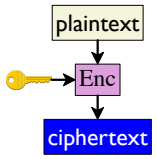
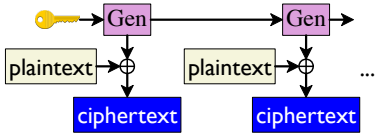
Cryptography Toolbox

block cipher



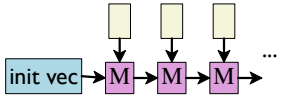
fixed-size

stream cipher



any length

hash function



$$H(x) = H(y)$$
$$\Rightarrow$$
$$x = y$$

Diffie-Hellman

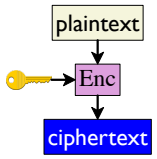
shared secret

RSA

authentication

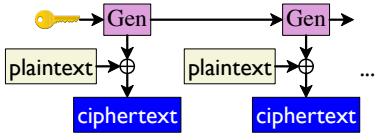
Cryptography Toolbox

block cipher



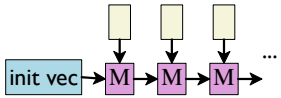
fixed-size

stream cipher



any length

hash function



$$H(x) = H(y) \Rightarrow x = y$$

Diffie-Hellman

shared secret

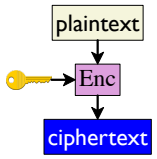
RSA

authentication

Cryptographic algorithms need to mix these ingredients correctly

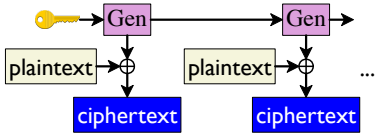
Cryptography Toolbox

block cipher



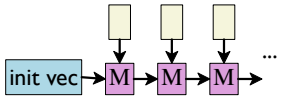
fixed-size

stream cipher



any length

hash function



$$H(x) = H(y) \Rightarrow x = y$$

Diffie-Hellman

shared secret

RSA

authentication

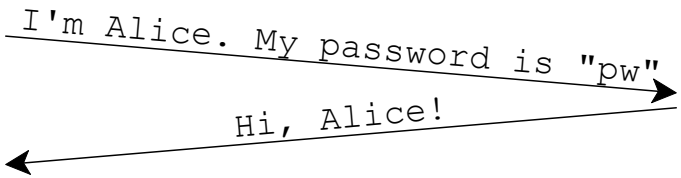
Cryptographic algorithms need to mix these ingredients correctly

... including for authentication

Sending a Password

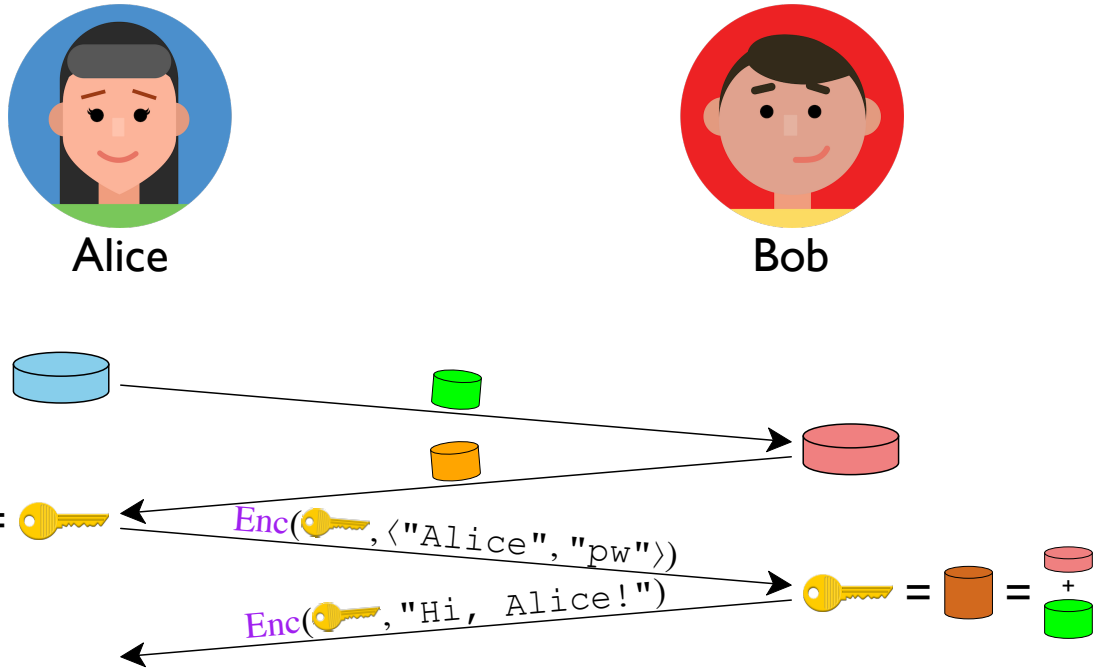
Alice

Bob



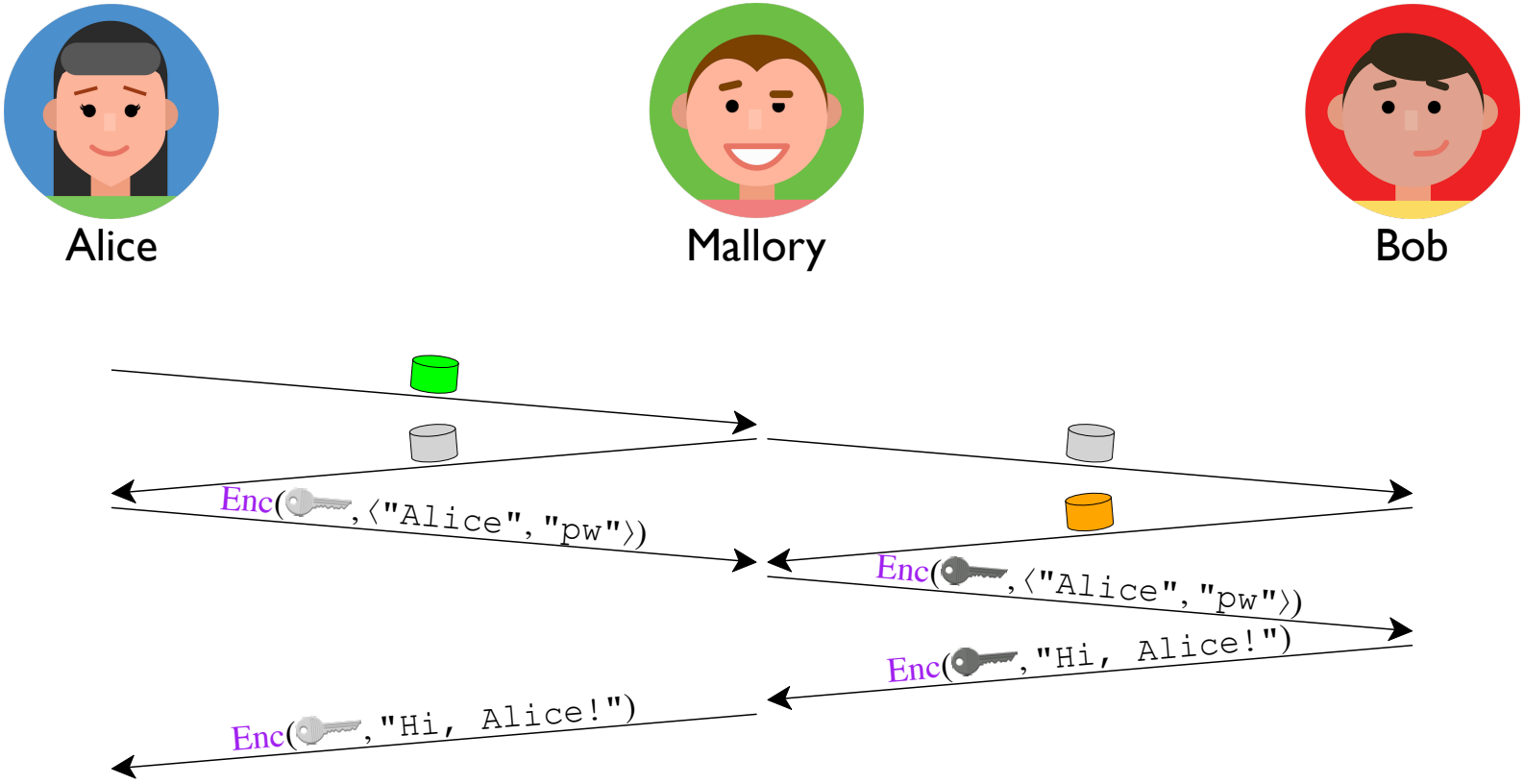
Obviously bad to send a password as plaintext

Sending a Password

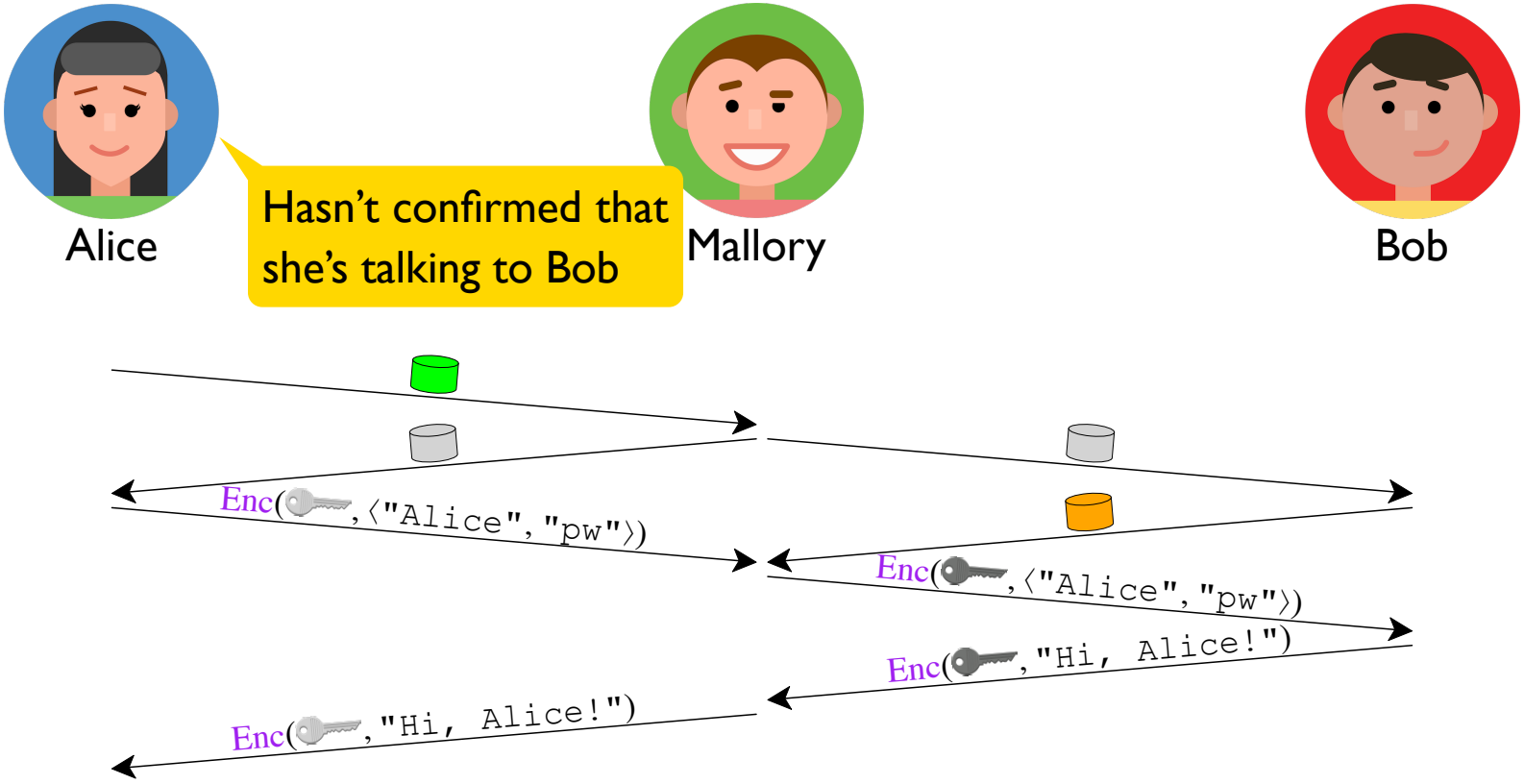


Use Diffie-Hellman to get a shared secret key?

Sending a Password



Sending a Password



Don't send a password without authenticating first!

