Performance Evaluation of Privacy-Preserving Policy Reconciliation Protocols

Jonathan Voris, Sotiris Ioannidis, Susanne Wetzel
Department of Computer Science
Stevens Institute of Technology
Castle Point on Hudson
Hoboken, NJ 07030, USA
{jvoris,si,swetzel}@cs.stevens.edu

Ulrike Meyer
Siemens Networks GmbH & Co
Otto-Hahn-Ring 6
D-81730 München, Germany
meyer.ulrike@siemens.com

Abstract

The process of policy reconciliation allows multiple parties with possibly different policies to resolve differences in order to reach an agreement on an acceptable policy. Previous solutions for policy reconciliation required the participants to reveal their entire security policy in order to reach an agreement. It was not until recently that new protocols were developed which take into account the privacy concerns of reconciliating parties. In this paper we present a performance evaluation of these privacy-preserving reconciliation protocols with a focus on quantifying the added cost due to the privacy guarantees.

1 Introduction

Enabling interaction between different parties that do not necessarily trust one another but each have their own set of requirements can be very challenging. The parties must not only agree on a set of rules that will govern their interaction but do so in an efficient manner. This can be accomplished by each party first defining a set of security policy rules that control *what* data the party is willing to share and *how* the data can be shared. Individual policy rules must then be expressed in some common format. Finally, the parties must go though a *reconciliation protocol* that determines a security policy which is consistent with the security requirements of all parties and will control their future interactions [11]. In case no common security policy can be found, the parties can either modify the requirements and repeat the protocol or decide not to interact.

To illustrate the process of security policy reconciliation let us consider a simple example from the field of mobile communications where the users associate themselves with different network providers to obtain service while roaming from one network to another. In order to form an association, the user and the network provider have to agree on a common security policy that will govern their interaction. In the context of mobile communication, policies may include *attributes* such as type of authentication, type of connection, underlying security, *etc.*. In general, each party will have a wide range of policies that satisfy its individual communication requirements, typically ranked in some order of preference. Consequently, this may result in participants having conflicting preferences. For example, while a mobile user might prefer to maximize the battery life of his mobile device and therefore prefer not to use encryption whenever possible, the network provider, on the other hand, may want to use strong encryption whenever possible, but might still be willing to support certain weaker encryption mechanisms, or no encryption at all, so that it can provide service to older mobile devices (see Figure 1).

Using traditional policy reconciliation protocols, the wireless provider and the user in the above example will be able to form an association since they both have a wide range of overlapping and acceptable policies. Unfortunately, they will both have to reveal their individual security policies in the process. Such a revelation might leave them vulnerable to attacks from malicious parties. Specifically, in our example, an attacker may mount a series of attacks [12] by exploiting the knowledge that the wireless provider is willing to provide service without enforcing the use of "strong crypto".

Our simple example clearly illustrates why preserving privacy during policy reconciliation is crucial. However, it is important to note that in this simple setting parties may easily discover each other's policies as they only involve a single attribute. Yet, one can envision real-world scenarios requiring the reconciliation of policies that include a large number of attributes (types of users, types of data, protocols, *etc.*). Obviously, meeting the privacy objective of parties in these scenarios is of even greater importance.

Eighth IEEE International Workshop on
Policies for Distributed Systems and Networks (POLICY'07)
0-7695-2767-1/07 $20.00 © 2007 IEEE

	Wireless Provider	User
Requirement	Security, compatibility with customer	Battery life, compatibility with network
High preference	Strong Crypto	No Crypto
Medium preference	Medium Crypto	Medium Crypto
Low preference	No Crypto	Strong Crypto

Figure 1. Requirements for the wireless provider and the network user. The wireless provider wants to maintain a secure network but still provide compatibility and service to customers with older devices, while the user wants to achieve maximum battery life and still get service.

In recent work [13], we have introduced new protocols that enable the participants to reconcile their security policies while maintaining strong privacy guarantees. However, the benefit of privacy comes at the cost of additional communication and computation. In this paper we present an evaluation of the performance of our *preference-maximizing* privacy-preserving policy reconciliation protocols with a focus on quantifying the added cost due to the privacy guarantees.

Outline: In Section 2 we first provide the formal definitions for policy representations and preference orders. We then review our preference-maximizing privacy-preserving policy reconciliation protocols. In Section 3 we detail the implementation of the protocols and provide a performance evaluation. We close this paper with a review of related work followed by some concluding remarks and future research directions.

