A Family of Collusion Resistant Protocols for Instantiating Security

Sandeep S. Kulkarni Bruhadeshwar Bezawada
Department of Computer Science and Engineering
Michigan State University
East Lansing MI 48824

Abstract

In this paper, we focus on the problem of identifying a family of collusion resistant protocols that demonstrate a tradeoff between the number of secrets that users maintain and the extent of collusion resistance. Towards this end, we define classes of collusion resistant protocols (modeled along the complexity classes in algorithmic complexity) and evaluate the membership of existing protocols as well as the protocols in the proposed family in these classes. We also show that this family contains existing protocols for instantiating security.

Keywords : Security, Instantiating security, Collusion Resistance

1. Introduction

One way to achieve security, including authentication and confidentiality, is to require that the sender and receiver share a collection of secrets such that no other user in the network knows all the secrets. An impediment in providing such security is the issue of collusion among users. Specifically, if a group of users collude then they can combine their collection of secrets and attempt to foil the security of the communication among the remaining users. One way to achieve such collusion resistance is to use the *full secret protocol*, where each pair of users maintains a unique secret that is only known to those two users. With such an approach, the collusion among some users does not affect the remaining users. However, in this approach, the number of secrets that users maintain is $n-1$, and it is difficult to maintain these secrets if the number of users is large or the user capability is low (e.g., in ad-hoc networks and sensor networks).

Another protocol in this context is the $square$ grid protocol from [1] (recalled in Section 2.1). This protocol guarantees security in the absence of collusion while maintaining only $O(\sqrt{n})$ secrets per user. Specifically, it guarantees that when two users communicate, the collection of (at most 2) secrets they use is such that no other user knows all secrets in that collection. Hence, they can use this collection of secrets to establish a session key. However, since other users may know (different) subsets of this collection, the collusion among such users can compromise the security.

Based on the above discussion, our goal is to identify a family of protocols that provide a continuum between the grid protocol (where number of secrets maintained is within a constant factor of the optimal (cf. [1])) and the full secret protocol (where the security is *as good as it gets*). Some of the desirable properties for the protocols in this family include: (1) the number of secrets should be proportional to collusion resistance, and (2) the number secrets used for establishing session key is small, i.e., $O(1)$, so that the session key can be established efficiently.

In developing such a family of collusion resistant protocols, one can use two approaches. In the first approach (as in [2]), in spite of collusion among any t users, where t is a threshold specified upfront, all users can communicate securely. Alternatively, in the second approach, colluding users may be able to compromise only a subset of communicating users. In other words, the communication between a pair of users is compromised only if a specific set(s) of users collude.

The family of collusion resistant protocols proposed in this paper focuses on the second approach. Such probabilistic security is useful in many scenarios, especially where the number of potentially colluding users is large enough so that the first approach cannot be followed. Furthermore, this approach is also useful in scenarios where some other external factors make it difficult for colluding users to obtain encrypted communication. For example, in ad-hoc networks or sensor networks, the colluding user(s) may be able to compromise communication between a pair of users only if it is in their listening range. Finally, it is straightforward to combine both approaches where secure communication

[1] Email: sandeep@cse.msu.edu, bezawada@cse.msu.edu
Web: http://www.cse.msu.edu/~{sandeep,bezawada}
Tel: +1-517-355-2387
This work was partially sponsored by NSF CAREER CCR-0092724, DARPA Grant OSURS01-C-1901, ONR Grant N00014-01-1-0744, NSF grant EIA-0130724, and a grant from Michigan State University.

is guaranteed if the number of colluding users is less than a pre-determined threshold t and probabilistic security is guaranteed if the number of colluding users exceeds t.

Contributions of the paper.

- We propose a family of collusion resistant protocols where the level of collusion resistance is proportional to the number of secrets that users maintain.
- We formally define the notion of collusion resistance. Using this definition, we identify classes of collusion resistant protocols. We evaluate the membership of existing protocols as well as the protocols in the proposed family in these collusion resistance classes.
- While the proposed family of collusion resistant protocols is based on the square grid protocol from [1], we show that other variations of the square grid protocol cannot be used to obtain such a family.

Organization of the paper. In Section 2, we consider the related work and recall the square grid protocol from [1]. Then, in Section 3, we define what we mean by collusion resistance of a secret instantiation protocol. Using this definition, in Section 4, we define collusion resistance classes. In Section 5, we identify the constraints that should be met in order to identify a family of collusion resistant protocols. Using these constraints, in Section 6, we present our family of collusion resistant protocols. In Section 7, we present simulation results and analyze them. In Section 8, we describe related work in this area and conclude in Section 9.

2. Classification of Protocols for Instantiating Security

The approaches for instantiating security can be broadly classified in terms of those that use asymmetric keys (e.g., public/private key) and those that use symmetric keys. In the former approach (e.g., [3–5]), certificates are used and initially each user is provided with a certificate signed by a trusted authority. However, this solution requires high computing power (100-1000 times when compared with symmetric keys). For this reason, we focus on solutions that use shared secrets instead of certificates.

The approaches for providing security with shared secrets can be further classified in terms of (1) availability (or, the lack thereof) of a trusted server *during communication between users*, and (2) trust (or, the lack thereof) in the intermediate users that may be required to facilitate routing of messages between communicating users.

Existing protocols such as [6–8] are designed for systems where a trusted server is available when two users need to communicate. However, in many systems, this approach is undesirable (respectively, impossible) as no trusted server is available *when two users need to communicate with each other*. The protocols in the proposed family assume that a trusted server is unavailable when users need to communicate.

Also, other protocols have been designed where intermediate users are trusted (cf. [9–12]). If two non-neighboring users need to communicate, they route the messages through the intermediate users that decrypt and re-encrypt the message. Thus, it suffices that the communicating users share a path such that every user on the path shares a secret with its predecessor and its successor. However, in this case, compromise of a small number of users (and, collusion among them) that act as intermediate users can severely compromise the security.

Another category of solutions includes solutions where intermediate users are not trusted. Clearly, in such a solution, compromise of intermediate users (or collusion among them) does not affect system security as long as the communicating users share a secret that is not known to the intermediate users. Therefore, especially in the context of developing collusion resistant protocols, it is desirable to follow this approach.

