A Threshold Cryptography-based Framework for Secure Group Communication

Principal Investigators:

Sai Ganesh, Rabi N. Mahapatra
Dept. of Computer Science, Texas A&M University
ssai@cs.tamu.edu

Abstract

Threshold system *(t, n)* schemes construct *t* related secret pieces of the original secret called shadows and distributes them to participants in the system such that *t* or more participants in the system are required to reconstruct the original secret using their shadows. Using threshold schemes for authentication, secret sharing and data integrity, enhances group security by protecting against insider attacks and external adversaries. We propose a secure IP-based cryptographic algorithms and secure protocols (we call them primitives) that implements threshold group authentication and key agreement, secret group communication protocols and threshold digital signatures schemes, all of them using threshold cryptography concept that strongly couples participants and prevents Byzantine-type attacks. We assume an unreliable, connectionless layer providing traditional routing to build our secure architecture. We propose two novel threshold algorithms and show they are foolproof against passive eavesdropping and insider attacks. We claim that these cryptographic primitives flawlessly provide entity and data origin authentication and guarantees reliable key establishment and maintenance.

Last updated: August 07, 2002

1. Introduction

Study of group communication techniques is currently one of the most active areas of research. Newer services are moving away from unicast point-to-point delivery mechanism to group multicast model introduced by Deering [1]. Securing such multicast group communication services is not a trivial extension of securing unicast applications. For instance, emerging group network services such as online teleconferences shared whiteboards and shared network file systems require real-time protection to prevent external adversaries from accessing the system and interpreting sensitive information intelligibly. Enterprises may adopt hierarchical information flow model in which information relevant for a supervisor may not be relevant to their subordinates and thus requires sound protocols to ensure the secrecy. Selected corporate group members (e.g. Vice Presidents and Board of Directors) maintain time-sensitive financial information and must be prevented from leaking out of the group during this time period. Many other high value practical requirements arise in daily life and some of them are discussed by Desmedt [6].

Securing large group communication systems requires several forms of protection mechanisms such as access control to bar unauthorized participants, encryption to protect against intelligible eavesdropping, exchange of certificates between members to establish confidentiality and digital signatures to prove integrity of data while in transit. Traditional secure group communications assume these protections are established with the help of centralized trusted agents (or key distribution center KDC). All participants in the system unconditionally trust the central trust agent. Thus protection of this trusted servers and ensuring their saneness is vital to the protection of the entire secure system.

Owing to its centralization, trusted servers naturally attract attention for attacks. Information gained by an attacker from such attacks is unimaginable. The attacker compromises central servers in several ways by analyzing repeated traffic patterns and crypt-analyzing the information exchanged. Thus in a distributed trust scheme, replication of the key information achieves significant robustness with little overhead. As emerging applications move towards an all-distrusted environment such as dynamic peer group (like ad-hoc networks), it becomes more important to establish mutual trust purely based on agreement protocols, sometimes even without the trusted server. We propose an architecture that is a perfect distributed scheme in which data among the participating members are never the same and hence attacks on selective targets is never successful. Further, the scheme universally fails if more than $1/3^{th}$ of them is compromised.

Several practical implementations of secure group communication services have developed in the past years. Traditionally, these implementations have always assumed a trusted server playing a key role to provide authentication or high-integrity services. Kerberos distributed authentication service [19] relies on an authentication server to provide the principal with Kerberos ticket that contains the session key and an expiration time after which the ticket is invalidated. The reliance on the central server in stressed by the fact that the authentication server dictates both the session key and ticket expiration times. Verifier trusts the session key that is sent as a part of Kerberos ticket. Such schemes suffer from dual disadvantage in which a central authenticated server is attack-prone and the participants use symmetric session key for information exchange. Symmetric keys are susceptible against brute force key recovery and known-plaintext attacks. Our proposed authentication services not only eliminate the need for a central authority but also determine the shared key through implicit calculation. Also, each participant equally contributes to the key generation process as against a centralized generation.

Recent developments of security services are well focused to only one protection mechanism such as confidentiality or data integrity. Reliable atomic multicast high-integrity distributed service is proposed by Rampart [20]. This scheme proposes key distribution, access control and other services using the high-value integrity service. SecureRing [21] provides reliable ordering of messages and group membership services despite Byzantine faults caused by corrupted participants. The scheme is attractive

in that it ensures integrity among group participants and demonstrates Byzantine problems in groups. However, at this time of this proposal, we are unaware of any toolkit or suite that implements all necessary primitives for a complete secure system. These may include access control, authentication, secure group membership services together with high-integrity schemes.

