Efficient, secure and reliable ring multicast in wired or wireless networks.

John J. Metzner, Jade Bissat, and Yuexin Liu
Pennsylvania State University
Dept. of Computer Science and Engineering
111 IST Building
University Park, PA 16802
(814) 863-1264, metzner@cse.psu.edu

Abstract. A multicast scheme is described which merges the key distribution and acknowledgment tasks, allowing simple acknowledgment and frequent key changing, if desired. The scheme, denoted SAM for Secure Acknowledging Multicast, requires a ring organization. The key change requires only slightly more than one ring revolution, and need not interrupt data flow. Leaving and joining key changes are similarly easy to handle. Any group member can be the source, and the same acknowledgment policy can be used for reliable communication. The new key is encrypted by the old key, and only new messages use the new key. The joining and leaving methods are somewhat like the "CLIQUES" strategy, but SAM is more symmetrical, and directly incorporates acknowledgments as an added bonus. It can be applied to virtual rings in switched networks, or to rings in wireless networks. The basic ring procedure is"stop and wait", but in a modified method, denoted MSAM, channels can be interlaced for near continuous transmission or simultaneous many-to-many communication. For some wireless networks, average transmitted power is a more severe limitation on average bit rate than bandwidth, and stop-and-wait transmission is practical. Broadcast information can be combined with ring acknowledgment for further efficiency reduction.

Key words: multicast, secure communication, reliable communication, ring network

I. INTRODUCTION.

Secure reliable multicast is difficult and complex to achieve. Various approaches to the multicast security problem are surveyed in [1]. Many papers have dealt with the reliable multicast problem and the difficulties of controlling the acknowledgment process [2-10]. A tree structure usually is preferred for scalability to large multicast groups. However a ring structure has advantages for small to medium size groups [11,12]. The most efficient schemes require a great deal of participation by and interaction with nonmember nodes. Reliable communication on a ring is much easier to solve, but ring multicast is only practical for small to moderate size groups. The ring can be a virtual ring.

Most of the security multicast work described [1,13] uses the tree structure because it is best suited to a large group. Exceptions to the tree structure are the "CLIQUES" method [1, 14] and a scheme using a logical ring [15]. It is well known by human experience that the larger the group that knows a secret, the harder it is to keep. Thus, highly secret data distribution is more practical for small groups, which favors the ring approach.

This paper proposes a new method, called SAM for "Secure Acknowledging Multicast", which merges the tasks of acknowledgment and key changes in a simple and direct manner. It encourages the use of frequent key changes, which could reduce the key size needed for a given degree of security. It also handles member joining and leaving, and fusing of ring multicast groups. The basic SAM scheme is of a stop-and-wait nature. For continuous or many-to-many communication over multiple channels, an additional method, denoted MSAM, is described. With MSAM, key changing can be implemented with the assurance that at any time, over all channels, only two keys are in existence: the old key and the new key. The old key is transient, but continues to exist until the new key is established over all channels.

The ring key establishment method has some resemblance to the CLIQUES method. The CLIQUES method has a linear exchange structure and uses similar subkeys, but unlike SAM, it is not symmetric and does not incorporate acknowledgments.

A. Review of the public/private key principle.

In the Diffie-Hellman method [16], there is a generator g, operations modulo a large prime p. Given a private secret number S, compute g^S mod p. Knowing g, p and g^S, it is extremely difficult to compute S. For one-to-one communication, suppose **a** has secret number Sa and **b** has secret number Sb. **a** computes and sends g^{Sa} and **b** computes and sends g^{Sb}. g^{Sa} and g^{Sb}. are the public keys. Both can compute g^{SaSb}, as $(g^{Sa})^{Sb}$ by **b** and as $(g^{Sb})^{Sa}$ by **a**. But no one else can compute g^{SaSb} without great difficulty, even though they might observe both g^{Sa} and g^{Sb}.

B. Review of the new word policy for ring multicast acknowledgment.

The new word policy for ring communication is described in [17,18]. The rule is: send a new word (packet) when and only when a new word is received. A new packet is distinguished from an old packet by an alternating bit. This is sufficient to prevent ambiguity if the link does not reorder signals. One station by prior agreement is the changer of the bit. For multicast, it is natural for the source to be the alternating bit changer. The ring can be a virtual ring, with non-participating stations or routers simply forwarding what is received to the next group member. The packet goes from the source to all group members on the ring and is returned to the source, although the last hop wouldn't actually have to include all the data. If a group member does not receive a new packet, whether by error detected, unrequested repeat or by timeout, it repeats the prior transmitted packet. Thus, a group member is required to remember one prior transmitted packet. Figure 1 illustrates where station 1 is the source and bit changer, M1 is the new packet, and L0 is the old packet. There

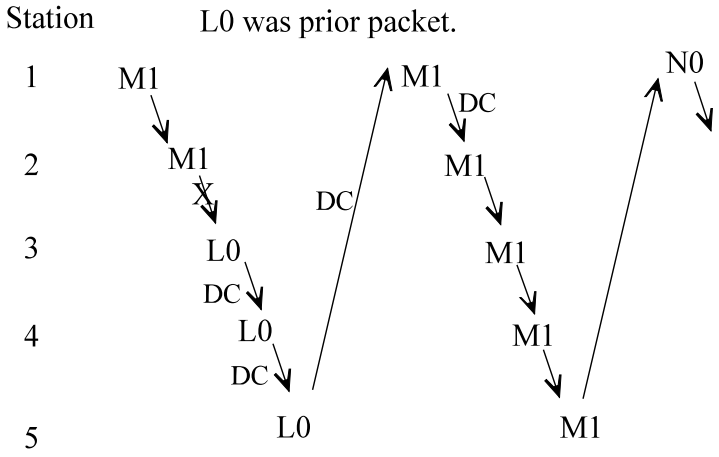


Figure l. The new word policy for a multicast ring

is an error from station 2 to station 3, indicated by an X. The next four hops are labeled DC for Don't Care, since the receiving station will take the same retransmission action (except for possible timeout delay), no matter what channel event occurs. Communication is stop-and-wait, and a group member

must check each packet for error prior to forwarding it. However, continuous communication can be achieved by interlacing.

