Multi-channel Protocols for Group Key Agreement in Arbitrary Topologies

Ford-Long Wong and Frank Stajano
University of Cambridge
Computer Laboratory

Abstract

We consider group key agreement (GKA) protocols, used by a group of peers to establish a shared secret key for multicast communications. There has been much previous work to improve the security, efficiency and scalability of such protocols. In our work, we describe secure schemes which utilize auxiliary channels in addition to that afforded by the open medium of radio. Such channels are often present in a human-centric pervasive ad-hoc networking scenario, though often neglected. We show that auxiliary channels can reduce public-key operations, reduce computational complexity, and strengthen security against an active adversary on the open channel, and against an eavesdropper on the auxiliary channels. Group key agreement protocols are usually often contextualized by their topology. We applied multi-channel schemes to different topologies, and found that the ideal topology may be different for different channels.

1. Introduction

A group of people in a face-to-face business meeting wish to establish a common key to protect multicast transmissions among their mobile phones or laptops. Since they are all there together, the problem appears at first trivial. Can't one of them just broadcast a random key to everyone around? No, because people outside the room might overhear it. Can't he write the key on the whiteboard for all to see[1]? No, because firstly we assume that a good key will be too long to transcribe, and secondly the spy with binoculars, or the cleaner, will also learn the key. And also because, in both cases, the key is generated by only one participant.

So, informally, our aim is to build a contributory protocol that will produce a strong shared key, known only to the people at the meeting, even in the presence of active attackers on the radio channel and passive attackers on the other channels.

[1]This would be an instance of using a different channel [21].

How can the protocol recognize who is at the meeting (for the purpose of excluding others)? Some previous GKA protocols have assumed that all legitimate participants share pairwise keys. Some proved to be vulnerable.

Our protocol has no need to recognize pre-established shared keys: it recognizes the participants by the fact that they can influence button presses on each other's devices during the protocol's run. It is therefore an instance of a multi-channel protocol that exploits physical presence, ideally suited to the pervasive computing scenario of an ad-hoc group of human players equipped with personal devices.

2. Related Work

For two participants, the 2-party Diffie-Hellman key exchange is well-studied. Over the years, researchers have made multi-party extensions to DH for group communications [6, 18, 19, 3]. Some use passwords [1] or public keys [12] to bootstrap; others make no assumptions about the topology [10]. But straightforward multi-party extensions to the 2-party DH can turn up subtle vulnerabilities [14, 15].

The use of auxiliary channels in key agreement has been studied. Balfanz et al [2] assume a high-bandwidth auxiliary channel, and Gehrmann et al [9] assume that the channel is low-bandwidth but confidential. Hoepman [11], and Wong and Stajano [21] refute the common implicit assumption that the auxiliary channel is confidential. These work on auxiliary channels have covered mainly the 2-party case, and only very briefly the multi-party case [2].

3. Passwords

Asokan and Ginzboorg [1] gave a good overview of different topologies for ad-hoc multi-party key agreement, and provided password-based solutions.

Their first protocol is a multi-party extension of the 2-party EKE [4], and the group has a 'star' topology. They modified it to make it 'contributory'.

Definition 1 : A key agreement protocol is *contributory* if each party contributes equally to the key.

The second of their protocols uses a Diffie-Hellman type of key agreement, where the group topology is essentially a linear chain. Their third protocol uses DH multi-party key exchange on a 'Hypercube' [3].

The drawback of using passwords lies not so much in the limits of human memorability, since the password would probably be disclosed to all participants immediately before the protocol is run. The main problem is the presence of an eavesdropper on the channel on which the password is shared, be it a visual, audio or other channel which has no privacy. This problem was first raised by Hoepman [11]. This weakness extends likewise to multi-party computations. If the password is compromised, the three above-mentioned protocols are all vulnerable to active attacks. Apart from that, the second protocol is related to *Cliques*, which is susceptible to an interesting generic insecurity, to be considered next.

4. Cliques-Type Authenticated Group Key Agreement

The Cliques Group Key Agreement protocol suite [19] uses basically a linear chain structure. There are many variants in the Cliques family: some are basic group key agreement (GKA) protocols, secure against a passive adversary, while others are authenticated group key agreement (AGKA) protocols, secure against an active adversary. In the latter group of variants, the group members are assumed either to initially share strong secure pairwise secrets with the group leader, or else they initially share pairwise secrets with all other members.

