Experiments in Encrypted and Searchable Network Audit Logs

Bhanu Prakash Gopularam
Cisco Systems India Pvt. Ltd
Nitte Meenakshi Institute of Technology
Bangalore, Karnataka, INDIA
Email: bhanprak@cisco.com

Sashank Dara
Cisco Systems India Pvt. Ltd
IIIT-Bangalore
Bangalore, Karnataka, INDIA
Email: sadara@cisco.com

Nalini Niranjan
Dept. of Computer Science & Engg.
Nitte Meenakshi Institute of Technology
Bangalore, Karnataka, INDIA
Email: nalinaniranjan@hotmail.com

***Abstract*—We consider the scenario where a consumer can securely outsource their network telemetry data to a Cloud Service Provider and enable a third party to audit such telemetry for any security forensics. Especially we consider the use case of privacy preserving search in network log audits. In this paper we experiment with advances in *Identity Based Encryption* and *Attribute-Based encryption* schemes for auditing network logs.**

***Keywords*—network telemetry; identity based encryption; audit log privacy;**

I. Introduction

Network telemetry in the form of NetFlow, IPFix, Syslogs and others, constitute important data about network activity. Such telemetry is considered valuable for network planning, security forensics and audits. The NetFlow records are generated by real-time flow monitoring devices like router, switch, gateway.

The volume of NetFlow data generated is humongous due to the increase in network based applications and Cloud services. A network interface with 100 $Mbps$ speed would collect 12.5 MB every second or 45 $Terabytes$ per hour [1]. Assuming a size-able portion randomly sampled for any security analytics would also need high amounts of storage and computation. Due to such voluminous nature of these logs consumers may prefer to outsource storage of such logs to a Cloud Service Provider for later use. But network telemetry contains sensitive information pertaining to internal IP addresses. In order to prevent unauthorized access to safeguard privacy of the consumers, these records need to be encrypted.

Traditional encryption techniques like AES, RSA ensure stronger privacy guarantees but the utility of the logs for forensics and audit is severely constrained. Especially for enabling any third party audit the techniques should aid for privacy preserving search on encrypted logs.

Recent advances in *Identity Based Encryption* and *Attribute Based Encryption* techniques look promising in providing capabilities of privacy preserving search over encrypted data. These techniques are gaining popularity in research community but exhaustive experimental evidence is lacking on their viability in real world scenario.

A. Key Contributions

In this paper we experiment with

1) *Identity Based Encryption* techniques proposed in [2] and referred as *BB* scheme hereby.
2) *Ciphertext Policy Attribute Based Encryption* techniques proposed in [3] and referred as *BSW* scheme hereby.

We evaluate the performance of operations like setup, key-generation, encryption, privacy preserving search and decryption under various configurations in a network audit log scenario.

II. Preliminaries

A. Identity Based Encryption

Shamir first proposed concept of Identity-Based public key cryptography [4]. The first practical IBE scheme was presented by Boheh and Franklin [5]. These techniques allow a unique identifier of the user (e.g. email address, phone number) to be the public key and generates the corresponding private key. This reduces the over head and complexity involved in key management especially in Public Key Cryptography.

B. Attribute Based Encryption

Sahai and Waters first introduced concept of Attribute-Based encryption in [6]. In 2006 Goyal et al. [7] introduced notion of Key-Policy Attribute based encryption in which monotone access structures is embedded into key-pair and ciphertext is described using user attributes or tags. In 2007 Ostrovsky et al. [8] proposed KP-ABE scheme with non-monotonic structures. In the same year Bethencourt et al. [3] presented first construction of Ciphertext-Policy Attribute-Based encryption which we refer as *BSW*.

Using CP-ABE it is possible to embed role based access control policies into the ciphertext. The decryption depends not only private key and ciphertext but also on certain attributes priorly defined (e.g. role, country, age).

C. NetFlow, IPFix

NetFlow is a technology to collect IP network traffic and perform analytics, measurements and security monitoring of the network. It is been found very useful in both network security monitoring [1] and security incident response [9]. An example packet format is given in Fig 1. NetFlow allows to capture all the characteristics of the network traffic flows like source address, destination address, port, number of packets etc. It is supported by many modern networking devices.

