Sqoop2 Securing Passwords

Support reading password from password file

- Sqoop2 should support reading passwords from password file specified by the user. When running jobs, this file should be copied over to HDFS using owner readonly permission.
- Even better would be supporting copying a password protected keystore (maybe via distributed cache mechanism) that contains the key (since it is often managed by the DBAs). Then the job would pass the decryption password under some owner r/o permission as you describe.
- For my use case, passwords need to be retrieved dynamically from a separate system. So at run time I would need to generate a file with the passwords in it, or create a keystore (from Java). Ideally I'd make an HTTPS request to the server that knows about the passwords. Which means needing to be able to specify a "password provider" class as a Sqoop parameter, along with an optional parameter to pass in to the constructor so it knows which password I need (depending on the database I'm accessing).
- We'd like to keep our database passwords as secure as possible, but currently in Sqoop1, the password is plainly visible from the jobtracker xml.
- Require that the passwords are not transmitted in clear text when Sqoop makes the JDBC connection to Teradata.
    - What the JDBC driver does is beyond Sqoop's control. Unless there is a special API that the driver exposes (which I am not aware of at the moment), the only way to reasonably achieve this would be to use JDBC over SSL. Doing this does not require any changes on Sqoop side as long as the the driver and the server are able to work with the secure layer.
- Less concerned with how the password is exposed during the running job and more concerned with how the password is persisted on disk and transmitted when it is to be used by the Sqoop jobs as part of an automated Oozie workflow.