4.1. Basic Cliques Design

We review the Cliques design. Each group member M_i selects a random key contribution r_i, and the final group key is $\alpha^{r_1 r_2 \cdots r_n}$ where α is a generator. For the AGKA variants, the group leader M_n shares with each of the other members M_i a pre-established secret key K_{in}.

Round *i* $(1 \leq i < n)$:

$$\begin{aligned} M_i \rightarrow M_{i+1} \quad &: \quad \{\alpha^{\frac{r_1 \cdots r_i}{r_j}} | j \in [1,i]\},\ \alpha^{r_1 \cdots r_i} \\ &\equiv \quad C_{i,1}, \cdots, C_{i,i}, C_{i,i+1} \quad \equiv \quad C_i \end{aligned}$$

Round *n* :

$$\begin{aligned} M_n \rightarrow All\ M_i \quad &: \quad \{\alpha^{\frac{r_1 \cdots r_n}{r_j} K_{jn}} | j \in [1, n-1]\} \\ &\equiv \quad C_{n,1}, \cdots, C_{n,n-1} \quad \equiv \quad C_n \end{aligned}$$

In Round i, M_i generates and sends to M_{i+1} a set of exponentials — we write these as $C_{i,1}, \cdots, C_{i,i}, C_{i,i+1}$, and the whole set as C_i. The set C_i is not independently generated by M_i, but is generated from the earlier set C_{i-1} which has been received from M_{i-1}. In Cliques, exponentiating a value by r_i is termed the 'r_i-service'. In Round n, M_n adds his contribution r_n and pairwise keys K_{in}, and broadcasts the set of sub-keys. All members can calculate the group key. For the unauthenticated case, K_{in} is omitted.

4.2. An Attack against the IKA Property

Pereira and Quisquater [14, 15] have discovered and proved generic insecurities of Cliques AGKA protocols, whenever the group size is at least 3, using a strand spaces analysis approach. They found that the implicit key authentication (IKA) security property, for instance, is not achieved.

Definition 2 : *Implicit key authentication* is the property that one party is assured that no other party aside from a specifically identified party may gain access to a particular secret key.

Consider a group size of 3. Say, the intruder M_I wants to fool member M_2. M_1, M_2 and M_3 are legitimate participants in the first protocol run, while M_I, M_2 and M_3 are participants in the second run. In the *second* run, M_I replaces the input values of M_3's $r_3 K_{I3}$-service and $r_3 K_{23}$-service with a random value he knows, say α^y. M_3 then broadcasts $\alpha^{y r'_3 K_{I3}}$ and $\alpha^{y r'_3 K_{23}}$. M_I replaces the input of M_2's r_2-service with $\alpha^{y r'_3 K_{I3}}$, then M_2 would send $\alpha^{y r'_3 K_{I3} r_2}$. Intruder M_I hears this, and can exponentiate it by K_{I3}^{-1} to obtain $\alpha^{y r'_3 r_2}$. He now has possession of a pair $(\alpha^{y r'_3 K_{23}}, \alpha^{y r'_3 r_2})$. Finally, M_I replaces $\alpha^{r_1 r_3 K_{23}}$ with $\alpha^{y r'_3 K_{23}}$ in M_3's broadcast message in the *first* protocol run. M_2 would be fooled into computing $\alpha^{y r'_3 r_2}$ as the group key, which M_I knows, for the first protocol session. M_2 ends up sharing a key with the attacker, hence the IKA property is violated.

4.3. Other Attacks

Attacks on the following properties were also described by Pereira and Quisquater [14, 15].

Definition 3 : *Perfect forward secrecy* is the property that the compromise of long-term keys does not compromise past session keys.

Definition 4 : *Resistance to known-key attack* is the property that compromise of past session keys does not allow compromise of future session keys, nor allow imper-

sonation by an adversary.

Briefly, in the first attack, say a long-term pairwise key K_{13} is compromised by intruder M_I, and he can replace the input of the r_3K_{13}-service with $\alpha^{r_1r_2}$. When M_3 adds r_3 to the sub-key and broadcasts it, M_I can hear the message sent to M_1, and he can compute the key $\alpha^{r_1r_2r_3}$ established between M_2 and M_3.