A similar IETF standard format is IPFIX. For all practical purposes in rest of the paper, the challenges and techniques outlined for NetFlow hold good for IPFIX without loss of generality.

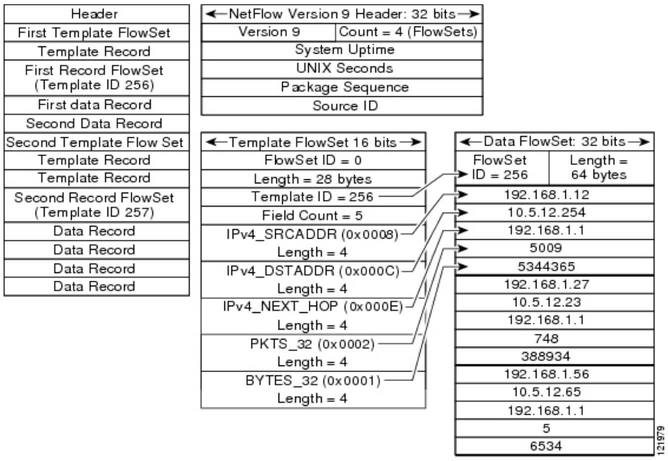

Fig. 1: Example NetFlow v9 Packet

D. *Privacy Preserving Audit*

Searchable Encryption techniques allow a client to store encrypted records at a remote server. Subsequently the client can *Search* the server using a token called *TrapDoor*. Server uses it in order to match the encrypted records and returns the matching ones.

First practical techniques were *Symmetric Searchable Encryption (SSE)* techniques proposed in [10]. Later many other *Searchable Encryption* techniques have been proposed [11],[12]. Privacy Preserving Search using Public Key encryption techniques were proposed in [13].

Privacy Preserving Search for auditing on database queries was presented in [14]. Our experiments are based on their techniques for NetFlow records. We give a brief over view of their work in the next section.

III. Methodology

Here we describe the procedure used for encryption of encrypting network logs. The steps mentioned here can be applied to any *Identity and attribute-based encryption* scheme and procedure for setup and search remains the same except for parameters used.

A. *Setup*

The algorithm initialization depends on bilinear pairing and elliptic curve used. The key server then generates master key MK, and public key PK.

1) Selection of bilinear-pairing a bilinear group G0 of prime order p with generator g. We have used elliptic curve with bilinear maps (or pairings) like SS512 which is a symmetric curve with a 512-bit base field using this private key is generated and asymmetric curve pairings like MNT159 and MNT224 having 159-bit and 224-bit base field respectively.
2) Curve selection We used Type-A curve such as $y^2 = x^3 + x$ to compute the pairings

B. *Key generation and sharing*

The user secret key is generated using (PK, MK, search_keyword). Like in traditional CP-ABE scheme, attributes are associated with public and access policy is associated with ciphertext. Here instead in place of public identifiers the search_keyword is used. The secret key is communicated to interested parties using a secure channel like TLS/SSL.

C. *Encryption*

The data records are read from SiLK repository. For each log entry m comprising search keywords $w_1, w_2, ..w_n$ (keywords could be ip-address, subnet-mask, protocol using which user would like to filter the data)

1) The server encrypts the entry using random symmetric encryption key K, to get $E_K(m)$. For each keyword wi, the server computes the CP-ABE encryption ci of string $(FLAG|K)$ using search_keyword as access policy and PK public key
2) The choice of symmetric encryption for data encryption is attributed to the fact that these exhibit high performance and more suitable while encrypting large data. We have used AES in CBC (cipher-block chaining) mode with 16 byte block size (with PKCS5 padding) and HMAC_SHA1 algorithm as a PRP generator (for randomization).

D. *Match and Decrypt*

If the data owner wants to provide controlled access to third party auditor who wish to search and retrieve of particular data from encrypted records. The data owner with help of key authority constructs private key with capability, then for each encrypted record MatchAndDecrypt operation is run:

- As part of match routine the data record is decrypted using (PK, sk, ciphertext) and the decrypted text is if it has FLAG has prefix.
- The match returns true then decrypt the ciphertext c using previously generated secret-key sk and public key PK. The symmetric encryption key is extracted from decrypted text and one more round of decryption happens but this time it is done using symmetric key.
- If match is false then the record is not processed further.

