RanchiMall

Single Data Authentication

Technology

RANCHIMALL
INTRODUCTION

The real adoption of blockchain will come only when we improve on the Internet access convenience for masses.

How about elimination of Usernames and Passwords ?

Username and Password system has been responsible for massive compromises and hacks.

What if we just used **digital signatures** for access to data.

What is we had the end user just enter his **private key** on his local browser, and send send digitally signed requests to servers.

What if we had the end user just enter his private key on his local browser, and send digitally signed requests to servers.

That's what RanchiMall calls Single data authentication.

Ability to just use private keys instead of usernames and passwords.

RanchiMall has created web applications where user needs to just enter his private key locally, and he can get access to his resources from distributed servers.

One of the major reasons big websites get hacked is because they need to store password hashes of the users.

Our authentication architecture totally eliminates any server side storage or password hashes which improves the system security massively.

Our architecture also eliminates the need for a new user to confirm email addresses.

Email address confirmation is needed to make sure users are not giving someone else email ID in registration process.

With blockchain private key based authentication, the need to have email address itself is eliminated.

Our architecture also eliminates need to do session mangemement in password based applications.

Our applications can simply send signed requests again and again to different distributed servers for different kinds of data.

Elimination of session management totally makes the applications safer by removing threat of session hijacking.

With a private key based authentication system, we get another feature: Mutilple applications can be integrated without any commonalities.

Like we can have intern management system, and intern application systems totally independent, yet linked through common private key signed data.

It is impossible to achieve integration of multiple applications without common database or common server code in password based systems.

A key precaution that needs to be taken that users should not enter their private key in unknown and untrusted websites.

Also users should take precautions to their local browsers safe in order to prevent theft of private key.

We can build massively scaleable applications one small piece at a time using single data authentication just by building small application pieces at one time.

We can build massively scaleable applications using single data authentication just by building small application pieces at one piece at a time.

A whole web based bank can be created by having deposit function created and activated first, and collateral functions created and activated next independently, and lending functions similarly being independently constructed later.

The ability to do private key authentication at client side represents dependence away from rigidities of server side design.

It represents massive abilities to unify very distant applications just using user data signatures.

Private Key based authentication eliminates the need to enforce password policies for the user and private keys are very strong passwords.

It reduces the security perimeter of the system just to the private key, making it easier to keep the system safe.

Single data authentication

1. Improves usability for users.
2. Increases scalability of applications.
3. Improves security of the system.

Typically when usability improves, security becomes worse.

Having both of them to improve together is a rare phenomenon in any design change.

Single data authentication also permits users to chose their own login identities.

In current password based systems, the server has to allocate a free username.

That makes the user ID unblockable globally which regularly happens in centralized applications today.

It improves user empowerment.