Authentication

Authentication requires some prior arrangement

Some options:

- previously shared secret 
- previously acquired RSA 

and 

For now, assume a previously shared 

Some Bad Authentication Ideas

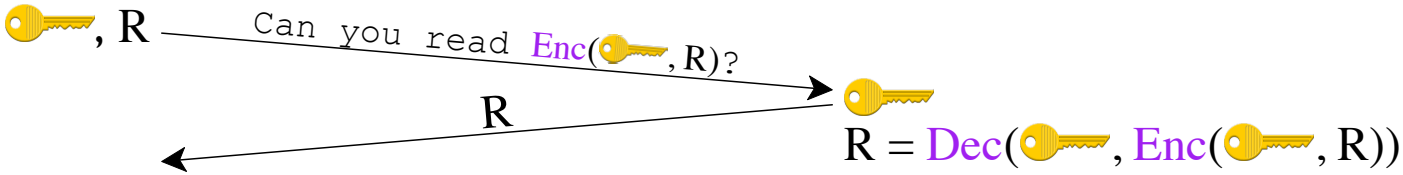
Alice

Bob

Some Bad Authentication Ideas

Alice

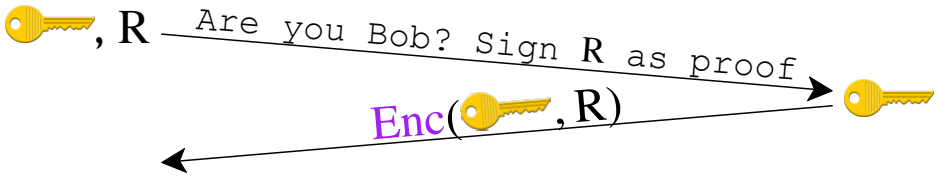
Bob



Some Bad Authentication Ideas

Alice

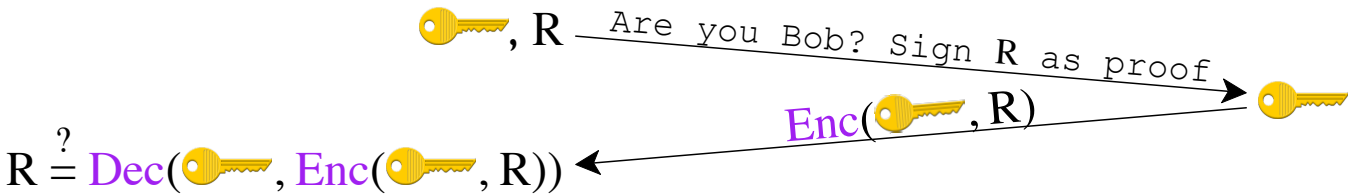
Bob



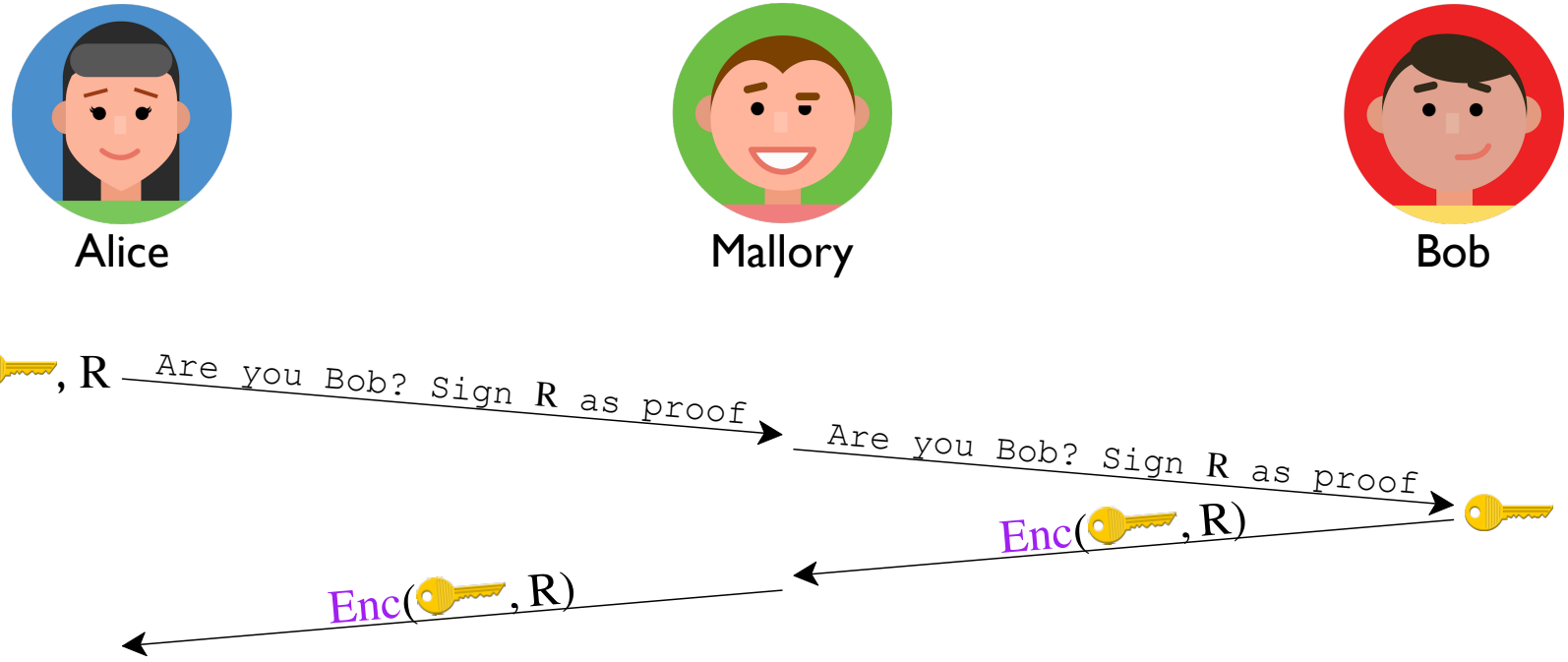
Some Bad Authentication Ideas

Alice

Bob



Some Bad Authentication Ideas



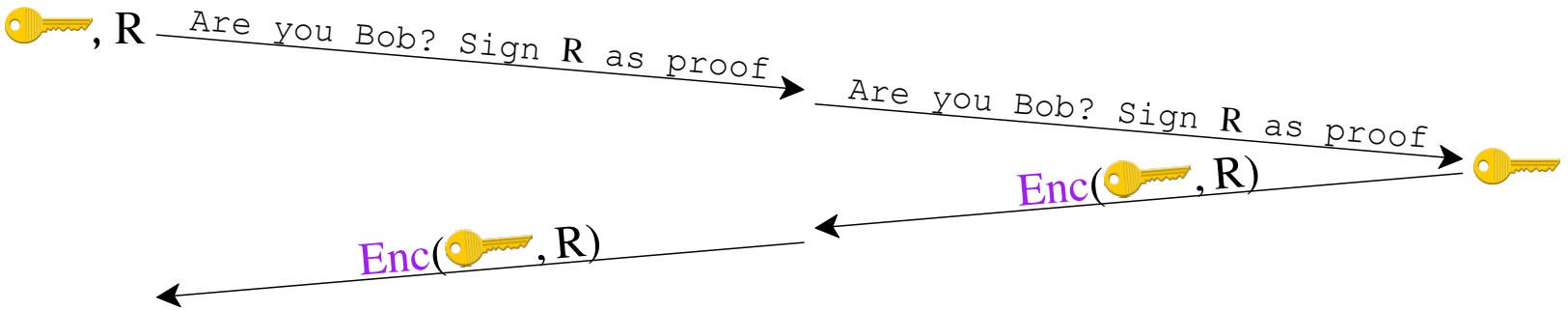
Whether this is immediately bad depends on whether the rest of the conversation encrypts with

Some Bad Authentication Ideas

Alice

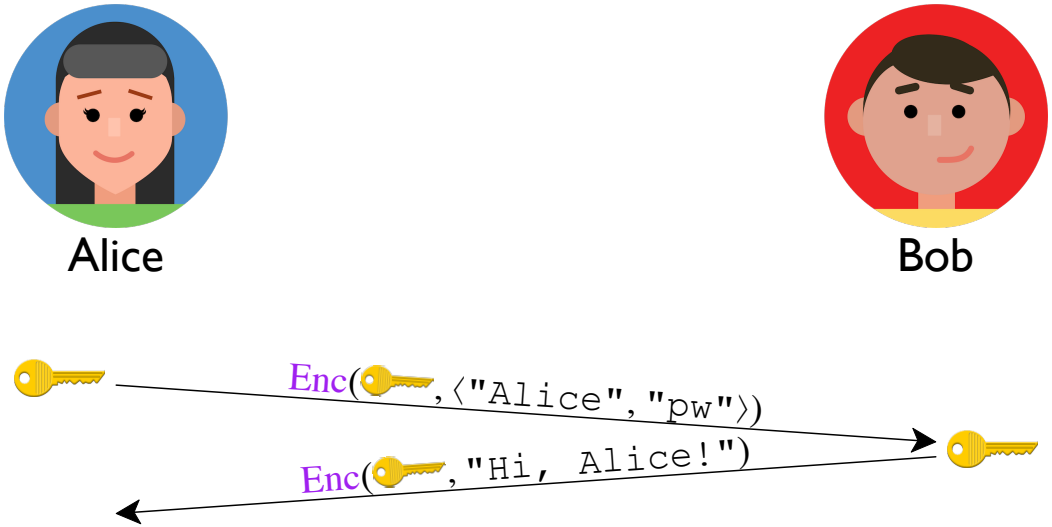
Mallory

Bob

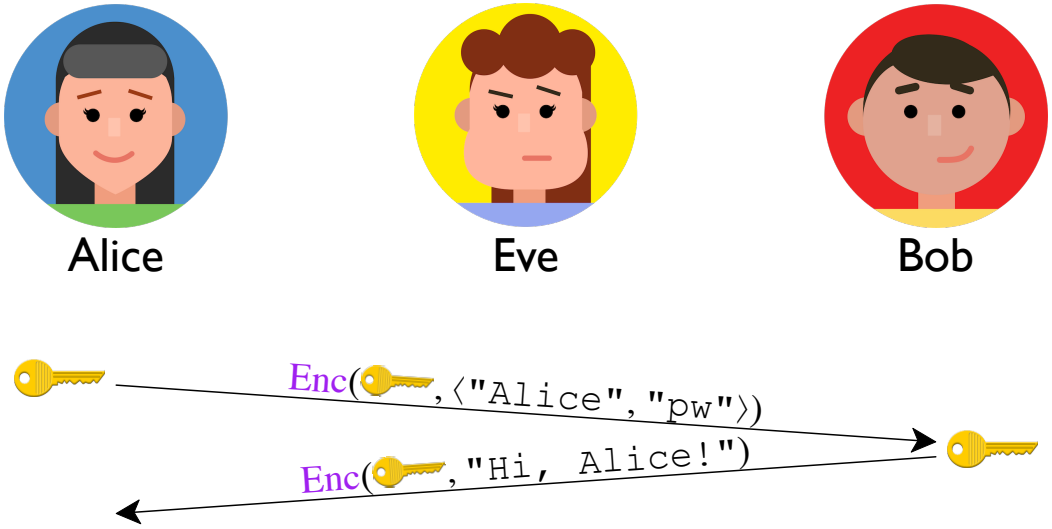


More generally, the problem here is that Bob has blindly signed a value handed to him

Some Bad Authentication Ideas

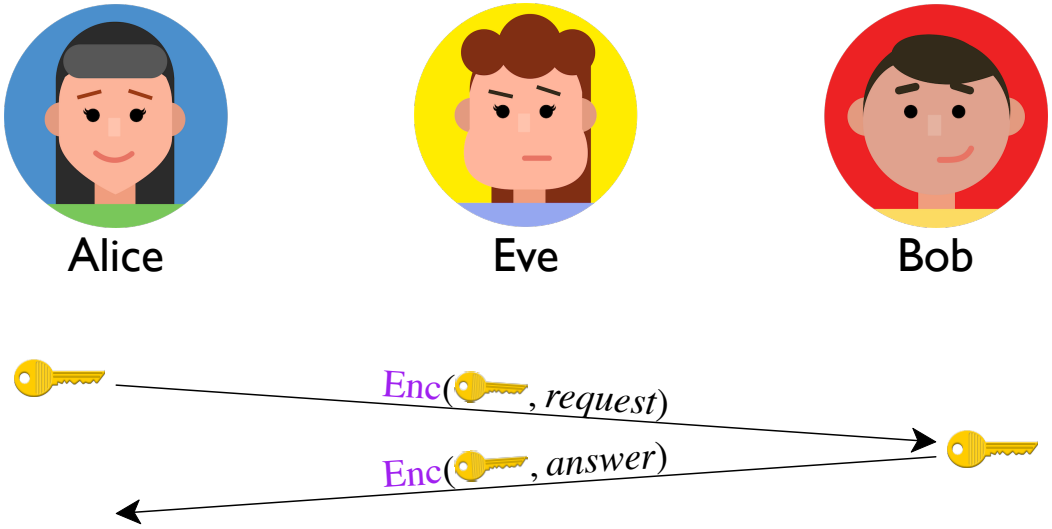


Some Bad Authentication Ideas



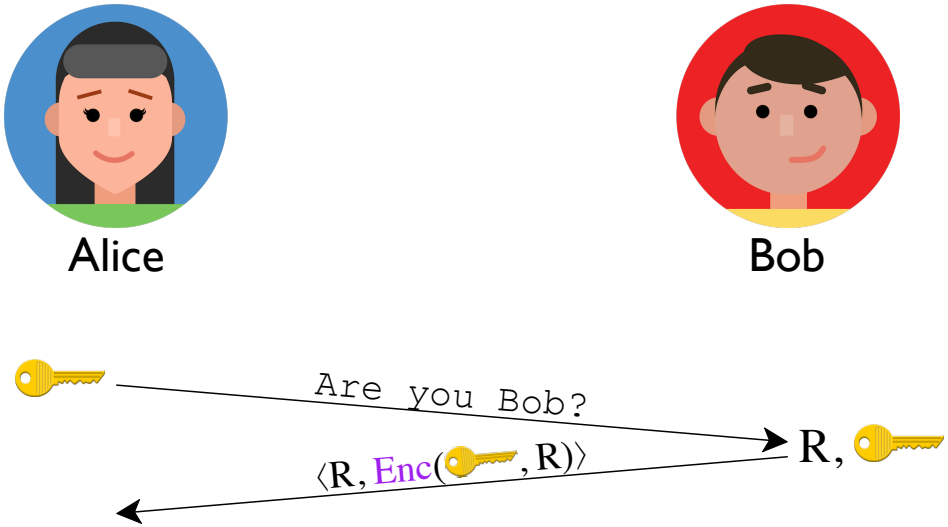
Replay attack: Eve can record Enc(, ⟨"Alice", "pw"⟩) and play it back later

Some Bad Authentication Ideas



Anyway, no need for a separate password if you have 🔑, but beware of the same problem with other predictable messages