E. *System Details*

We have used *CHARM*[15] library *v0.43* for prototyping. At a very high-level the library provides a protocol engine for many cryptographic operations and an adapter architecture which bridges gaps necessary for building a complete crypto system.

TABLE I: Security Parameters

Key Size	$\lvert p \rvert$	$\lvert q \rvert$
80	512	160
112	1024	224
128	1536	256

In addition we used other open source libraries including *OpenSSL 1.0.1, GMP 6.0.0a* and *Pairing-Based Cryptography* library version *0.5.14* of Stanford.

The experiments were carried on X86 based platform using Ubuntu 12.04.4 LTS (precise) 32-bit server with 8 GB RAM and Intel Core i5-3470 CPU with 3.2 GHz 4 core processor

F. Test Data

The original sample data is from anonymized enterprise packet header traces obtained from Lawrence Berkeley National Laboratory and ICSI[16]. Further it is converted into *SiLK Flow* format with their permission [17]. We chose the data sets that contain scanning activity as needed by our use cases.

We further *split* these data sets into chunks of approximately 200000 records each carefully ensuring each such data set contains traces of scanning activity.

We perform our experimental analysis on these chunks of data sets in order to observe the performance and space requirements.

IV. Experiments and Evaluation

A. Security Parameters

Table I shows the various security parameters used in both the schemes with type-1 super singular curve.

B. Set up

For key-generation we have used elliptic curve having following bilinear pairing setting

1) SS512 symmetric curve pairing with 512-bit base field
2) MNT159 asymmetric curve pairing with 159-bit base field
3) MNT224 asymmetric curve pairing with 224-bit base field

In general the time taken for operations like setup, key-generation, the asymmetric pairings took almost 50% higher than for symmetric pairing and it is double for MNT224 curves. The BSW scheme took almost twice the time for initialization than the BB scheme.

C. Encryption

Encryption involves generating a random symmetric key and running encryption with keyword in access policy string. The time for both the schemes was in the range 200-1000 milliseconds. The encryption time for BSW scheme is slightly less (10-15%) when compared to BB scheme.