Based on the above discussion, we focus on the security protocols where (1) shared secrets are used, (2) a trusted server is not available when two users communicate (such a trusted server could exist for users to obtain their shared secrets in an offline manner), and (3) intermediate nodes are only trusted as far as routing is concerned. They should not be able to decrypt any communication they are forwarding. Examples of such protocols include the protocols from [1,2]. Since our proposed family of collusion resistant protocols is based on the protocol from [1], we recall this protocol, next.

2.1. The Square Grid Protocol

In this section, we recall the *square grid protocol* [1] for instantiating security. In this protocol, n users are arranged in a *logical* square grid of size $\sqrt{n}$ x $\sqrt{n}$. Each location, $\langle i, j\rangle$, $0 \leq i, j < \sqrt{n}$, in the grid is associated with a *user* $u_{\langle i,j\rangle}$ and a *grid secret* $k_{\langle i,j\rangle}$. Each user knows all the grid secrets that are along its row and column. For example, in Figure 1, the grid secret associated with $\langle 1, 1\rangle$ is known to users at locations $\langle j, 1\rangle$, $\langle 1, j\rangle$, $0 \leq j \leq 3$. Additionally, each user maintains a direct secret with users in its row and column. This direct secret is not known to any other user. For example, user $u_{\langle 1,2\rangle}$ shares a direct secret with user, $u_{\langle 1,3\rangle}$, which is located in the same row (cf. Figure 1).

Now, consider the case where user A wants to set up a session key with user B. Let the locations of A and B be $\langle j_1, k_1\rangle$ and $\langle j_2, k_2\rangle$ respectively. In this case, A selects the session key and encrypts it using the following secret selection protocol. Along with the encrypted session key, it also

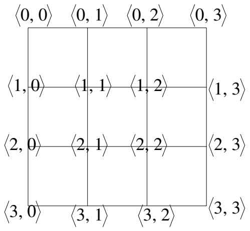


Figure 1. Single grid protocol: A node marked $\langle j,k\rangle$ is associated with user $u_{\langle j,k\rangle}$ and *grid secret* $k_{\langle j,k\rangle}$

sends its own grid location (in plain text). If multiple secrets are selected by A then a combination of those secrets (using hash functions like MD5) is used to encrypt the session key.

```
Secret selection protocol for session key establishment
for users at $\langle j_1, k_1\rangle$ and $\langle j_2, k_2\rangle$
// If users are neither in same row nor in same column
If ($j_1 \neq j_2 \wedge k_1 \neq k_2$)
  Use the grid secrets $k_{\langle j_1,k_2\rangle}$ and $k_{\langle j_2,k_1\rangle}$
Else
// If users are in the same row or column
  Use the direct secret between $u_{\langle j_1,k_1\rangle}$ and $u_{\langle j_2,k_2\rangle}$
```

Theorem 2.1 The above protocol ensures that the collection of secrets used by two communicating users is not known to any other user in the system. Hence, in the absence of collusion, the above protocol can be used for establishing the session key. (cf. [1] for proof.) □

Remark. Note that in this paper, we only focus on the issue of secret distribution and secret selection. Once the secret(s) is selected, existing approaches can be used to establish a session key while providing resistance against attacks such as replay of old messages. These approaches include the use of 'time' and/or 'nonces',

3. Defining Collusion Resistance of Security Protocols

In this section, we precisely define how we count the secrets maintained by users and how we compute the collusion resistance of a protocol. We are interested in protocols where the secrets maintained by a user are independent, i.e., even if an attacker knows a subset of the secrets that a user has, it is not be possible for the attacker to discover the other secrets (e.g., through cryptanalytic attacks) that the user has. Thus, even if two users use a set of secrets to ensure security and the attacker is aware of all but one of those secrets, the attacker cannot compromise that communication.

Furthermore, to compute the space requirement for secrets, we count all secrets that are needed to be stored by the user. To illustrate this issue, consider the case where a small number, say x, of secrets are used initially to generate a large number, say y, of *new* secrets by some mathematical manipulation (e.g., using those in evaluating certain polynomials) of the original secrets. In such a case, if these y secrets are stored by the user then the space requirement for this case is y. However, if these secrets are computed on-the-fly then the space requirement is x.

Intruder/Attacker/Colluder Model. We assume the standard node-compromise attacker model (e.g., [2, 9, 10, 12–14]). If a user has been compromised by an attacker then the attacker can utilize all the secrets that the user had. It can do so either passively, i.e., by just listening to messages and attempting to decrypt them if possible. Or, it can do so actively, for example, it can attempt to impersonate another user. The colluding users (attackers) can pool together their secrets in order to break communication security.

Also, we make no assumptions about mobility in the network. Thus, the users may be mobile or static. We only assume that an orthogonal approach is used to route messages (even in the presence of mobility) and to deal with denial of service attacks. In other words, we only assume that any message sent by legitimate users is delivered (even if users are mobile or the system is subject to a denial of service attack). The approaches used for routing or for dealing with denial of service attacks are outside the scope of this paper. (Note that routing does not need to be secure; it only needs to be reliable.)

When users collude, they can combine the secrets they know in order to compromise communication among the remaining users. To study the effect of such collusion, consider a system with n users. In such a system, there are a total of $n(n-1)$ pairs of communicating users. (For simplicity, we consider $\langle j,k\rangle$ to be a different pair than $\langle k,j\rangle$). Hence, when a set of users collude, some of these pairs can no longer communicate securely, as all the secrets they use for achieving security are known to the colluding users. For example, in Figure 2, if users $u_{\langle 0,0\rangle}$ and $u_{\langle 1,1\rangle}$ collude then user $u_{\langle 2,0\rangle}$ cannot communicate securely with $u_{\langle 3,1\rangle}$. Our definition of collusion resistance is based on the effect of the collusion on these pairs. First, we consider two such plausible definitions, and argue that they are inappropriate because either they do not capture the *true* collusion resistance of the protocol or they are difficult to use.