In the known-key attack, two protocol runs are required. In the first run, M_I modifies the input of the r_3K_{13}-service to $\alpha^{r_1r_2}$. M_2 and M_3 share the key $k = \alpha^{r_1r_2r_3}$, while M_1 computes the key $k_1 = \alpha^{r_1r_1r_2r_3}$. We assume k is compromised by M_I. In the second run, each member generates new contributions. M_I modifies the input of the r_3K_{13}-service to $\alpha^{r_1r_2r_3}$ (known from earlier), and also alters the cardinal value $\alpha^{r_1'r_2'}$ to $\alpha^{r_1r_2r_3K_{13}}$ (overheard earlier). M_3 then computes the group key as $k_2 = \alpha^{r_1r_2r_3K_{13}r_3'}$, and at the same time also sends M_1 the sub-key $\alpha^{r_1r_2r_3K_{13}r_3'}$, which are equal. M_I hears this, and now can impersonate M_1 to M_3.

4.4. Cliques Assumptions Re-visited

We re-visit the assumptions underlying the Cliques design. It is observed in the AGKA variants that strong pairwise keys are assumed to have been pre-established between the group controller and the $n-1$ members, or even among all members. Applying these keys in Round n is meant to achieve authentication. One may guess that these keys must have been established via authenticated 2-party DH between pairs of members, before the AGKA process.

Despite the presence of these keys, the designers decided not to use conventional cryptography. In retrospect, we consider this is an unnecessary barrier for achieving authentication. Encryption is today not a prohibitively costly operation, and some MACs use cipher algorithms at their core. We highlight the curious situation of not leveraging these keys in conventional cryptography to guarantee confidentiality and authenticity.

4.5. Multi-Channel Augmentation

Our main contribution is to address the vulnerabilities mentioned in Sections 4.2 and 4.3, with recourse to auxiliary channels. We argue that the auxiliary channels often exist in a pervasive computing environment, though they have often been not well-recognized or well-modelled, but may now be leveraged profitably to bootstrap authenticated group key agreement. Our approach is to augment both Round i and Round n with auxiliary channels.

Protocol Objective : To assure implicit key authentication, perfect forward secrecy and resistance to known-key attack for contributory group key agreement under conditions of an active adversary operating on the open channel.

The MACs used in our solution are keyed from randomly generated keys. The basic building block is derived from the surprising result of the asymmetric pairing situation given in Protocol Trace 5 in Wong and Stajano [21]. We adapt that to Round i of the original protocol, as shown in Figure 1.

I_i and I_{i+1} are M_i's and M_{i+1}'s identifiers respectively. M_i chooses a short random nonce R_i, a long one-time key K_i, and produces MAC_i based on

$$MAC_i \quad = \quad MAC_{K_i}(I_i \mid I_{i+1} \mid C_i \mid R_i)$$

Assumption 1 : Auxiliary channels (such as 'visual' and 'pushbutton' channels), possessing the property of *data-origin authenticity*, exist between group members.

Assumption 2 : The adversary acting on these auxiliary channels is limited to be a *passive* adversary, who can eavesdrop on messages but cannot modify them.

The protocol does not rely on long-term passwords (as in Cliques AGKA) nor the confidentiality of the auxiliary channels (as in MANA III [9]). Values visually exchanged this way today run the risk of being eavesdropped by pervasive CCTVs [21].

#	Ch	M_i	msg	M_{i+1}
1	RF		$- C_i \mid MAC_i \rightarrow$	
2	PB		$\leftarrow \text{ack} -$	
3	V		$- R_i \rightarrow$	
4	RF		$- K_i \rightarrow$	
				Verify MAC_i
5	PB		$\leftarrow \text{outcome} -$	

Figure 1. Augmented Round *i*

The column 'Ch' refers to the type of channel utilised. The 'RF' channel has high bandwidth, but is vulnerable to an active attacker, who can eavesdrop on as well as modify messages. The 'V' refers to a low-bandwidth *unidirectional* visual channel of limited bandwidth, commonly found in devices as a screen and keypad, and including two human operators. The 'PB' channel is a 'push-button' unidirectional channel that is allowed to have bandwidth as low as 1 bit, and whose operation is also mediated by human operators. It can use the same 'V' channel too if providing an additional channel is expensive. Under the assumptions, we believe:

Proposition 1 : The advantage of an active adversary who modifies $\{C_i|MAC_i\}$ and attempts to fool M_{i+1} into believing it is from M_i, is of the order of the probability of M_I correctly guessing R_i, i.e. the inverse of R_i's length.