II. ESTABLISHMENT OF THE GROUP KEY AND SUBKEYS.

Consider a unidirectional ring of 4 stations: a sender and 3 receivers. The method assumes that there is a separate process of authentication to ensure the group members are legitimate.

The four secret keys are Sa for **a**, Sb for **b**, Sc for **c**, and Sd for **d**. The common secret key will be $g^{SaSbScSd}$. Other auxiliary public keys will have to be generated, such as g^{SaSbSc}, g^{SaSb}, g^{SbSc}, etc. For example, if **d** sees g^{SaSbSc}, **d** can compute the common secret key $g^{SaSbScSd}$.

To make the notation less cumbersome, ABCD will denote $g^{SaSbScSd}$, ABC will denote g^{SaSbSc}, etc. ABCD is a common key for the four group members. The generation of a common key for groups in this way is well-known [1,12-14]. In SAM, the keys are distributed in a way that merges neatly with acknowledgments and allows simple key changes at little overhead cost. Consider the transfers to find a common key from 4 secret keys. **a** starts by sending g^{Sa}, abbreviated as A, to **b**. The steps are shown in Figure 2.

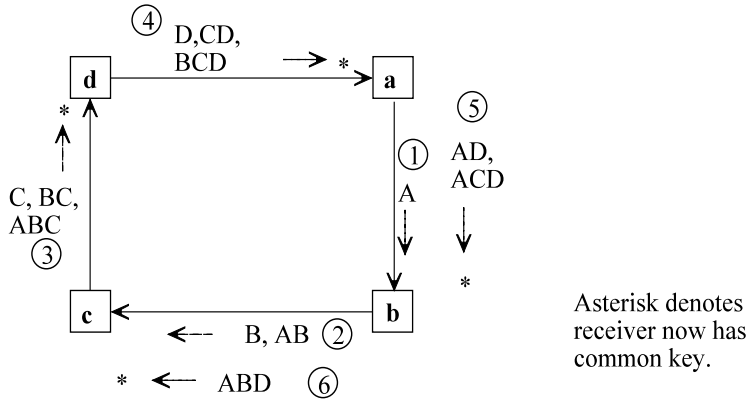


Figure 2. Establishing the group key

After 6 transfers all know key ABCD. For N members, 2(N–1) transfers are required. This is the minimum number of transfers for establishing a group common key [19]. Each member stores N keys, given N members (–2 subgroup keys, its own key and the group key used for encryption).

In the CLIQUES method, one station has the responsibility to store all the subgroup keys. Figure 3 shows the key establishment transfers in CLIQUES, using notation similar to Figure 2.

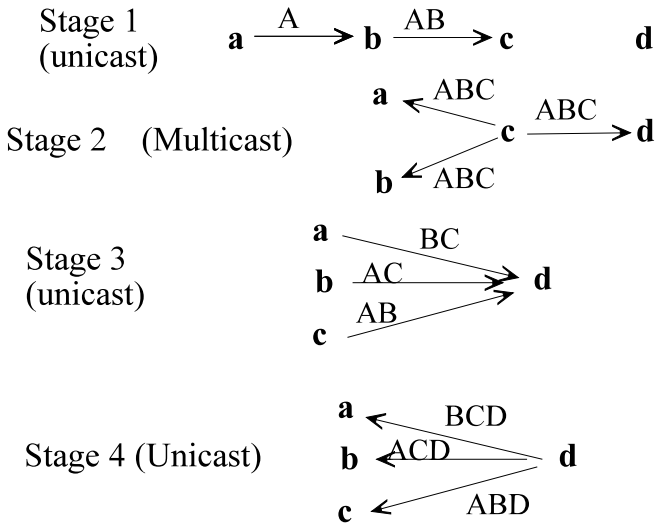


Figure 3. Message transfer in CLIQUES, four members.

The common key is ABCD, as in the ring protocol, but the topology is arbitrary. Member **d** has the subgroup store responsibility. There are four stages for any number N of members. The number of transfers are: N–2 for stage 1; N–1 for stage 2 (a multicast, but takes at least N–1 link transfers except for a broadcast network); N–1 unicast transfers to the same station in step 3 (may have implosion problems); and N-1 unicast transfers in step 4. The 4N-3 total transfers compare with 2N-2 with SAM. Also, acknowledgment is not included in CLIQUES. In [15], a system called Conference Key Distribution System is introduced (CKDS), which uses a logical ring similar to SAM. However, in CKDS the common key, ABCD...., is established in N–1 rounds, compared to less than 2 rounds in SAM, and there is no provision for acknowledgments. Also, stage 3 in CLIQUES requires the use of the inverse of a station's secret key. SAM does not require use of an inverse key