2 Policy Representation and Reconciliation Protocols

A prerequisite for successful policy reconciliation between organizations is to agree upon a common policy representation.[1] We assume here that policy rules are represented as bit-strings in which each bit represents a particular attribute. A policy as a whole can then be represented as a matrix in which the rows are the individual policy rules (Definition 2.1).

Continuing the example from Section 1, two participants in a policy reconciliation protocol first have to agree upon a set of attributes. In our example, three attributes are defined for the use of encryption, namely *3DES*, *DES*, and *None*. Each participant represents its policy rules for encryption in order of preference. In our example, the wireless provider prefers the use of *3DES* over *DES* but is willing to accommodate older devices, that do not support encryption at all. The user on the other hand wants to maximize battery life and therefore prefers no encryption over *DES* and would rather use *DES* than *3DES* although he supports these algorithms.

[1] The process of agreeing upon such a common representation is out of scope of this paper.

	Wireless Provider			User		
	3DES	*DES*	*None*	*3DES*	*DES*	*None*
Rule 1	1	0	0	0	0	1
Rule 2	0	1	0	0	1	0
Rule 3	0	0	1	1	0	0

Figure 2. Policy rules expressed as bit-strings and ranked in order of preference.

Definition 2.1 *A policy rule $a_i = (a_{i1}, \ldots, a_{in}) \in \{0,1\}^n$ is a bit-string of length n indicating whether a field a_{ij}, $j = 1, \ldots, n$ is defined or not. A policy P_A is a set of rules $a_1, \ldots, a_k$ represented as a matrix $P_A = (a_{ij})_{1 \le i \le k, 1 \le j \le n} \in \{0,1\}^{k \times n}$. Attribute $\mathcal{A}_j$ is a string of characters that uniquely represents the j-th fi eld of the policy rules.*

In our example from Figure 2, *3DES*, *DES* and *None* are the attributes defined in the security policy. Rule a_1 for the wireless provider is the bit-string $(1, 0, 0)$ specifying that he requires the use of *3DES*.

Remark 2.1 *Assuming parties A and B with policies P_A and P_B, attributes $\mathcal{A}_1, \ldots, \mathcal{A}_u$ and $\mathcal{B}_1, \ldots, \mathcal{B}_v$, policy reconciliation for these parties requires $u = v$ and $A_j = B_j$ for $j = 1, \ldots, u$.*

Definition 2.2 *A policy preference order of party A on its policy P_A is a total order $\le_A$ of its policy rules.*

Definition 2.3 *Let $\{a_1, \ldots, a_k\}$ be the set of policy rules of A in decreasing order of $\le_A$. The ranking of a_i corresponding to $\le_A$ is defined as $\text{rank}_A(a_i) := k - i + 1$ $(i = 1, \ldots, k)$.*

Definition 2.4 *A policy preference order composition scheme takes two policy preference orders $\le_A$ on a policy P_A and $\le_B$ on P_B as input and combines them to a joint (partial or total) order $\le_{AB}$ on $P_A \cap P_B$.*

Examples for preference order composition schemes are:

Definition 2.5 *The* **sum of ranks** *composition scheme combines $\le_A$ on P_A and $\le_B$ on P_B to the joint total preference order $\le_{AB}$ defined as follows: If $x, y \in P_A \cap P_B$, then*

$$x \le_{AB} y :\Leftrightarrow \text{rank}_A(x) + \text{rank}_B(x) \le \text{rank}_A(y) + \text{rank}_B(y).$$

Definition 2.6 *The* **maximized minimum of ranks** *composition scheme combines $\le_A$ on P_A and $\le_B$ on P_B to*

Eighth IEEE International Workshop on
Policies for Distributed Systems and Networks (POLICY'07)
0-7695-2767-1/07 $20.00 © 2007 IEEE

the joint total preference order $\leq_{AB}$ as follows: If $x, y \in P_A \cap P_B$, then

$$x \leq_{AB} y :\Leftrightarrow \min\{\mathit{rank}_A(x), \mathit{rank}_B(x)\} \leq \min\{\mathit{rank}_A(y), \mathit{rank}_B(y)\}.$$

Different composition schemes implement different notions of fairness.