Attempt 1: A protocol is collusion resistant to x users if at least **one pair** of users can communicate securely even if any subset of x users collude.

This definition is inappropriate for the following reason: Consider any secret distribution protocol for users $1 \ldots n$. Without loss of generality, let n be even. In this protocol, add an additional secret between $(1,2)$, $(3,4)$, ..., $(n-1,n)$. With such modification, if the number of colluding users is less than $n/2$ then at least one pair of users can communicate securely. Thus, if we were to use the above definition then the protocol will be collusion resistant to $n/2$ users. Since this modification involves addition of one secret to each user, any secret distribution protocol can be trivially modified to get resistance to $n/2$ colluding users. In other words, the above definition fails to identify the true collusion resistance of different protocols. Therefore, the definition of collusion resistance should require a 'significant number' of pairs to be unaffected. Hence, we consider the following definition.

Attempt 2: A protocol is collusion resistant to x users if at least **half of the pairs** of users can communicate securely even if any subset of x users collude.

Although this definition does allow one to distinguish between collusion resistance of different protocols, the choice of half is arbitrary. Moreover, using this definition, it may be difficult to compute the collusion resistance of a particular protocol. Also, the collusion resistance varies widely if we change the requirement about the percentage of unaffected pairs.

Yet another problem with the above definition is that it does not allow us to identify the trend in the effect of collusion. Specifically, as the number of users in a system grows, it is desirable that the number of colluding users required to inflict the same disruption, computed in terms of the percentage of pairs affected, should also increase. Therefore, the definition of collusion resistance should be such that we can say 'a protocol is collusion resistant to $f(n)$ users if n is the total number of users in the system'. With this intuition, we now define the notion of a collusion resistance function.

Definition. *A function $\mathscr{C} : N \longmapsto N$, is a collusion resistance function for a protocol $\mathscr{P}$ iff*

$\lim_{n\to\infty} \frac{NumUnAffected(n,\mathscr{C}(n))}{TotalPairs(n)} > 0$*, where,*

$NumUnAffected(n, \mathscr{C}(n)) =$ *the minimum number of pairs in a system of n users that can communicate securely even if any subset of $\mathscr{C}(n)$ users collude, and,*

$TotalPairs(n) =$ *All possible pairs in a system with n users* $=n(n-1)$ □

Definition. *We say that a protocol $\mathscr{P}$ with n users is collusion resistant to $\mathscr{C}(n)$ users iff $\mathscr{C}$ is a collusion resistance function of $\mathscr{P}$.* □

Remark. Note that, $\frac{NumUnAffected(n,\mathscr{C}(n))}{TotalPairs(n)}$ is always in the range $[0, 1]$. Thus, the above definition requires that, as the system size increases, the ratio converges to a non-zero constant. A reader may wonder if we could have defined that a protocol is collusion resistant to $\mathscr{C}(n)$ users if the number of affected pairs is insignificant, i.e., $\lim_{n\to\infty} \frac{NumUnAffected(n,\mathscr{C}(n))}{TotalPairs(n)} = 1$. We note that such a definition would be viable. We consider this issue after identifying the collusion resistance of the square grid protocol.

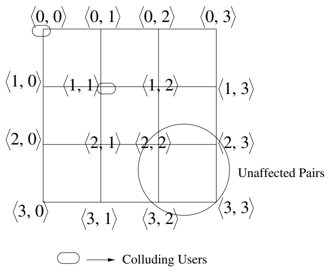


Figure 2. Effect of collusion on square grid protocol

Example: Collusion resistance of the square grid protocol. Observe that, in the square grid protocol in [1], the worst case disruption due to collusion occurs if the colluding users are in different rows and different columns. Moreover, without loss of generality, the grid locations of such colluding users can be renamed so that they lie along the diagonal of the square grid as such renaming does not affect the nature of secrets known to the colluding users. Thus, if r users collude, we can assume that they are at locations, $\langle 0, 0\rangle$, $\langle 1, 1\rangle \ldots \langle r-1, r-1\rangle$, along the diagonal.

Now, we show that the function, $\mathscr{C}(n) = \lfloor\frac{\sqrt{n}}{2}\rfloor$ is a collusion resistance function for the square grid protocol. To verify this, consider the case where the first $\lfloor\frac{\sqrt{n}}{2}\rfloor$ users along the diagonal collude (cf. Figure 2). The grid secrets of users in the top $\lfloor\frac{\sqrt{n}}{2}\rfloor$ rows and left $\lfloor\frac{\sqrt{n}}{2}\rfloor$ columns are compromised. However, the users in the lower right quadrant (cf. Figure 2) are not affected if they communicate within themselves. Since the number of users that are not affected is at least $\frac{n}{4}$, the number of unaffected pairs is at least $\frac{n}{4}.(\frac{n}{4}-1)$. Now,

$lim_{n\to\infty} \frac{\frac{n}{4}.(\frac{n}{4}-1)}{n.(n-1)} = \frac{1}{16} > 0$

From the above result, $\mathscr{C}(n) = \lfloor\frac{\sqrt{n}}{2}\rfloor$ is a collusion resistance function of the square grid protocol. In general, a function $c.\sqrt{n}$ is a collusion resistance function for the square grid protocol if $0 < c < 1$. Moreover, $1.\sqrt{n}$ is not a collusion resistance function because if all $\sqrt{n}$ users along the diagonal collude then, all the grid secrets are compromised. Thus, a user is able to communicate securely only with those users with which it maintains a direct secret. In other words, users will be able to communicate securely with users in their rows and columns. As each row (column) has $\sqrt{n}$ users, the number of pairs in a row (column) is at most n. Since there are $\sqrt{n}$ rows and $\sqrt{n}$ columns, the number of unaffected pairs is atmost, $2n\sqrt{n}$. And,

$lim_{n\to\infty} \frac{2n\sqrt{n}}{n.(n-1)} = lim_{n\to\infty} \frac{1}{\sqrt{n}} = 0$ □

Remark. If we had used the definition 'a protocol is collusion resistant to $\mathscr{C}(n)$ users if the number of affected pairs is insignificant, i.e., $\lim_{n\to\infty} \frac{NumUnAffected(n,\mathscr{C}(n))}{TotalPairs(n)} = 1$' then it could be shown that the square grid protocol is collusion resistant to $n^{(1/2)-\epsilon}$ users where ϵ is any positive number. We leave this proof as an exercise to the reader. Therefore, the conclusion reached about the collusion resistance of the grid protocol is essentially the same as that reached with our definition. We note that this observation is true for all the protocols considered in this paper.