Yet another security architecture IPSec [8] studied by IETF attempts to provides security services at IP layer. It is intended to provide cryptographically secure interoperable security services for upper layer services. Some of these services include: access control, data origin authentication, and confidentiality. IPSec architecture does not however specify standards for cryptographic algorithms that form the heart of the system. In addition, development of one primitive does not guarantee a seamless operation in multiple environments. For instance, defects in OS and poor quality of random number generation may degrade the performance of these algorithms. In contrast, our model proposes exact specifications of the cryptographic algorithms and the protocols making use of these algorithms in a layered approach.

We propose a secure IP-based toolkit that implements these three main cryptographic primitives over a reliable, group communication layer – threshold authentication and key agreement, secret sharing protocols and threshold digital signature scheme. Our solution is the only known currently that integrates all three together in a single unique framework. With the framework in mind, we intend to build a robust cryptosystem for problems of the type mentioned in the first paragraph and [6]. Interrelations among the primitives can also be studied during the course of developing the prototype.

The paper is organized as follows: section 2 gives a background of secure group communication, secret sharing and early study of threshold schemes; section 3 describes our proposed model, secure IP architecture and its components; section 4 details each cryptographic primitives in detail. We summarize the proposed research and give milestones of development in section 5 and conclude with our thoughts on implementation details in section 6.

2. Background

Secure group communication framework and applications has been studied for numerous years now. Several secret sharing schemes and protocols are analyzed since the early literature by Shamir [1]. In his cryptographic study, Shamir considers the problem of sharing cryptographic keys among a group that intends to protect a secret such that it is easy to construct the key with a fewer trusted group members but never with just one. Shamir calls the scheme (t, n)-threshold scheme whenever at least *t*-trusted members together are able to recover the secret. Applications reliably choose t that is safe and convenience to use. In a similar treatise, Blakley [9] simultaneously introduced the notion of safeguarding the cryptographic keys from geometrical perspective. Shamir and Blakley addressed the same problem but from different perspectives.

Group secret sharing schemes started with an introductory of the problem by Desmedt in [6]. Desmedt extends the 2-party public key cryptosystem to a group in which responsibilities of members differ. This arises due to the differing needs of the group members at different moments. Desmedt is the first to introduce a common group public key that maintains anonymity of the members. However, scalability and reliability of the group key is to be protected whenever the group changes.

Many problems on broadcast and multicast key agreement schemes in a secure group are being studied. Group agreement schemes protect the integrity of the communication using symmetric or asymmetric [2], [3] group key exchange schemes. Introduction of asymmetric RSA digital signature schemes [4], [5] removed the need to distribute key symmetrically, ensured privacy of communication and protects individuals against any forgery..

IETF Secure Multicast (MSEC) working group [22] studies the problem first introduced by Desmedt for securing group communications over internet where IP-layer multicast routing problems are deployed. A single trusted entity sets the security policy of the group and controls the group memberships. Functionalities implemented by these standards include: group and source authentication, cryptographic key generation and distribution and establishing group memberships.

Our proposed model addresses all of these problems. We think that an application can provide better secure service by integrating some of the important cryptographic elements in a toolkit such as authentication, mutual trust, group secret sharing and establishing data integrity.

3. Proposed Model

The proposed model assumes communication between two or more parties over an insecure channel that operates in peer-to-peer mode with no centralized trusted agent. Distributed trust among operating members is becoming popular paradigm in the face of attack to centralized trusted server resulting in a network-wide failure. For environments like dynamic peer groups (such as ad-hoc networks), it is even impossible to have a centralized entity that issues trust certificates.

3.1 Distributed Framework

Secure distributed group communication is studied recently in several literatures [29], [30], [31] and [32]. Our proposed model consists of n parties that participate in distributed communication of which at least t trusted participants are necessary to collaborate in a threshold scheme. All communication is assumed to be in an insecure channel with eavesdropper passively or actively monitoring channels. In other words, we do not differentiate between secure key-channel and unprotected data channel. All trusted parties are assumed capable of performing polynomial-time computation. We assume this holds for adversaries as well.