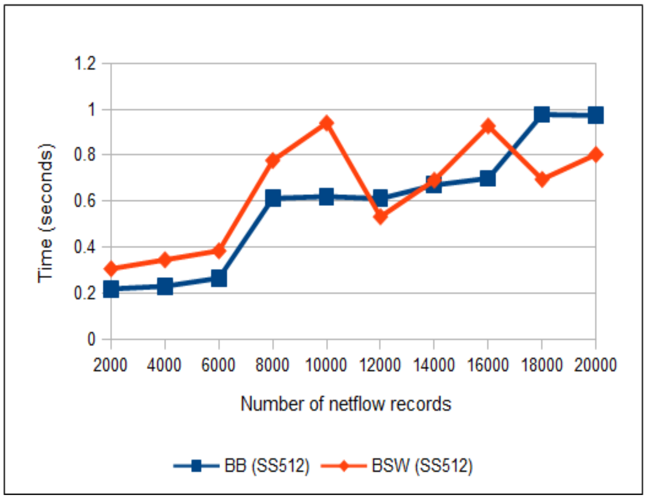

Fig. 2: Encrypt using 512-bit field symmetric curve

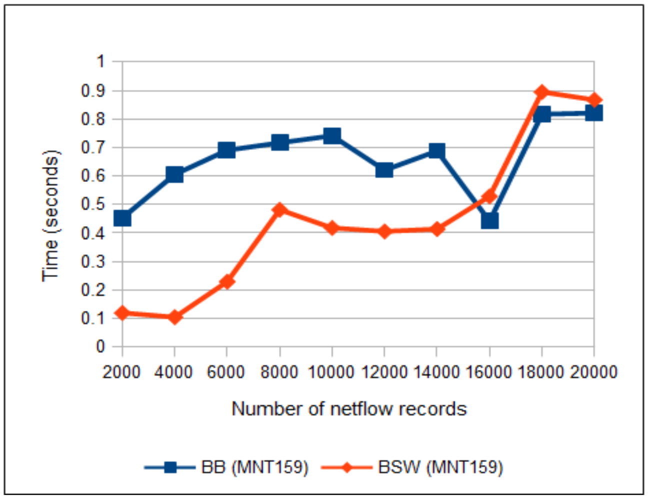

Fig. 3: Encrypt using 159-bit field asymmetric curve

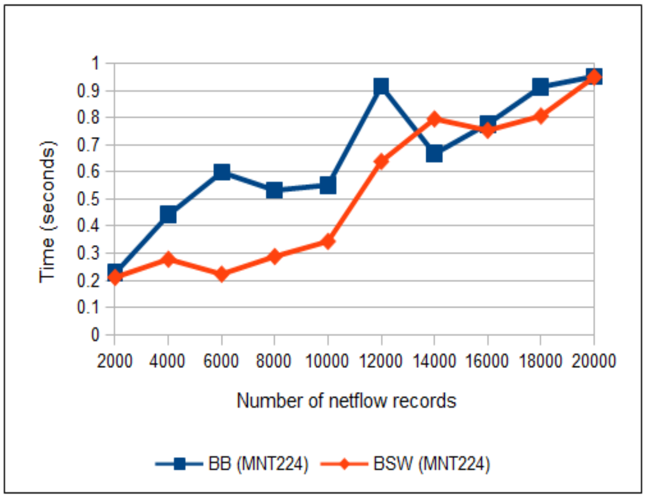

Fig. 4: Encrypt using 224-bit field asymmetric curve

D. Search

The match operation for BB scheme took time 5-120 seconds with SS512 pairing and [10-250] seconds with MNT224 pairing. BSW scheme took between [1-20] seconds only. This is significantly less when compared with BB scheme.

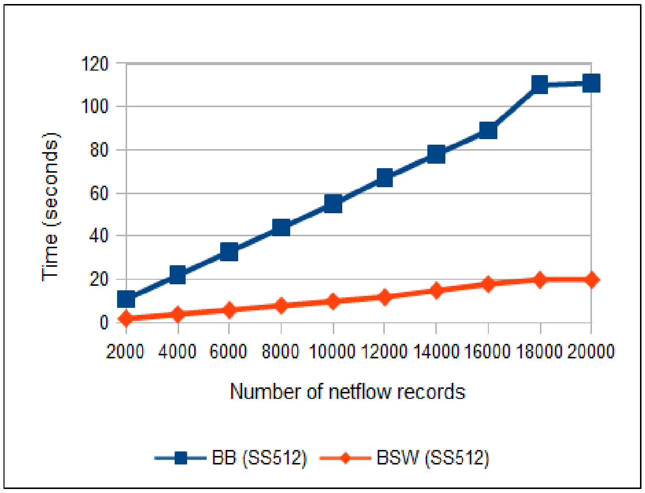

Fig. 5: Search using 512-bit field symmetric curve

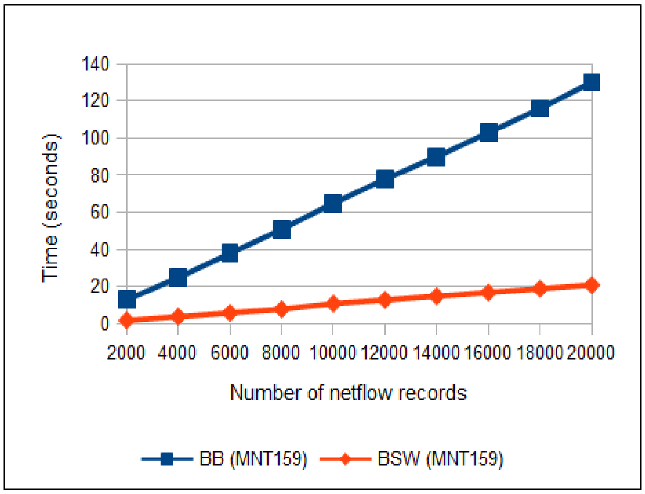

Fig. 6: Search using 159-bit field asymmetric curve

E. Decryption

Once match is found symmetric key is extracted and is used for decryption and it the time was linearly increasing with size of records and for both the schemes with SS512, MNT156, MNT512 pairings.