4. Collusion Resistance Classes

Based on the above discussion of the square grid protocol, we now define, along the lines of complexity classes for algorithms, the notion of collusion resistance classes.

Definition. $\Omega_r(f(n))$ is the set of key distribution protocols for which $c.f(n)$ is a collusion resistance function for some (positive) value of c. In other words,

$\Omega_r(f(n))$ = $\{P \mid \exists c : c > 0 : c.f(n)$ is a collusion resistance function of P $\}$ □

Definition. $O_r(f(n))$ is the set of key distribution protocols for which $c.f(n)$ is ***not*** a collusion resistance function for some (positive) value of c. In other words,

$O_r(f(n))$ = $\{P \mid \exists c : c > 0 : c.f(n)$ is not a collusion resistance function of P $\}$ □

Definition. $\Theta_r(f(n))$ is the set of key distribution protocols that are both in $\Omega_r(f(n))$ and $O_r(f(n))$. In other words,

$\Theta_r(f(n))$ = $\Omega_r(f(n)) \cap O_r(f(n))$ □

Now, based on these definitions and the above discussion about the square grid protocol, we have:

Observation 4.0 Given ϵ and δ such that $0 \leq \delta \leq \epsilon$, we have

- $\Omega_r(n^\epsilon) \subseteq \Omega_r(n^\delta)$
- $O_r(n^\delta) \subseteq O_r(n^\epsilon)$

Theorem 4.1 The square grid protocol $\in \Theta_r(\sqrt{n})$. □

Remark. Although the above definitions are modeled along the complexity classes for algorithms, not all results about complexity classes may apply in this context and vice versa. For example, the above definitions are meaningful only if $f(n)$ is linear or a slower growing function. Also, since $f(n) = n$ is not a collusion resistance function for any protocol (because if all users collude then none can communicate securely), $O_r(n)$ consists of all secret distribution protocols.

Example: Collusion resistance of the full secret protocol. Now, we describe the full secret protocol and its collusion resistance. In this protocol, there is a unique secret associated with every pair of users. Thus, for a system of n users, each user maintains $n-1$ secrets.

The function $\mathscr{C}(n) = \lfloor \frac{n}{2} \rfloor$ is a collusion resistance function for the full secret protocol. To verify this, we note that the pairwise secrets shared by the remaining $\lceil \frac{n}{2} \rceil$ users are not compromised due to the collusion. Thus, the number of pairs that are unaffected due to collusion is at least $\frac{n}{2}.(\frac{n}{2}-1)$. Now,

$lim_{n\to\infty} \frac{\frac{n}{2}.(\frac{n}{2}-1)}{n.(n-1)} = \frac{1}{4} > 0$

From this result, we note that, $\mathscr{C}(n) = \lfloor \frac{n}{2} \rfloor$, is a collusion resistance function for the full secret protocol. Thus, we have,

Theorem 4.2 The full secret protocol $\in \Theta_r(n)$. □

Tradeoff between number of secrets and collusion resistance. The full secret protocol is in $\Theta_r(n)$ and requires each user to store $\Theta(n)$ secrets. The collusion resistance provided by the full secret protocol is, in some sense, the maximum collusion resistance offered by any secret distribution protocol. However, this protocol also requires the users to store the maximum number of secrets. On the other hand, the square grid protocol is in $\Theta_r(\sqrt{n})$ and requires each user to store $\Theta(\sqrt{n})$ secrets. The number of secrets stored by a user in the square grid protocol (cf. [1]) is within a factor of the minimum number of secrets that need to be stored in any secret distribution protocol that guarantees authentication and confidentiality in the absence of collusion. However, the collusion resistance provided by the square grid protocol is also lower than the full secret protocol.

Now, consider the following question: Is it possible to identify a family of protocols that provide a tradeoff between the number of secrets that users maintain and collusion resistance. Specifically, are there protocols in $\Theta_r(n^\delta)$, $1/2 \leq \delta \leq 1$, where the number of secrets maintained by users is $\Theta(n^\epsilon)$, $1/2 \leq \epsilon \leq 1$. Our approach to identify this family is based on variations of the square grid protocol. We consider three variations of the square grid protocol in Section 5. Then, in Section 6 we present the family of protocols that achieves the above requirements.

5. Constraints on Family of Collusion Resistant Protocols

In this section, we consider three variations of the square grid protocol. While these variations fail to identify the desired family of collusion resistant protocols, they identify three of the desired properties, (1) need to use a single grid instead of multiple grids, (2) need to use 2D grids instead of higher dimensional grids, and (3) need to use symmetric grids where the number of users in a row is (approximately) the same as that in column, that should be met by protocols in this family. Hence, a reader who is willing to take these properties for granted can skip this section.

Use of multiple grids. One way to increase collusion resistance of the grid protocol is to use multiple grids, as in [2]. Specifically, in [2], authors consider multiple square grids. In each grid, the locations in that grid are associated with a user and a secret. Furthermore, every user is assigned a location in every grid. Now, when two users communicate, they exchange their location in all the grids. Using these locations, they identify the grid secrets (if any) from every grid. (The protocol in [2] does not maintain direct secrets.)

In [2], authors show that if the number of rows (respectively, columns) is prime, then the users can be arranged in such a way that if two users are in the same row or column in one grid then they cannot be in the same row or column in any other grids. Thus, if t grids are maintained then any two users can use grid secrets from at least $t-1$ grids. Thus, when two users communicate, they need to use upto $2t$ secrets to establish the session key. It follows that if we want to keep the number of secrets used for establishing session key to be $O(1)$, t must also be $O(1)$.