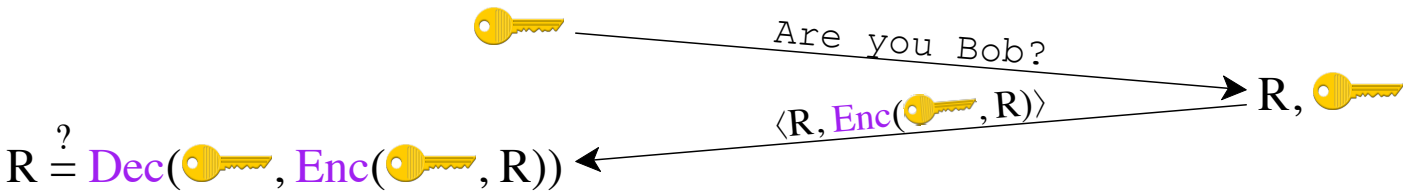
Authentication



Authentication

Alice

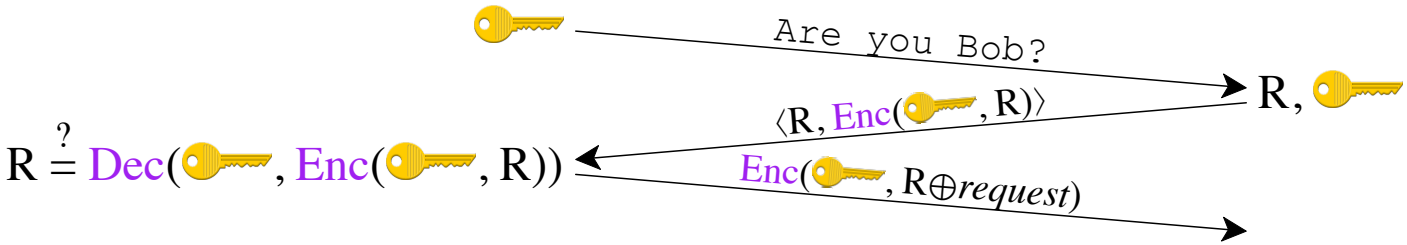
Bob



Authentication

Alice

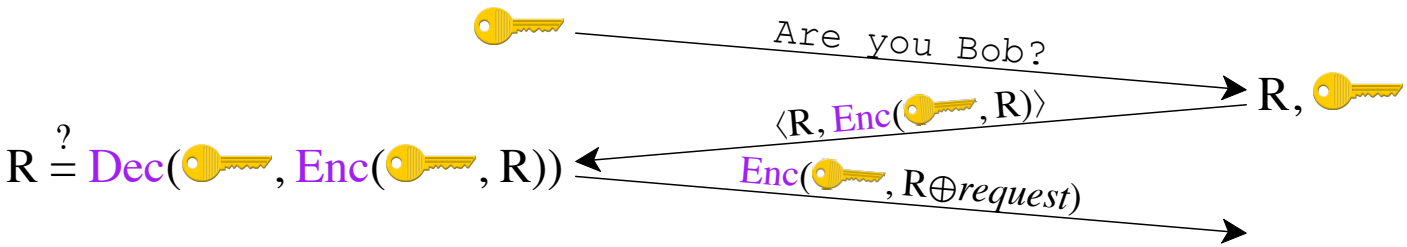
Bob



Authentication

Alice

Bob



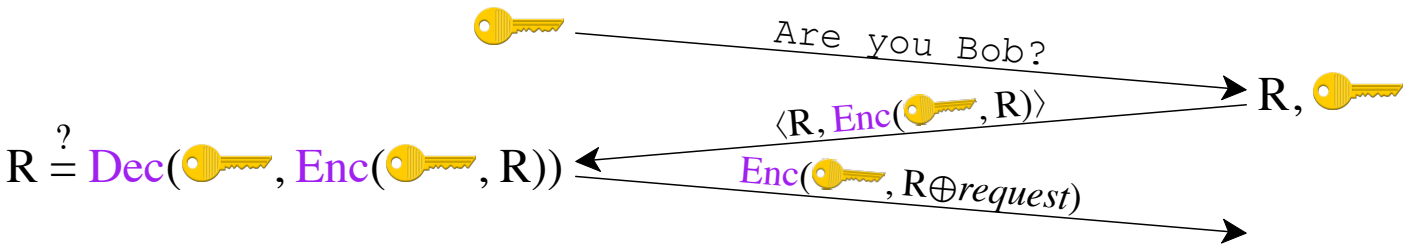
As long as Bob picks a unique random R each time, this will work ok

... but there is still some room for improvement

Authentication

Alice

Bob



Gives attackers lots of examples for

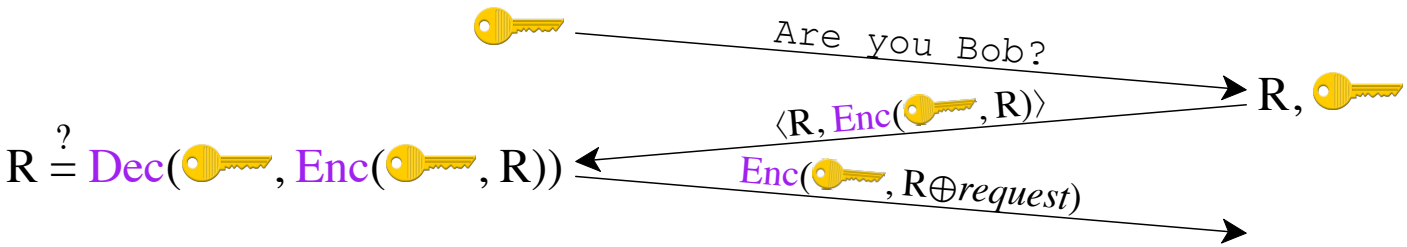
As long as Bob picks a unique random R each time this will work ok

... but there is still some room for improvement

Authentication

Alice

Bob



If is compromised, all history is readable

As long as Bob picks a unique random R each time this will work ok

... but there is still some room for improvement

Distributing Public Keys

Suppose that Alice acquires Bob's public key 

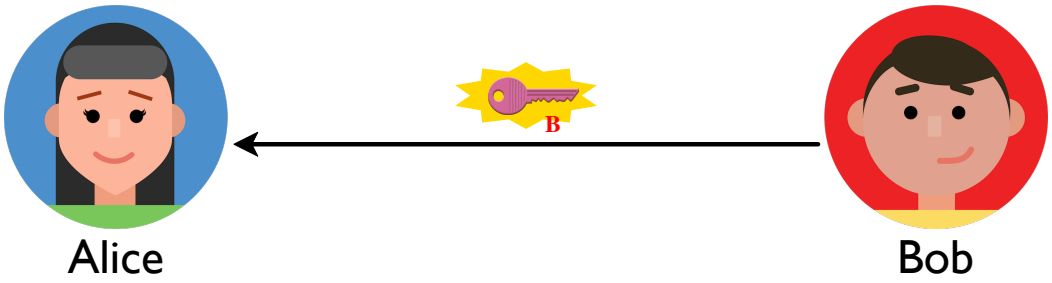
Is it *really* Bob's key?

