphic encryption. Finally, we plan to work on improving the performance of unification modulo the FVP theories we have identified, and integrating the unification implementation more closely with the Maude-NPA tool.

References

[1] Maude Formal Environment. `http://maude.lcc.uma.es/MFE/`.

[2] Siva Anantharaman, Hai Lin, Christopher Lynch, Paliath Narendran, and Michaël Rusinowitch. Cap unification: application to protocol security modulo homomorphic encryption. In Dengguo Feng, David A. Basin, and Peng Liu, editors, *ASIACCS*, pages 192–203. ACM, 2010.

[3] Myrto Arapinis, Sergiu Bursuc, and Mark Dermot Ryan. Reduction of equational theories for verification of trace equivalence: Re-encryption, associativity and commutativity. In Pierpaolo Degano and Joshua D. Guttman, editors, *POST*, volume 7215 of *Lecture Notes in Computer Science*, pages 169–188. Springer, 2012.

[4] Franz Baader. Unification in commutative theories, Hilbert's basis theorem, and Gröbner bases. *J. ACM*, 40(3):477–503, 1993.

[5] Franz Baader and Werner Nutt. Adding homomorphisms to commutative/monoidal theories or how algebra can help in equational unification. In Ronald V. Book, editor, *RTA*, volume 488 of *Lecture Notes in Computer Science*, pages 124–135. Springer, 1991.

[6] Franz Baader and Klaus U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. In *CADE*, volume 607 of *LNCS*, pages 50–65. Springer, 1992.

[7] David Basin, Sebastian Mödersheim, and Luca Viganò. An on-the-fly model-checker for security protocol analysis. In *Proceedings of Esorics'03, LNCS 2808*, pages 253–270. Springer-Verlag, 2003.

[8] Bruno Blanchet. An efficient cryptographic protocol verifier based on Prolog rules. In *CSFW*, pages 82–96. IEEE Computer Society, 2001.

[9] Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated verification of selected equivalences for security protocols. *J. Log. Algebr. Program.*, 75(1):3–51, 2008.

[10] Christopher Bouchard, Kimberly A. Gero, Christopher Lynch, and Paliath Narendran. On forward closure and the finite variant property. In Pascal Fontaine, Christophe Ringeissen, and Renate A. Schmidt, editors, *FroCos*, volume 8152 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2013.

[11] Felix Brandt. Efficient cryptographic protocol design based on distributed El Gamal encryption. In *ICISC*, pages 32–47, 2005.

[12] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

[13] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.

[14] Andrew Cholewa, Jose Meseguer, and Santiago Escobar. Variants of variants and the finite variant property. Technical report, University of Illinois at Urbana-Champaign, http://hdl.handle.net/2142/47117, 2014.

[15] Hubert Comon-Lundh and Stéphanie Delaune. The finite variant property: How to get rid of some algebraic properties. In Jürgen Giesl, editor, *Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2005.

[16] Serdar Erbatur, Santiago Escobar, Deepak Kapur, Zhiqiang Liu, Christopher Lynch, Catherine Meadows, José Meseguer, Paliath Narendran, Sonia Santiago, and Ralf Sasse. Asymmetric unification: A new unification paradigm for cryptographic protocol analysis. In Maria Paola Bonacina, editor, *CADE*, volume 7898 of *Lecture Notes in Computer Science*, pages 231–248. Springer, 2013.

[17] Santiago Escobar, Deepak Kapur, Christopher Lynch, Catherine Meadows, José Meseguer, Paliath Narendran, and Ralf Sasse. Protocol analysis in Maude-NPA using unification modulo homomorphic encryption. In Peter Schneider-Kamp and Michael Hanus, editors, *PPDP*, pages 65–76. ACM, 2011.

[18] Santiago Escobar, Catherine Meadows, and José Meseguer. A rewriting-based inference system for the NRL Protocol Analyzer and its meta-logical properties. *Theor. Comput. Sci.*, 367(1-2):162–202, 2006.