III. KEY CHANGES, AND COORDINATION WITH ACKNOWLEDGMENTS

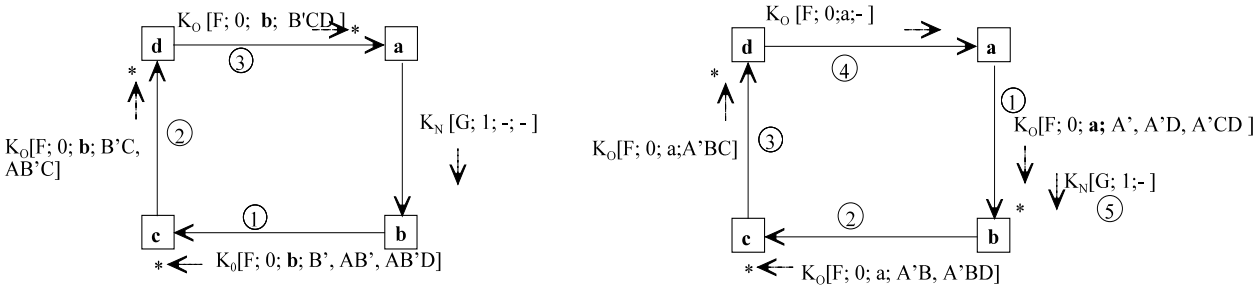
By frequent changes of a key coupled with acknowledgments, shorter or simpler key change instructions can be used. This has been noted in [20] for one-to-one communication.

Once the key ABCD is established, data transmission can flow encrypted by the key until the key is changed. Consider the transition from an old key (K_O), to a new key (K_N). A packet is defined in the form:

$$K_O[\ -\ ;\ -\ ;\ \ -;\ -\ ],$$

where the first place in the bracket denotes the packet data, the second place denotes the alternating bit value, the third place denotes the initiator of the new key, and the fourth place denotes the new subkeys.

Figure 4a shows where **a** is the source and **b** initiates the key change. F and G are successive packets.After 3 steps (as numbered in Figure 4a), every station knows AB'CD, and **a** is free to use the new key for the new packet. If a packet is received in error or lost, the state remains unchanged as all senders resend what they sent the last time, until a correct packet is delivered on the link that had the error.



a) **b** changes the key

b) **a**, the source. changes the key

Figure 4. Key changes

Note the following:

1. Key changes don't interrupt the data and acknowledgment flow.

2. Any station could be the key change initiator The initiator always knows the new key immediately, since it has the necessary old subgroup key already stored.

3. To use the new key at the earliest possible time, the node before the node that made the change would need to decrypt the packet, compute the new key, and then encrypt the outgoing packet with the new key. This can create some additional delay. Even if the changer is the node right after the source, the source needs to decrypt, use the reception to compute the new key and encrypt the new packet with the new key. To avoid this extra delay, the use of the new key should be delayed one more round. This way the destinations will have computed the new key while waiting, and will know exactly when to expect to use the new key.

4. No node should be allowed to initiate a further change until at least the second packet encoded with the new key arrives. Only then does a station know that all nodes have received a packet encoded with the new key.

Considering item 3 above, figure 4b shows a change initiated by source **a**, and when the new key is used. Stations b, c, d have learned the new key after steps 1,2, and 3, as indicated by the asterisks. They will be ready to decrypt with the new key for the new packet G.

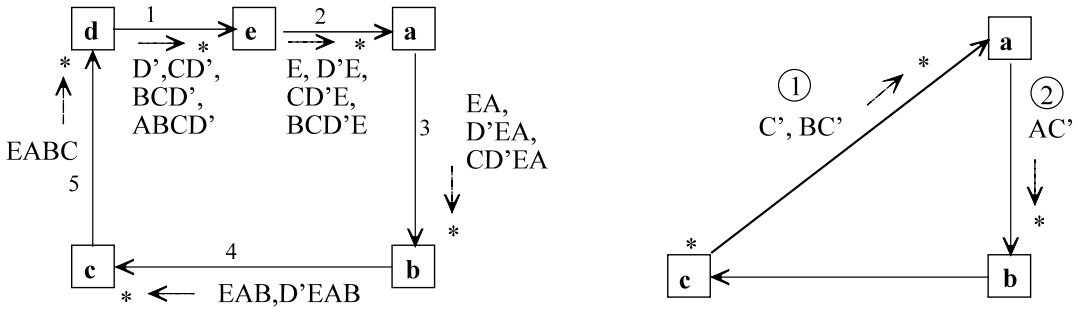
IV. JOINING AND LEAVING.

A. Joining the multicast ring.

Suppose a fifth station **e** wishes to join the ring. The old key ABCD might not have to be changed, since it will not be valid after **e** joins. However, if **e** is to be prevented from decrypting any prior messages, at least **d** should change the key to D'.

Figure 5a shows the message transfer to include **e**. Only one circulation distributes the new multicast

key ABCD'E. Note that no link transmits ABCD'E, and there is no easy way to compute ABCD'E from the transmitted items.



a) **e** joins the ring.

b) **d** leaves the ring.

