ISSN: 1832-5505

Applied GIS

Vol-8 Issue-04 Oct 2020

Shared Secret Key Agreement to increase in bandwidth and computation

V.Naresh[1], P.Anand[2], R.Raju[3]
Assistant professor[1,2,3]
NPV Nagar, Srimushnam, uddalore

Abstract— *Without worrying about network or node failures, a group of people may agree on a secret key with the help of a group key agreement protocol (GKA). The standard encryption-based group key agreement protocol can be robust against an arbitrary number of node faults, but the size of the messages broadcast by each player is proportional to the number of players, making current constant-round GKA protocols either efficient but nonrobust or robust but inefficient. In contrast, nonrobust group key agreement may be reached with just constant-size messages being broadcast by all participants. Using O(T)-sized messages for any T, we offer a unique 2-round group key agreement mechanism that can survive the loss of up to T nodes. We demonstrate that under the assumption of random node failures, the novel protocol implies fully-robust group key agreement with logarithmically-sized messages and predicted round complexity close to 2. Small constant factor increases in bandwidth and compute may be used to expand the protocol so that it can survive hostile insiders. The suggested protocol is secure under the (standard) Decisional Square Diffie-Hellman assumption.*

Index Terms— *Safety, redundancy, algorithms, and shared-secrets systems.*

1. INTRODUCTION

Group-oriented security procedures across unsecured network channels are becoming more important as the use of group applications grows. IP telephony, shared office space, encrypted meetings, and dynamic coalitions in emergency response settings are just some of the use cases. If all members of the group share a group-wide secret key, the standard security services necessary in such a situation, such as secrecy of group-wide broadcasts, may be provided in a highly efficient manner. Protocols for initial group key agreement (GKA) are designed with efficiency in mind. Computational, computational, and round complexities are all examples of efficiency measures. While it is true that all metrics have practical importance, round complexity might be more important, especially in a distributed system.

Proposed are many well-known, efficient two-round GKA procedures. However, if errors occur while the protocol is being executed, their performance suffers. When errors occur, the typical procedure (which lacks resilience) must begin over from the beginning. Current GKA procedures need to be strengthened to increase efficiency. To finish the procedure successfully is what we mean by "robustness" here.
despite flaws in play and/or communication.
An important practical issue is robust GKA. Loss of contact is possible among wirelessly communicating mobile nodes. The likelihood of a network failing also rises when routers fail and fragment the network or when malicious assaults occur.Imagine you're in a circumstance where there's an immediate need to have a secret meeting for rescue operations and military discussions before a crucial deadline. Then, effective GKA is required for
lessen the blow. Real-time communication in a group is how things work. Therefore, strong GKA is essential to raise QOS generally. Group keys are need to be renewed at regular intervals as per most security regulations.Therefore, it is necessary to re-run a GKA protocol (potentially often), and enhancement of GKA performance is crucial.
Imagine a set of nodes (routers or servers) operating in deep space, where there is no reliable connection to the outside world. Restarting a GKA treatment when just one patient has failed to respond

If it doesn't work, it's going to cost a lot of money. Assuming a stable broadcast channel, making a GKA protocol resilient to node failures by restarting is a piece of cake.

every time a problematic player is found, the process must start again from the beginning. However, doing so would increase the round complexity of the protocol, as well as all other protocol expenses, by a factor of the number of errors. By running many copies of a nonrobust constant-round GKA protocol in parallel, one for each conceivable subset of nonfaulty players, it is possible to create robust constant-round GKA protocols. A resilient and constant-round protocol would have unacceptable increases in communication and computation costs of 2n. Because of this, one may wonder whether there are constant-round GKA methods that are

more efficient and resistant against node failures. A malevolent player that sends random messages that don't adhere to the protocol is another source of robustness issues. The enemy's objective is to cause a malfunction in the protocol. Unlike DH and other two-party key agreement protocols, GKA requires the reuse of contributions. The key is not agreed upon if any of the messages deviates from the protocol structure, such as when one message uses a different contribution to calculate a value than the others. Some may believe that random message transmission may be prevented with message/player authentication. Authentication, however, just checks the message/player's legitimacy and not whether or not the player used the right format. However, the protocol disruption attack brought on by the rogue player is not dealt with by popular authenticated GKA protocols.

1.1 Contributions

1. This investigates the issue of efficiency versus robustness to node failures, for constant-round GKA protocols working in a reliable broadcast communication medium. We describe how to achieve a natural trade-off between message size and the desired level of fault-tolerance in a GKA protocol.

2. It proposes a new 2-round GKA scheme, which tolerates up to T node failures, using O(T) sized messages, for any T. To exemplify the usefulness of this flexible trade-off between message size and fault tolerance, we demonstrate that in a realistic setting of random node faults. This protocol implies a fully robust GKA protocol with O(np) sized messages and expected round complexity close to 2.

3. This extends robust GKA protocol to withstand the disruption attack by the malicious insider. Our extension efficiently not only identifies the malicious. Player who does not follow any protocol step, but also allows the rest of the players to agree upon a key.

4. It proves the security of the proposed protocols under the standard Decisional Diffie-Hellman and Decisional Square Diffie-Hellman assumptions.