Alice

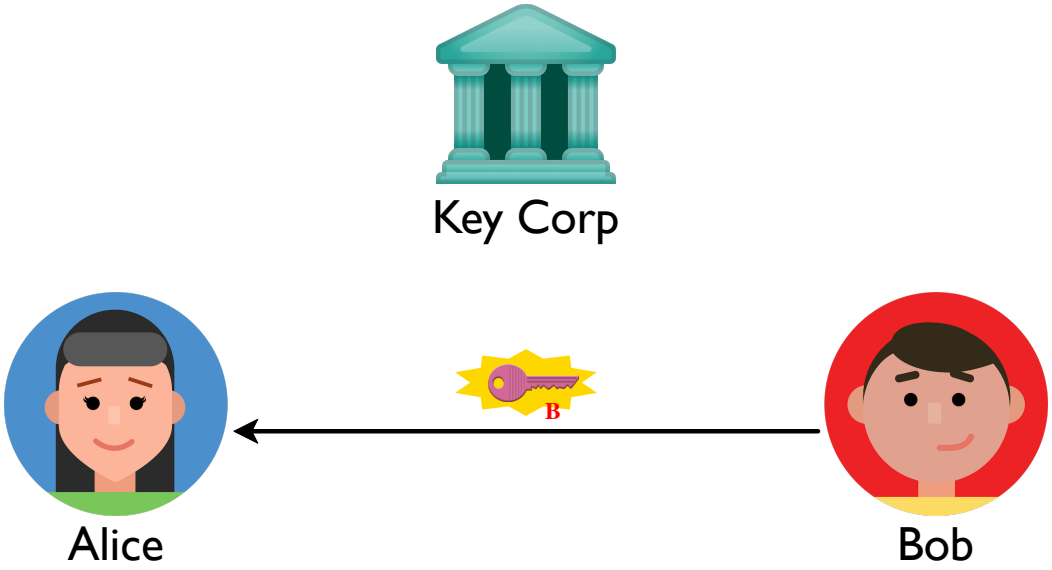
Mallory

Bob

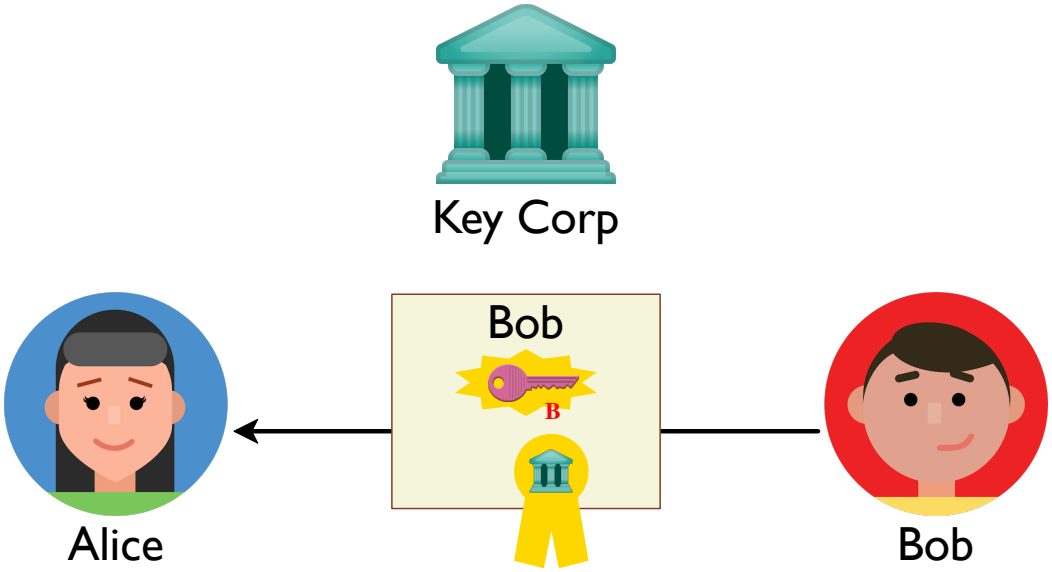
Distributing Public Keys



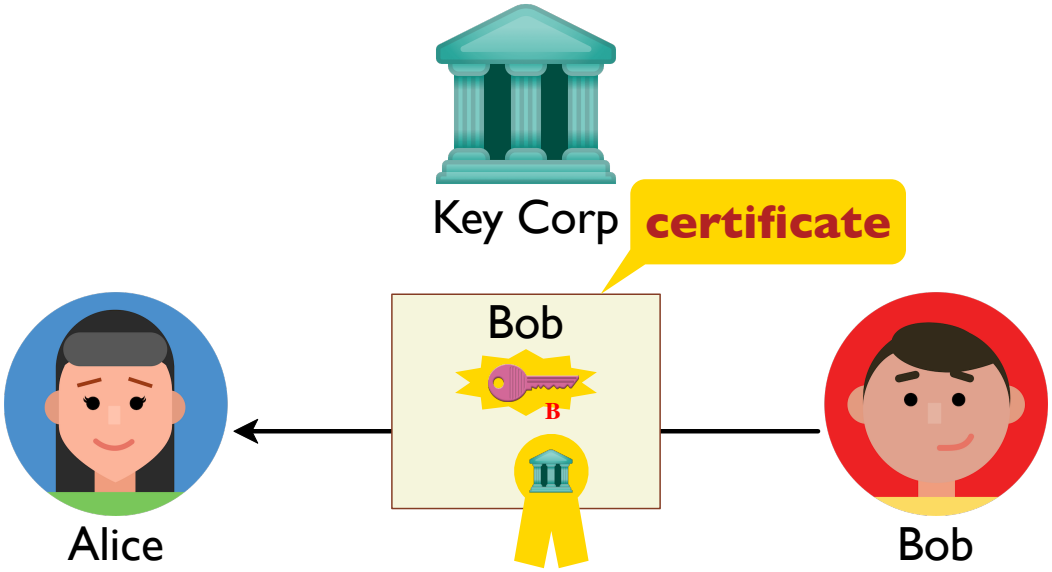
Distributing Public Keys



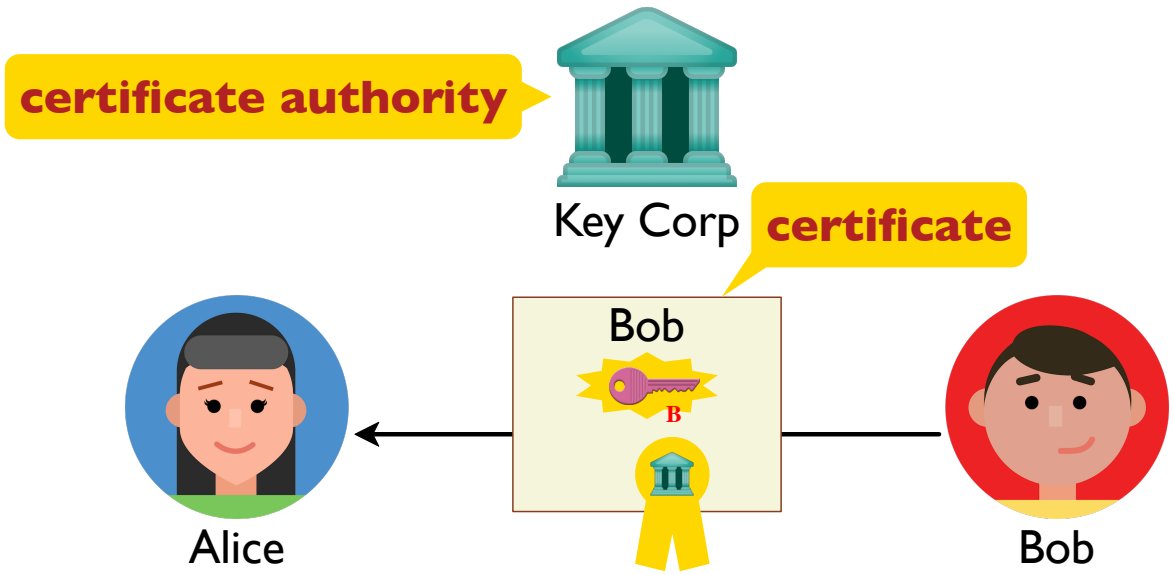
Distributing Public Keys



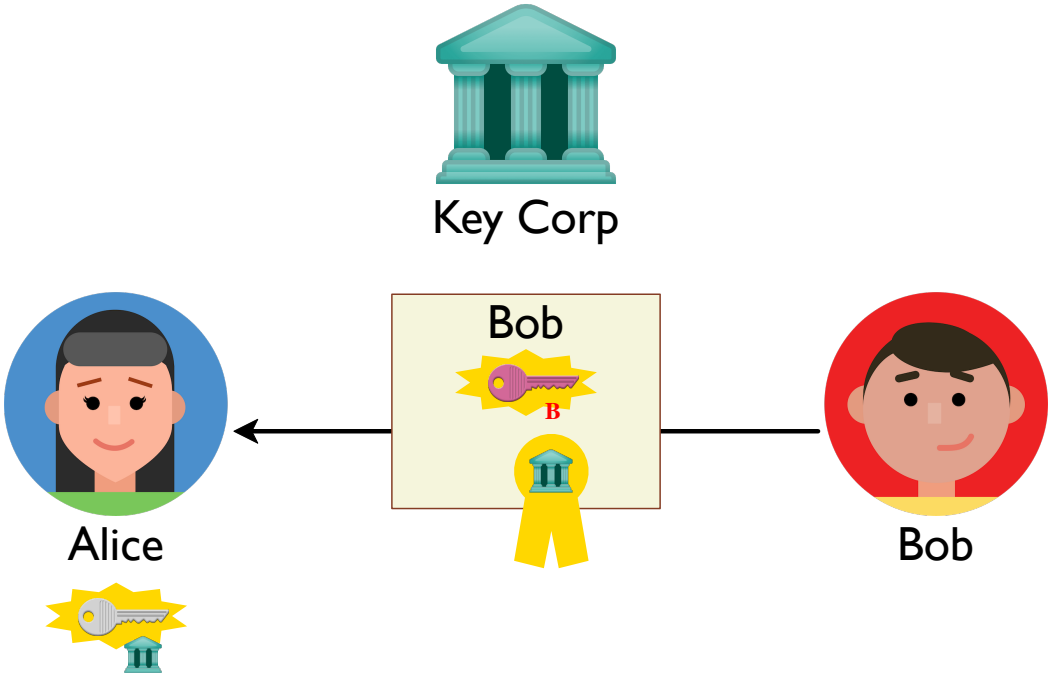
Distributing Public Keys



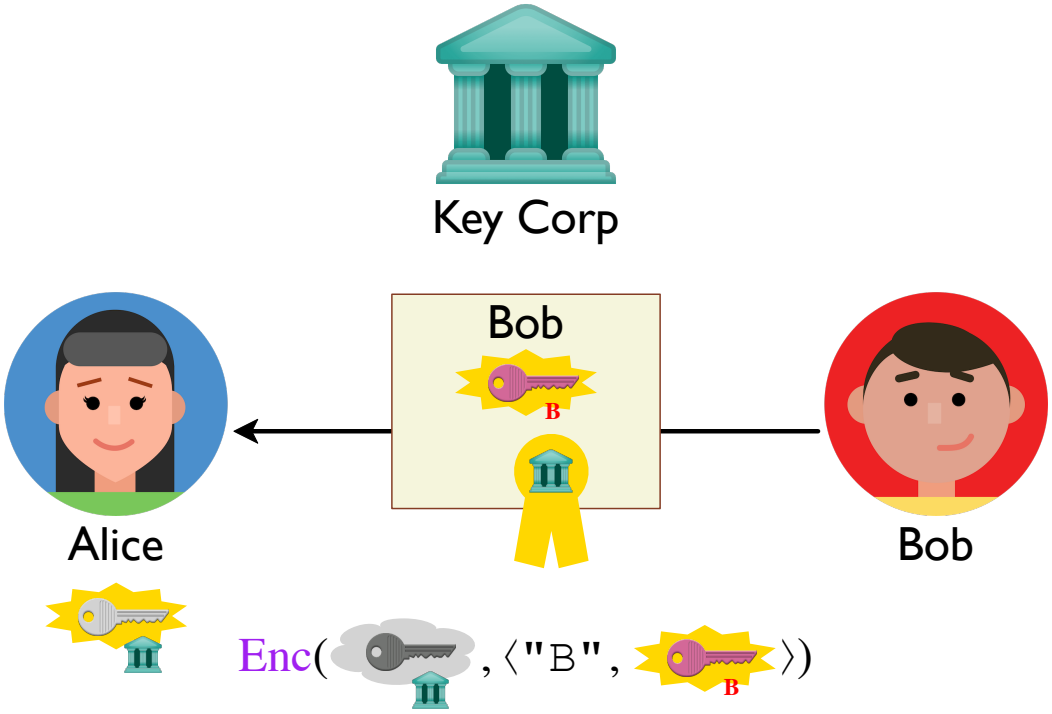