Now, consider the collusion resistance class for the protocol in [2] where t grids are maintained. If the diagonal users from all grids collude then they know all the secrets in the system. Hence, no two users can communicate securely. Hence, $t\sqrt{n}$ is not a collusion resistance function for [2]. Similar to the square grid protocol, we can show that $\frac{1}{2}\sqrt{n}$ is a collusion resistance function for [2]. Thus, we have

Theorem 5.1 The multiple grid protocol from [2] where $O(1)$ grids are maintained is in $\Theta_r(\sqrt{n})$. □

Grid Protocol for Higher Dimensions. In the context of higher dimensional grid, we consider 3D grids. Each location $\langle i,j,k\rangle$, $0 \leq i,j,k < n^{1/3}$, is associated with a user $u_{\langle i,j,k\rangle}$ and a grid secret $k_{\langle i,j,k\rangle}$. Each user gets the grid secrets associated with the (3) planes it is in.[2] Also, the user maintains a direct secret with users in its planes. Thus, the number of secrets maintained by the user is $\Theta(n^{2/3})$. Now, when two users communicate, if they are in the same plane, they use the direct secret. Otherwise, they use all the grid secrets shared by them. Now, consider the case where the users on the diagonal of this grid, i.e., $\langle 0,0,0\rangle$, $\langle 1,1,1\rangle, \ldots \langle n^{1/3}-1, n^{1/3}-1, n^{1/3}-1\rangle$, collude. Since all grid secrets are thus compromised, users can communicate with only those with whom they maintain direct secrets. In other words, a user can communicate securely with only those users in its planes. As each plane has $n^{2/3}$ users, the number of pairs that can communicate securely is atmost, $n^{2/3}.n^{2/3}$. Since there are $3n^{1/3}$ planes in the grid, the total number of unaffected pairs is atmost, $3n^{1/3}.n^{2/3}.n^{2/3}$. Now,

$$lim_{n\rightarrow\infty}\frac{3n^{1/3}.n^{2/3}.n^{2/3}}{n(n-1)} = lim_{n\rightarrow\infty}\frac{1}{n^{1/3}} = 0$$

As in the case of the square grid protocol, if we consider that $\frac{n^{1/3}}{2}$ users on the diagonal collude then, we have at least $n/8$ users who can communicate securely within themselves. Therefore, similar to the square grid protocol, we can show that $\frac{n^{1/3}}{2}$ is a collusion resistance function for the 3D grid. Thus,

Theorem 5.1 The 3D grid protocol $\in \Theta_r(n^{1/3})$. □

Moreover, the number of secrets maintained by the users is $\Theta(n^{2/3})$. Thus, the collusion resistance of the 3D grid protocol is not as good as the square grid protocol even though it requires the users to store more secrets.

Rectangular Grid Protocol. Based on the discussion about protocols with higher dimensional grid, it is clear that to identify a family of collusion resistant protocols, we should focus on 2D grids. One variation of the 2D grid is a rectangular grid of size lxb, where $l \neq b$. In this case, the number of users is $l.b$. The square grid protocol from [1] can be trivially extended to such rectangular grids. The reason for considering such protocols comes from the observation that if $b = 1$ then this protocol is identical to the full secret protocol. Specifically, if $b = 1$ then all users are in a single row. Hence, there are no grid secrets and there is a direct secret between every pair of users.

Now, we evaluate the collusion resistance of such a protocol. Without loss of generality, we consider a grid with $l > b$. Consider the case where the users on the diagonal, $\langle 0,0\rangle, \langle 1,1\rangle, \ldots \langle b,b\rangle$, collude. Thus, all the grid secrets are compromised and a user can communicate securely with only those users with which it maintains a direct secret. Thus, secure communication is possible only along the rows and columns of the rectangular grid. As each column has b users and the number of columns is l, the number of unaffected pairs in the columns is atmost $l.b^2$. Likewise, the number of unaffected pairs in the rows is atmost $b.l^2$. Now,

$$lim_{n\rightarrow\infty}\frac{l.b^2+l^2.b}{n.(n-1)} = lim_{n\rightarrow\infty}\frac{\frac{n}{b}.b^2+(\frac{n}{b})^2.b}{n.(n-1)} = lim_{n\rightarrow\infty}\frac{1}{b}$$

Thus, if b is $O(1)$ (i.e., independent of n) then the above limit is non-zero. However, if b is of the form $f(n)$ where $lim_{n\rightarrow\infty}f(n) = \infty$ then the above limit is zero. Thus, the above protocol is collusion resistant to b users only if $b = O(1)$. Moreover, since $l > b$ then the collusion resistance of the rectangular grid protocol is no better than that of the square grid protocol. In other words, to identify the family of collusion resistant protocols, we should focus on protocols where the number of users in a row is (approximately) equal to the number of users in a column. We identify such a protocol family in Section 6.

6. Proposed Family of Collusion Resistant Protocols

Based on our discussion in Section 5, to identify a family of collusion resistant protocols, we should focus on 2D grids where the number of users in a row is approximately

[2] A reader may wonder if we could have allowed a user to only maintain the grid secrets in the (3) lines (instead of planes) it is in. With such an approach, we could reduce the number of secrets to $n^{1/3}$. However, with this approach, it is not possible for all users to communicate securely even in the absence of collusion. For example, users located at $\langle 0,0,0\rangle$ and $\langle 1,1,1\rangle$ do not have common secrets that they can use.

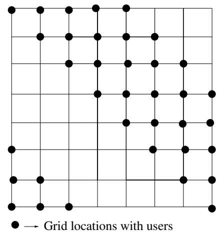


Figure 3. User assignment in the diagonal protocol

the same as that in a column. With this intuition, in this section, we propose a family of collusion resistant protocols that (1) are in $\Theta_r(n^\epsilon), 1/2 \leq \epsilon \leq 1$, (2) maintain $\Theta(n^\epsilon), 1/2 \leq \epsilon \leq 1$, secrets, and (3) the level of collusion resistance is proportional to the number of secrets that users maintain. Specifically, in Section 6.1, we present our *diagonal* protocol family and in Section 6.2, we identify its collusion resistance.