Figure 5. Joining and leaving the multicast ring

B. Leaving the multicast ring.

Say **d** leaves the original 4-station ring. Since C, BC, and ABC are known by d, at least one of the secret keys must change. One solution is for all keys to change. This would be the initial key exchange, and would take 2(N–1) transfers (4 in this case), where N is the new number. However, if the node just before the node that leaves makes the change, only N–1 transfers (2 in this case) are needed. Why just this node? Note that **c** in this case knows the subgroup AB, but **a** doesn't know BC and **b** doesn't know AC. Thus **c** knows ABC' immediately, and only two steps are needed as shown in Figure 5b**.**

C. Mass Join [21]

If a number of members to join the group, it is more efficient for them to join in one operation rather than one at a time. This can be done in a way similar to the CLIQUES method [14], namely

chain the new members, in our case into the ring. Figure 6 shows how this can be done for the case of two new members.

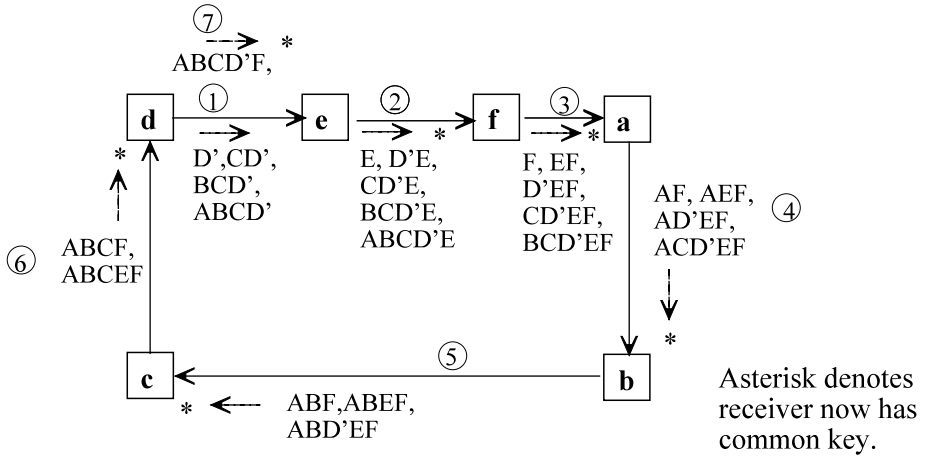


Figure 6. Mass join for two new members, e and f.

In general, if M new members join a group that previously had N members, the member just before the insertion point is responsible for changing its key, and the total number of transfers for all to know the common secret key is M + N + M - 1 = N + 2M - 1. This is less than the 2(N + M - 1) steps that would be needed to establish the key for an N + M -member ring.

D. Group Fusion [21].

By a technique similar to one developed in [19], two groups can fuse based on their separate common group keys, as illustrated in Figure 7. In Figure 7 the common key for the two groups is $g^{ABCDEFG}$. Each group maintains its subgroup key, Group 1 has the key ABCD, and Group 2 has the key EFG. If the groups split, they could revert to their separate keys.

It remains to describe how this fusion can fit into the ring acknowledgment structure. One solution is to merge into one ring in the data flow. Figure 8a illustrates key change for this case, where **d** wants to change the key to Sd' –> D'. **d** can immediately compute ABCD' and $g^{ABCD'EFG}$, and

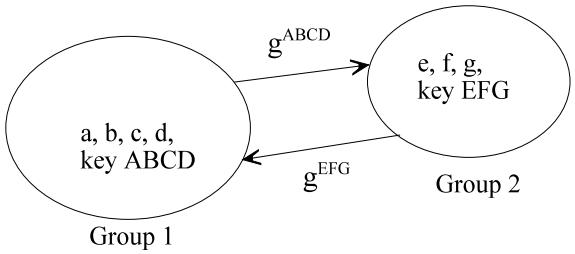


Figure 7. Group fusion to realize a common key

send $g^{ABCD'}$ to allow the group 2 members to compute $g^{ABCD'EFG}$, but it also must transfer D' to **a, b**, and **c**, so they can compute the new group 1 key of ABCD' that they need to compute $g^{ABCD'EFG}$. Still, both items can be passed around the expanded ring in one revolution. The transfers are encrypted with the old key, $g^{ABCDEFG}$. Future transfers will be encrypted with the new key, $g^{ABCD'EFG}$. If the groups separate, they can operate securely with their subgroup key.

There is another solution - have two separate acknowledgment flow rings. Since acknowledgment ensures only correct transfer on the ring itself, one station on the ring containing the source (it could be the source) needs to have the ability to cache all data needing acknowledgment on the second ring. This node, say **a**, could join both rings, as shown in Figure 8b. If **a** is the source, it is not difficult for it to prepare two different encryptions of each packet for the two rings, and key changes for the two rings could proceed independently.

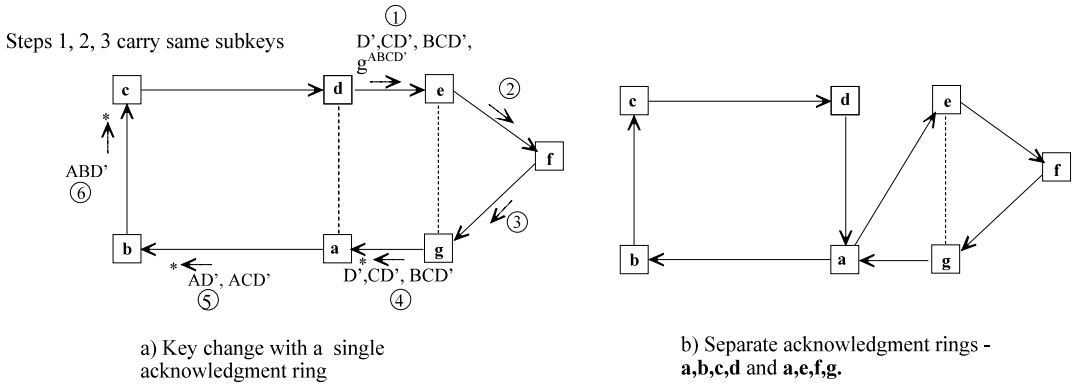


Figure 8. Groups fused into one or two rings.

