Shared Secret Key Agreement to increase in bandwidth and computation

R.Rampriya M.E., K.S.Giriprasath Asst Prof.
Department of CSE, DSCET, Mahabalipuram
Chennai

Abstract:
A robust group key agreement protocol (GKA) allows a set of players to establish a shared secret key, regardless of network/node failures. Current constant-round GKA protocols are either efficient and nonrobust or robust but not efficient; assuming a reliable broadcast communication medium, the standard encryption-based group key agreement protocol can be robust against arbitrary number of node faults, but the size of the messages broadcast by every player is proportional to the number of players. In contrast, nonrobust group key agreement can be achieved with each player broadcasting just constant-sized messages. We propose a novel 2-round group key agreement protocol, which tolerates up to T node failures, using O(T)sized messages for any T. We show that the new protocol implies a fully-robust group key agreement with logarithmic-sized messages and expected round complexity close to 2, assuming random node faults. The protocol can be extended to withstand malicious insiders at small constant factor increases in bandwidth and computation. The proposed protocol is secure under the (standard) Decisional Square Diffie-Hellman assumption.

Index Terms—Group key agreement, fault- tolerance, algorithms, and security.

1. Introduction

The growth of group applications triggers the need for group-oriented security mechanisms over insecure network channels. The applications include IP telephony, collaborative workspaces, secure conferences, as well as dynamic coalitions common in law enforcement and disaster rescue scenarios. Standard security services required in such group settings, e.g., confidentiality of group wide broadcasts can be very efficiently achieved if all group members share a group-wide secret key. The early design of contributory group key agreement (GKA) protocols focuses on the efficiency of initial GKA. Efficiency metrics Include computation, computation and round complexities. Although each metric is important in practice, the round complexity can be more crucial, particularly in the distributed computing environment.

Several well known efficient two-round GKA protocols are proposed. However, their performance degrades if faults occur during the protocol execution. Faults cause the normal protocol (without robustness) to be restarted from the scratch. To improve performance, current GKA protocols must be made robust. In this context, robustness refers to the ability to complete the protocol, despite player and/or communication faults.

Robust GKA is a serious concern in practice. Mobile nodes that communicate over a wireless medium can loose connectivity. Router failures, causing network partitioning as well as malicious attacks, also increases the failure probability.

Consider an emergent situation where some secure meeting for rescue missions and military negotiations must be held prior to a special time. In that case, robust GKA is prerequisite to minimize damage. Group communication operates on a real-time setting. Thus, robust GKA is crucial to improve the overall QOS. Security policies usually dictate that group keys must be refreshed periodically.Thus, a GKA protocol needs to be re-run (perhaps often), and improving GKA performance is essential.

Consider a group of entities (routers or servers) in extreme environments, such as deep-space, that lack continuous network connectivity. In such a setting, re-starting a GKA protocol, because a single participant failed, results in inordinately expensive costs. Assuming a reliable broadcast medium, a GKA protocol can trivially be made robust to node failures by restarting the protocol from scratch, whenever a faulty player is detected. However, this would multiply all protocol costs by the number of faults, including the round complexity of the protocol. Robust constant-round GKA protocols can be achieved by executing parallel instances of any standard, i.e., nonrobust, constant-round GKA protocol, one instance for every possible subset of nonfaulty players. Such protocol would be robust and constant-round, but its communication and computation costs would grow by an inadmissible factor of 2n. This gives rise to the question whether there exist constant-round GKA protocols that are robust to node failures at more reasonable efficiency costs. Another robustness problem is caused by a malicious player, who sends arbitrary messages not correctly following the protocol. The goal of the adversary is to disrupt the protocol. Unlike two party key agreement protocol, such as DH, GKA requires to use contributions more than once in the protocol. If any of the messages is not following the protocol structure, e.g., inconsistently computed values by using a different contribution, then the key is not agreed upon. One may think that message/player authentication can prohibit from sending random messages. However, authentication examines only authenticity of message/ player, but does not determine if the player has sent the correct form of messages. In fact, well-known authenticated GKA protocols do not address the protocol disruption attack due to the malicious player.

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.