We assume a Byzantine-type distributed fault-tolerance environment in which k participants $(k < t)$ may be corrupted from exchanging their correct threshold shares. Theoretical solutions for Byzantine generals problem [23] show that if $k < n/3$, uncorrupted parties can establish consistent results amidst corrupted agents. Shoup [24] considers a similar problem in which a group of participants can robustly sign using non-interactive threshold signature with at most $t = k-1$ corrupted players in the system. A systematic proof of the protocol demonstrates the non-forge ability of threshold signature scheme with corrupted players assuming RSA scheme is secure. Cachin et al [25] presents a random oracle model that solves an asynchronous Byzantine agreement problem for number of participants $n > 3k$, a theoretical result already demonstrated in [23] for any general Byzantine agreement problem. Both [24], [25] however assume a trusted agent disseminate initial state information.

Our proposed system addresses Byzantine-type failures of the participants and fault tolerance applicable to threshold cryptosystem. We plan to study the following problems.

Byzantine Resilience: For a general Byzantine agreements problem, theoretical results show the maximum corrupted parties $k < n/3$ against which honest parties agree upon a consistent result. Articles [24] and [25] apply Byzantine model to threshold secret share system and demonstrate the same upper bound. Our proposed plan seeks to study Byzantine protocols for threshold key agreement and data integrity in proving the existence of upper bound of corrupted agents with no trusted agents mediating the protocol.

Protection against insider attacks. Protection of secret shadows of trusted entities is required against insider attacks. Parties inside the system may collude to deduce the shadows of others in a threshold system. It is thus required to study the optimal construction of concurrence system [15] such that no single party significantly losses. For instance, nodes that are relatively less mobile in an ad-hoc

network may considered to be trustworthy and hence may own more than one shared shadow whereas others own just one shadow each. Such a system thus is robust against attack but is not perfect [15] (unequal distribution of keying information).

Scalable secure group communication: Articles [26], [27], [28] study scalable secure group communication techniques. The implications of large group size on implementation of robust and fault-tolerant cryptographic primitives are an important aspect of our planned course of study. Thus, construction of large group threshold scheme with appropriate distribution of shadows to protect against insider attacks is a challenge. We plan to study the message and time complexities of such a large group size.

3.2 Secure IP –based toolkit

We propose an IP-based layered secure toolkit to implement cryptographic primitives. Most established network services today exist over IP and even attacks are prevalent in such networks. Thus it makes sense to establish strong security framework directly over IP that protects these upper layer services. Interoperability between existing IP nodes is thus guaranteed. Figure below show the components of our proposed model and their interfaces. Shaded blocks indicate our proposed component. Each of these components in our model is described in detail below.

