Evaluating Encrypted Boolean Functions on Encrypted Bits: Secure Decision-making on the Black side

Rajesh Krishnan*[a], Ravi Sundaram[b]
[a]Cosocket LLC, 464 Shenandoah Valley Drive, Front Royal, VA 22630; [b]College of Computer and Information Science, Northeastern University, 360 Huntington Avenue, Boston, MA 02115

ABSTRACT

We present a novel approach for secure evaluation of encrypted Boolean functions on encrypted bits. Building upon Barrington's work to transform circuits to group programs and the Feige-Kilian-Naor cryptographic protocol, our novel Fixed Structure Group Program construction for secure evaluation eliminates the need for an expensive Universal Circuit to hide the function. Elements on the Black side weave together and multiply two coordinated streams of random sequences of elements from an unsolvable group; the Boolean decision is recovered while preserving the confidentiality of the decision function and the input bits. The operation is fast and can be further sped up using parallel computation. Our approach can handle expressions with NC^1 complexity, which is the class of Acyclic Boolean Circuits with polynomial width and logarithmic depth in the size of the input. This efficiently parallelizable class includes non-monotone Boolean expressions of equality, inequality/range, Hamming distance, Boolean matrix multiplication, and k-of-m threshold matching operations. The combined benefits of scaling and expressivity of our approach enables secure decision-making on the Black side. Envisioned applications include confidential publish/subscribe systems (with empirically validated performance), secure content-oriented internetworks, confidential forwarding and firewalling rules, and cross-domain guards.

Keywords: Encrypted Operations on Encrypted Data, Barrington's Theorem, Feige-Kilian-Naor Protocol, Confidential Publish/Subscribe, Secure Decision-making on Black side, Fixed Structure Group Program, Secure Cloud Infrastructure

1. INTRODUCTION

1.1 Motivation

Traditional cryptographic architectures—with binary Red-Black separation—are in tension with the need for increased sharing of both information and computational resources across organizational boundaries while preserving fine-grained need-to-know under limited trust. For example, in today's architectures, queries to databases are only protected in transit. They are decrypted for processing, and the administrator of the RDBMS has need-to-process, but not necessarily the need-to-know; with traditional architectures, the two are needlessly bound together. From the operational command and control perspective, edge wireless technologies are providing increased local bandwidth and are increasingly content-centric, while the back-haul often remains stressed in terms of availability and capacity. From the ISR perspective, sensors are getting increasingly more capable in terms of resolution and frame rate, with more opportunities for in situ processing. Limiting exposure of vital information to open-source intelligence gathering is a related consideration. With new technology that can support secure processing on encrypted data, critical processing can be pushed out to the edge and/or distributed into the cloud securely and confidentially [8].

We have developed a novel algorithm for securely evaluating an encrypted Boolean function on encrypted bits. We have implemented and evaluated our solution in the context of a confidential content-based publish/subscribe system, in which a server (Broker) is able to match publications to subscriptions without learning the content of either the publications or the subscriptions. The exact time complexity of our solution is a low-order polynomial of the input size. Any subscription expression within NC^1 [10] complexity is supported. NC^1 is the class of Acyclic Boolean Circuits with polynomial width and logarithmic depth in the size of the input. This efficiently parallelizable class includes non-monotone Boolean expressions of equality, inequality/range, Hamming distance, Boolean matrix multiplication, and k-of-m threshold matching operations.

* krash@cosocket.com; phone +1 540 622-7331; http://cosocket.com

Using our approach, computational resources within the cloud can perform contextual decision-making in the encrypted domain without decryption; elements within the cloud compute the Boolean decision to be performed—for example, whether to forward or cache the encrypted content—without learning anything about the operation performed, or the metadata/content upon which the operation is performed. The fundamental question our technology addresses is: how to perform computational decision-making operations securely on the Black side!

1.2 Related Work

We provide an overview of areas of cryptography that investigate secure processing on encrypted data; these areas have been in existence for several decades, yet, these areas have seen limited mainstream applications focused on traditional computing, communications and networking architectures. These areas, however, are gaining interest (and results) from the research community in recent years; researchers have identified numerous applications in private information retrieval, private searching, secure voting, and other areas including secure signal and image processing.