The performance of decrypt operation is somewhat more interesting in *BSW* scheme. It is slightly more difficult to measure in the absence of a precise application, since the decryp tion time can depend significantly on particular access trees and set of attributes involved. The implementation uses a 160-bit elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over a 512-bit finite field

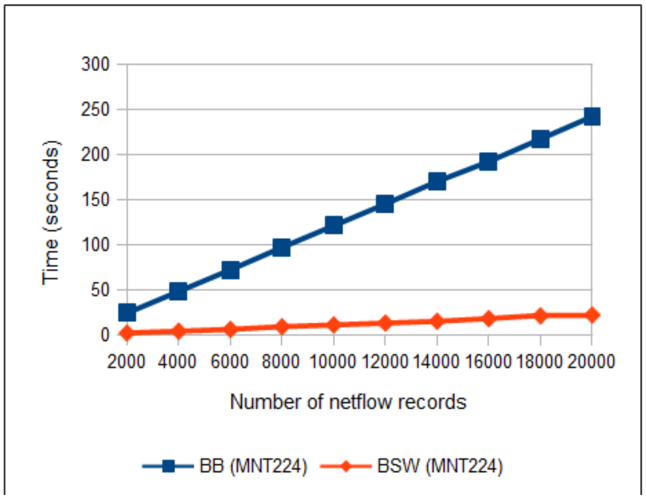

Fig. 7: Search using 224-bit field asymmetric curve

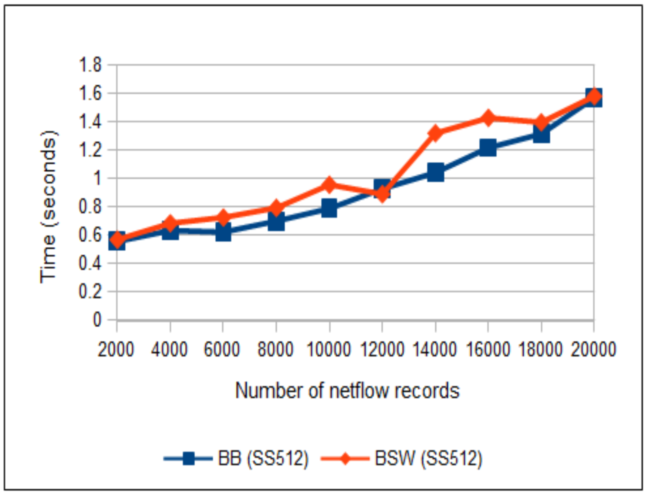

Fig. 8: Decrypt using 512-bit field symmetric curve

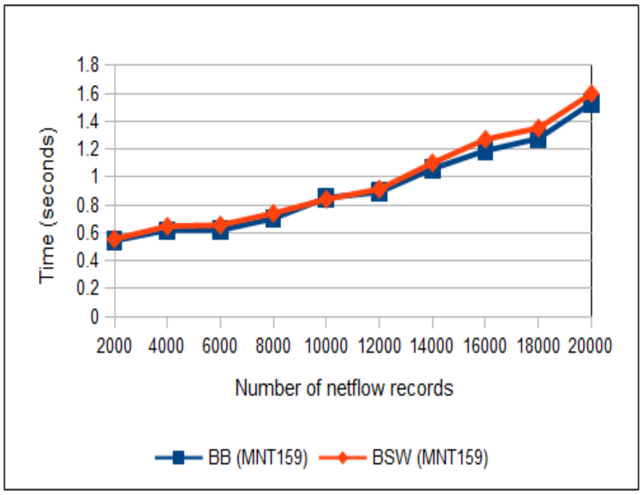

Fig. 9: Decrypt using 159-bit field asymmetric curve

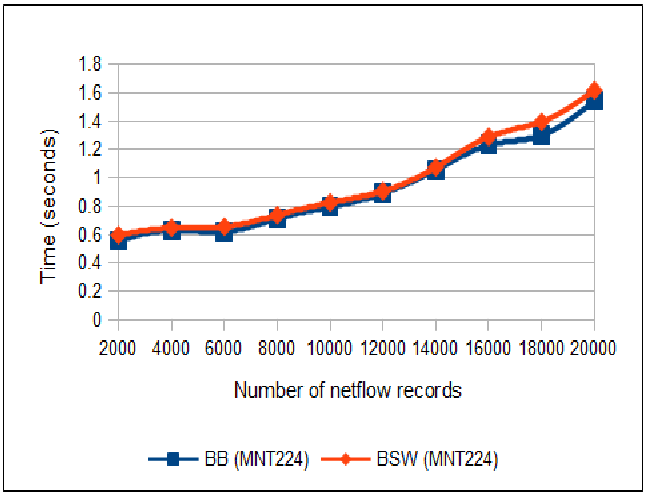

Fig. 10: Decrypt using 224-bit field asymmetric curve

V. Limitations and Recommendations

- Encrypting the network logs with identity-based schemes introduces key-escrow problem. The communication between the data owner of key-escrow agent and key generation server should be secured using traditional SSL schemes.

- The length of FLAG parameter is inversely proportional to false positive ($1/2^l$) chances for the match operation might result in failure). We have chosen SHA_HMAC algorithm for generating the FLAG that is a 128-bit which is fairly enough to execute match operation on large set of keywords but based on requirements one need to carefully choose this setting.

- Regardless of message size, the ciphertext begins at 630 bytes which is not desirable if data size is small. Each attribute would generate 300 byte data and this becomes a problem in representing large access structures.

- Also more importantly notice that we could experiment only with 20,000 records which is very minuscule data sample in reality. This shows that the schemes are not practically viable for larger data sets.

VI. Conclusion

Network Telemetry data generated by flow monitoring devices reveal interactions and business operations of consumers. The identity based encryption techniques provide privacy without overhead of certificate management. Our experiments reveal that *BB* scheme is less perfomant when compared to *BSW* scheme for search and other operations like encrypt and decrypt both the schemes are comparable. In the future we plan to experiment with anonymized data search or hidden keyword search when data is encrypted using attribute-based encryption schemes with monotonic (access structure represented with *AND*, *OR*, or *threshold gates*) and non-monotonic access structures (one with *negative* constraints) and with predicate encryption.