6.1. Diagonal Protocol Family

For a given set of n users, a protocol in this family arranges these users in a grid of size kxk, where $k \geq \sqrt{n}$. The value used to instantiate k identifies different members in this family. Similar to the square grid protocol, each grid location $\langle i, j\rangle$ is associated with a grid secret, $k_{\langle i,j\rangle}$. However, as there are more grid locations than users, some grid locations are not associated with users. We assign the users to grid locations along the diagonal. First, we arrange k users along the diagonal, i.e., these users are at locations $\langle x, y\rangle$ where $(y-x) \equiv 0 \ \ mod \ \ k$. Then, another k users are assigned to grid locations $\langle x, y\rangle$, where, $(y-x) \equiv 1 \ mod \ k$. We continue assigning the remaining users, to grid locations, $(y-x) \equiv 2 \ mod \ k$, $(y-x) \equiv 3 \ mod \ k$, and so on, until all the users are assigned a grid location. (cf. Figure 3 where 36 users are arranged in a 8x8 grid). Observe that with such assignment, the number of users in a row is approximately the same as the number of users in a column.

The secret distribution is identical to that of the square grid protocol. Specifically, a user gets the grid secrets in its row and in its column. And, each user shares a direct secret with users in its row and column. Moreover, the secret selection protocol is the same as that of the square grid protocol (recalled in Section 2.1). Now, we show that by appropriate instantiation of k, we can obtain the square grid protocol and the full secret protocol.

Obtaining the square grid protocol. If we instantiate $k = \sqrt{n}$ in the diagonal protocol family then all grid locations will be associated with users. Thus, the resulting protocol is the same as the square grid protocol. □

Obtaining the full secret protocol. If we instantiate $k = n$ in the diagonal protocol family then all users will be arranged along the diagonal. Thus, no direct secrets are maintained and each user maintains $2(n-1)$ grid secrets. Consider the secrets maintained by a user, say at location $\langle j, j\rangle$. When this user communicates with a user at location $\langle l, l\rangle$, it uses secrets at locations $\langle j, l\rangle$ and $\langle l, j\rangle$. Observe that while communicating with any other user, $\langle j, j\rangle$ (respectively, $\langle l, l\rangle$) uses neither of these secrets. Hence, instead of maintaining the secrets at locations $\langle j, l\rangle$ and $\langle l, j\rangle$, the user at locations $\langle j, j\rangle$ (respectively, $\langle l, l\rangle$) can only maintain a combination of these secrets. With this revision, the protocol is the same as the full secret protocol. □

Remark. In the subsequent discussion, for brevity, we use the term 'a diagonal protocol' to mean 'a member in the diagonal protocol family'.

6.2. Collusion Resistance of Diagonal Protocol

In the diagonal protocol, if a kxk grid is used for a group of n users then users will be assigned to locations $\langle x, y\rangle$ where $y - x \equiv w \ mod \ k$, where $0 \leq w < \lceil n/k \rceil$. For simplicity, we assume that n/k is an integer and ignore the fact that some locations where $y - x \equiv (\lceil n/k \rceil - 1) \ mod \ k$ are not associated with users. This assumption is reasonable as we are interested in asymptotic behavior of the protocol while computing its collusion resistance.

Now, we consider the collusion resistance of such a protocol. Similar to the protocol in [1], the colluding users cause the maximum disruption when they are in different rows and columns. Hence, without loss of generality, we can assume that if there are r colluding users then they are at locations $\langle 0, 0\rangle$, $\langle 1, 1\rangle \ldots \langle r-1, r-1\rangle$, along the diagonal.

We consider the case where there are k/2 colluding users. Similar to our discussion in Section 3, we can see that that the users in the lower right quadrant can communicate securely within themselves. Now, we compute the number of users in this quadrant. The users in the lower right quadrant are at locations $\langle x, y\rangle$ where $x \geq k/2$ and $y \geq k/2$. The number of such users where $y - x = 0 \ mod \ k$ is $k/2$. Also, the number of such users where $y - x = 1 \ mod \ k$ is $k/2 - 1$, and so on. Thus, the number of users in the lower right quadrant is:

$$\begin{array}{ll} & k/2 + (k/2-1) + ... + (k/2 - n/k + 1) \\ \geq & (k/2 - n/k) * n/k \\ \geq & n/2 - (n/k)^2 \end{array}$$

Now, if $k = n^\epsilon$ where $1/2 < \epsilon < 1$ then $(n/k)^2$ is o(n). Therefore, the number of users in the lower right quadrant is n/2 $-$ o(n). Thus, if $k = n^\epsilon$, $1/2 < \epsilon < 1$, then the number of users in the lower right quadrant is $\Theta(n)$. And, the number of unaffected pairs is $\Theta(n^2)$. It follows that the diagonal protocol with a kxk grid is resistant to collusion of $k/2$ users.

Furthermore, if all k users along the diagonal collude then all grid secrets are compromised. Thus, a user can securely communicate with only users in its row/column. There are n/k users in each row/column. And, there are k rows and columns. Hence, the number of unaffected pairs is at most $2k(n/k)^2$. If $k = n^\epsilon$ then the number of unaffected pairs is atmost $2(n^{2-\epsilon})$. It follows that the diagonal protocol is not collusion resistant to k users.

Based on the above discussion, we have:

Theorem 6.1 The diagonal protocol with a n^ϵ x n^ϵ grid $\in \Theta_r(n^\epsilon)$. □

Thus, the diagonal protocol family consists of protocols (one for each value of ϵ) such that the number of secrets maintained by users in these protocols is proportional to the collusion resistance provided by them.

7. Analysis of Collusion Resistance

In this section, we compare the collusion resistance of different protocols in the diagonal protocol family. Specifically, we consider the protocols in the diagonal protocol family where the grid size is kxk, where k is either $\sqrt{n}, n^{2/3}, n^{3/4}$ or $n^{4/5}$. Of these, $k = \sqrt{n}$ corresponds to the protocol from [1]. We compare collusion resistance of these protocols in two cases (1) where the colluding users are selected in such a way that they cause the maximum number of user pairs to be affected, and (2) where colluding users are selected randomly. The former corresponds to the case where colluding users can select their grid locations whereas the latter corresponds to the case where the colluding users do not have such capability.