In cryptography, secure multi-party computation is a problem that was suggested by Andrew C. Yao in 1982 [16]. Yao proposed a solution after introducing the millionaire problem: Alice and Bob are two millionaires who want to find out who is richer without revealing the precise amount of their wealth. This problem and result gave way to a generalization called multi-party computation (MPC) protocols [17]. In an MPC, a given number of participants (p_1, p_2 ... p_N) each with private data (respectively d_1, d_2 ... d_N). The participants want to compute the value of a public function F on N variables at the point (d_1, d_2 ... d_N). An MPC protocol is dubbed secure if no participant can learn more from the description of the public function and the result of the global calculation than what he/she can learn from his/her own entry—under particular conditions depending on the model used.

Oblivious Transfer (OT) is a related cryptographic primitive proposed by Rabin[12], where one party can transfer a subset of bits to a second party, without learning which specific bits were retrieved by the second party. Even, Goldreich, and Lempel provided an early OT protocol [3]. The Naor-Pinkas OT protocol [9] is popular in current systems. The related area of zero-knowledge proofs and verifiable secret sharing is presented in a simple way by Quisquater et al.[11].

Two major approaches to secure computation are popular in state-of-the-art literature. The first approach is based on Garbled Circuits. Huang, Evans, Katz, and Malka [6] have recently developed a practical fast pipelined implementation of Yao's Garbled Circuits that uses the Naor-Pinkas OT protocol. In this approach, the function is not confidential, only the inputs are. The second approach uses Universal Circuits to provide confidentiality of the function as well as the inputs. Valiant, who received the ACM Turing Award in 2010, is well-known for his seminal work in Universal Circuits[15]. Kolesnikov, Sadeghi, and Schneider use Universal Circuits to make the function confidential [7].

Our solution, presented in this paper, overcomes limitations of the extant approaches through a novel Fixed Structure Group Program construction that eliminates the need for a Universal Circuit. We build on work by Feige, Kilian and Naor [4] who showed how Group Programs can be used as a basis for secure evaluation by a third party; note their approach does not keep the function confidential, while our approach does. We also use Barrington's work [1] which shows how to convert an Acyclic Boolean circuit to a Group Program.

Homomorphic encryption is a form of encryption where one can perform a specific algebraic operation on the plaintext by performing a (possibly different) algebraic operation on the ciphertext. For example, it is possible for a third party to compute encrypted circuits on encrypted inputs without gaining any information about the inputs or the circuits being computed. Rivest, Adleman, and Dertouzos [14] posed the problem of whether any arbitrary computation—with both addition and multiplication operations in a field—can be performed using homomorphic encryption. A solution proved more elusive; for more than 30 years, it was unclear whether fully homomorphic encryption was even possible. In 2009, the first fully homomorphic cryptosystem was constructed by Gentry [5] which removed theoretical barriers to fully homomorphic encryption. The existence of a fully homomorphic cryptosystem would have great practical implications in the outsourcing of private computations, for instance, in the context of cloud computing. Effort is underway to create practical and scalable implementations. All homomorphic schemes are malleable, and cannot provide indistinguishability under adaptively chosen cipher-text (IND-CCA2) attacks. In contrast, secure evaluation approaches such as the one we provide are non-malleable.

1.3 Problem

We formulate the problem in the context of a confidential content-based publish/subscribe system. For an introduction to content-based publish/subscribe systems, see Carzaniga and Wolf [2], and the subsequent extensions by Raiciu and

Rosenblum [13] to keep inputs (but not subscription functions) confidential. There are three classes of participants in a confidential content-based publish/subscribe system—Publishers, Subscribers, and Brokers. The Publisher produces content and metadata and encrypts them using separate mechanisms. The Subscriber expresses interest in content by declaring and encrypting Boolean expressions that can operate on the metadata bits. The Broker performs the matching and relaying entirely in the encrypted domain without decryption. In other words, the Publisher and Subscriber are Red-side entities that receive a secure content-based decision service from the Broker residing within the Black side of the network. The participants and interactions are illustrated in Figure 1.

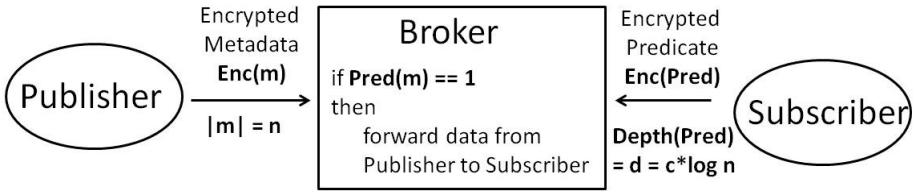

Figure 1. The Confidential Publish/Subscribe Problem

In our formulation, the identities of all participants are known to each other. The Publisher and Subscriber have a shared secret which is not known to the Broker. The Publisher and Broker do not collude with any other party. Collusion among Subscribers are tolerated; collusion will collapse the colluding Subscribers to be treated as a single Subscriber. The parameters of the problem are: (i) the number of bits, n, in the Publisher's metadata, m, and (ii) the depth of the circuit encoding of the Subscriber interest expression, Pred. For efficiency reasons, the interest expressions are limited to circuit depth d = c * log n, where c is a constant. The objective is to create an efficient secure function computation at the Broker which will preserve the confidentiality of the metadata and the Subscriber predicate. The only information learned by the Broker is whether there is a match, in which case, the content payload is forwarded to the Subscriber.