Acknowledgment

We would like to thank Cisco Systems for supporting this work. We would also like to thank *crypto.stackexchange* community for having many insightful discussions around this topic.

Sashank would like to thank his research supervisor Dr. V.N. Muralidhara for his support and guidance.

References

[1] C. Mike, “Netflow security monitoring for dummies.” Wiley.

[2] D. Boneh and X. Boyen, “Efficient selective-id secure identity-based encryption without random oracles,” in *Advances in Cryptology-EUROCRYPT 2004*. Springer, 2004, pp. 223–238.

[3] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in *Security and Privacy, 2007. SP’07. IEEE Symposium on*. IEEE, 2007, pp. 321–334.

[4] A. Shamir, “Identity-based cryptosystems and signature schemes,” in *Advances in cryptology*. Springer, 1985, pp. 47–53.

[5] D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” in *Advances in CryptologyCRYPTO 2001*. Springer, 2001, pp. 213–229.

[6] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Advances in Cryptology–EUROCRYPT 2005*. Springer, 2005, pp. 457–473.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 89–98.

[8] R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-based encryption with non-monotonic access structures,” in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 195–203.

[9] C. Mike, “Incident response with netflow for dummies.” Wiley.

[10] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*. IEEE, 2000, pp. 44–55.

[11] R. Brinkman, J. Doumen, and W. Jonker, *Using secret sharing for searching in encrypted data*. Springer, 2004.

[12] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: Improved definitions and efficient constructions,” *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[13] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in *Advances in Cryptology-Eurocrypt 2004*. Springer, 2004, pp. 506–522.

[14] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters, “Building an encrypted and searchable audit log.” in *NDSS*, vol. 4, 2004, pp. 5–6.

[15] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, “Charm: a framework for rapidly prototyping cryptosystems,” *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013. [Online]. Available: http://dx.doi.org/10.1007/s13389-013-0057-3

[16] http://www.icir.org/enterprise-tracing/Overview.html.

[17] https://tools.netsa.cert.org/silk/referencedata.html.