V. MSAM - MULTIPLE CHANNELS

A limitation of SAM as described above is the stop-and-wait nature of the process. A full frame delay is needed to decode before making a response at each station, so that the effective data rate tends to go down inversely with N, the number of group members.

If greater use of the virtual links in the ring is possible, the stop-and-wait limitation can be alleviated by having interlaced channels, somewhat as a slotted ring described in [22]. It would be too complex to have N channels each changing keys independently. However, there are keying schemes where at any time only one of two keys are in existence: the old key and the new key. Two keys would only be needed during the transition to a new key. The methods of doing this are called MSAM [23], for Multiple Channel Secure Acknowledging Multicast. This section explains MSAM methods.

If there are the same number of channels as group members and the ring is exclusively for the group, then each member can have a channel and they all could send reliably to all the others at the same time. Other possibilities are that one station could be the source for all channels, or that the channels could be apportioned in various ways among the senders. The interlaced channels need to be distinguished by some channel number. The order between interlaced channels does not have to be preserved, but order within a channel needs to be preserved if the alternating bit protocol is to be used. If order is not preserved, multi-bit sequence numbering is required.

Assume code changes are made only by sources, but in many-to-many communication there may be multiple sources. To prevent confusion, only one source can be allowed to change the key at a time. A solution to this [23], is to pass around a token, and only a station possessing the token is allowed to exercise the next key change.

A. All channels used by one source - initiator, bit changer and key changer.

Realize first, that not all channels will be in step due to varying acceptance and delay. In figure 9, each row represents the sequenced frames for the numbered channel at the station that can change the key. The boxed numbered frames are the newest frames that the station has sent, encoded by the old key, prior to inserting a new key instruction. It is now desired to encrypt with the new key. As soon as one of the boxed frames can be discarded by the station, that station sends the new frame, using the old key, but with information for its new key embedded. This will appear in the new frames numbered **5,7,4,8** (bold) on channels 1,2,3,4, respectively. As each of these frames are acknowledged, frames 6,8,5,9 on channels 1,2,3,4, respectively will be encoded with the new key. When all four

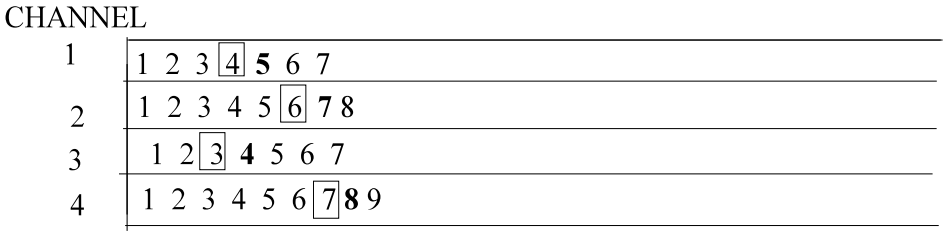


Figure 9. Changing the key on all channels - one source

of these frames (6,8,5,9) get acknowledged at the changing station, the old key can be discarded everywhere, since everyone will be expecting a new key encryption on every channel. After this, introduction of another new key change is allowed.

B. Multiple sources.

The channels can be independent, one for each group member's transmissions. Their rates can be different. They can be statistically time multiplexed on the same or different virtual paths, but each channel must follow a particular virtual path to preserve order. A common feature is that all are encrypted with the same key or, during the transition, with one of two possible keys. Acknowledgment needs only one bit per frame, assuming order is preserved within a channel. But

in this case we must modify the approach described in Figure 9. Figure 10 shows a case where there are two sources, X and Y, and two channels, CX and CY. Each source is the bit changer on its

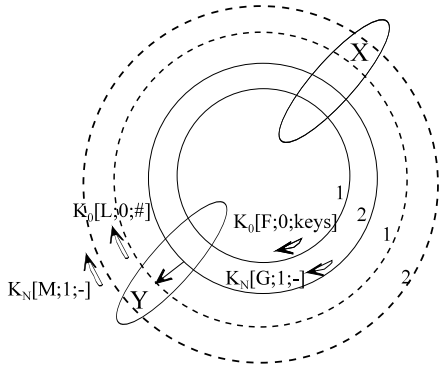


Figure 10. Changing the key with two different sources and channels.

channel. X is controlling the change to a new key. Channel CX circulations are shown in solid circles; Channel CY circulations are shown in dashed circles. The first circulation indicated on CY is the first circulation beginning when the second circulation of CX reaches Y (see arrow).

When circulation 1 of CX arrives at Y, Y has learned the new key, but can't use it yet, since those from Y to X don't have the new key yet. Thus Y continues its old key Channel CY behavior. When circulation 2 of CX arrives at Y, Y knows that all have the new key. However, other stations don't know that Y knows this, so Y sends its next new message on circulation 1 of CY with the old key and a # indication that the next new message on CY will have the new key. The new key is used on the labeled circulation 2 of CY. The transition could be made quicker if Y used the new key on CY in circulation 1, but then stations would have to try to decrypt with both the new key and the old key.