2. PRELIMINARIES

2.1 Group Programs

Permutations of (1 2 3 4 5) form a group called the symmetric group S5, with well-defined identity, multiplication, and inverse. There exist several 5-cycles such that their commutator is a 5-cycle. S5 is a non-solvable group of order 120.

For any Boolean expression on n bits, we can construct an equivalent Group Program, which consists of a series of sequence of Group Program elements. A Group Program element. shown in Figure 2, depends on a Boolean bit, and has two values from the Group, and the value of the bit determines which of the two values are chosen for that element.

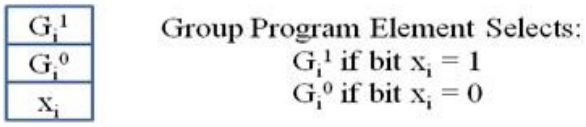

Figure 2. Element of a Group Program

A Group Program is said to "α-compute" a Boolean expression if it evaluates to a 5-cycle α if the expression evaluates to true for the given input bits, and to identity I otherwise. Evaluation consists of selection of the value of each element based on the corresponding input bit, and group multiplication of the selected elements. S5 is a non-commutative group, therefore, the order of multiplication matters. A Group Program in S5 with a particular choice of α is shown in Figure 3.

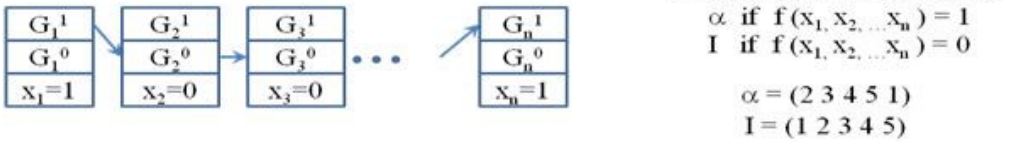

Figure 3. Structure of a Group Program in S5

2.2 Barrington's Transform

Barrington [1] showed how to convert any Acyclic Boolean Circuit to a Group Program and proved that any depth-d circuit can be converted to an equivalent Group Program of length 4^d. A log-depth circuit with n inputs will therefore result in a program of length n^2. Barrington Transforms for an AND gate and a NOT gate are shown in Figure 4. Elements in the construction for an input wire of the AND or NOT gate can be replaced with a Group Program segment for a sub-circuit. We can convert a Group Program from a γ-compute form to an α-compute form, for any cycles α, γ ∈ S5. We note that a NAND gate is a Universal gate that can be composed from an AND gate and a NOT gate, and therefore, any acyclic Boolean expression can be transformed into a Group Program using this approach.

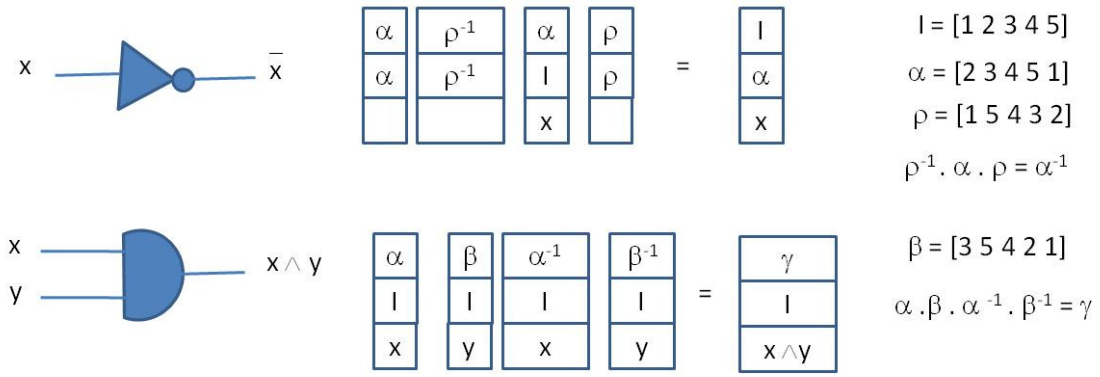

Figure 4. Barrington's Transform of an Acyclic Boolean Circuit to a Group Program in S5

2.3 Feige-Kilian-Naor Protocol

Feige, Kilian, and Naor [4] presented a minimal model for secure computation in which a third party Carol can compute a public function on private inputs from two other parties Alice and Bob. They show the algorithm can apply to any public function in the complexity class NC^1 [10], the complexity class of circuits with depth logarithmic in the input size and width polynomial in the input size.