Protocol. *A* **privacy-preserving preference-maximizing policy reconciliation** *scheme for a preference order composition scheme $\mathcal{C}$ (3PR$^{\mathcal{C}}$) is a multi-round two-party protocol between parties A and B. Their respective policies P_A and P_B are drawn from the same domain of policy rules (see Remark 2.1). Upon completion of the protocol, A and B learn nothing about each other's policies but one common policy rule max that maximizes the combined preference order $\leq_{AB}$ of $\leq_A$ and $\leq_B$ under $\mathcal{C}$ as well as in which round of the protocol max was determined. I.e., if 3PR$^{\mathcal{C}}$ takes P_A, P_B, $\leq_A$, and $\leq_B$ as input, then A and B learn* $\max_{x \in \leq_{AB}}\{x\}$.

In [13] we introduced two privacy-preserving preference maximizing protocols: one protocol for the maximized minimum composition scheme and one protocol for the sum of ranks composition scheme. The mathematical details and a thorough analysis of the security properties of these protocols can be found in [13]. Here, we only provide an intuitive description of the two protocols.

Our protocols are built on the following basic procedure. The two participants utilize a semantically secure homomorphic cryptosystem [15, 2]. The homomorphic encryption function allows a party A to encrypt a polynomial corresponding to one of its own policy rules in a way that the encrypted polynomial does not reveal any information to party B. In addition, if B evaluates the encrypted polynomial on one of its own policy rules, then its policy rule is blinded and encrypted. From this blinded ciphertext A learns nothing about B's policy rule, unless the policy rule A used for the creation of the polynomial and the policy rule on which B evaluated the polynomial coincide. This is because, if and only if the policy rules coincide, the blinded ciphertext decrypts A's policy rule. A and B can thus determine a common policy rule without revealing rules to each other that they do not have in common. Our protocols make use of this basic procedure in multiple rounds.

In the following figures, which describe our protocols in greater detail, we use the following notation: $E_A^{(i)}$ ($E_B^{(i)}$) denotes the encrypted polynomial computed by A (B) for its policy rule a_i (b_i). $c_{A,j}^{(i)}$ denotes the evaluation of $E_B^{(i)}$ at a_j leading to the blinded ciphertext. Equivalently, $c_{B,j}^{(i)}$ denotes the evaluation of $E_A^{(i)}$ at b_j.

The first protocol is illustrated in Figure 3. In each round, parties A and B exchange polynomials and blinded ciphertexts corresponding to previously received polynomials. They each decrypt the blinded ciphertext and compare the result to the policy rules corresponding to the previously sent polynomials. The order in which the information is exchanged guarantees that the protocol terminates when a common policy rule is found that maximizes the sum of ranks of the common policies of A and B. For example, in the first step of the protocol, A can check whether the two most preferred policy rules of both parties are the same ($a_1 \stackrel{?}{=} b_1$). If this is not the case, A does not learn anything about b_1. As a general rule, in Step i of the protocol, A and B compare all policies rules that result in the same sum of ranks in the combined preference order of A and B.

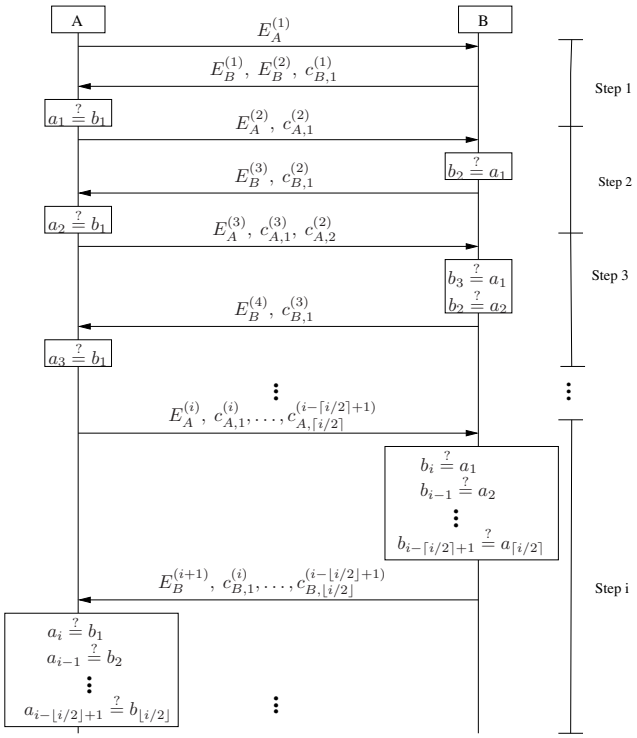

Figure 3. Commitment Phase of the sum of ranks 3PR$^{\mathcal{C}}$. In Step i, A for $j = 1, \ldots, \lfloor i/2 \rfloor$ compares a_{i-j+1} with b_j and B for $j = 1, \ldots, \lceil i/2 \rceil$ compares b_{i-j+1} with a_j without obtaining knowledge of these values (unless they are equal), that is, in a privacy-preserving manner. Note that in Step i all policy rules are compared that result in the same sum of ranks.