[19] Santiago Escobar, Catherine Meadows, and José Meseguer. Maude-NPA: Cryptographic protocol analysis modulo equational properties. In *Foundations of Security Analysis and Design V, FOSAD 2007/2008/2009 Tutorial Lectures*, LNCS vol. 5705, pages 1–50. Springer, 2009.

[20] Santiago Escobar, José Meseguer, and Ralf Sasse. Variant narrowing and equational unification. *Electr. Notes Theor. Comput. Sci.*, 238(3):103–119, 2009.

[21] Santiago Escobar, Ralf Sasse, and José Meseguer. Folding variant narrowing and optimal variant termination. *J. Log. Algebr. Program.*, 81(7-8):898–928, 2012.

[22] Jean-Marie Hullot. A catalogue of canonical term rewriting systems. Technical Report CSL-113, SRI International, 1980.

[23] Deepak Kapur, Paliath Narendran, and Lida Wang. An E-unification algorithm for analyzing protocols that use modular exponentiation. In Robert Nieuwenhuis, editor, *RTA*, volume 2706 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 2003.

[24] Deepak Kapur, Paliath Narendran, and Lida Wang. A unification algorithm for analysis of protocols with blinded signatures. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning*, volume 2605 of *Lecture Notes in Computer Science*, pages 433–451. Springer, 2005.

[25] Ralf Küsters and Tomasz Truderung. Using ProVerif to analyze protocols with Diffie-Hellman exponentiation. In *CSF*, pages 157–171. IEEE Computer Society, 2009.

[26] Ralf Küsters and Tomasz Truderung. Reducing protocol analysis with xor to the xor-free case in the horn theory based approach. *Journal of Automated Reasoning*, 46(3-4):325–352, 2011.

[27] Hai Lin. *Algorithms for cryptographic protocol verification in presence of algebraic properties*. PhD thesis, Clarkson University, 2009.

[28] Zhiqiang Liu. *Dealing Efficiently with Exclusive OR, Abelian Groups and Homomorphism in Cryptographic Protocol Analysis*. PhD thesis, Clarkson University, 2012.

[29] Andrew M. Marshall. *Equational Unification: Algorithms and Complexity with Applications to Cryptographic Protocol Analysis*. PhD thesis, Univ. at Albany–SUNY, Albany, NY, USA, 2012. AAI3512570.

[30] Andrew M. Marshall and Paliath Narendran. New algorithms for unification modulo one-sided distributivity and its variants. In Bernhard Gramlich, Dale Miller, and Uli Sattler, editors, *IJCAR*, volume 7364 of *Lecture Notes in Computer Science*, pages 408–422. Springer, 2012.

[31] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[32] J. Meseguer. Membership algebra as a logical framework for equational specification. In F. Parisi-Presicce, editor, *Proc. WADT'97*, pages 18–61. Springer LNCS 1376, 1998.

[33] Sebastian Mödersheim. *Models and methods for the automated analysis of security protocols*. PhD thesis, ETH Zurich, 2007.

[34] Paliath Narendran. Solving linear equations over polynomial semirings. In *LICS*, pages 466–472, 1996.

[35] Werner Nutt. Unification in monoidal theories. In Mark E. Stickel, editor, *CADE*, volume 449 of *Lecture Notes in Computer Science*, pages 618–632. Springer, 1990.

[36] H. O.. Plugfelder. *Quasigroups and Loops: Introduction*. Heideman, 1990.

[37] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphism. In R. DeMillo, R. Lipton, D. Dobkin, and A. Jones, editors, *Foundations of Security Computation*. ACM, 1978.

[38] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[39] Benedikt Schmidt, Simon Meier, Cas J. F. Cremers, and David A. Basin. Automated analysis of Diffie-Hellman protocols and advanced security properties. In *CSF*, pages 78–94, 2012.

[40] Manfred Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *J. Symb. Comput.*, 8(1/2):51–99, 1989.

[41] TeReSe, editor. *Term Rewriting Systems*. Cambridge University Press, Cambridge, 2003.

[42] Erik Tidén and Stefan Arnborg. Unification problems with one-sided distributivity. *J. Symb. Comput.*, 3(1/2):183–202, 1987.