Using the Feige-Kilian-Naor protocol, Alice and Bob prepare their inputs for the known public function by encoding it as a Group Program. Inputs from Alice and Bob will be interleaved in the resulting Group Program. Alice and Bob pre- and post-multiply their inputs with random group elements and their inverses chosen from a coordinated random sequence unknown to Carol. They coordinate the multipliers such that the random element post-multiplied by Bob will cancel the pre-multiplier Alice applies to the next element, and vice versa. Carol determines the result by multiplying the elements together, and does not learn the private inputs of Alice and Bob. The protocol is illustrated in Figure 5.

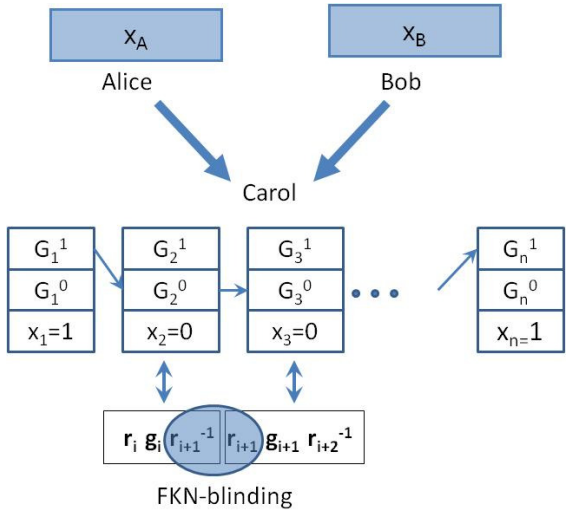

Figure 5. Using Feige-Kilian-Naor protocol, Alice and Bob use a shared secret to coordinate randomness for preparing their private inputs so that Carol can compute a public function securely

2.4 Universal Circuits

Valiant [15] showed that there is a combinational acyclic Boolean circuit of complexity O(s log s), that can be made to compute any Boolean function of complexity s by setting its specially designated set of control inputs to appropriate fixed values. As shown in Figure 6, a Universal Circuit takes a smaller circuit along with the input bits of the smaller circuit and computes the same output as the smaller circuit would have computed on the input bits. Valiant also showed how to construct such a Universal Circuit.

Figure 6. The concept of a Universal Circuit

Universal Circuits have been used in secure function evaluation to hide the function being evaluated (see for example, Kolesnikov et. al. [7]). For example, in the previous section, Carol was able to compute a public function on private inputs from Alice and Bob. We can use a Universal Circuit as a means to make the function private as well. In practice, the Universal Circuit will be large relative to the size of the circuit to be simulated, which places a limitation on the size of the input function that can be computed by the Universal Circuit.

3. SOLUTION APPROACH

3.1 Overview

Our overall solution approach is summarized in Figure 7. In our approach the Publisher and Subscriber agree on a shared randomness and the structure of a Universal Group Program described later. The Universal Group Program is of fixed structure and length with elements from Publisher and Subscriber that alternate strictly. The Subscriber converts its predicate to a circuit, applies Barrington Transform to convert it into a Group Program, and then embeds it into a Universal Group Program. The Publisher encodes its metadata into bits, which are then converted into Group Program elements that fit into the agreed upon structure. Both the Publisher and Subscriber blind their elements using Feige-Kilian-Naor protocol. The Publisher prepares the metadata to be matched for each subscription separately. Likewise the Subscriber prepares the interest separately to be matched with each publication. The Broker multiplies the sequences and if the result is alpha, then the content is forwarded to the Subscriber.