Proposition 2 : The advantage of a passive adversary who attempts to compute the session key from C_i is of the order of the Computational Diffie-Hellman problem on the group.

Thus, without requiring a confidential channel, nor pre-established pairwise keys between members M_i and M_{i+i}, the augmented Round i guarantees the data-origin authenticity of the exponentials. An active adversary cannot re-write a chosen C_i at will, required for an attack.

A variant of the augmented Round i can also be derived from Hoepman's protocol [11], as shown in Figure 2. H_1 and H_2 are different hash functions. The two variants are largely equivalent: one uses a MAC and a RNG call, while the other uses two hashes.

#	Ch	M_i	msg	M_{i+1}
1	RF		$- H_1(C_i) \rightarrow$	
2	PB		$\leftarrow \text{ack} -$	
3	V		$- H_2(C_i) \rightarrow$	
4	RF		$- C_i \rightarrow$	
				Verify $H_1(C_i), H_2(C_i)$
5	PB		$\leftarrow \text{outcome} -$	

Figure 2. Augmented Round *i* variant derived from Hoepman's protocol

Proposition 3 : On successful completion of a Round i, M_{i+1} has assurance that the received C_i originates from a human-verifiable member M_i with high probability.

In a similar vein with Figure 1, we augment Round n with the trace in Figure 4.

#	Ch	M_n	msg	All M_i
1	RF		$- C_n \mid MAC_n \rightarrow$	
2	PB		$\leftarrow \text{ack}_1 -$	
	...		...	
	PB		$\leftarrow \text{ack}_{n-1} -$	
3	V		$- R_n \rightarrow$	
4	RF		$- K_n \rightarrow$	
				Verify MAC_n
5	PB		$\leftarrow \text{outcome}_1 -$	
	...		...	
	PB		$\leftarrow \text{outcome}_{n-1} -$	

Figure 3. Augmented Round *n*

Succesfully verifying the authenticity of M_n's multicast message requires $n-1$ 'ack' and $n-1$ 'outcome' messages to be properly transmitted and registered via human-verifiable 'V' and 'PB' channels. M_n must wait for all the 'acks' to be received before releasing R_n. This series of protocol steps assure that C_n cannot be modified.

Having data-origin authenticity enforced on the point-to-point Round i messages, and the multicast Round n message, renders these messages unforgeable by Pereira et al's active adversary, and completely foils the attacks.

Proposition 4 : Assuming no colluding members, if C_i's *and* C_n cannot be modified by an intruder without detection, then the attacks against the IKA, PFS and resistance to KKA properties cannot succeed with high probability.

The augmentation of Round n is in fact recommended more for the GKA scheme (i.e. no pairwise keys) than for the AGKA scheme. Doing so yields the twin benefits of saving the computation and latency of at least $n-1$ pairwise key establishment rounds, and transforming an otherwise *unauthenticated* scheme into an *authenticated* scheme.

4.6. Costs and Savings

Enhancement to security notwithstanding, the disadvantages of the technique include the increased latency per round and increased user intervention. The increased latency is mitigated by the fact that the scaling per round is by a constant factor. The attendant message complexity has been necessarily increased, though this is not usually a significant performance metric.

Topology-wise, the proximity requirements of the auxiliary channels entail that the group members be arranged in a form of a physical linear chain, so it is not just that the flow of group key contributions is in a linear chain. In other words, each successive member M_{i+1} needs to be positioned to be within a human visual range of M_i that allows M_{i+1}'s human-owner to distinguish the visual message transmitted in message 3 of Round i by M_i.

Hardware requirement-wise, as auxiliary channels (such as screen and keypad) are often already present in devices, provisioning this should not be a major barrier.

Strong security can also be achieved via an alternative method which uses private/public key pairs for signature and verification, as described, for example, in the scheme of Katz and Yung [12]. However, this is achieved through higher computational complexity, of which signature verification is particularly expensive.

4.7. Augmented Group Operations

Group membership is often dynamic. Members can leave or join, sub-groups may leave or fuse. Augmenting

with multiple channels allow all the group operations defined in the original Cliques suite, such as member addition, mass join, group fusion, member exclusion and subgroup exclusion, to be essentially retained (but they will not be presented here due to space constraints).

