ABSTRACT:

While OSNs today provide some form of privacy controls to protect a user’s shared content from other users, these controls are not sufficiently expressive to provide fine-grained protection. In this article, we introduce Twitsper, to support fine-grained control over who sees a user’s messages. Twitsper provides privacy controls to the users of Twitter today without relying on Twitter to make changes. This is because it is a wrapper around Twitter that enables private group communication while preserving Twitter’s commercial interests. It preserves privacy both from the Twitsper server as well as from undesired Twitsper users.

EXISTING SYSTEM:

Currently, Twitter provides its users only two kinds of privacy controls—the ability to share content with all their followers (using tweets) or with exactly one follower (using direct messages)

DISADVANTAGES OF EXISTING SYSTEM:

- Controls aren’t sufficiently expressive to provide fine-grained protection.
- No private group communication.
PROPOSED SYSTEM:

In this article, we introduce Twitsper, a wrapper over Twitter that provides privacy controls to the users of Twitter today without relying on Twitter to make changes. Currently, Twitter provides its users only two kinds of privacy controls—the ability to share content with all their followers (using tweets) or with exactly one follower (using Direct Messages). Twitsper enables users to share content with any subset of their followers, without requiring the followers to migrate to a new application or OSN.

ADVANTAGES OF PROPOSED SYSTEM:

- Sharing private information with some followers, Twitsper can also be used to segment information and send only relevant information to subsets of followers.
- While Twitsper does not hide content from Twitter itself, some users may desire to do so. This raises a different set of challenges and may impact Twitter’s revenue model.
- We created Twitsper to enable users to share content with any subset of their followers without requiring the followers to migrate to a new application or online social network (OSN).
MODULES:

- Twitsper Interface Module
- Protecting Privacy
- Threat Model
- Protection from Undesired Users

MODULES DESCRIPTION:

**Twitsper Interface Module:**

Our first module is to develop the Twitsper Module. In addition to sharing messages with all followers or precisely one follower, Twitsper lets users privately share messages with a subset of their followers by extending Twitter’s API with additional functionalities. Note that Twitter associates every tweet, direct message, user, and list with a unique ID.

**Protecting Privacy Module:**

Although users don’t directly send content to the Twitsper server, the mapping of direct message IDs to list IDs is provided to the server to support group conversations. This metadata could reveal the identities of the members involved in the private
conversation or the size of the group, which users might desire to keep private. Hence, we incorporated several features that hide this information both from the Twitsper server and other undesired users.

**Threat Model**

Twitsper’s components are Twitter, user devices, the Twitsper server, and the channel connecting these entities. We trust Twitter not to leak a user’s private information; this was the premise of our work. We assume that users’ personal devices don’t compromise their privacy. Thus, the two potential leakage sources are the Twitsper server and the channel. The Twitsper server is the only new addition to the preexisting Twitter architecture. As we discussed, in Twitsper’s architecture, private content is always posted to Twitter’s servers, thus ensuring that this content isn’t leaked due to the Twitsper server. The threat is then the leakage of the metadata associated with private content that might be exposed to the server. Because we administer the Twitsper server, we assume that the server Won’t modify or delete metadata stored on it.

**Protection from Undesired Users Module**

Curious users who aren’t privy to a private conversation might want to trick the Twitsper server into disclosing whether one of their friends has initiated a private conversation. Recall that an initiator of a private conversation sends direct messages to
a private group, then seeks to create a mapping on the supporting server between the unique IDs for those messages and the recipient list. This way of storing the mappings on the server has two benefits. First, because the hash function is noninvertible, the server can’t infer the identity of the users involved. (The text input to the hash function is only known to the group members and thus not available to the server.) Second, even if undesired users guess the IDs of the posted messages, they can’t retrieve the desired mapping, again because they don’t know the text provided as input to the hash.

**SYSTEM CONFIGURATION:-**

**HARDWARE CONFIGURATION:-**

- Processor - Pentium –IV
- Speed - 1.1 Ghz
- RAM - 256 MB(min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA
SOFTWARE CONFIGURATION:-

✓ Operating System : Windows XP
✓ Programming Language : JAVA/J2EE.
✓ Java Version : JDK 1.6 & above.
✓ Database : MYSQL