# 164.https://stackoverflow.com/questions/72613920/how-do-i-securely-sign-a-transaction-within-my-solana-dapp

**T:**How do i securely sign a Transaction within my Solana Dapp

**Q:**An NFT project I'm working with was recently rugged and I'm setting up a dapp with goal of derugging the project. We are trying to achieve this by destroying all of the NFTs in the old collection and replace them with a new collection of NFTs created with our update authority. With this in mind, the dapp has two basic functions for the user.  
  
 ● Burn the old NFTs from our collection held in the user's wallet.  
  
 ● Mint a new copy of old NFT using our update authority.  
  
The problem I've run into is that in order to mint the new NFT using our update authority, our wallet must also sign the transactions along with the user. However, it seems very insecure to sign the transaction from within our dapp because a malicious actor could get ahold of our keypair.  
  
What is best practice for this type of problem? How do we sign for transactions within our dapp without exposing our private keys?  
  
The project code is living here: https://github.com/Giffen-good/fomo-bombs-dapp

1 **Answer**

**A1:**First: If you want to have your users exchange an old NFT with a new one you should have a look at the Token entangler. It's a metaplex tool which provides exactly this feature. Unfortunately it's not well documented though. You might know it from exiled apes / SOL Sanctuary  
  
But in regards to your question: The only "safe" way would be to have the user send a API Request to your backend where you build and sign the transaction, then give it back to the user to sign it:  
  
 ● Have the user send a API request with the necessary data to your backend  
  
 ● Build your transaction in the backend  
  
 ● Sign the TX in your backend (It can NOT be changed afterwards)  
  
 ● send the tx object back to your user and have them sign it  
  
 ● send the signed tx to the RPC  
  
With this you don't expose the PK to the frontend. You should still be very careful when hosting your PK in the backend.