Distributing Public Keys



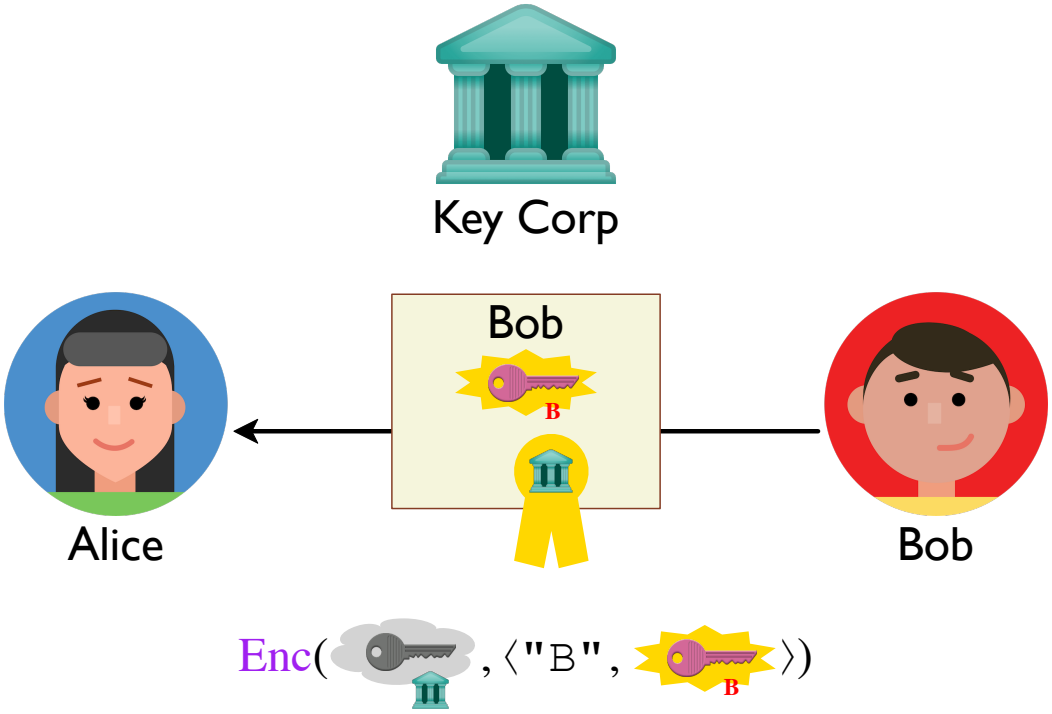
Distributing Public Keys



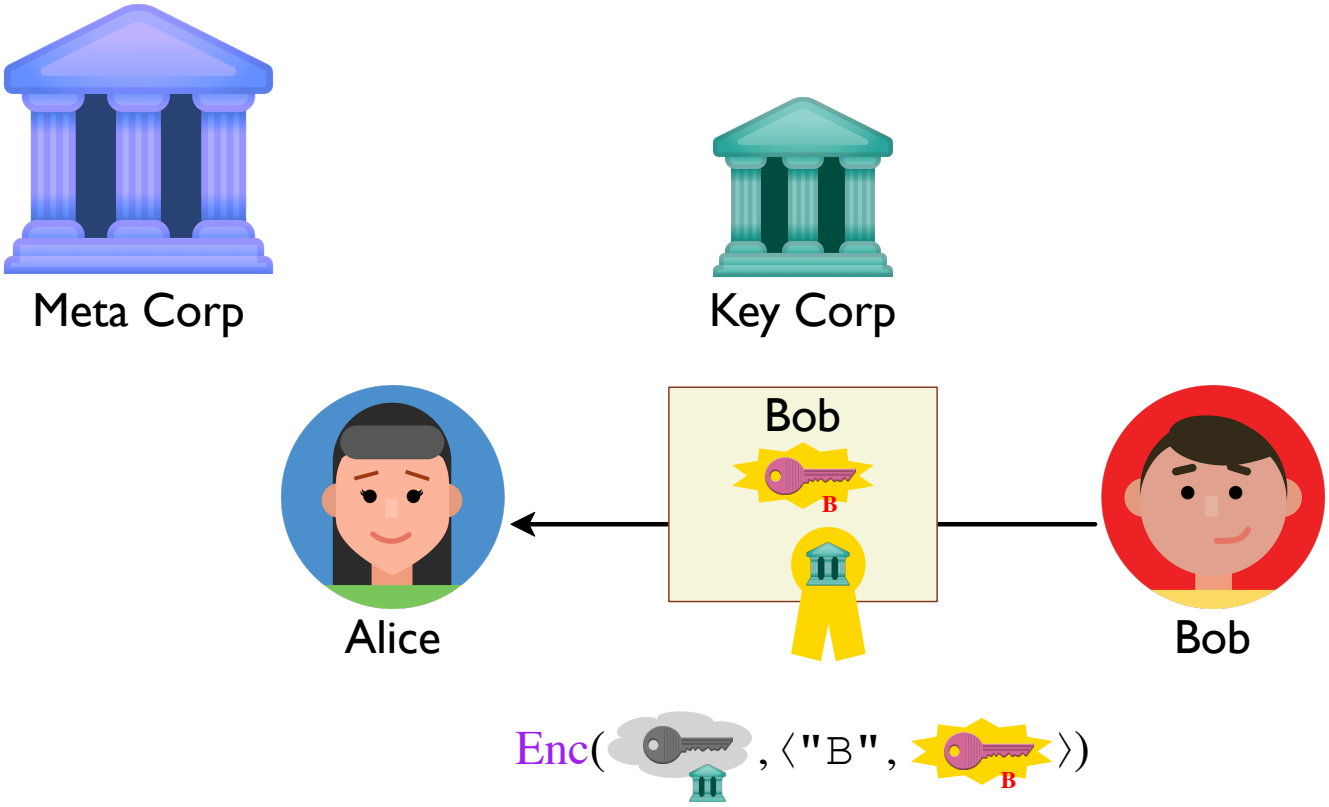
Distributing Public Keys



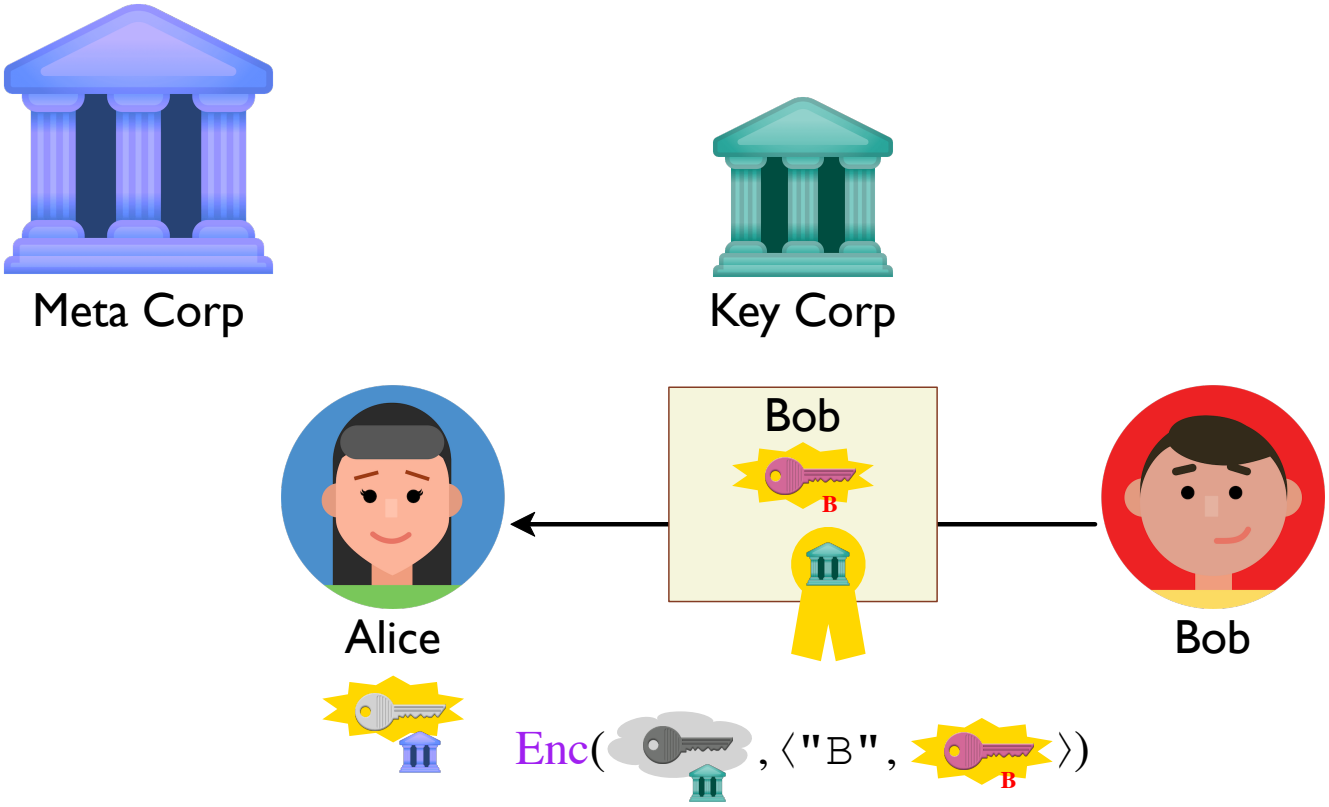
Distributing Public Keys



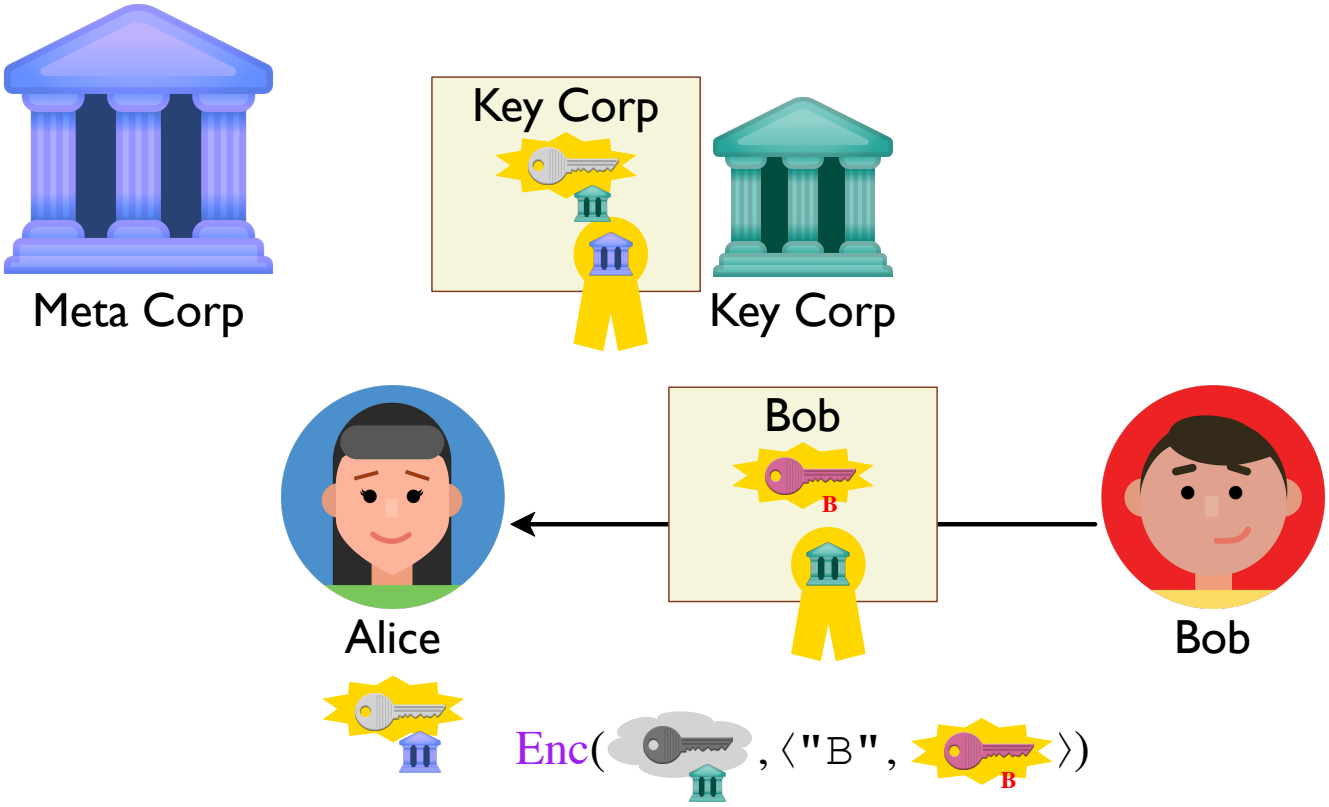
Distributing Public Keys



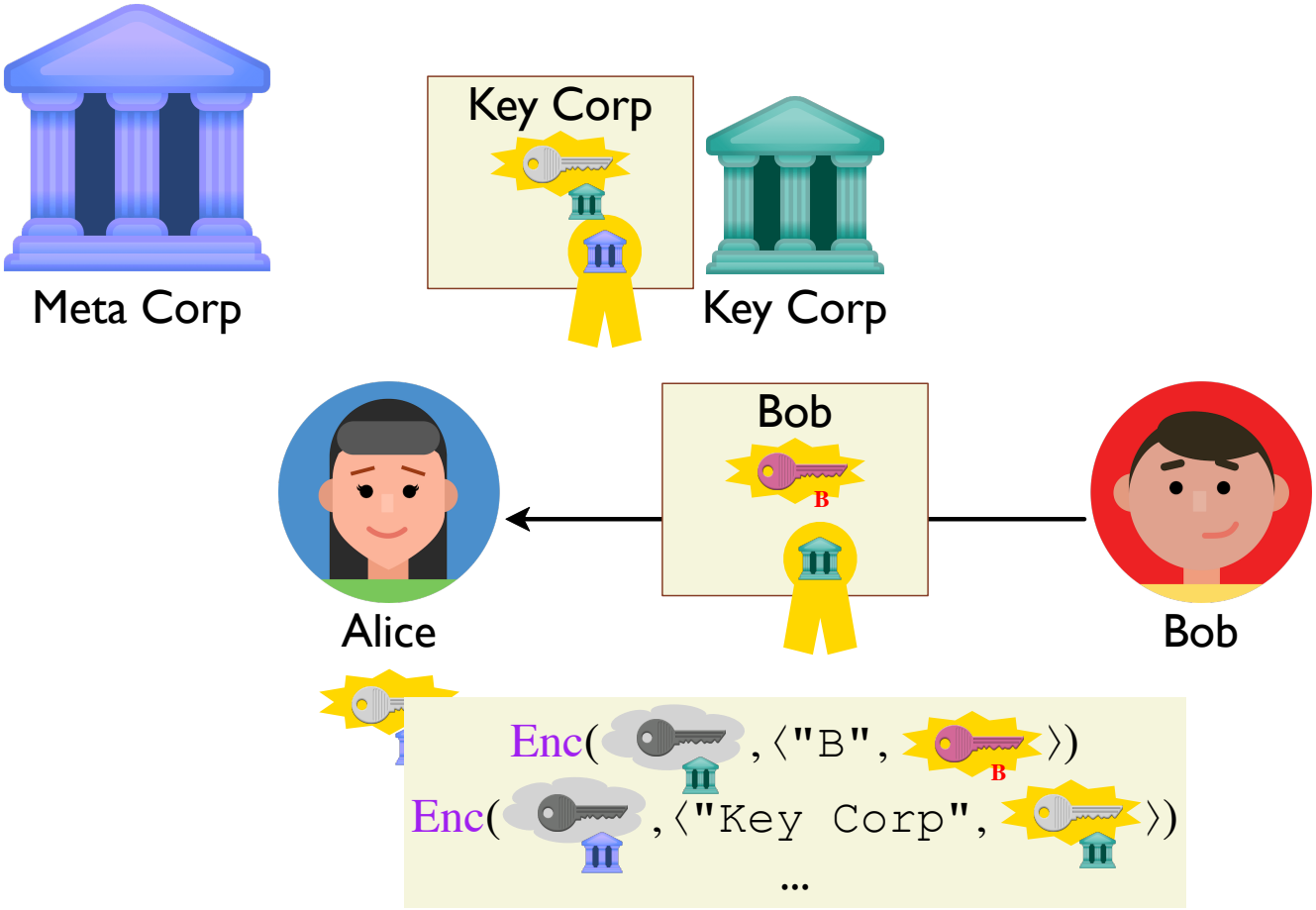
Distributing Public Keys



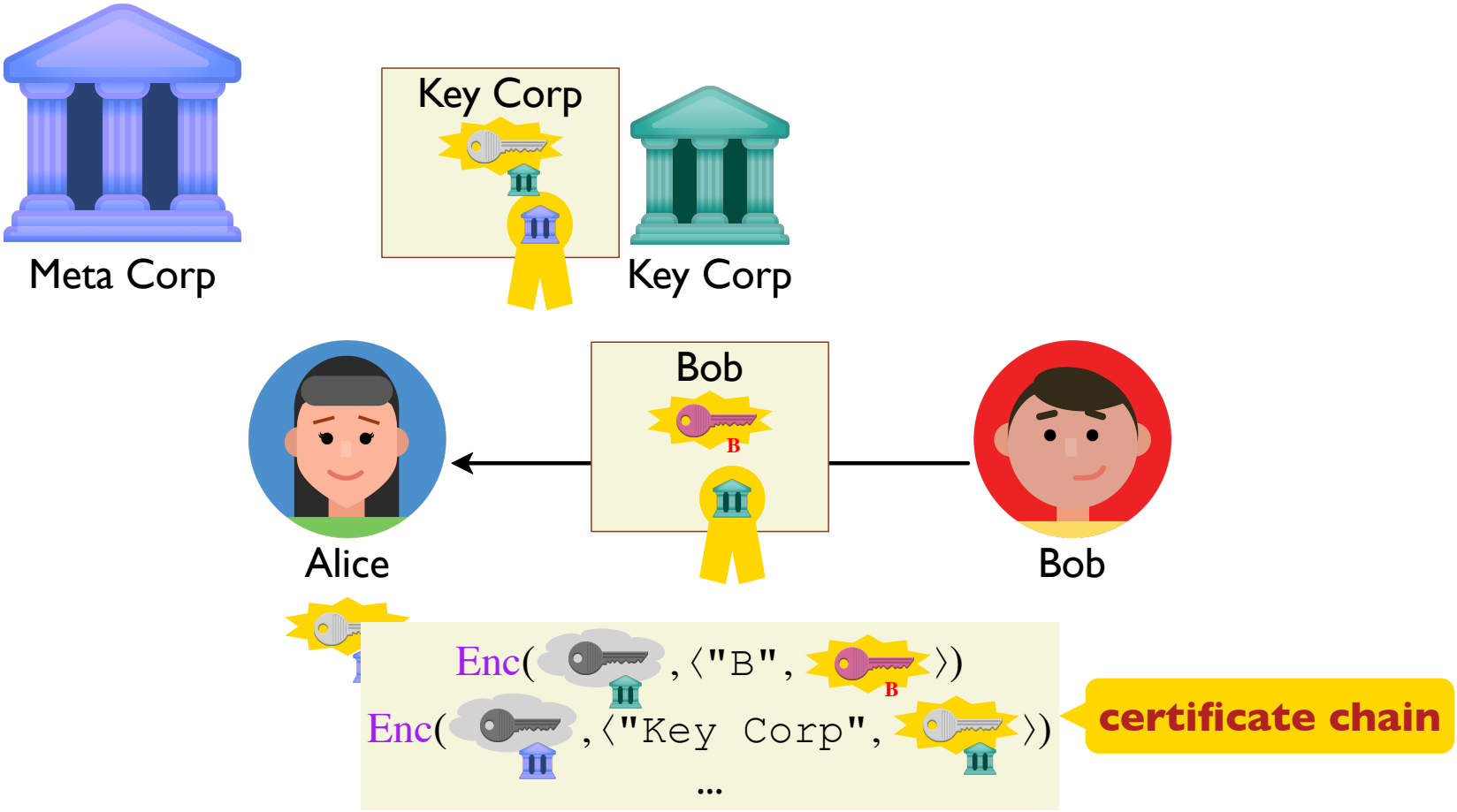