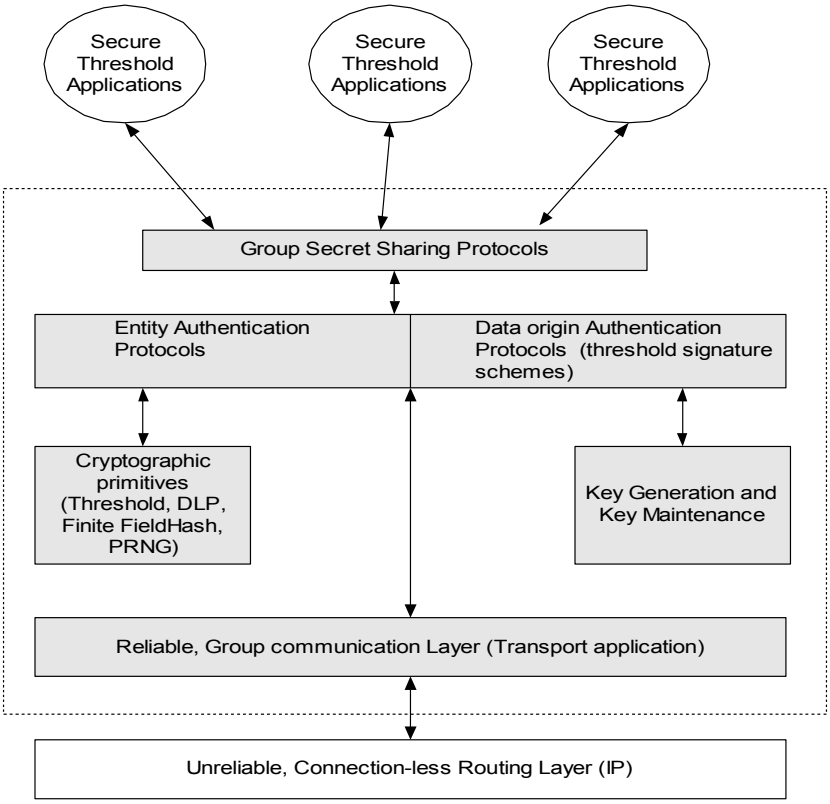

Fig. Secure IP-based Threshold Architecture

We propose a framework that depends on a reliable transport layer capable of providing time guarantees on packet deliveries (such as TCP). Message timeliness guarantees additional data integrity.

Reliable group communication layer manages group membership information (identity, network, public key, etc.) and member state information (session key, link reliability, last authenticated time, etc.)

of all participants participating in a session. Each host enters multiple sessions and each such session and session member information is maintained in a table in this layer. Reliable multicast may be implemented if required using existing protocols developed by IETF [13]. Sessions themselves are maintained by transport layers such as TCP and our layer becomes a transport application.

At the heart of the framework lie two black boxes: Cryptographic primitives & Key generation and maintenance. Cryptographic primitives provide a set of cryptographic algorithms for encryption, decryption, pseudo-random number generation (PRNG), message authentication code (hash functions), threshold encryption/decryption, threshold digital signature calculation etc. The interaction of these cryptographic primitives with other modules is maintained such a way that new algorithms may be developed and tested independently. Key generation and maintenance allow modular way of generating host authentication key, session keys and safe guarding private keys corresponding to the host. Hosts dynamically create and terminate membership and session information. Session information are maintained in local tables.

Entity authentication protocols are implemented for those sessions that request high reliability and trust among the members. Protocol themselves do not form a part of cryptographic primitives but rely on underlying group communication layer to implement the threshold authentication schemes. Entity authentication schemes rely on the possessed key as proof of their identity. This is termed as authentication-by-possession. The proposed authentication schemes exploit threshold cryptography concept to authenticate and establish group agreement keys. To demonstrate the criticality of the scheme, we present two examples of authentication primitives in section 4 that are modifications of existing schemes.

Data origin authentication (or data integrity) scheme uses threshold digital signing techniques that are novel and protects strongly against forgery. Data origin authentication combined with entity authentication forms cryptographically strong robust system in the absence of a centralized trusted agent. Data integrity is maintained per message between hosts in a session. Signatures are functions of message and authentication key corresponding to the host. Using session keys known to other participating members, security may be enhanced by applying additional transformation to the threshold signature scheme.

Group secret sharing protocols establish membership in a secure session. These protocols authenticate each other, exchange session keys, elect leader, vote for membership addition or removal, circulate trust information and terminates sessions. Secret sharing protocols use threshold scheme to establish session keys and trust information. The protocols may directly use the underlying cryptographic primitives and key management modules. Cryptographic details of secret sharing protocols are given in section 4.2.

Finally, several secure threshold applications may be developed using few or all of the features described earlier. Since the framework is modular and highly reusable such that the applications do not burden themselves with complex cryptographic implementations. Instead, the applications only add few additional lines of code in order to ensure the three important security considerations – authentication, secret sharing and data integrity.

4. Cryptographic Primitives

Our model proposes a set of core cryptographic algorithms that are based on threshold cryptography and these forms the heart of strong security. We use these algorithms to develop distributed applications. The goals of these algorithms are to provide: entity and data origin authentication, protection against message replay, and protection against forgery, data integrity and timeliness, and protection against insider attacks. These are implemented through authenticating principles and group secret sharing between members.

4.1 Authentication

Authentication establishes trust and identity of entities in the system and protects against transferability and impersonation irrespective of the previous number of operations with the claimant [9]. In our case, authentication and identification deals with establishing the entity authentication. Message authentication, which ensures trusted data origin and protection against message tamper, is taken care of through data integrity and digital signature schemes.

