As shown in Figure 4.3, an AC is exchanged along Agt_1, Agt_2, ..., Agt_k. Since those agents are strangers, the only relationship between two adjacent agents is that they encounter each other somewhere. The relationship between Agt_1 and Agt_k becomes weaker when we increase k. In other words, attackers can hardly infer the identity of Agt_k when he only knows the identity of Agt_1, and his difficulty increases with the increase of parameter k. As a result, it is hard for attackers to infer the original requester, even though Agt_k is in the social tie of the original requester. Since the reply message must go through a series of agents, a long obfuscation distance also lengthens the path of the reply message. Therefore, a large k makes the original requester safer, while making it harder and longer for the reply to be delivered.

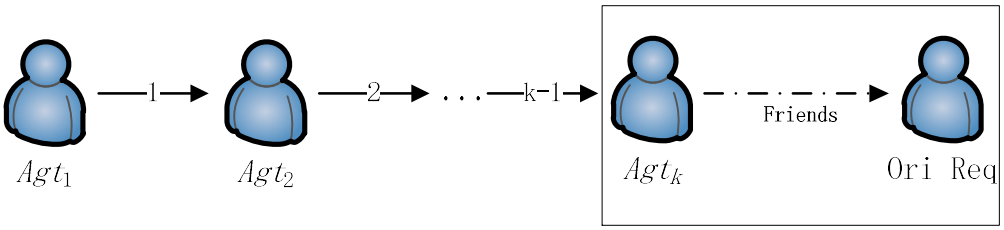

Figure 4.3: Obfuscation Distance

4.5.2. Friends Obfuscation Distance

Since Agt_{k-1} is a stranger for Agt_k, it is possible that Agt_{k-1} is exactly the attacker. We assume that the attacker knows that Agt_k is in the social tie of the original requester. The attacker can assume that Agt_k is a close friend of the original requester, so the identity of Agt_k gives the attacker a good tip to infer the original requester. A solution to prevent the agent Agt_{k-1} from learning the original requester easily is that the last m agents are friends, as shown in Figure 4.4. Here, m is called *friends obfuscation distance*, and the last m agents are trusted agents.

It is true that Agt_i is a friend of Agt_{i+1}, $i > k - m$, but two nonadjacent trusted agents (e.g., Agt_i and Agt_{i+2}) might have a weak relationship. Since there are at least m trusted