Distributing Public Keys



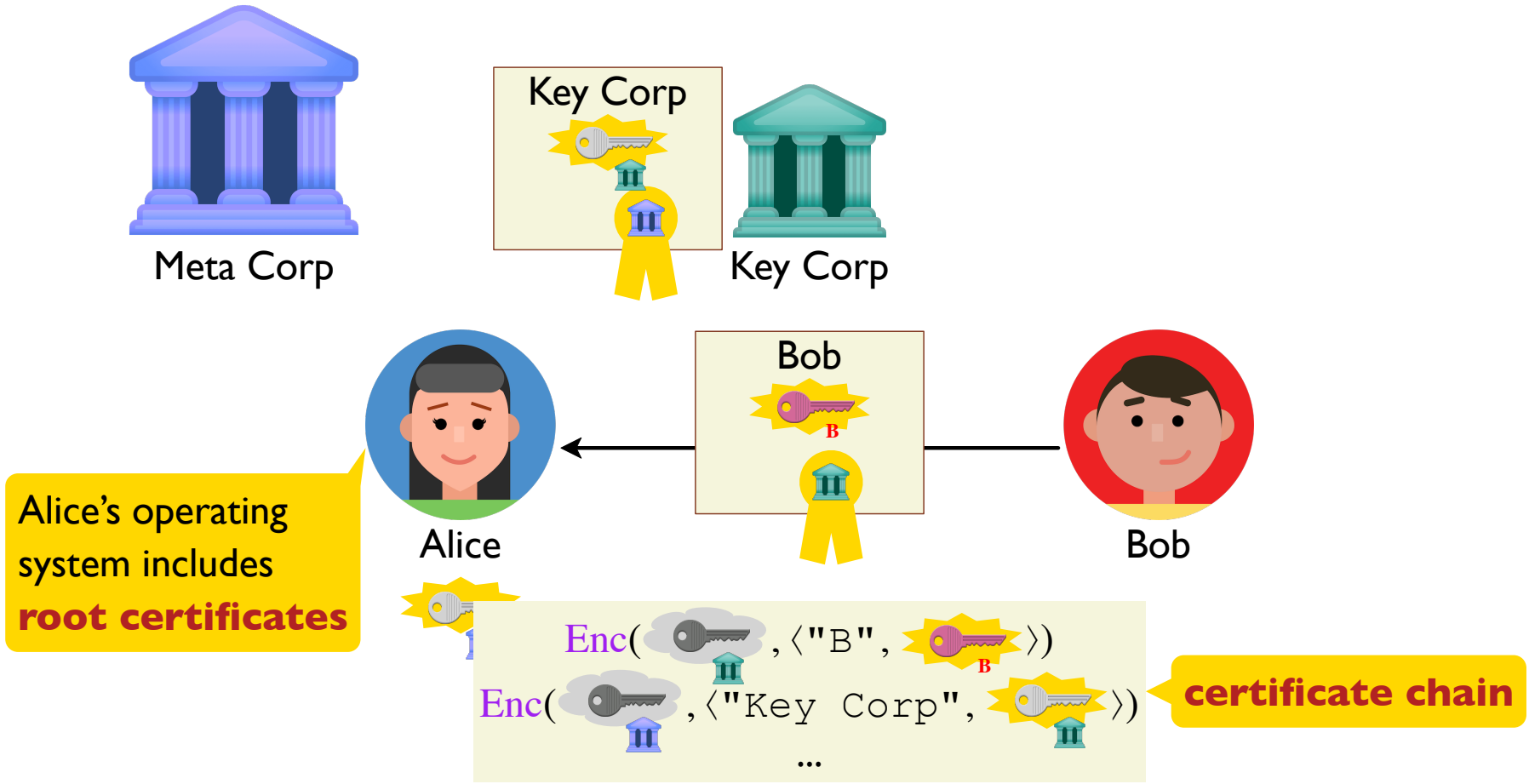
Distributing Public Keys



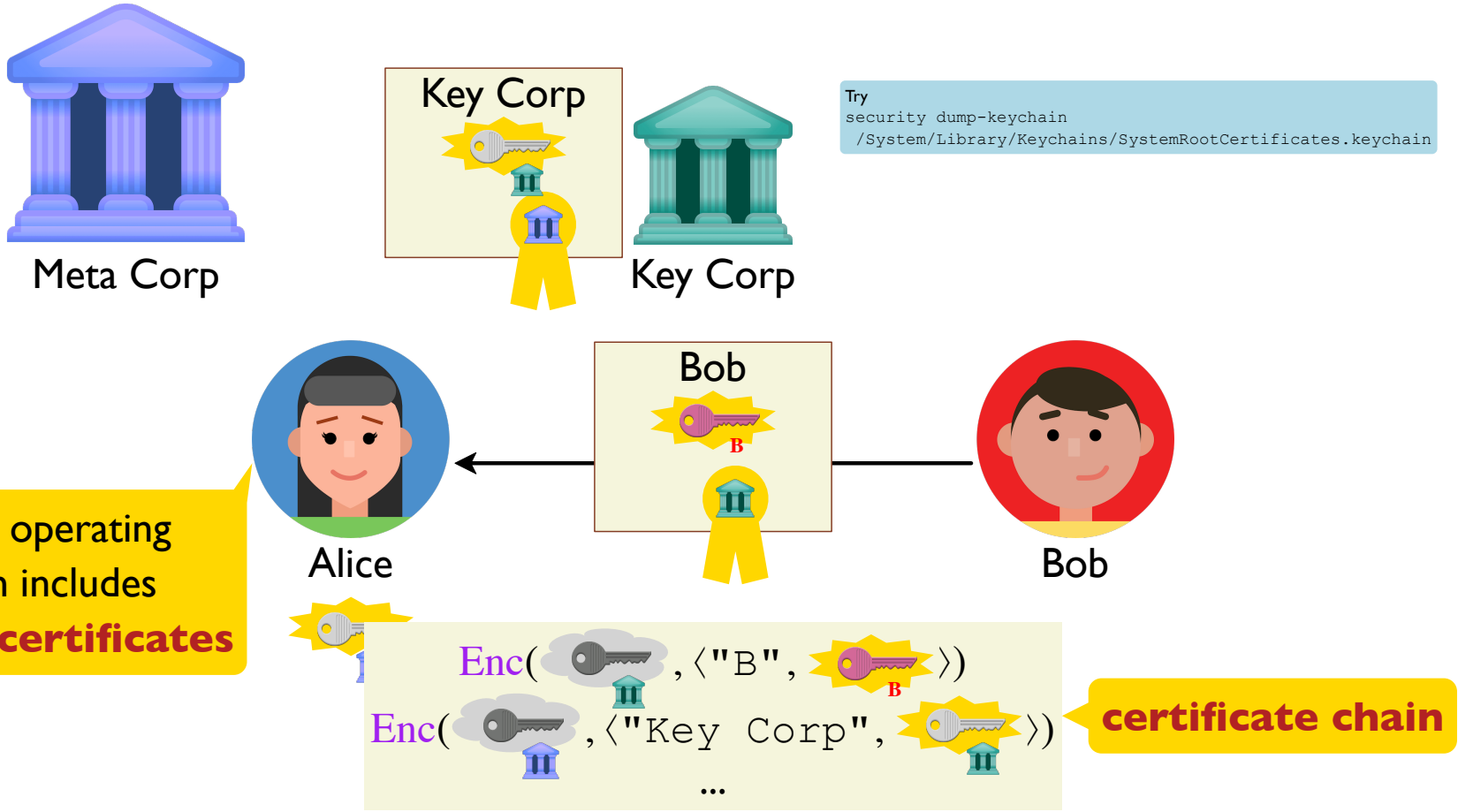
Distributing Public Keys



Distributing Public Keys



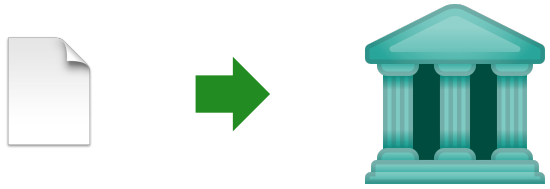
Distributing Public Keys



Getting a Certificate

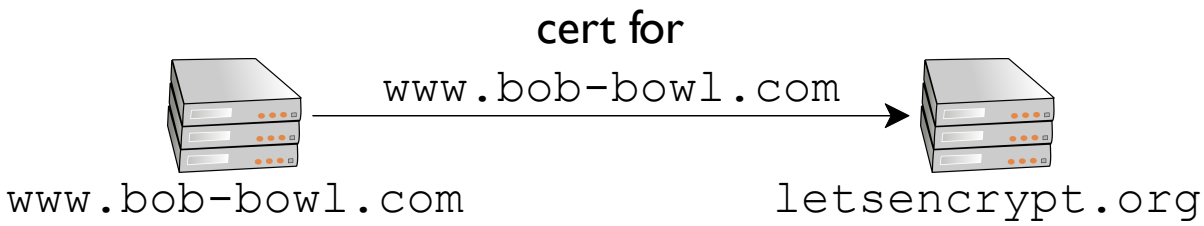
How does a service provider obtain a certificate?