Figure 4 illustrates the privacy-persering policy reconciliaion scheme for the maximized minimum of ranks composition scheme. The difference between this and the previously described protocol is that the order in which A and B exchange information and compare policies is changed. As

Eighth IEEE International Workshop on
Policies for Distributed Systems and Networks (POLICY'07)
0-7695-2767-1/07 $20.00 © 2007 IEEE

a general rule, in Step i of the protocol A and B compare all policy rules that lead to the same maximum of minimum of ranks.

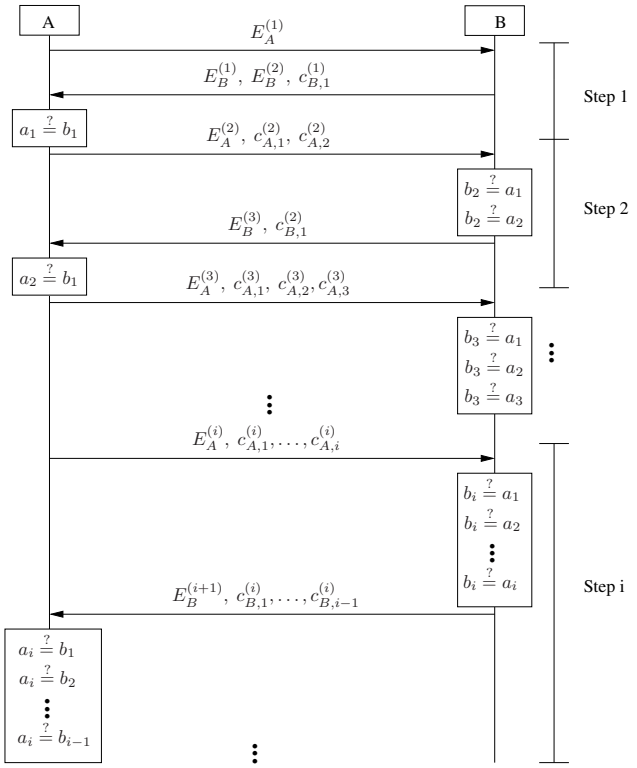

Figure 4. Commitment Phase of 3PR^C for the maximized minimum of ranks composition scheme. In Step i, A compares a_i with $b_1, \ldots, b_{i-1}$ and B compares b_i with $a_1, \ldots, a_i$ without obtaining knowledge of these values, that is, in a privacy-preserving manner.

3 Implementation

3.1 Protocol Implementation Details

Our implementation of the privacy-preserving policy reconciliation protocols required several base components. Since our scheme involves the generation of cryptographic keys using the Paillier cryptosystem [16], we began by adapting the system from [24] into a library we could use for our system. Participants in the policy reconciliation protocols begin by using this library to generate cryptographic keys of the desired length. The next step involves the participants exchanging their public keys and policy size. The other base tool we require for our protocol implementations was a multiple precision arithmetic library. This is used to both perform cryptographic operations as well as the necessary algebraic manipulation of the encrypted policy polynomials. We chose to incorporate OpenSSL's [14] Big Number library because it was written specifically to address the needs of public key protocols similar to ours.

With these base tools in place, our privacy-preserving policy reconciliation systems can perform their core protocol operations. Policy rule sets are stored as bit-strings in flat storage files. At the beginning of the protocols, the policy rules are read from their files and converted to decimal to facilitate future computations. The participants connect to each other using standard Internet sockets and proceed in running the protocols in an iterative fashion as described in Section 2.

To isolate the cost that the privacy operations incur on the reconciliation process, we also implemented a version of the protocols that skips the cryptographic operations that hide the policy rule set belonging to each participant. They are simply exchanged in the clear instead. All other operations remain the same, that is, the policy rule exchange messages and the policy composition scheme occur as in the private case. This ensures that in both cases the protocols will output the same results and that the structure, length, and number of messages sent is kept the same in order to isolate the privacy-related overhead.