Several subtle forms of authentications are possible among entities and they are reported in [9] (see section 12.2). In our scheme, authentication is always coupled with secure key agreement. Entity authentication is performed by the initial key establishment protocol and possession of this agreed key is considered proof of the entity. We propose two authentication & key establishment techniques and claim that these are sufficient to mutually authenticate all members in the group and establish session keys between them. The framework guarantees: protection against passive key deduction, key confirmation and integrity and protection against insider attacks.

In an unfamiliar environment, no entity can be trusted. Instead, the entities mutually authenticate themselves using strong challenge-response authentication system. In addition to the strong challenge-response, we apply *t*-threshold scheme to protect against any passive eavesdropping.

In section 4.1.1 and 4.1.2 below, we demonstrate two example authentication schemes based on existing approaches. These are two-party mutual and group threshold authentication.

4.1.1 Two-party mutual authentication

Merkle [8] presented an authentication protocol between two directly talking peers assumed over an insecure channel. Passive eavesdropping does not prove to be effective especially for a circulation of a large number of *puzzles [8]*.

We propose a direct extension to this two-party scheme but with few changes. Claimant needs to be authenticated and verifier authenticates or rejects. Their roles are reversed in the second phase for mutual authentication.

To achieve this, both claimant and verifier choose a prime p and a generator under Z_p field. All key operations are assumed to occur under Z_p. Claimant chooses a set of N random keys $r_1,\ldots, r_N$ that are used to encipher a puzzle. Notation E_{ri} is used to represent this encryption function. Further, N puzzle keys $PK_1,\ldots PK_N$ are chosen by claimant of which *t* will eventually be used to calculate the authenticated key K between entities A and B. The steps are detailed as follows.

A B: $E_{r1}(PID_1, PK_1)$, $E_{r2}(PID_2, PK_2)$, …, $E_{rN}(PID_N, PK_N)$
A B: $E_{PK1}(PID_i, PK_i)$, $E_{PK2}(PID_j, PK_j)$,…, $E_{PKN}(PID_N, PK_t)$
A, B calculates $K = {}^{(PK_1 * .. * PK_t)} \pmod p$

At the end of these 2 steps, both A and B implicitly calculate the key K using only *t* puzzle keys out of N. Note that, the factor *t* is never published anywhere and B chooses arbitrarily. Hence an eavesdropper is unable to find out which puzzles keys are used cannot guess how many keys out of N is useful. For an eavesdropper to do brute force analysis requires $O(N^2)$ operations, due to the fact that the puzzles sent back from B to A in turn are encrypted using puzzle keys found in previous exchange from A to B. For large N, this factor grows strongly making eavesdropper brute force difficult.

The encryption function $E_r(PID, PK)$ is a simple, effective but solvable function having a reduced key space *k* [8]. This function is required by both A and B to solve the puzzle brute force (and hence

increases the eavesdropper complexity). To give an example, a discrete logarithmic (DLP) encryption function will require O(N) multiplications before obtaining one ID-Puzzle pair (by A, B or eavesdropper). Incorporation of DLP as the possible encryption function thus improves the total eavesdroppers complexity to $O(N^3)$.

4.1.2 Multi-party Authentication and Key Agreement

Several attempts have been made to extend two-party public key protocols to a more general *n*-party case. Some of the important schemes include the study made in [3], [6], [11] and [14]. Steiner et al [3] makes extends two-party Diffie-Hellman public key systems and presents multi-party key agreement protocols. They further add authentication using symmetric key (KDC assumed) authentication protocol between participating agents. Article [14] uses *(t, n)*-threshold scheme to authenticate and distribute conference keys between parties. In both the schemes, trusted key server is always assumed, making it difficult to apply in our context. Further, their protocols lead to asymmetric key distribution as the procedure converges towards the final key generation. This results in an unfair distribution of key-related information to certain participating nodes (especially the last node to receive all key information) making them suitable targets for attack.

Developing multi-party authentication scheme based on threshold cryptography can be challenging. Exchange of partial key information or *key projections* should be equally contributory from all participating nodes to avoid additional hints about key deduction. Also, if a polynomial interpolation type threshold scheme is used (as formulated in [1]) we expose *t* coordinates of the polynomial corresponding to the participants for deducing the secret key K. Once a key is deduced, it may be reused for that session and care should be taken to execute session key termination process. If not, a non-participating node may pretend to be a trusted share and misuse secure communication channels.