- Some kinds of certificates: **paperwork**

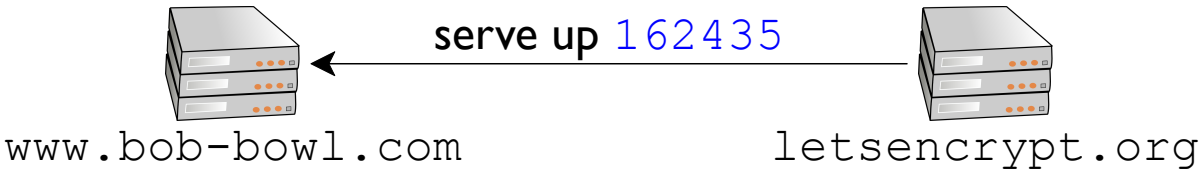


- Web sites: **Let's Encrypt**

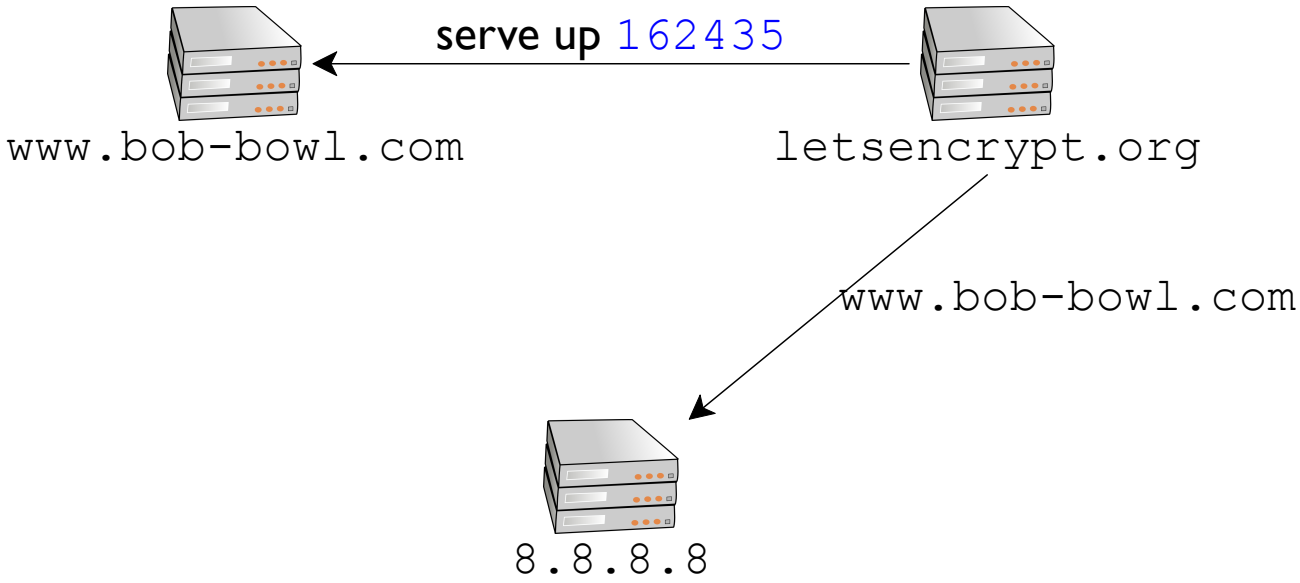
Getting a Certificate via Let's Encrypt



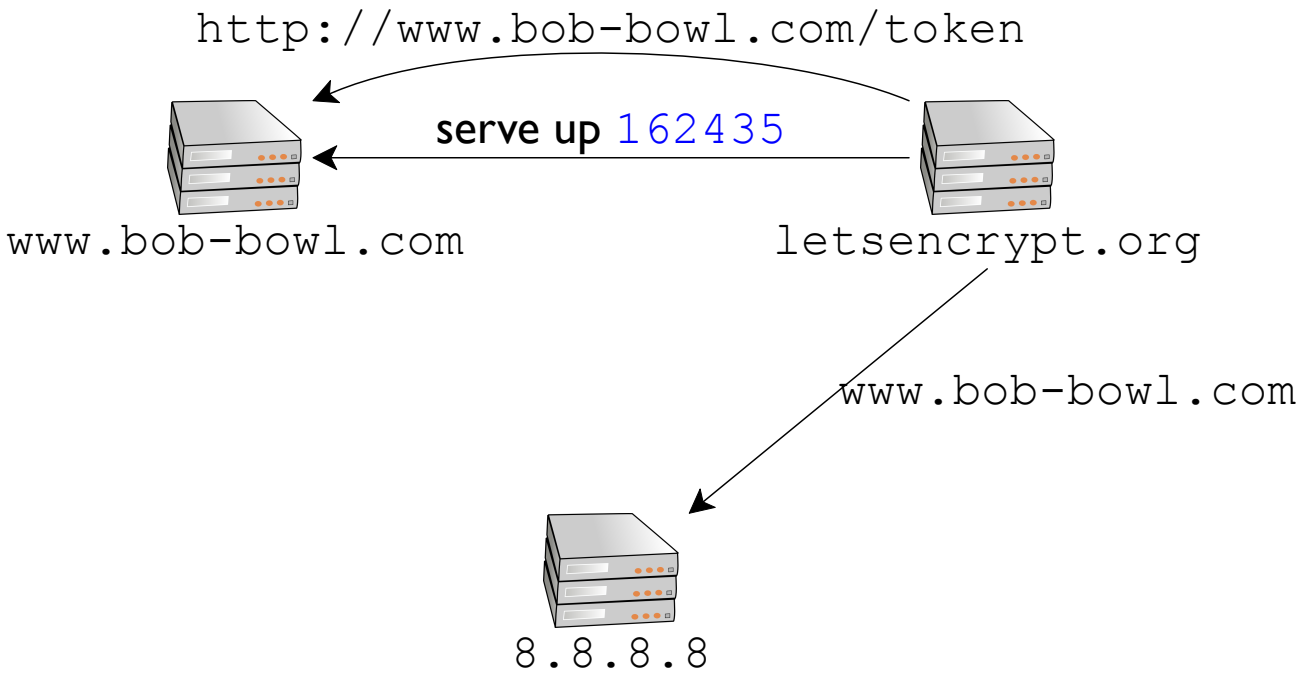
Getting a Certificate via Let's Encrypt



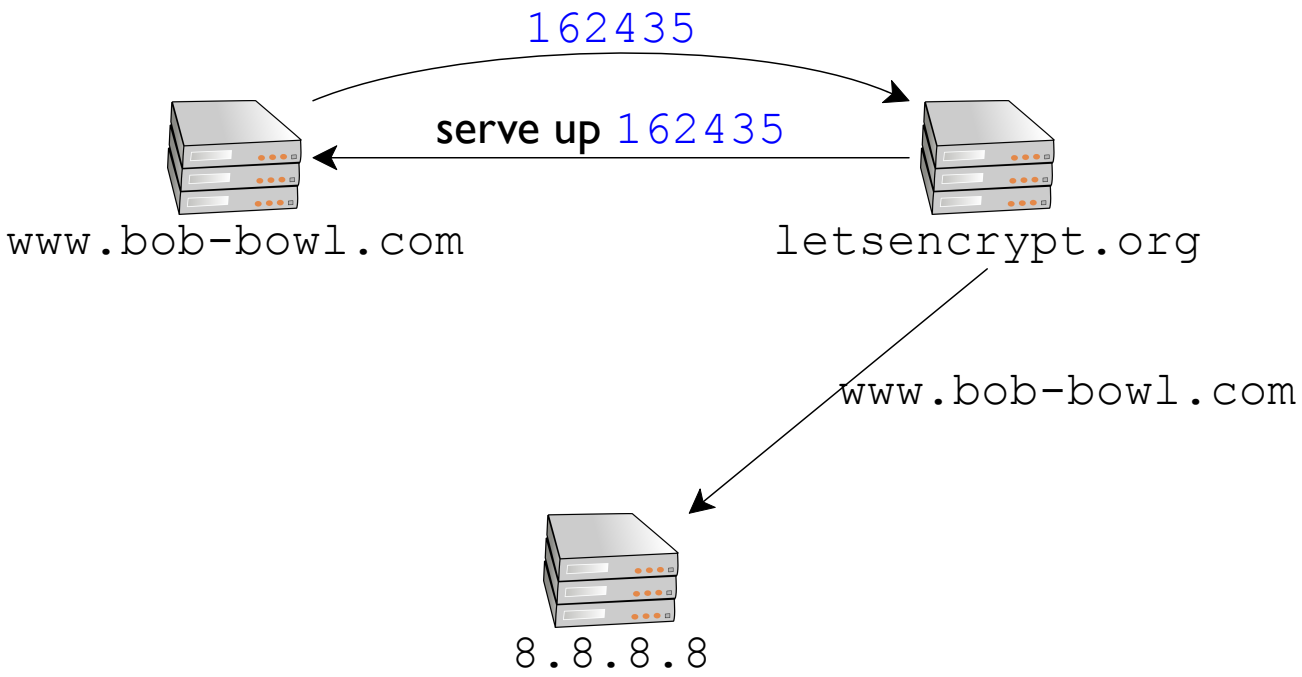
Getting a Certificate via Let's Encrypt



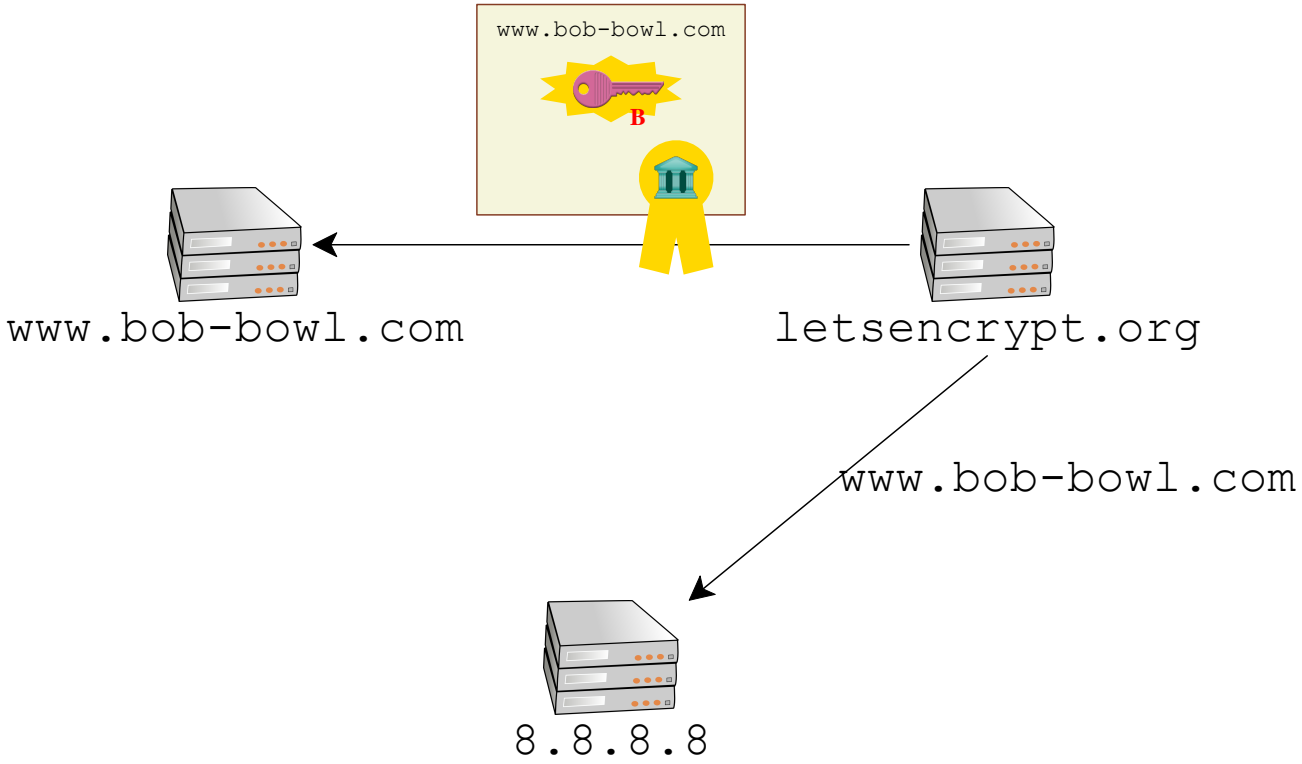
Getting a Certificate via Let's Encrypt



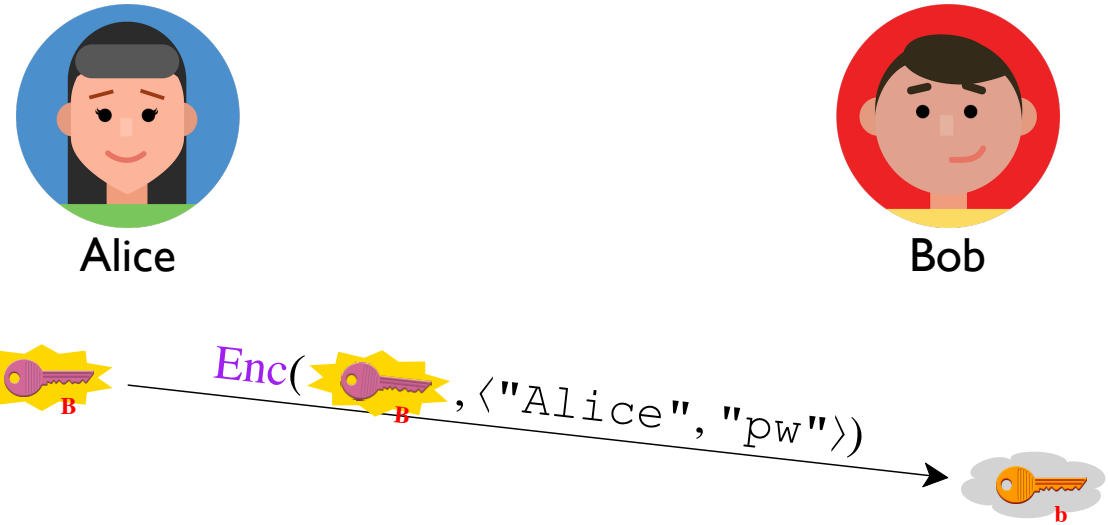
Getting a Certificate via Let's Encrypt



Getting a Certificate via Let's Encrypt



More Bad Authentication Ideas

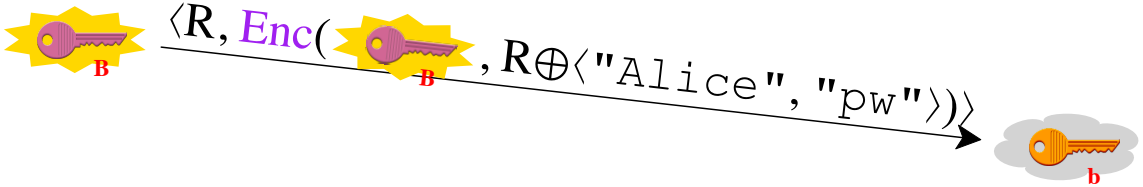


This is the replay-attack example, again

More Bad Authentication Ideas

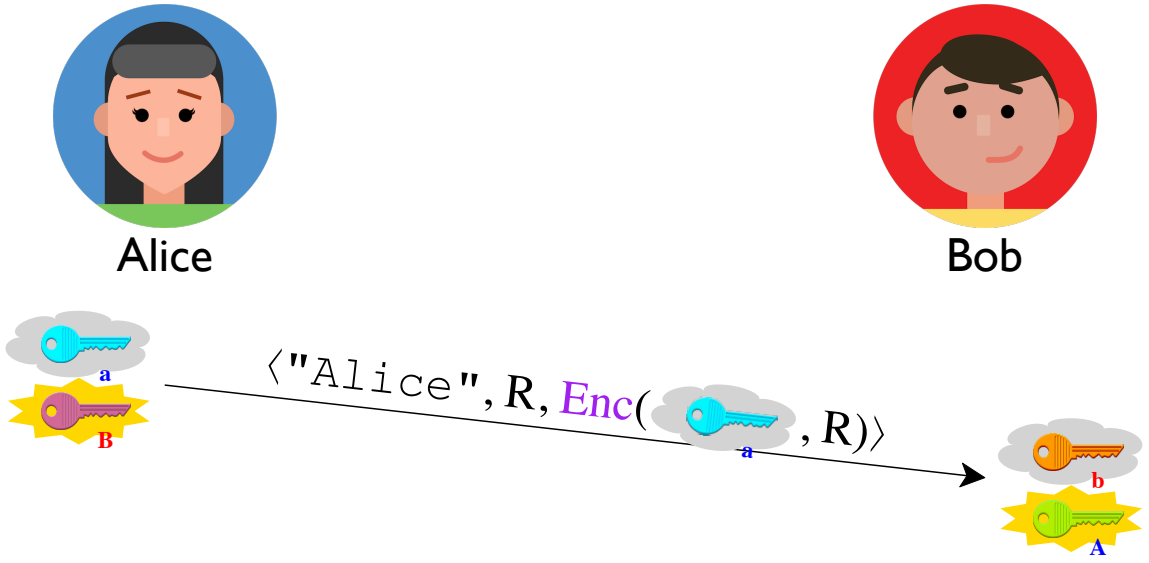
Alice

Bob



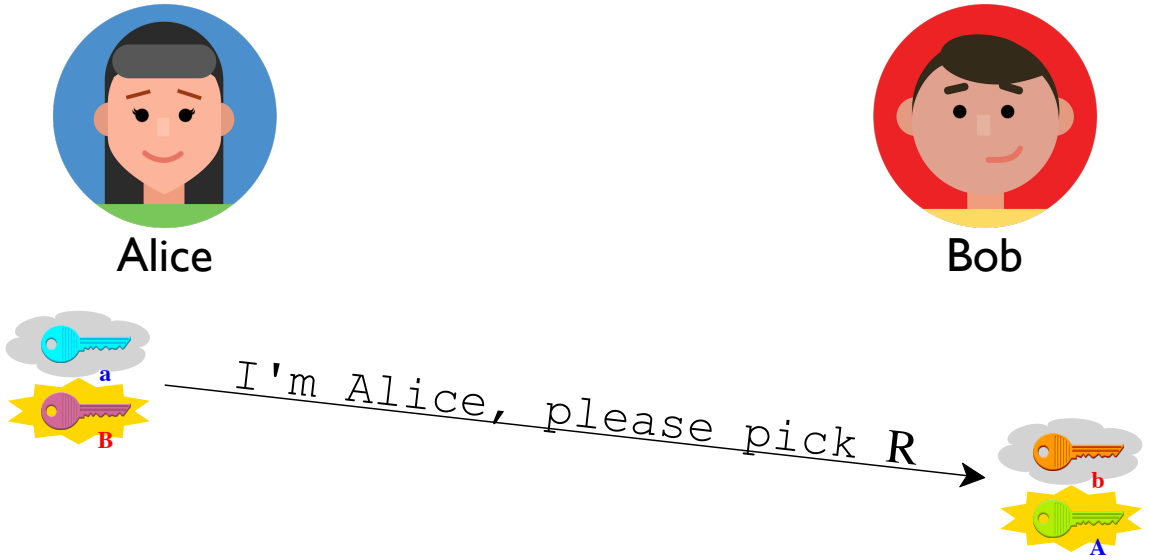
Still subject to a replay attack if Alice picks R