3.2 Experimental Setup

Our experimental test-bed was deployed on a private network and the computers were dedicated to running our protocol tests to eliminate network interference. We ran experiments under two configurations. The first one consisted of two AMD Athlon 2.0GHz machines with 1GB of RAM running Gentoo 3.4 connected via a Netgear EN104TP 10Mbps hub, creating a one hop topology. In the second configuration, we included a Netgear FS605 v2 switch to the chain. Specifically, one machine was connected to the hub, the second machine was connected to the switch, and the two network elements were connected to each other, creating a two hop topology.

We performed a series of experiments to verify the correctness and analyze the performance of our implementation of the privacy-preserving policy reconciliation protocols. We generated random policy rule sets ranging in size from 10 to 1 million entries in powers of 10 increments for each participant. The length of each policy rule, which in our system we represent with a bit-string (see Section 2), grows linearly in the number of entries in order to generate unique rules. Each execution of the protocols was repeated ten times on the same policy input to eliminate any timing deviations.

0-7695-2767-1/07 $20.00

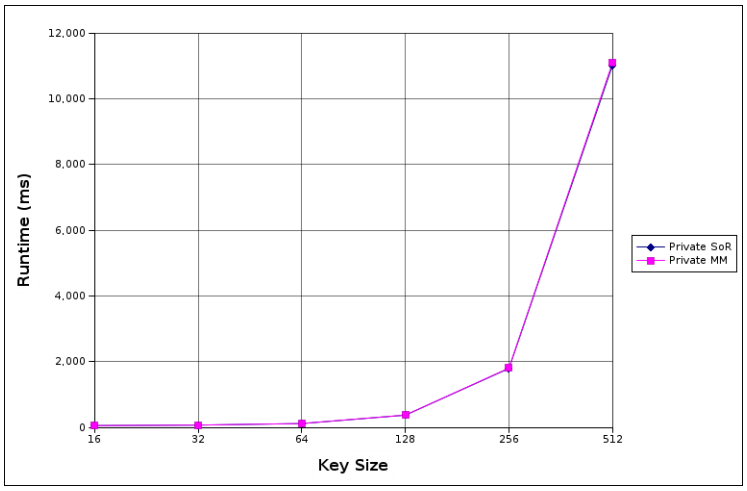

Figure 5. **Protocol execution time as a function of the key-size used for a policy size of 100 over one network hop.**

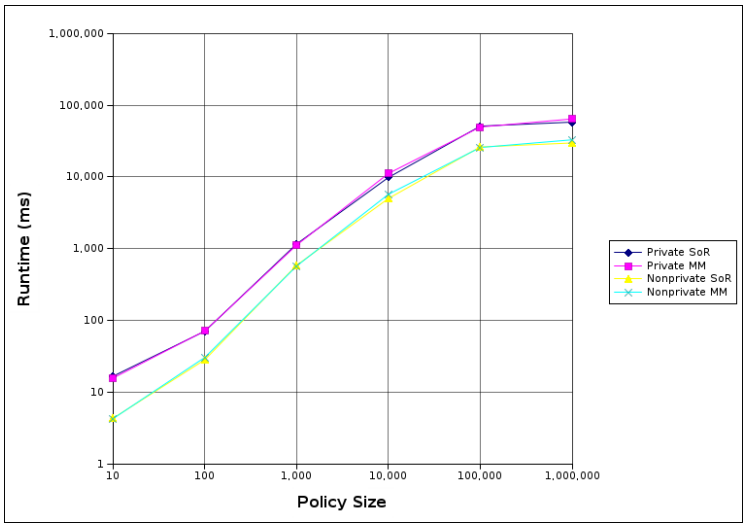

Figure 6. **Protocol execution time as a function of the policy-size for a 32 bit key over one network hop.**

3.3 Results

In Figure 5 we present the execution time of our protocols as a function of the key-size used for a fixed policy size of 100 rules. Our key selection ranges from 16 to 512 bits in length, increasing by a power of 2 each step. We plot two curves, one for the *sum of ranks* (SoR) composition scheme and one for the *maximized minimum of ranks* (MM). As expected, performance degrades as the key length increases.