We propose a generic multi-party authentication and key agreement scheme that neither assumes a trusted key server nor distributes key information unevenly. Using Lagrange's interpolation formula as mentioned by Shamir [1], all nodes simultaneously calculate the keys with only partial information. In order to protect against information leakage against keys, we apply Chinese Remainder Theorem (CRT) as further level of indirection to calculate ordinate values (values along y-axis). At the beginning of the algorithm, each node randomly generates values along x-axis. The algorithm is described below.

As before, we assume every node chooses a prime p and a generator under Z_p field. All key operations are assumed to occur under Z_p. Traditionally choosing of a common prime and generator is considered to be the role of central key distribution server but here we assume this is given information, without loss of generality.

Threshold authentication and key agreement algorithm:

1. Node M_i generates non-prime integer X_i (mod p) at the beginning and broadcasts encrypted using a reduced keyspace with key k_i: $E_{ki}(ID_i, X_i)$.
2. Having received $E_{ki}(ID_i, X_i)$, each node searches the reduced key space and recovers X_i. M_i calculates $\text{Product\_}X_{x=0} = \prod_{k=1, k \neq i}^{n} \frac{x_k}{(x_k - x_i)}$, the partial Lagrange's interpolation at $x = 0$.
3. M_i M_j: $E_{xi}^{xj} \pmod p (g^{ri})$, where random r_i (mod p) generated by M_i and exponentiated g^{r_i}.
4. M_i then calculates $\text{CRT\_X} = g^{r1+r2+\ldots+rn} \pmod p$. M_i uses Chinese Remainder Theorem (CRT) to calculate its own ordinate value Y_i corresponding to X_i. That is, $Y_i = \text{CRT\_X} - (X_i * t) \pmod p$, where integer $t = g^s \pmod p$ and s = 1,2,3.. is the execution sequence number. This sequence number is used to change the value of Y_i after every round of session key, for added security.

5. All nodes simultaneously calculate the group session key using the Lagrange's interpolation formula as per [1]. Nodes M_i increment their sequence number by one (modulo p).

$$K \left(\prod_{i=1}^{n} Yi\right) * \text{Product\_X}_{x=0} = K \left(\prod_{i=1}^{n} Yi\right) * \prod_{k=1, k \neq i}^{n} \frac{x_k}{(x_k - x_i)}$$

Analysis of the algorithm shows that the eavesdropper require O(N) times the decryption complexity E_{ki}. If eavesdropper fails to deduce any of the X_i's, he is not successful in finding $\text{Product\_X}_{x=0}$ and CRT_X and hence the final key K. A successful deduction of all X_i's requires him to gather all partial exponents g^{ri} to calculate CRT_X adding an additional overhead of O(N). For eavesdropper to calculate the only possible ordinate value Y_i for this session key requires to keep track of the sequence number s and integer $t = g^s$. Like our previous two-way authentication procedure, incorporation of DLP as the possible encryption function improves the total eavesdroppers complexity to $O(N^2)$.

Although we claim these algorithms are robust against attacks, they are not verified analytically. Our next step is to prove that they are robust against insider and external attacks mathematically and we seek the worst-case time and message complexities of the algorithm and best-case eavesdropper attack probabilities.

4.2 Group Secret Sharing

Secret sharing schemes were devised for robust key management for cryptosystems. Given a cryptographic key, the secret piece has *t* related *shadow* pieces of which set of *k* will suffice to recover the original secret but no subset of *k-1* or fewer is able to reveal the secret. We have already devised two-party and multi-party schemes that combined authentication with key agreement. Several important secret sharing schemes are studied under [17] and [18] where no trusted agent is assumed.

Using a multi-party threshold scheme, each node contributed equally in determining the final key of the shared secret scheme. Nodes that distrust participants undertake a consensus agreement protocol that is democratic and involves equivalent transactions to determine the final secret K. We do not trust anyone in the sequence of steps but the consensus guarantees mutual authentication.

We propose to establish scalable protocols for group member changes to take place. All membership changes occur with democratic voting schemes such as the one described by Simmons in [18]. A member may be removed from the group and new session key formed upon voting so by majority of those having shares. The protocols are secure against insider Byzantine-type failure due to the fact that members are not able to deduce other's share with collusion. Due to extreme distrust among members, session keys are also short-spanned.