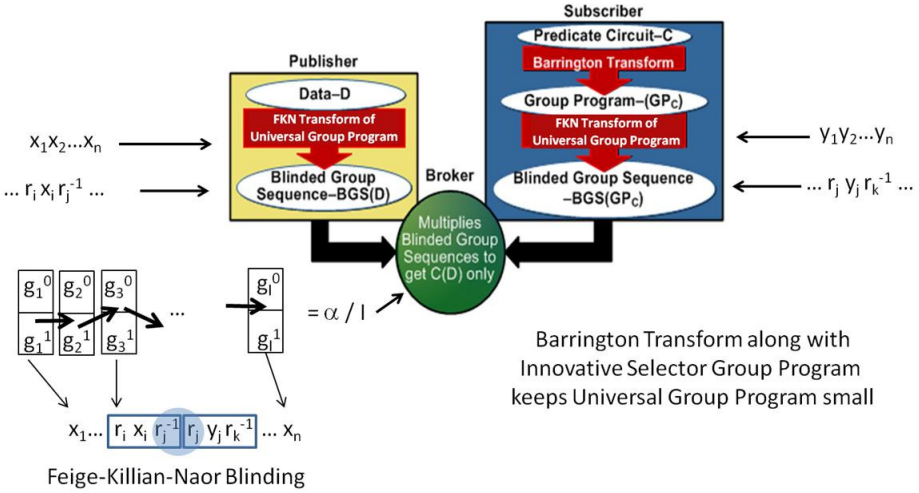

Figure 7. Overview of the Solution Approach

Three key insights make the above-described solution approach practical. These are described below.

3.2 Insight 1 – Simulate Group Programs, Not Circuits

Suppose we let Alice's input encode a circuit, and let Carol's public function be a Universal Circuit (UC) that simulates Alice's circuit on Bob's input. Then suppose we apply Barrington's transform to the UC and apply Feige-Kilian-Naor Protocol. Have we solved the confidential content-based publish/subscribe problem? Not quite!

Recall that Barrington's transform requires the Boolean function, in this case the UC, to be in NC^1. A Universal Circuit of size N_U and depth $O(\log N_U)$ can simulate size-s depth-d circuits such that $d \log s = O(\log N_U)$. Since s can be as large as 2^d observe that this restricts d, the depth of the input circuit, to be at most $(\log N_U)^{1/2}$. In other words, simulating a circuit of depth 10 will require a UC of depth 100, and the corresponding Group Program from the UC will have 4^{100} elements. This naive approach is not tractable.

Our first insight toward a tractable solution was to construct a Universal Circuit that simulated a Group Program instead of a Circuit (as shown in Figure 8). The intuition is that the Universal Circuit to evaluate a Group Program has a simple structure that requires only selectors and group multipliers.

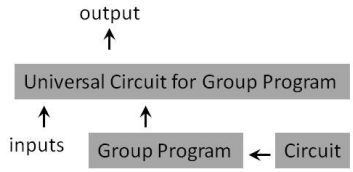

Figure 8. A Universal Circuit that Simulates a Group Program

We state without proof that the resulting solution results in a Group Program of length that is a polynomial of degree 12c + 1 in the metadata size n. Although this insight leads to a tractable solution, in theory, we note that a high-degree polynomial solution is not practical for implementation. For example, for a Subscriber GP of length s, just the GP for the multiplier pyramid portion (within the Universal Circuit for Group Program shown in Figure 9) grows as fast as $112^{\lg s}$, when using an efficient low-depth implementation of the multiplier for S5.

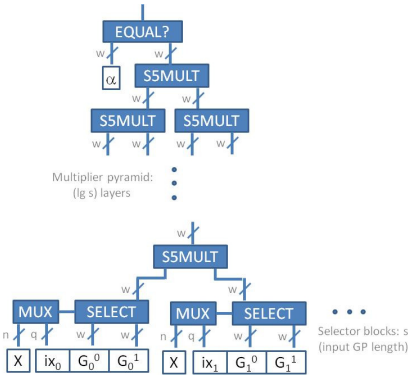

Figure 9. Practical Construction of a Universal Circuit that Simulates a Group Program in S5

3.3 Insight 2 – Fixed Structure Group Program

It became clear that we need to eliminate the UC for GP from the UGP construction. A second and important insight paved the way. We observed that the Broker learns nothing as long as the structure of the group program is fixed; the 2-decomposable randomized encoding guarantees privacy.

A Fixed Structure Group Program can be constructed as follows. We compute Barrington's Transform of the Subscriber's predicate. We convert to a "canonical" form where Publisher and Subscriber elements strictly alternate, which can be accomplished by inserting and multiplying constants as needed. We replace the Publisher elements in the Subscriber's Group Program with a Selector Group Program that picks one out of n Publisher bits. The Selector Group Program can be constructed from a 1-of-n selector circuit by applying Barrington's Transform. We pad the resultant GP to a fixed length with additional blocks that always compute to identity. Without proof, we state that this leads to an FSGP of length $4n^{2c+2}$, which leads to practical implementation.