The next two experiments, presented in Figures 6 and 7 show the effects of the policy size at fixed key lengths of 32 and 512 bits. The graphs present two pairs of curves, the first pair shows the performance of the normal protocols, whereas in the second pair we have disabled the cryptographic operations that guarantee the privacy of the participants as described in Section 3.1. This isolates the cryptographic costs from the communication and computation costs. It is immediately noticeable that for large key lengths the limiting factor is the cost of cryptography. However. we believe that for small policy rule sets, *e.g.,* less than 100 entries, performance is still reasonable. Furthermore, we expect that for large rule sets (more than 10 thousand entries) reconciliation operations will occur rather infrequently.

Finally, Figure 8 illustrates the effect that adding an extra network step between the two machines performing the reconciliation process has on the operation of the protocols. The added network latency is hardly visible in the performance of the protocols.

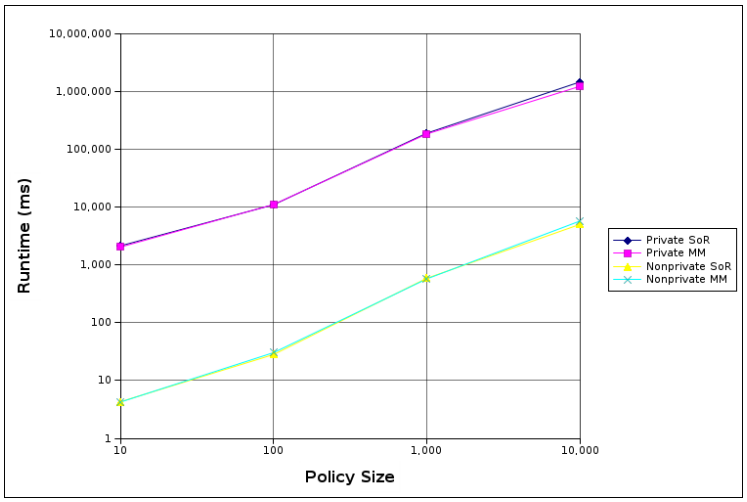

Figure 7. **Protocol execution time as a function of the policy-size for a 512 bit key over one network hop.**

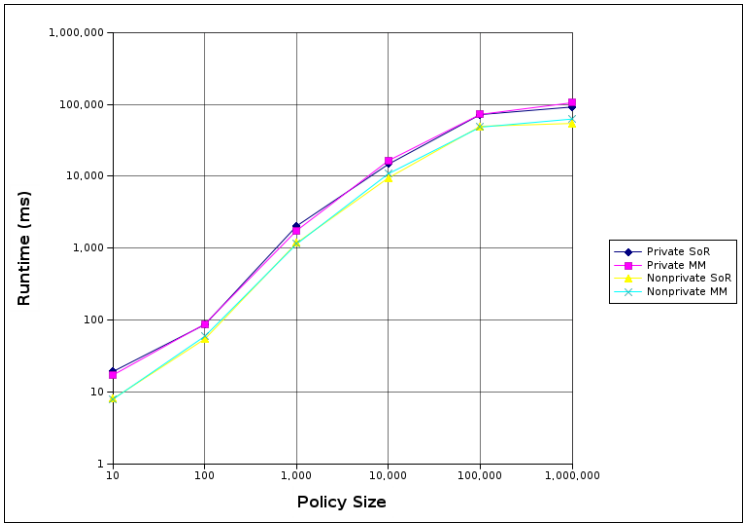

Figure 8. **Protocol execution time as a function of the policy-size for a 32 bit key over two network hops.**

In summary, our results show that even though the privacy operations add significant overhead to the reconciliation protocol, there are scenarios where their use can be practical. More specifically, small rule sets can benefit from our privacy-preserving policy reconciliation protocols without adverse performance impact. Large rule sets, on the other hand, will face a more pronounced slowdown. However, this is proportional to the policy size.

4 Related Work

Security Policy Reconciliation: Systems with heterogeneous access control structures present a challenge whenever there is a need for interoperation. In [3], Gong and Qian analyze the complexity of secure interoperation in such systems. They proved that composing authorization policies is NP-complete. The Ismene policy language, defined by McDaniel and Prakash [10] permits the specification and reconciliation of group security requirements. In follow-up work, McDaniel *et al.* show that general purpose provisioning policy reconciliation is intractable [11]. Wang *et al.* demonstrate that it is possible to structure security policies in such a way that policy reconciliation becomes tractable [19]. Specifically, they represent policies as directed acyclic graphs. Using this structure, they are able to achieve efficient security policy reconciliation.