5. Arbitrary Topologies

Topology became a subject of interest for group key agreement protocols mainly for round efficiency reasons. We have been able to augment protocols having topologies such as the star [1], Hypercube [3], Octopus [3], and tree [6] topologies with multiple uni- and bidirectional [21] channels (not described here due to space constraints). In multi-channel protocols, we are interested in topology mainly because data-origin authentic channels are often limited in reachability. For group key agreement protocols run purely on an open radio medium, if all members are within easy radio range, then members' relative spatial arrangement and positions on a given topology are somewhat unimportant. Not so for multi-channel protocols, where relative proximity and line-of-sight affect usability. Perhaps, multi-channel group key agreements ought to be used with algorithms that can decide the best topologies and how to populate them.

6. Further Work

Different channel properties and different topologies need to be investigated to discover further useful interactions. As multiple channels may increase overheads, studies could be done to consider what are the best topology combinations to achieve high security at the least expense. Work also remains to be done to formalize these protocols.

7. Conclusions

We have applied multiple channels in pervasive computing environments to resolve security vulnerabilities in group key agreement protocols, in particular against active adversaries. Using multiple channels can readily transform an unauthenticated scheme into an authenticated scheme, and can modify previously password-based schemes to be strong against even a passive adversary which is increasingly pervasive on some bandwidth-limited auxiliary channels which however possess data-origin authenticity.

References

[1] N. Asokan and P. Ginzboorg. Key agreement in ad-hoc networks. *Proceedings of Nordsec 1999*, Nov 1999.

[2] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: authentication in ad-hoc wireless networks. *NDSS Symposium*, Feb 2002.

[3] K. Becker and U. Wille. Communication complexity of group key distribution. *ACM Conference on Computer and Communications Security*, pages 1–6, 1998.

[4] S. M. Bellovin and M. Meritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 72–74, May 1992.

[5] E. Bresson, O. Chevassut, and D. Pointcheval. Dynamic group group diffie-hellman key exchange under standard assumptions. *Proceedings of Advances in Cryptology - EUROCRYPT 2002, LNCS 2332*, pages 321–336, 2002.

[6] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. *Advances in Cryptology - EUROCRYPT '94, LNCS 950*, pages 275–286, 1994.

[7] R. Delicata. A security analysis of the cliques protocol suite. Master's thesis, Oxford University, 2002.

[8] D. Dolev and A. C. Yao. On the security of public key protocols. *Proceedings of the IEEE 22nd Annual Symposium of Computer Science*, pages 350–357, 1981.

[9] C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *Cryptobytes*, 7(1):29–37, 2004.

[10] M. Hietalahti. Efficient key agreement for ad-hoc networks. Master's thesis, Helsinki University of Technology, Department of Computer Science and Engineering, May 2001.

[11] J.-H. Hoepman. The ephemeral pairing problem. *8th International Conference on Financial Cryptography*, pages 212–226, Feb 2004.

[12] J. Katz and M. Yung. Scalable protocols authenticated group key exchange. *Proceedings of CRYPTO 2003, LNCS 2729*, pages 110–125, 2003.

[13] A. J. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[14] O. Pereira and J.-J. Quisquater. Generic insecurity of cliques-type authenticated group key agreement protocols. *17th IEEE CSFW*, Jun 2004.

[15] O. Pereira and J.-J. Quisquater. Some attacks upon authenticated group key agreement protocols. *Journal of Computer Security*, (4):555–580, Jan 2004.

[16] F. Stajano. *Security for Ubiquitous Computing*. John Wiley and Sons Ltd., 2002.

[17] F. Stajano and R. Anderson. The resurrecting duckling — security issues for ad-hoc wireless networks. *Proceedings of the 7th International Workshop on Security Protocols*, 1999.

[18] M. Steiner, G. Tsudik, and M. Waidner. Diffie-hellman key distribution extended to group communication. *Proceedings of the 3rd ACM conference on Computer and communications security*, pages 31–37, March 1996.

[19] M. Steiner, G. Tsudik, and M. Waidner. Cliques: A new approach to group key agreement. *Proceedings of IEEE International Conference on Distributed Computing Systems*, May 1998.

[20] M. Steiner, G. Tsudik, and M. Waidner. Key agreement in dynamic peer groups. *IEEE Transactions on Parallel and Distributed Systems*, 11(8):769–780, 2000.

[21] F. L. Wong and F. Stajano. Multi-channel protocols. *13th International Workshop in Security Protocols*, Apr 2005.