3.4 Insight 3 – Optimize using Hand-crafted Selector Group Program

We had a third and final insight that made the solution more practical. We were able to construct a Selector Group Program by hand, of length 4n elements that selects 1-of-n inputs if the index is known in advance. In contrast, applying Barrington's Transform to a 1-of-n selector circuit leads to a selector Group Program $4n^2$. The construction is a sequence of AND blocks that AND the input to be selected with 1 and the other inputs with 0 as shown in Figure 10.

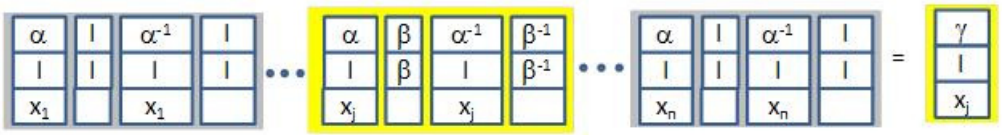

Figure 10. Hand-crafted Selector Group Program to select one out of n inputs

We state without proof the following theorem on the length of the Optimized Fixed Structure Group Program.

Theorem 1: Any depth-d (d=c*log n) formula in n variables can be confidentially matched by the Broker in $4n^{2c+1}$ steps.

We note that the length and structure of the Optimized Fixed Structure Group Program is independent of the inputs, and is uniformly distributed over the space of equal-length Group Programs. The privacy guarantee follows from 2-decomposable randomized encoding arguments similar to those from Feige, Kilian, and Naor [4]. Detailed complexity and correctness proofs are beyond the scope of this paper, and will be addressed in a separate publication.

3.5 Protocol Summary

The above insights lead to a practical distributed protocol implementation with three classes of participants – Publisher, Subscriber, and Broker. The identities of all participants are known to each other. The Publisher and Subscriber coordinate a shared randomness which is not known to the Broker. The Publisher and Broker do not collude with any other party. Collusion among Subscribers will collapse the colluding Subscribers to a single Subscriber. The overall protocol can be summarized by the following steps:

- Each Subscriber and Publisher agree on shared secret randomness, syntax and semantics for the clear-text metadata and Boolean expressions for the subscription, length of the metadata in bits, length of the group program in number of elements, and the (universal) fixed structure for the group program.
- Publisher prepares and sends encrypted metadata (along with opaque payload separately protected using end-to-end encryption) to the Broker; metadata bits are replaced with pairs of alpha or pairs of identity if the respective bit is 0 or 1, the sequence is replicated to the length of the program, and blinded using Feige-Kilian-Naor approach using the shared randomness.
- Subscriber prepares and sends the encrypted subscription to the Broker; preparation involves parsing the expression, expressing as an acyclic circuit of binary gates, converting to a Group program using Barrington's theorem, converting to the fixed structure by adding selector blocks, and blinding using Feige-Kilian-Naor approach using the shared randomness.
- Broker interleaves the encrypted group program sequences from the Publisher and Subscriber, multiplies the resulting sequence, and forwards payload to the Subscriber if and only if the result is alpha. A dummy payload may be send upon failure for additional protection against traffic analysis.

4. EMPIRICAL EVALUATION

4.1 Implementation

We integrated our solution within the Siena content-based publish/subscribe framework due to Carzaniga and Wolf [2]; we use the Siena framework as our clear-text baseline for comparison purposes. An earlier effort to extend the Siena framework to add confidentiality with limited expressivity was done by Raiciu and Rosenblum [13]. Our solution supports any predicate within NC^1, and furthermore provides stronger (information-theoretic) security guarantees.

We make use of an FIPS 140-2 approved cryptographic pseudo-random generation algorithm such as SHA-1 (or later) for the blinding. For practical reasons, we do not use one-time pads for the implementation.

The algorithm along with an end-to-end example was originally rapidly prototyped using jScheme. The prototype helped compare/validate alternatives quickly and refine the solution. The declarative functional programming style kept the overall program short. The seamless integration with Java was beneficial, for example, we make use of the SHA1PRNG random number generator from the Java security library. The implementation covers functionality to encode metadata to bits and expressions to circuits, Barrington's transform, Feige-Kilian-Naor blinding, selector group program blocks, and optimized fixed structure group construction. Once refined, the algorithm was re-implemented in Java. The performance evaluation reported below was conducted with the Java re-implementation.