The method described for two sources and two channels can be extended to any number of sources with one or more channels per source. The key-changing source handles its own channels as in Figure 9, and the other sources can treat their channels as in Figure 10. The first circulation 2 event on any channel of the key-changing source triggers the start of circulations 1 and 2 of the other source. The transition to the new key is complete when new key messages have circulated back to the key changer on all channels in use.

In token passing, when the source has received new key encoded frames from all active

channels, it is free to pass the token to the next source. A token can be passed only when the previous change has completed on all active channels. The token passing can be carried in a frame similar to key change instructions. If a source does not wish to participate in key changing, it can pass the token at any time to the next source.

VI. COMMUNICATION MEDIA FACTORS

A. Wired, switched networks.

For group members a, b, c, d, ..., assume there is some path between any two members. From the point of end system multicast [24], there are N(N - 1)/2 connections, from which we wish to construct a ring employing N of these paths. Finding a least cost multicast tree among the N(N - 1)/2 connections is very easy (Prim-Dijkstra Algorithm). The problem of finding the best ring is in the Traveling Salesman category, which is known to be intractable for large N. However, SAM is not intended for use with very large N, so it should not be difficult to find a good or possibly least cost ring. If a least cost tree is established with two-way paths, equal-cost in both directions, a ring can be constructed readily with $\leq 2\cdot$(minimum multicast tree cost) [9].

Here is a rough scenario of how the ring might be established .by an end system. Say A knows the addresses of B, C, D, and E, and A learns the $5\cdot 4/2 = 10$ least cost routes between pairs and picks the 5 links that form the best ring. (There are only 4! = 24 possibilities for this case.) Then A does individual authentication exchange with each, giving each by secure 1:1 communication the ordered address list and a component or operation to include in a verification round. Then A can initiate the common key establishment over the ring. With the first encrypted packet sent after the common key is established, each group member can decrypt/encrypt to include its secret operation, to verify that all and only the desired A, B, C, D, E have contributed to the ring circulation.

The data rate depends on the number of group members and the transmission time of each

member. If we ignore the input-output delay of relaying nonmembers or include it in the propagation time, the data rate in the absence of errors is:

$$R = \frac{m}{\sum_{i=1}^{N} (T_i + T_{pi})} \qquad (1)$$

where R is in bits/second, m is the number of bits per packet, T_i is the packet transmission time for member i, and T_{pi} is the propagation time from i to i + 1. If all transmission times are equal and all propagation times are equal or negligible, the rate goes down as 1/N. But if there is one extremely slow link with transmission time T_s, the rate is about m/T_s. In the case where exactly one case is extra slow, the number of members is not too critical. Limitation by the slowest link is a problem common to all secure, acknowledging multicast methods. If the transmission times are roughly equal, multiple channels with MSAM can provide substantial rate gains.

Stop-and-wait communication has often been maligned as inefficient, since continuous communication is much faster. However, in a multiuser network, stop-and-wait is not necessarily inefficient, since other communications can use the wait period.

With a fixed path, order is preserved. Thus the alternating bit, with a channel indicator for multiple channels, is adequate to prevent ambiguity. Another possibility for the case of one slow path is to provide multiple parallel paths to help supplement the slow path. However, this creates the risk of packets arriving out of order, since an old packet already acknowledged over one of the parallel paths, might turn up so late as to be misinterpreted as a new packet expected two rounds later, which would have the same alternating bit value. This problem could be solved with the use of a multi-bit sequence number.

B. Wireless networks

Wireless networks tend to be more error-prone and in many cases a sender is limited in

transmitted energy. Where energy per bit is a key factor, as in much wireless multiaccess communication, a given sender does not have a need or ability to send continuously at the highest possible rate. Consider for example the recent interest in Ultra-Wideband communication [25], where a huge bandwidth is available. A sender can send at an extremely high bit rate, even gigabits/second, over a very brief time interval, but doesn't have the energy or permission to send continuously at that rate. One technique which could be used is Wideband ALOHA [26]. A packet of say 10,000 bits could be sent in 10^{-5} seconds at a 1 gigabit/second rate. If any needed decrypt/encrypt conversion could be done in 10^{-3} seconds per station, a ring with < 10 stations and short distances might complete the ring round trip in about 10^{-2} seconds, which would allow a long term average throughput of about 10 megabits/second despite stop-and-wait. If there are 5 stations, total time in transmission is $5*10^{-5}$ seconds out of a 10^{-2} second cycle, which is a fraction 0.005 of the time.

Because of the error-prone factor, acknowledgment is essential for reliability. Any packet in error is retransmitted, and partial progress around the ring is maintained. If p_i is the failure of a packet transmission on link I of the ring, The average number of circulations per success is

$$N_S = \sum_{i=1}^{N} \frac{1}{1-p_i} - (N-1) \qquad (2)$$

If all pi = p, this reduces to

$$N_S = \frac{1+(N-1)p}{1-p} \qquad (3)$$

Total traffic on all links for each success would be

$$T = N \cdot \frac{1+(N-1)p}{1-p} \qquad (4)$$

If a packet were unicast to the last member, using N - 1 links with probability of failure p on each

link, and no memory at the intermediate points,

$$N_S\,(end-to-end) = \frac{1}{(1-p)^{N-1}} \qquad (5)$$

Ignoring end-to-end acknowledgment,

$$T\,(end-to-end) = (N-1)\cdot\frac{1}{(1-p)^{N-1}} \qquad (6)$$

Although (4) is considerably smaller than (6) for moderate to large p, it could be still better. The new word policy can suffer from unnecessary retransmission. Member i's packet could have been received correctly at i+ 1, but fail at a later point, but member i will retransmit on getting an old packet, unnecessarily. With hop-by-hop acknowledgment, assuming an error-free return channel on each hop, and equal error probability p per hop,

$$\mathrm{T\,(hop\text{-}by\text{-}hop)} = \mathrm{N}\cdot\frac{1}{1-\mathrm{p}} \qquad (7)$$