4.3 Data Integrity and Threshold digital signatures

Our third proposed primitive is data integrity, a property that detects any alteration of data during transmission. Using the basic threshold primitives, we develop a message authentication system (data integrity) that ensures that each message carries a valid identification (i.e. data origin authentication). Our data integrity approach provides the following: Data origin authentication, Detection of message loss or modification, and Protection against forgery from insiders. Timestamps may be used to guarantee timeliness and protection against message replay. Several threshold schemes have been investigated in [10], [11] and [12]. Article [12] gives various signature schemes and their security.

Conventional digital signatures are signed by a single entity that also holds its own secret key. The signature scheme resulting from threshold cryptography proves significantly stronger against internal and external adversaries. In threshold signature schemes, many parties collaborate each generating partial

signature and *t* of which are require to successfully sign the message. Parties verify their internal identities using their session keys. In a completely distrusted environment, secret keys are forgeable but threshold schemes protect strongly against both internal and external adversaries. In addition, signing a message by the entire group validates the global consensus among the group members.

Public key signature scheme is introduced by Diffie-Hellman in [2]. Rivest et. al [4] describes RSA digital signing procedure. Threshold based consensus and signing protocols are developed independently by Desmedt [6] and [7]. An authentication and shared signature scheme by Desmedt is studied in [18].

Although similar process employed in [18] can be directly borrowed and used in our model, we leave the topic open for further research and analysis.

5 Summary of Proposed Research and Milestones

As the current network moves towards newer group oriented services, secrecy and trust among the members becomes an important aspect. Thus a secure and robust group communication system depends on the strength of underlying cryptographic primitives. In our proposed toolkit, we have attempted to integrate three major cryptographic primitives and prove their strength through well-established threshold primitives. Our contribution to this study is many-fold: integrated toolkit for securing group communication based on threshold system that is tolerant against Byzantine-type attacks and novel threshold authentication and data integrity primitives. We seek to study scalability issues, generation & maintenance of key information and methods of constructing secure schemes as the group size becomes large.

6 Conclusion

Several implementation issues are to be considered in our model development. Although an unreliable, connectionless IP routing is assumed, a reliable transport that guarantees multiple session maintenance, reliability detection of loss and tampered packets, sequential delivery and timeliness are expected. TCP provides most of these except handling of session information by applications using TCP. Also, session state and other participant information need be maintained but TCP does not provide any explicit interface to do so. One possible solution is to explicitly query and exchange such information at the beginning of the secure sessions.

Our threshold primitives are not useful without applications. One can think of several applications ported to this model merely by interfacing with the primitives. Once nodes are authenticated with session keys, a number of useful features may be added to transport layer. For instance, several DNS domain servers in an autonomous systems group together forming a secure autonomous DNS servers that is resistant to DNS attacks. The root of this DNS master is elected the leader due to its position in the network and relative amount of information it maintains. If any of the nameserver slaves is attacked, it can be identified and eliminated from the group unianimously.

We claim that existing primitives can be implemented and tested and new ones added as required. The construction of the framework is modular and supports easier extension. Our next step is to prove the claims made here analytically and study the performance issues.

7 References

1. A.Shamir. How to share a secret. Communications of the ACM, 1979.

2. W.Diffe, M.Hellman, New directions in cryptography, IEEE Trans. Inform. Theory, Nov 1976.

3. M. Steiner, G. Tsudik, and M. Waidner. Diffie-Hellman Key Distribution Extended to Group Communication. In Proceedings of the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, March 1996.

4. R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and publickey cryptosystems, Comm. ACM,1978.

5. R.C. Merkle. Secure communication over insecure channels. Communications of the ACM, 1978.

6. Yvo Desmedt. Society and group oriented cryptography: A new concept, *Advances in Cryptology, CRYPTO 1987*.

7. Y. Desmedt, Y. Frankel, Threshold cryptosystems. In: Advances in Cryptology - Crypto '89, Proceedings, Lecture Notes in Computer Science 435 (G. Brassard, Ed.), Springer-Verlag, 1990.

8. S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, IETF RFC 2401, Nov 1998.

9. A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996. For further information, see http://www.cacr.math.uwaterloo.ca/hac

10. Y. Desmedt and Y. Frankel, "Threshold cryptosystems," in Advances in Cryptology - Proceedings of CRYPTO '89 (G. Brassard, ed.), vol. 435 of Lecture Notes in Computer Science, pp. 307-315, Springer-Verlag, 1990.