4.2 Example

We developed an end-to-end example to illustrate the solution (shown in Figure 11). In the example, a simple intelligence record with fields with their corresponding range of enumerated values are used. Subscription expressions allow Boolean combinations of equality. In this notional example, we show the Publisher providing a content item with metadata indicating "an intelligence report of an important cyber threat in the Asia-Pacific region anticipated to occur within days." The Subscriber has expressed interest in "intelligence reports of urgent threats and important cyber threats." The Broker performs the matching without learning either the metadata or the subscription expression, using a simple procedure that can be engineered for high-speed operation. We note that this example implementation covers only a small subset of possible expressions, and several other interesting functions such as k-of-m threshold, Hamming distance threshold, and others fall within the NC^1 class.

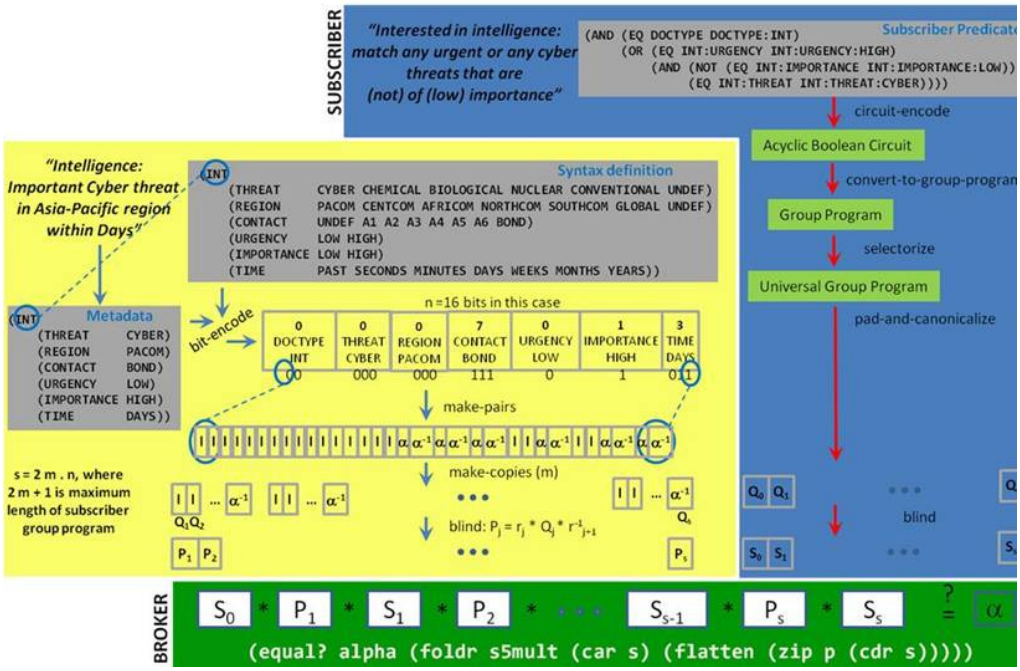

Figure 11. An end-to-end example illustrating the implementation of our algorithm in a publish/subscribe context

4.3 Performance

We evaluated our algorithm integrated within the Siena framework, and used Siena for the clear-text comparison. For convenience, all the processes for the distributed components—Publisher, Broker, and Subscribers—were run on a typical Linux workstation. The implementation, however, allows components to be run on different machines.

The experiment included one hundred publications and up to ten subscriptions each from ten Subscribers. The time is measured from the publication of a content item by the Publisher until delivery to all Subscribers whose subscriptions match the item. The metadata for publications included several fields with values from an enumerated range. The cost imposed by the confidential matching for a publish/subscribe system is shown in Figure 12. The security guarantees provided by our solution require that each metadata and interest are prepared separately for each match. As a result, the communication complexity grows with the number of content items and the number of active subscriptions.