The Security Policy System (SPS) was proposed by Zao *et al.* [25]. Its purpose is to resolve IPsec security associa-

tions between domains of communication. Reconciliation is achieved by intersecting sets of policy values. Dinsmore *et al.* [1] present the Dynamic Cryptographic Context Management project in which security policies are negotiated between dynamic groups of participants. Their protocol involves multiple rounds of negotiations between the participants, eventually producing a common policy that all of them agree upon. The protocols we propose in [13] are orthogonal to these other works in that the goal is to ensure privacy during policy reconciliation and not the reconciliation algorithm itself.

Automated Trust Negotiation was first proposed by Winsborough *et al.* [23]. The purpose is to build trust between participants by having the two parties exchange digitally signed credentials that contain attribute information in order to establish trust and make access control decisions. In environments like the Internet where there may be few or no pre-existing relationships, parties that seek to form a trust relationship might be unwilling to release sensitive credentials [20, 5]. To address this difficulty, a number of schemes have been proposed recently that use cryptography or multiple rounds of negotiations to protect credentials and attributes [21, 22, 9, 4, 17, 18]. Using an automated trust negotiation scheme, participants specify access control policies for the disclosure of credentials. They then enter a negotiation phase which consists of a sequence of exchanges that are controlled by the access control policies defined for the credentials. At each round, parties gain higher levels of mutual trust, permitting access control policies for more sensitive credentials to be satisfied, which in turn enable these credentials to be exchanged.

Our protocols [13] are designed to protect privacy in security policy reconciliation where two parties attempt to reach an agreement on a communication association that satisfies their security requirements. In contrast to automated trust negotiation, our approach does not require a sequence of credential exchanges in order to build trust (and in turn reveal more policies). Instead, we use cryptographic functions to protect the privacy of the policies themselves such that there is no need for additional access policies that control the release of the original security policy of each participant.

Privacy-Preserving Protocols: Freedman *et al.* [2] present a solution to the problem of computing the intersection of private datasets of two parties based on representing sets as roots of polynomials. Follow-up work by Kissner and Song [6, 7] extends these results by utilizing properties of polynomials beyond evaluation at given points. In their protocol, they present privacy-preserving operations for union, intersection, and element reduction. Our main contribution in [13] is that we extend on the privacy-preserving operations in [2] by constructing new privacy-preserving protocols for security policy reconciliation that allow the maximizing of various preference orders. In [8], Kursawe *et al.* address the similar problem of reconciling privacy policies in a privacy-preserving manner. While our work is based on privacy-preserving set operations, their solution is based on modeling the function to evaluate as boolean circuits and evaluating it on inputs encrypted with a threshold homomorphic cryptosystem.

In this paper, our work does not focus on the theoretical aspects of privacy-preserving protocols but explores their practicality by implementing them in a real system and evaluating their performance under a variety of policy configurations.

5 Conclusions

In this paper we presented an initial performance evaluation for privacy-preserving policy reconciliation protocols. Our results show that for key sizes of 512 bits, the cost of privacy comes at a two order of magnitude slowdown over the non-private reconciliation protocols. The slowdown, however, is directly proportional to the size of the policy. We believe that privacy-preserving policy reconciliation is practical in cases where the policy size is small or in cases where even though the policy size is large, updates are infrequent.

Our future research will focus on two main directions. First, we are planning on conducting a more extensive performance evaluation. This will include multiple participants, various network configurations, and more realistic security policies. Secondly, we intend to investigate alternate policy representations. We expect that a more efficient policy encoding will reduce the cryptographic overhead that we are currently experiencing.

Acknowledgments

We would like to thank the anonymous reviewers and Jared Cordasco for their valuable comments that helped us improve this paper. This work was partially supported by the National Science Foundation under Grants No. CCR-0331584 and DUE-0417085.

References

[1] P. Dinsmore, D. Balenson, M. Heyman, P. Kruus, C. Scace, and A. Sherman. Policy-Based Security Management for

Large Dynamic Groups: A Overview of the DCCM Project. In *Proceedings of DARPA Information Survivability Conference and Exposition*, 2000.

[2] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient Private Matching and Set Intersection. In *Proceedings of EUROCRYPT'04*, 2004.