11. V. Shoup, "Practical Threshold Signatures", Advances in Cryptology - Eurocrypt 2000, Springer-Verlag LNCS 1807, pp.207-220, 2000.

12. S. Goldwasser, S. Micali, and R.L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput., April 1988, pages 281--308.

13. IETF Muticast security (msec) working group charter: http://www.ietf.org/html.charters/msec-charter.html

14. Chi-Sung Laih, Sung-Ming Yen: On the Design of Conference Key Distribution Systems for the Broadcasting Networks. INFOCOM 1993.

15. G.J.Simmons, Contemporary Cryptology: The Science of Information Integrity, IEEE Press 1991 edition.

16. I.Ingemarsson, G.J.Simmons, How mutually distrustful parties can set up a mutually trusted shared secret scheme, Intl. Assoc. Cryptologic Research (IACR) Newsletter, Jan 1990.

17. I.Ingemarsson, G.J.Simmons, A protocol to set up shared secret schemes without the assistance of a mutually trusted party, Advances in Cryptology, Proc. Eurocrypt 1990.

18. Y. Desmedt, Y. Frankel, Shared generation of authenticators and signatures. Advances in Cryptology – CRYPTO 1991.

19. B. Clifford Neuman and Theodore Ts'o. Kerberos: An Authentication Service for Computer Networks, IEEE Communications, 32(9):33-38. September 1994.

20. M. Reiter. The Rampart toolkit for building high-integrity services. In Theory and Practice in Distributed Systems. Springer- Verlag, 1995.

21. K.P. Kihlstrom, L. E.Moser and P. M. Melliar-Smith , ``The SecureRing Protocols for Securing Group Communication,", Proceedings of the IEEE 31st Hawaii International Conference on System Sciences, Kona, Hawaii (January 1998).

22. IETF Multicast Security (MSEC) charter http://www.ietf.org/html.charters/msec-charter.html

23. L. Lamport, R. Shostak and M. Pease, "The Byzantine Generals Problem", ACM Transactions on Programming Languages and Systems, 4 (3), pp.382-401, July 1982.

24. V. Shoup, Practical Threshold Signatures, Advances in Cryptology - Eurocrypt 2000.

25. C. Cachin, K. Kursawe, and V. Shoup. Random oracles in Constantinople: Practical asynchronous Byzantine agreement using cryptography. In Proceedings of the 19th ACM Symposium on Principles of Distributed Computing (PODC 2000), Portland, OR, July 2000.

26. Suvo Mittra. Iolus: A Framework for Scalable Secure Multicasting. In Prodeesings of ACM SIGCOMM '97 Conference, pages 277--288. ACM, September 1997.

27. S. Setia, S. Koussih, S. Jajodia, and E. Harder. Kronos: A scalable group re-keying approach for secure multicast. In Proceedings of IEEE Symposium on Security and Privacy, Berkeley, CA, May 2000.

28. C.K. Wong, M. Gouda, and S.S. Lam. Secure group communication using key graphs. In ACM SIGGCOM. ACM, September 1998.

29. Christian Cachin, Klaus Kursawe, Frank Petzold, and Victor Shoup. Secure and Efficient Asynchronous Broadcast Protocols. In Joe Kilian, editor, *Advances in Cryptology - Crypto 2001*, Lecture Notes in Computer Science, vol. 2139, Springer-Verlag, 2001.

30. Christian Cachin, Klaus Kursawe, and Victor Shoup. Random oracles in Constantinople: Practical asynchronous Byzantine agreement using cryptography. In *Proc. 19th ACM Symposium on Principles of Distributed Computing (PODC 2000), Portland, OR*, pages 123-132, July 2000.

31. Christian Cachin, Klaus Kursawe, Frank Petzold, and Victor Shoup. Secure and efficient asynchronous broadcast protocols (extended abstract). In Joe Kilian, editor, Advances in Cryptology: CRYPTO 2001, volume 2139 of Lecture Notes in Computer Science, Springer, 2001.

32. L. E. Moser, P. M. Melliar-Smith and N. Narasimhan, "The SecureGroup Communication System", Proceedings of the IEEE Information Survivability Conference, Hilton Head, SC (January 2000).