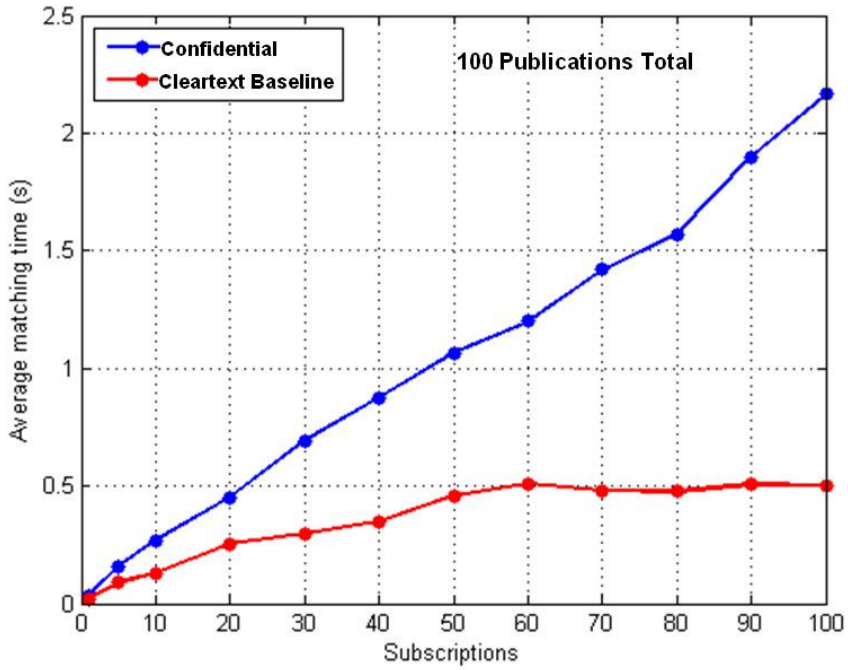

Figure 12. The average time taken to match and deliver new content items using our confidential matching approach, compared with the clear-text baseline

5. CONCLUSION

We presented a novel algorithm for secure evaluation of encrypted Boolean functions on encrypted bits. Our solution for confidential matching is:

- **Scalable**: our innovative selector group program construction in conjunction with Barrington's transform of circuits to group programs leads to a scalable Fixed Structure Group Program for secure matching; we require a smaller cipher-text length (low-order polynomial in input size) by eliminating the need for a Universal Circuit for hiding the function
- **Expressive**: we handle Subscriber interest expressions of NC^1 complexity, the class of Acyclic Boolean Circuits of polynomial width and logarithmic depth in input size
- **Secure**: we use the Feige-Kilian-Naor cryptographic protocol that provides information-theoretic security; most other approaches provide only semantic security

Envisioned applications include confidential publish/subscribe systems, secure content-oriented internetworks, confidential forwarding and firewalling rules, and cross-domain guards. We have implemented and evaluated the algorithm in the context of a confidential content-based publish-subscribe system. Applications of our algorithm to other use cases for secure decision-making in the Black side are the subject of future work.

ACKNOWLEDGMENT

The authors thank Mr. W. Konrad Vesey (IARPA) for his guidance and encouragement, and Mr. David Karnick (Argon ST) for the Java re-implementation of the algorithm.

REFERENCES

[1] Barrington, D. A. M., "Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1," STOC 1986

[2] Carzaniga, A. and Wolf, A. L., "Forwarding in a Content-Based Network," ACM SIGCOMM, 2003

[3] Even, S., Goldreich, O. and Lempel, A., "A Randomized Protocol for Signing Contracts," CACM, Jun 1985

[4] Feige, U., Kilian, J. and Naor, M., "A Minimal Model for Secure Computation," STOC 1994

[5] Gentry, C., "Fully Homomorphic Encryption using Ideal Lattices," STOC 2009

[6] Huang, Y., Evans, D., Katz, J., and Malka, l., "Faster Secure Two-Party Computation Using Garbled Circuits," USENIX Security Symposium, 2011

[7] Kolesnikov, V., Sadeghi, A.-R. and Schneider, T., "From dust to dawn: Practically efficient two-party secure function evaluation protocols and their modular design," Cryptology ePrint Archive, Report 2010/079, 2010

[8] Krishnan, R. and Sundaram, R., "Policy-agile Encrypted Networks Via Secure Function Computation," MILCOM 2010

[9] Naor, M., Pinkas, B., "Oblivious transfer with adaptive queries," CRYPTO 1999

[10] Papadimitriou, C., "Section 15.3: The class NC," Computational Complexity (1st ed.), Addison Wesley, pp. 375–381, ISBN 0-201-53082-1, 1993

[11] Quisquater, J.-J., Guillou, L., Annick, M., Berson, T., "How to explain zero-knowledge protocols to your children," CRYPTO 1989

[12] Rabin, M. O., "How to exchange secrets by oblivious transfer," Tech Report TR-81, Harvard U., 1981

[13] Raiciu C., and Rosenblum, D. S., "Enabling Confidentiality in Content-Based Publish/Subscribe Infrastructures," SecureComm 2006

[14] Rivest, R. L., Adleman, L. and Dertouzos, M. L., "On data banks and privacy homomorphisms," Foundations of Secure Computation, 1978

[15] Valiant, L. G., "Universal Circuits (Preliminary Report)," STOC 1976

[16] Yao, A. C., "Protocols for secure computations," FOCS 1982

[17] Yao, A.C., "How to Generate and Exchange Secrets," FOCS 1986