3. PRELIMINARIES

3.1 Cryptographic Setting

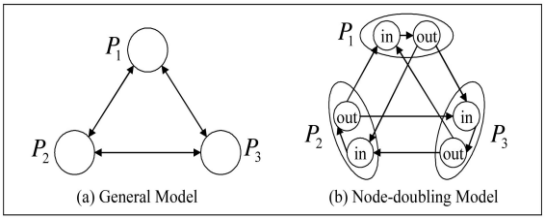
Let G be a cyclic group of prime order q, and let g be its generator. We assume the DDH and Square-DDH problems are hard in G. For example, G could be a subgroup of order q in the group of modular residues Zp*
S.t. p _ 1 divides q.

3.2 Signatures of Knowledge

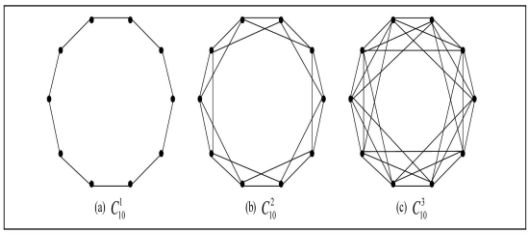
Zero-knowledge proofs of knowledge allow a prover to demonstrate the knowledge of a secret w.r.t. some public information such that no other information is revealed in the process. The interactive zero-knowledge proof protocols proven zero-knowledge in an honest-verifier model can be performed no interactively 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

In this section, we extend the robust GKA protocol to withstand the protocol disruption attack that the malicious player may attempt. While the basic robust GKA protocol considers missing gadgets (due to network or device failures), the extended robust GKA protocol additionally examines whether or not the gadgets generated by each player are consistent with the protocol algorithm. We refer to a gadget not correctly generated as a faulty gadget. Recalling that without a sequence of connectable gadgets, which covers the set of all nodes, the key agreement protocol fails, it is clear that a faulty gadget would lead to the protocol failure as well. However, the RGKA protocol can be still robust by excluding it, if a faulty gadget can be detected. The adversary can make each player stop at an arbitrary moment in the protocol execution to learn the key. We also consider a malicious adversary, who participates in a protocol but executes the protocol in an erratic way.

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.

6. CONCLUSION

In this paper, we proposed a novel 2-round GKA protocol that offers a natural trade-off between message size and the desired level of fault tolerance. The new protocol is also extensible to tolerate malicious insiders at small constant factor increases in communication and computation cost. The proposed protocol is secure under the (standard) Decisional Square Diffie-Hellman assumption.

REFERENCES

1. Y. Amir, C. Nita-Rotaru, J.L. Schultz, J.R. Stanton, Y. Kim, and G.Tsudik "Exploring Robustness in Group Key Agreement," Proc. Int'l Conf. Distributed Computing Systems (ICDCS), pp. 399-408,2001.
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. E. Bresson, O. Chevassut, and D. Pointcheval, "Provably Authenticated Group Diffie-Hellman Key Exchange—the Dynamic Case," Proc. Conf. Asiacrypt '01, Dec. 2001.M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System (Extended Abstract)," Proc. Conf. Advances in Cryptology (EUROCRYPT '94), pp. 275-286, 1994.
4. D. Boneh, "The Decision Diffie-Hellman Problem," Proc. Third Int'l Symp. Algorithmic Number Theory, pp. 48-63, 1998.
5. D. Chaum and T.P. Pedersen, "Wallet Databases with Observers,"Proc. 12th Ann. Int'l Cryptology Conf. Advances in Cryptology, pp. 89-105, 1992.
6. 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.
7. A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," Proc. Conf. Advances in Cryptology (CRYPTO), pp. 186-194, 1986.
8. J. Katz and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange," Proc. Conf. Advances in Cryptology (CRYPTO), pp. 110-125, 2003.
9. A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996.
10. C.-P. Schnorr, "Efficient Identification and Signatures for Smart Cards," Proc. Conf. Advances in Cryptology (CRYPTO), pp. 239-252, 1989.
11. D.G. Steer, L. Strawczynski, W. Diffie, and M.J. Wiener, "A Secure Audio Teleconference System," Proc. Conf. Advances in Cryptology (CRYPTO), pp. 520-528, 1988.
12. M. Steiner, G. Tsudik, and M. Waidner, "Key Agreement in Dynamic Peer Groups," IEEE Trans. Parallel and Distributed Systems, vol. 11, no. 8, pp. 769-780, 2000.
13. 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.