ISSN: 1832-5505

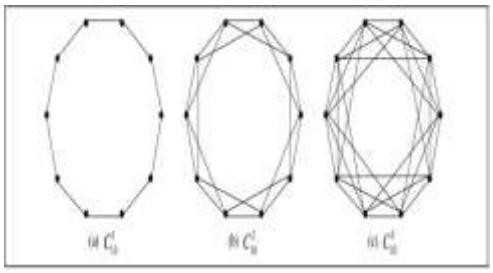
Applied GIS

Vol-8 Issue-04 Oct 2020

3. PRELIMINARIES

3.1 Cryptographic Setting

Let G be a cyclic group of prime order q, and let g be its generator. We assume the DDH and Square-DDH problems are hard



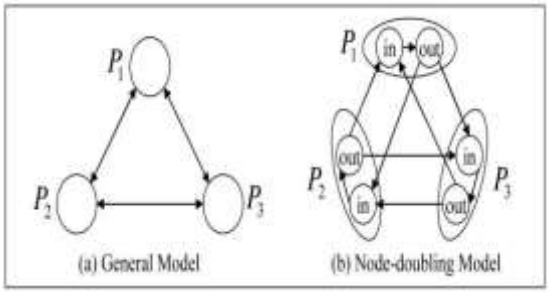
in G. For example, G could be a subgroup of order q in the group of modular residues Zp*
S.t. p _ 1 divides q.

3.2 Signatures of Knowledge

Zero-knowledge proofs of knowledge allow a prover to demonstrate the knowledge of a secret
w.r.t. some public information such that no other information is revealed in the process. The interactive zero-knowledge proof protocols proven zero-knowledge in an honest-verifier model can be performed no interactively with the help of an ideal hash function H.We refer to the resulting constructs as signatures of knowledge. One example is the Schnorr signature scheme, where a signature can be viewed as a proof of knowledge of the discrete logarithm of the signer's public key made no interactive. In the following, we introduce a variant of the Schnorr signature.

4. ROBUST GROUP KEY AGREEMENT PROTOCOLS

It describe two-rounds robust GKA protocol that tolerates T faults with O(T)-sized messages. This section purposes, we explain how the nonrobust GKA protocol of Burmester-Desmedt (BD) generalizes to a (fully) robust 2-round GKA protocol at the



cost of increasing the length of the constant-sized messages of the BD protocol to $O(n^2)$-sized messages. We call this robust generalization of the BD protocol BD-RGKA and show that the protocol remains secure under the same DDH assumption required for the underlying BD protocol.

Next Section, using the technique of node-doubling, we show that the BD-RGKA protocol can be modified to retain full robustness with message size reduced to 2n group elements. Moreover, with randomness reuse, we can further reduce the message size to just n group elements per player. We call the resulting protocol RGKA and show that it is secure under the Square-DDH

assumption. This leads to our main contribution, the T-RGKA protocol, which is a version of the above RGKA protocol in which each player broadcasts only 2T group elements.

5. ROBUST GROUP KEY AGREEMENT EXTENSION

Here, we strengthen the already-robust GKA protocol such that it can resist a protocol disruption assault launched by an adversarial player. While the basic robust GKA protocol considers missing gadgets (due to network or device failures), the extended robust GKA protocol also checks whether or not the gadgets created by each participant are compatible with the protocol methodology. A defective gadget is one that was not produced in the proper way. Remembering that the key agreement protocol fails in the absence of a connected gadget chain that encompasses all nodes, it follows that a broken gadget would also cause the protocol to fail. However, if a

malfunctioning device can be identified, the RGKA protocol can still function well without it. An opponent may force a participant to pause the protocol at any time during its execution and steal the key. We also take into account a malevolent opponent who takes part in the protocol but acts in an unpredictable manner.

5.2 Robust GKA Extension with O(n) Batch Verification

The RGKA-EXT protocol, where n players generate n-1 gadgets on a common exponent, requires n^2- n instances of EPDL verifications of gadgets. Verification of correct gadgets is the greatest factor, contributing to computational cost in the protocol. The technique proposed in addresses batch verification of common exponent in a threshold decryption scheme based on the following theorem.

CONCLUSION

A unique 2-round GKA protocol was presented in this study, providing a straightforward trade-off between message size and the required degree of fault tolerance. The new protocol may be modified to allow for the presence of malevolent insiders while only increasing the cost of communication and computation by a minor constant factor. The suggested protocol is safe when the (default) Decisional Square Diffie-Hellman model is used.

REFERENCES

1. 1) Y.-M. Tseng, "A Robust Multi-Party Key Agreement Protocol Resistant to Malicious Participants," The Computer J., vol. 48, no. 4, pp. 480-487, 2005.
2. 2. R. Aditya, K. Peng, C. Boyd, E. Dawson, and B. Lee,"BatchVerification for Equality of Discrete Logarithms and Threshold Decryptions," Proc. Second Int'l Conf. Applied Cryptography and Network Security (ACNS), pp. 494-508, 2004.
3. 3. "Provably Authenticated Group Diffie-Hellman Key Exchange—the Dynamic Case," Proc. Conf. Asiacrypt '01, December 2001, authors: E. Bresson, O. Chevassut, and D. Pointcheval.
4. "A Robust Multi-Party Key Agreement Protocol Resistant to Malicious Participants," by Y.-M. Tseng, published in The Computer J., volume 48, issue 4, pages 480-487, 2005.
5. C. Cachin and R. Strobl, "Asynchronous Group Key Exchange with Failures," Proc. 23rd Ann. ACM Symp. Principles of Distributed Computing (PODC), pp. 357-366, 2004.