In these simulations, we also consider a randomized version of the protocol family from Section 6. To compare this with the diagonal protocol family, observe that in a diagonal protocol, deterministic approach is used to ensure that the number of users in a row is approximately equal to the number of users is a column. An alternative approach is to randomly assign grid locations to users with uniform probability. With such an approach, the expected number of users in a row is the same as the expected number of users in a column. Hence, it is expected that such a protocol can be used in identifying a family of collusion resistant protocols. Unfortunately, in this protocol, it is difficult to identify the worst-case disruption that can occur due to colluding users. For this reason, in this protocol, we consider the case where a random collection of users is selected to collude.

In the following simulation results, the term 'Grid' denotes the protocol in [1] (also equivalent to the diagonal protocol with $k = \sqrt{n}$). The term 'Diagonal $k = n^\epsilon$' denotes the diagonal protocol with n^ϵxn^ϵ grid where colluding users are selected along the diagonal. And, as discussed in Section 3, this causes maximum user pairs to be affected. The term 'Random' denotes the above random distribution of users where colluding users are selected at random. Finally, the term 'Diagonal $k = n^\epsilon$ with random collusion' denotes the diagonal protocol with n^ϵxn^ϵ grid where colluding users are selected randomly. For the experiments with random collusion, we repeated the simulation five times and took the average of these simulations. For other experiments, repetition is not required as the calculation is deterministic.

In Figures 4(a), (b) and (c), respectively, we show the effect of, 5, 10 and 20 colluding users. From these figures, we see that, as the number of users increases, the number of affected pairs in the diagonal protocol with $k = n^{2/3}$ is less than that in the grid protocol. The percentage of user pairs affected due to random collusion is slightly less than the case where colluding users are along the diagonal. This is due to the fact that with random colluding users, some of the colluding users are in the same row/column. Moreover, even with random colluding users, the diagonal protocol provides slightly better collusion resistance when compared to the case where users are randomly distributed in the grid. In other words, even if the colluding users were to be selected at random, it is beneficial to arrange the users deterministically.

Figure 5 compares collusion resistance of different protocols in the diagonal protocol family. When users are arranged in a n^ϵxn^ϵ grid, the number of pairs affected due to collusion decreases as the value of ϵ increases. Thus, the level of collusion resistance is proportional to the number of secrets that users maintain.

Finally, Figure 6 evaluates the effect of different collusion resistance functions for the protocols in the diagonal protocol family. Based on the discussion in Section 6, for a diagonal protocol with n^ϵxn^ϵ grid, $c.n^\epsilon$ is a collusion resistance function if $c < 1$. We consider the number of affected user pairs for the case where c equals 1/2,1/3, ..., 1/6. As shown in Figure 6, the number of user pairs affected reaches a constant as the number of users is increased. This validates the expected result that in a diagonal protocol with n^ϵxn^ϵ grid, the percentage of pairs unaffected by collusion reaches a non-zero limit if the number of colluding users is $c.n^\epsilon$ where $c < 1$.

8. Related Work

Existing work on secret instantiation can be classified based on whether intermediate users are trusted or not. In [9–12], where the intermediate users are trusted, when two users communicate, their communication is decrypted and re-encrypted by intermediate users. Thus, in these protocols, even in the absence of collusion, some (several) pairs of users cannot communicate securely. By contrast, the pro-

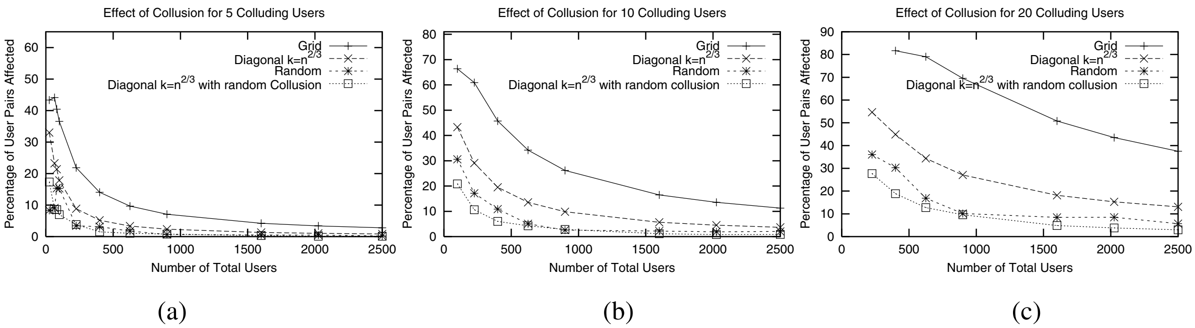


Figure 4. Effect of collusion on various protocols. (a) 5 colluding users (b) 10 colluding users and (c) 20 colluding users

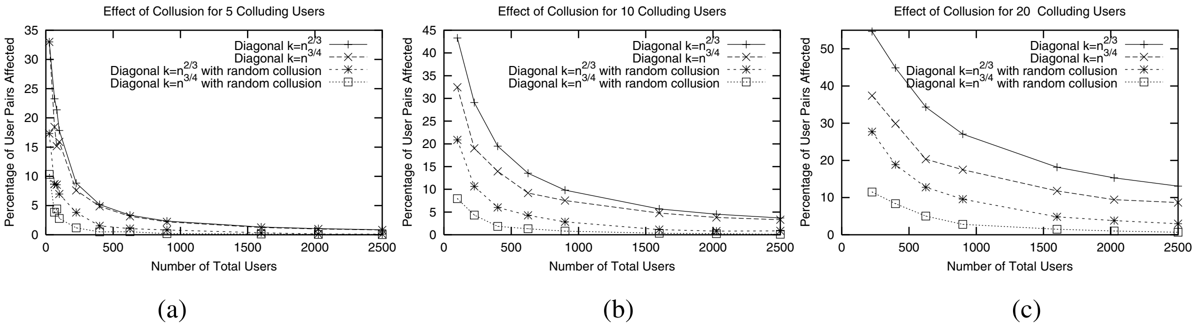


Figure 5. Collusion in diagonal protocol for different k values. (a) 5 colluding users (b) 10 colluding users and (c) 20 colluding users

tocols in [1, 2], ensure that any two pair of users can communicate securely.