Traffic from (4) is higher than from (7) by a factor of 1 + (N - 1)p. But this would come at the expense of having an acknowledgment returned for each hop. A successful reception is always forwarded immediately, so there is no extra ring delay. NACKs can also be used to speed retransmission. If the acknowledgment is lost, member i will timeout and retransmit. There are two advantages: unnecessary retransmissions are avoided, and with the aid of NACKs, an erroneous transmission could be sent sooner than if the sender waited for a ring circulation before retransmission.

To keep the features of the alternating single bit and that a member never needs to cache more than one prior packet, the sender needs to get a response through the ring circulation. However, if two-way hops have been set up, the return acknowledgment from the final receiver could follow the N-hop path back to the receiver to complete the ring, if no better return path is available. This doesn't have to return the whole packet; it just needs to carry the error-protected alternating bit and a

distinction from the one-hop acknowledgment.

As mentioned in part A of this Section, multiple paths could be set up between successive ring members, but at the expense of using multi-bit sequence numbers to counter out-of-order arrivals. In wireless communications, alternate paths are valuable because particular links may be lost or encounter fading. It is not clear whether multi-bit sequence numbers would always be needed, because the time difference of arrival of transmissions of the same packet on parallel channels would normally be shorter than the round trip ring circulation time. A new packet can't come into station i until i's old packet transmission has succeeded, not just to i+1, but all the way around the ring, and, prior to this happening, any old packet reception at i+1 would be ignored. Even after the new packet comes to i+1, there is a minimum ring circulation time until the old alternating bit value is used for the following new packet.

C. Broadcast assistance in ring acknowledgment protocols

In wireless networks, often a node can broadcast to many receivers in one transmission. The research in multicast over wireless Ad Hoc networks [27] emphasizes hop-by-hop forwarding over two-way links, rather than simultaneous broadcast to multiple group receivers. We will look at the possibility of using the broadcast feature for efficiency, while retaining a ring structure for acknowledgment and security.

Consider first where one source node is a broadcast node to all members, and also is a member of a ring, where the other members are more energy limited. The ring could be used to pass acknowledgments and key information around the ring. If the acknowledgment did not pass around the ring, the broadcast node could retransmit. This is illustrated in Figure 11. The ring return could be used in key establishment and key changes, but key changes could be infrequent to save energy at the non-broadcast nodes. With the ability of two-way communication between neighboring nodes, this idea can be generalized to save energy by taking advantage of any broadcast ability on the ring.

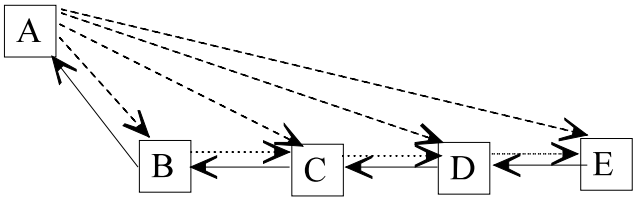


Figure 11. Wireless broadcast of data by A with acknowledgments returning by ring.

Acknowledgment packets are shorter than regular packets, and take much less energy to transmit. When node j sends a new packet it addresses it to its next ring node j+1, but it may also be received by other nodes. If node j+1 receives a correct packet from j, node j+1 sends an alternating bit acknowledgment only packet to j and j+2 (it could be the same to both, assuming both in range), indicating it has the new packet. If j+2 has the new packet, it acknowledges the fact to j+1 and j+3. If j+3 does not acknowledge back to j+2, node j+2 will send the new packet, addressed to j+3. Acknowledgment thus advances to the next ring member only in two ways: by receiving a new packet addressed to itself, or by receiving a new alternating bit message combined with having received the packet via some broadcast. For the case of key establishment, however, the key information for the next node must be inserted in the new alternating bit message for the next node and in the packet addressed to the next node, because each transmission contains information needed specifically for the next node.

Key changes could be done via broadcast of packets encrypted with the old key if a simpler key change was used after the first key was established. This could be justified if changes were frequent. For example, The original key K could be constructed through the common key method described in Section II. Then, at times of the broadcast source's choosing, a transform T_1 could be broadcast, to a new key T_1K, encoded by K. When the acknowledgment around the ring completes, the broadcast source and all other transmissions can use the key T_1K. At the next change it could become T_2 T_1K, etc.. These transforms could be simple, to ease decrypt/encrypt conversions, but would still be effective since they would occur too fast for an eavesdropper to follow.

VII. DISCUSSION AND CONCLUSIONS.

The SAM method provides both simple acknowledgment and simple key changing. The ability to change keys with any new reception allows shorter and simpler keys. Furthermore, the key change does not have to interrupt the data exchange and acknowledgments. Ring multicast also is good if any member can be the source, since the topology doesn't need to change depending on the source. Multiple channels can be interlaced for near continuous communication. These channels can all be used by one source for highest speed, or different sources can have different channels. Despite key changing and different rates of progress on the different channels, only at most two common keys are needed at any one time (only one key except during a key change period).