[3] L. Gong and X. Qian. The Complexity and Composability of Secure Interoperation. In *Proceedings of the IEEE Symposium of Security and Privacy (S&P'94)*, 1994.

[4] J. E. Holt, R. W. Bradshaw, K. E. Seamons, and H. Orman. Hidden Credentials. In *WPES'03: Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*, 2003.

[5] K. Irwin and T. Yu. Preventing Attribute Information Leakage in Automated Trust Negotiation. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS'05)*, 2005.

[6] L. Kissner and D. Song. Private and Threshold Set-Intersection. In *Proceedings of CRYPTO'05*, 2005.

[7] L. Kissner and D. Song. Private and Threshold Set-Intersection. Technical Report CMU-CS-05-113, Carnegie Mellon University, February 2005.

[8] K. Kursawe, G. Neven, and P. Tuyls. Private Policy Negotiation. In *Proceedings of Financial Cryptography'06*, 2006.

[9] J. Li, N. Li, and W. H. Winsborough. Automated Trust Negotiation Using Cryptographic Credentials. 2005. CERIAS TR 2005-59.

[10] P. McDaniel and A. Prakash. Ismene: Provisioning and Policy Reconciliation in Secure Group Communication. Technical Report CSE-TR-438-00, University of Michigan, 2000.

[11] P. McDaniel and A. Prakash. Methods and Limitations of Security Policy Reconciliation. In *Proceedings of the IEEE Symposium of Security and Privacy (S&P'02)*, 2002.

[12] U. Meyer and S. Wetzel. On the Impact of GSM Encryption and Man-in-the-Middle Attacks on the Security of Interoperating GSM/UMTS Networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'04)*, 2004.

[13] U. Meyer, S. Wetzel, and S. Ioannidis. Distributed Privacy-Preserving Policy Reconciliation. In *Proceedings of IEEE International Conference on Communications (ICC'07)*, 2007.

[14] OpenSSL: The Open Source Toolkit for SSL/TLS. `http://www.openssl.org`.

[15] P. Paillier. Public-key Cryptosystems Based on Composite Degree Residuosity Classes. In *Proceedings of EUROCRYPT'99*, 1999.

[16] P. Paillier. Composite-Residuosity Based Cryptography: An Overview. Technical report, RSA Laboratories Cryptobytes, Winter/Spring 2002.

[17] K. Seamons, M. Winslett, and T. Yu. Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation. In *Proceedings of Network and Distributed System Security Symposium*, 2001.

[18] K. E. Seamons, M. Winslett, T. Yu, L. Yu, and R. Jarvis. Protecting Privacy During On-line Trust Negotiation. In *Proceedings of the 2nd Workshop on Privacy Enhancing Technologies (PET'02)*, 2002.

[19] H. Wang, S. Jha, P. McDaniel, and M. Livny. Security Policy Reconciliation in Distributed Computing Environments. In *Proceedings of the IEEE Workshop on Policies for Distributed Systems and Networks (Policy'04)*, 2004.

[20] W. Winsborough and N. Li. Towards Practical Automated Trust Negotiation. In *Proceedings of IEEE 3rd International Workshop on Policies for Distributed Systems and Networks (Policy'02)*, 2002.

[21] W. H. Winsborough and N. Li. Protecting Sensitive Attributes in Automated Trust Negotiation. In *Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society (WPES'02)*, 2002.

[22] W. H. Winsborough and N. Li. Safety in Automated Trust Negotiation. In *IEEE Symposium on Security and Privacy (S&P'04)*, 2004.

[23] W. H. Winsborough, K. E. Seamons, and V. E. Jones. Automated Trust Negotiation. In *Proceedings of DARPA Information Survivability Conference and Exposition*, 2000.

[24] R. Wright, O. Kardes, R. Ryger, and J. Feigenbaum. Implementing Privacy-Preserving Bayesian-Net Discovery for Vertically Partitioned Data. In *Proceedings of the ICDM Workshop on Privacy and Security Aspects of Data Mining*, 2005.

[25] J. Zao, L. Sanchez, M. Condell, C. Lynn, M. Fredette, P. Helinek, P. Krishnan, A. Jackson, D. Mankins, M. Shepard, and S. Kent. Domain Based Internet Security Policy Management. In *Proceedings of DARPA Information Survivability Conference and Exposition*, 2000.