Existing work on collusion resistance [14, 15] has focused on approaches for ensuring that all pairs of users can communicate securely even if some threshold (t) number of users are compromised. They do not focus on the situation where the colluding users exceed this threshold. By contrast, our work has focused on what happens if the number of colluding users exceeds such threshold and therefore only a subset of user-pairs can communicate securely. Our approach can be combined with previous work so that if colluding users is less than t then all pairs of users can communicate securely. However, when a larger number of users collude, only some pairs can communicate securely.

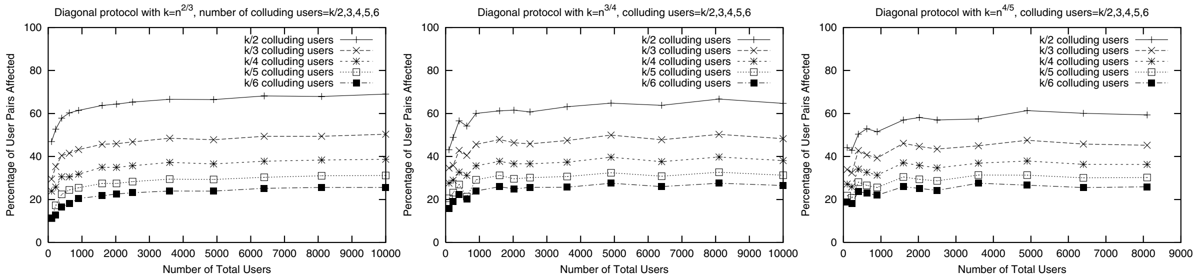
9. Conclusion

In this paper, we presented a family of collusion resistant protocols, the diagonal protocol family, where the level of collusion resistance is proportional to the number of secrets that users maintain. The proposed protocol family is based on the square grid protocol from [1]. We showed that other variations of this protocol, however, failed to identify the family of collusion resistant protocols.

We defined the notion of collusion resistance classes. We showed that these collusion resistance classes could be effectively used to compare the collusion resistance of different protocols. We identified membership of existing protocols as well as protocols in the proposed family in these classes. We also validated these results through simulation. Specifically, we showed that given a collusion resistance function $\mathscr{C}(n)$ of a protocol, the percentage of unaffected pairs due to collusion of $\mathscr{C}(n)$ users in a system of n users is unchanged as the number of users is increased.

For reasons of space, we did not discuss in detail *how* the user obtains the initial secrets as this issue is orthogonal to the issue of *what* secrets a user should get. A user may obtain these initial secrets in several ways, e.g., a user may obtain these secrets by initially visiting (respectively, periodically revisiting) a trusted server. Also, the problem we discussed is orthogonal to the issue of secret maintenance [16], where users change their secrets periodically to thwart cryptanalytic attacks.

One of the open questions from this work is the optimality of the number of secrets maintained in order to provide the required level of collusion resistance. In the proposed diagonal protocol family, if $\Theta(n^\epsilon)$ secrets are maintained then the resulting protocol is in $\Theta_r(n^\epsilon)$. For $\epsilon = 1/2$ (where we obtain the protocol from [1]), the number of secrets maintained is within a constant factor of the optimal. Also, for $\epsilon = 1$, the number of secrets maintained by a user is within a constant factor of optimal. However, the optimality is not known for intermediate values.



(a) (b) (c)

Figure 6. Collusion in diagonal protocol for k/X colluding users. (a) k = $n^{2/3}$ (b) k = $n^{3/4}$ (c) k = $n^{4/5}$

References

[1] Sandeep S. Kulkarni, Mohamed G. Gouda, and Anish Arora. Security instantiation in ad-hoc networks. *Special Issue of Elsevier Journal of Computer Communications on Dependable Wireless Sensor Networks*, 2005. A Preliminary Version Appears in Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS), June 2004, Florence, Italy.

[2] Li Gong and David J. Wheeler. A matrix key-distribution scheme. *Journal of Cryptology*, 2(1):51–59, 1990.

[3] J. Kong, P. Zefros, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiqutious security support for mobile ad-hoc networks. *IEEE International Conference on Network Protocols*, 2001.

[4] J. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad-hoc networks. *ACM Symposium on Mobile Ad Hoc Networking & Computing*, 2001.

[5] L. Zhou and Z. Haas. Securing ad hoc networks. *IEEE Network*, 13(6), 1999.

[6] M. Tatebayashi, N. Matsuzaki, and D.B. Newman Jr. Key distribution protocol for digital mobile communications systems. *Advances in Cryptology*, 1990.

[7] V. Varadharajan and Y. Mu. Design of secure end-to-end protocols for mobile systems. *Wireless*, 1996.

[8] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communications of ACM*, 21:993–999, 1978.

[9] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. *IEEE Symposium on Security and Privacy*, 2003.

[10] L. Eschenauer and V. Gilgor. A key management scheme for distributed sensor networks. *ACM Conference on Computer and Communications Security (CCS)*, pages 41–47, 2002.

[11] W. Du, J. Deng, Y. Han, and P. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. *ACM Conference on Computer and Communications Security (CCS)*, pages 42–51, 2003.

[12] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. *ACM Conference on Computer and Communications Security (CCS)*, pages 52–61, 2003.

[13] R. Blom. Non-public key distribution. *Advances in Cryptology: Crypto*, pages 231–236, 1982.

[14] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly secure key distribution for dynamic conferences. *Advances in Cryptology*, pages 344–355, 1992.

[15] R.Blom. An optimal class of key generation systems. *Lecture Notes in Computer Science*, pages 335–338, 1984. Advances in Cryptology –Eurocrypt '84.

[16] V. Naik, S. Bapat, A. Arora, and M. Gouda. Whisper: Local secret maintenance in sensor networks. *Workshop on Principles of Dependable Systems*, 2003.