The SAM method seems practical for small groups. It can be used in switched network end system multicast, or in wireless multicast systems such as Ad Hoc networks, if both high reliability and security are needed . In wideband, low energy networks, stop-and-wait transmission is hardly a drawback. In wireless networks it is possible to use broadcast ability for multicast energy conservation, while also allowing a ring structure for the acknowledgments. It also is possible to set up multiple member-to-member paths to augment or bypass slow or failed links on the ring, but this would likely require multi-bit sequence numbering to resolve out-of-order receptions..

REFERENCES

[1] K. Chan and S. Chan, "Key management approaches to offer data confidentiality for secure Multicast," *IEEE Network Mag.*, v. 17, pp. 30-39, Sept./Oct. 2003.

[2] D. Towsley, J. Kurose, and S. Pugali, "A comparison of sender-initiated and receiver-initiated reliable multicast protocols," *IEEE J. Selec. Areas Comm*., v. 15, pp. 398-406, April 1997.

[3] S. Paul, K. Sabnani, J. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," *IEEE J. Select. Areas Commun.*, v. 15, pp. 407-421, April 1997

[4] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission", *IEEE/ACM Trans. on Networking*, v. 6 ,no. 4, pp. 349-361, August 1998.

[5] B. Whetten and G. Taskale, "An overview of reliable multicast transport protocol II ," *IEEE Network Mag*., pp. 37-47, Jan/Feb 2000.

[6] Kasera, G. Hjalmtysson, D. Towsley, and J. Kurose, "Scalable reliable multicast using multiple multicast channels," *IEEE/ACM Trans. on Networking*, v. 8 ,no. 3, pp. 294-309, June 2000

[7] J. Gemmell, et al, "The PGM reliable multicast protocol," IEEE Network Magazine, v. 17, pp. 16-23, Jan/Feb 2003.

[8] X. B. Zhang, S.S. Lam, D.-Y. Lee, and Y. R. Yang, "Protocol design for scalable and reliable rekeying,"*IEEE/ACM Trans. on Networking*, v. 11 ,no. 6, pp. 908-922, December 2003.

[9] J.-H. Cui, M. Faloutsos and M. Gerla, "An architecture for scalable, efficient, and fast fault-tolerant multicast provisioning," *IEEE Network Mag.*, pp. 26-34, March/April 2004.

[10] C. Papadopoulos , G. Parulkar, and G. Varghese, "Light-weight multicast services (LMS): a router-assisted scheme for reliable multicast, *IEEE/ACM Trans. on Networking*, v. 12, pp. 456-468, June 2004.
[11] M. Baldi, and Y. Ofek, "A comparison of ring and tree embedding for real-time group multicast", IEEE/ACM Trans. on Networking, v. 11,no. 3, pp. 451-464, June 2003.
[12] M. Baldi, Y. Ofek, and B. Yener, Adaptive group multicast with time-driven priority," *IEEE/ACM Trans. on Networking*, v. 8, pp. 31-43, February 2000.
[13] Y. Sun, W. Trappe, and K.J.R. Liu, "A scalable multicast key management scheme for heterogeneous wireless networks, *IEEE/ACM Trans. on Networking*, v. 12, pp. 653-666, August 2004.
[14] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," IEEE Trans. on Parallel and Distributed Systems, v. 11, pp. 769-779, August 2000.
[15] I. Ingemarsson, D. T. Tang, and C. K. Wong, "A conference key distribution system," *IEEE Trans. on Inf. Thy.,* v. IT-28, no. 5. pp. 714-720, September 1982.
[16] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Thy.*, vol. IT-22 (6): 644-654. 1976.
[17] J. J. Metzner, "The new-word policy and decision feedback in loop data communication networks", *IEEE Trans. Commun.*, vol. COM-21, pp. 727-730, June 1973.
[18] J. J. Metzner, *Reliable Data Communications.* Academic Press. 1998. Chapter 12.
[19] K. Becker and U. Wile, "Communication complexity of group key distribution," *Proc. ACM Conf. Comp. And Commun. Security*, New York, N.Y., pp. 1-6, 1998.
[20] J. B. Marco, "Efficient re-keying using the sliding window protocol," Master's paper at Pennsylvania State University, July 1998.
[21] Y. Liu, " A new key management scheme in ring multicasting and its comparison to SAM, CLIQUES and CKDS schemes," M.S. Thesis in Computer Science and Engineering, Pennsylvania State University, May 2004.
[22] J. J. Metzner, "A high efficiency acknowledgment protocol for the slotted Pierce ring," *Proceedings of IEEE Infocom 85,* pp. 333-339, March 1985.
[23] J. Bissat, "Multiple Channel SAM (MSAM): description, extensions and performance analysis," M.S. Thesis in Computer Science and Engineering, Pennsylvania State University, August 2004.
[24]Y. Chu, S. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE J. Selec. Areas Comm*., V. 20, pp.1456-1471, October 2002
[25] W. C. Chung, N.J August, and D. S. Ha; "Signalling and multiple access techniques for ultra wideband 4G wireless communication systems," *IEEE Wireless Commun. Mag*, v. 12, April 2005, Page(s):46 - 55.
[26] N. Abramson, "Multiple access in wireless digital networks", *Proc. IEEE*, vol. 82, pp. 1360-1369, September 1994.
[27] C. Cordeiro, H. Gossain, and D. Agrawal, "Multicast over wireless mobile Ad Hoc networks: Present and Future Directions," *IEEE Network Magazine*, v. 17, pp. 52-59, Jan/Feb 2003.