More Bad Authentication Ideas

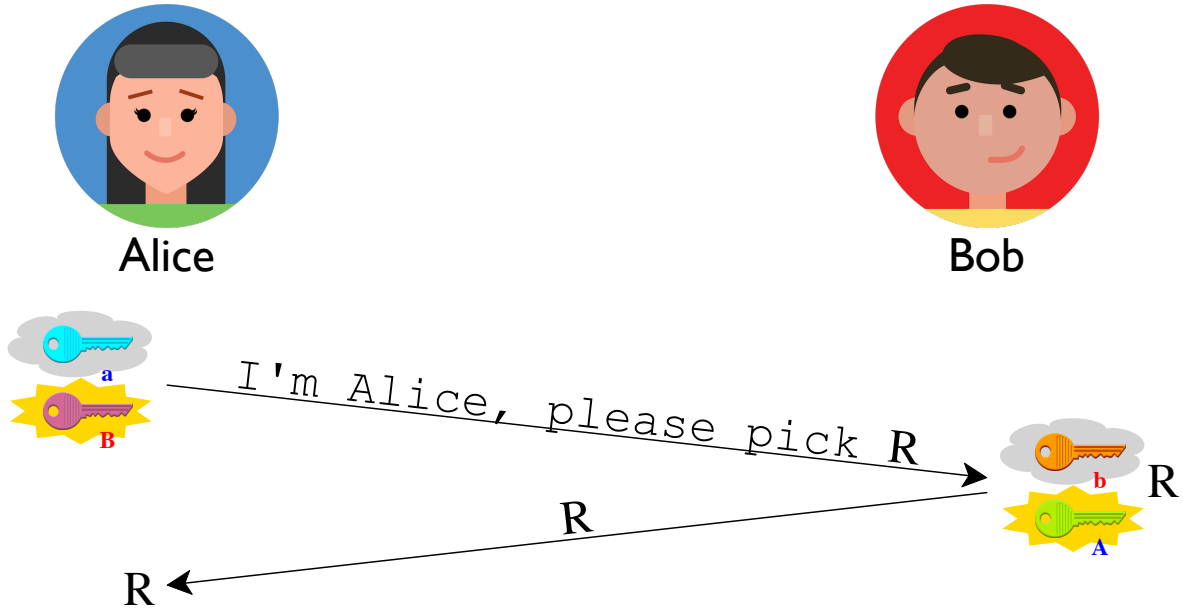


Still subject to a replay attack if Alice picks R

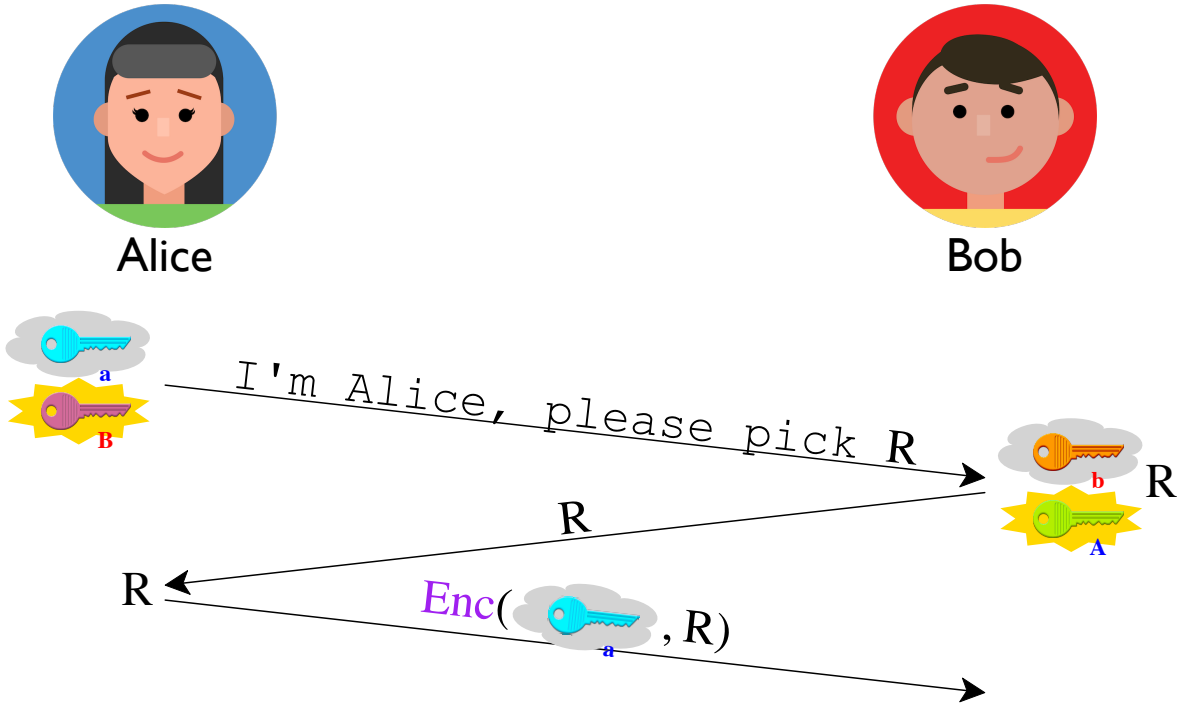
More Bad Authentication Ideas



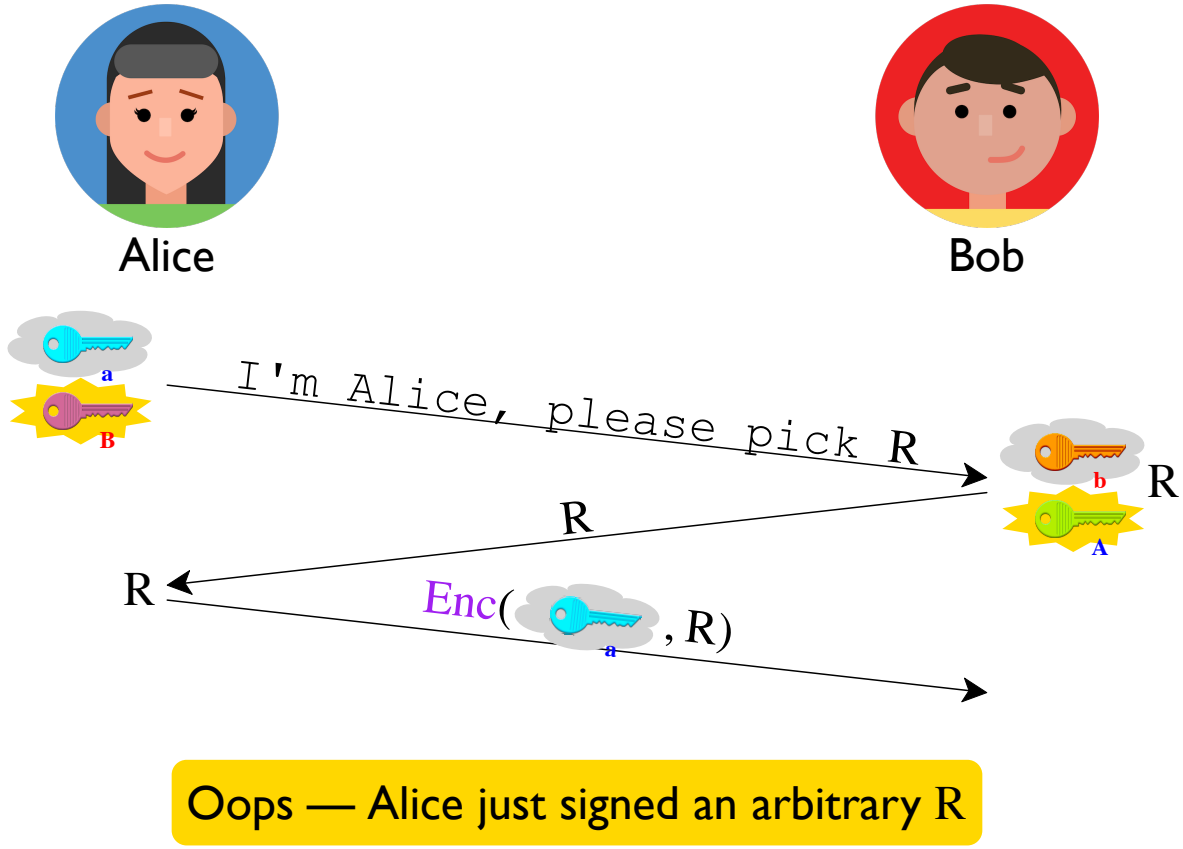
More Bad Authentication Ideas



More Bad Authentication Ideas

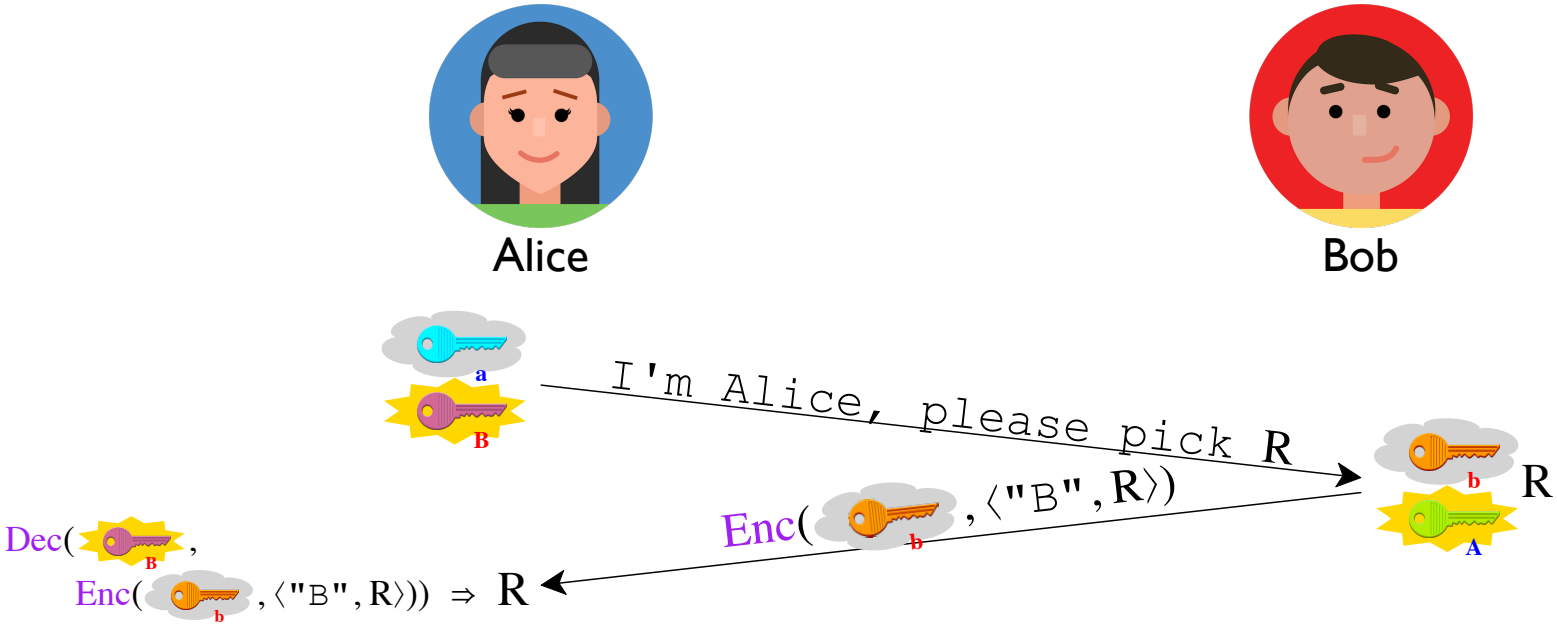


More Bad Authentication Ideas

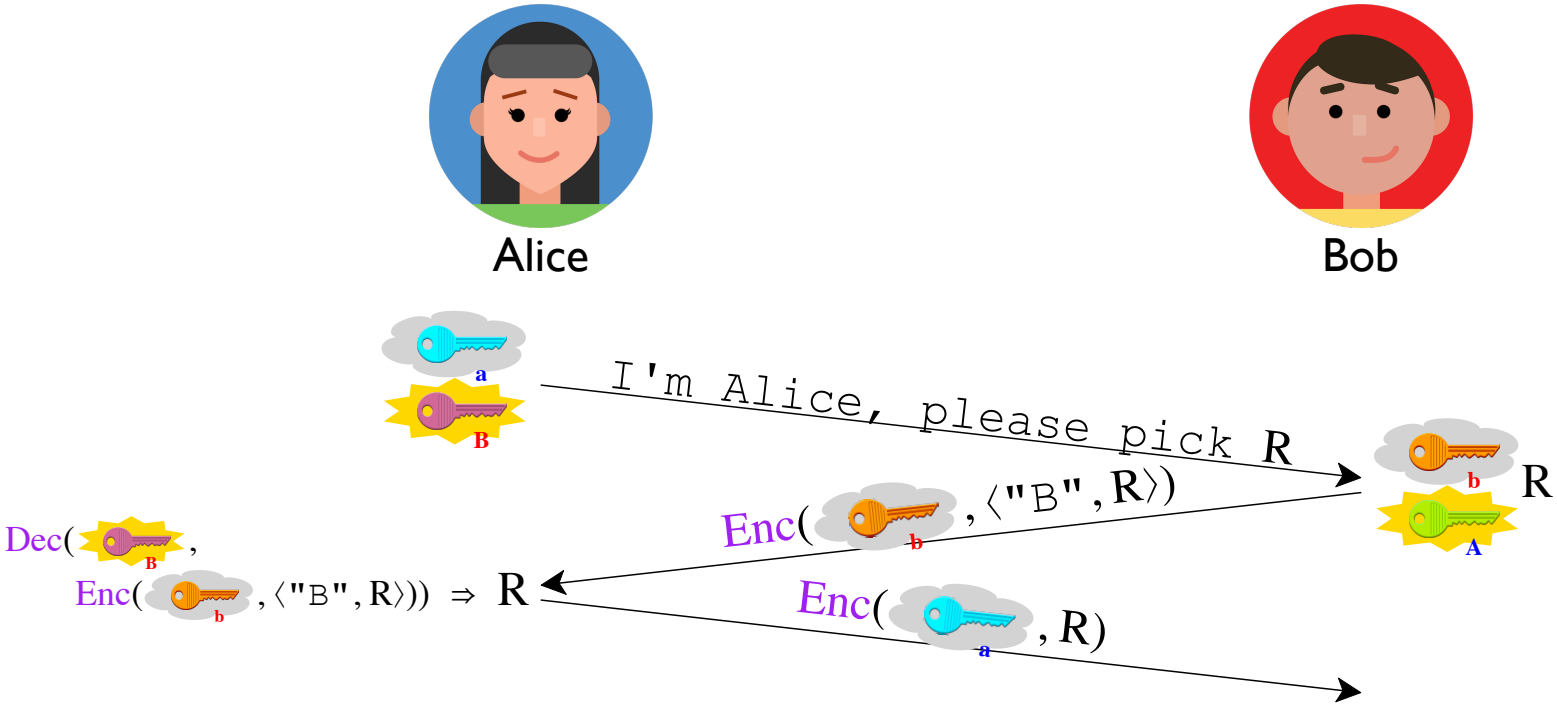


Oops — Alice just signed an arbitrary R

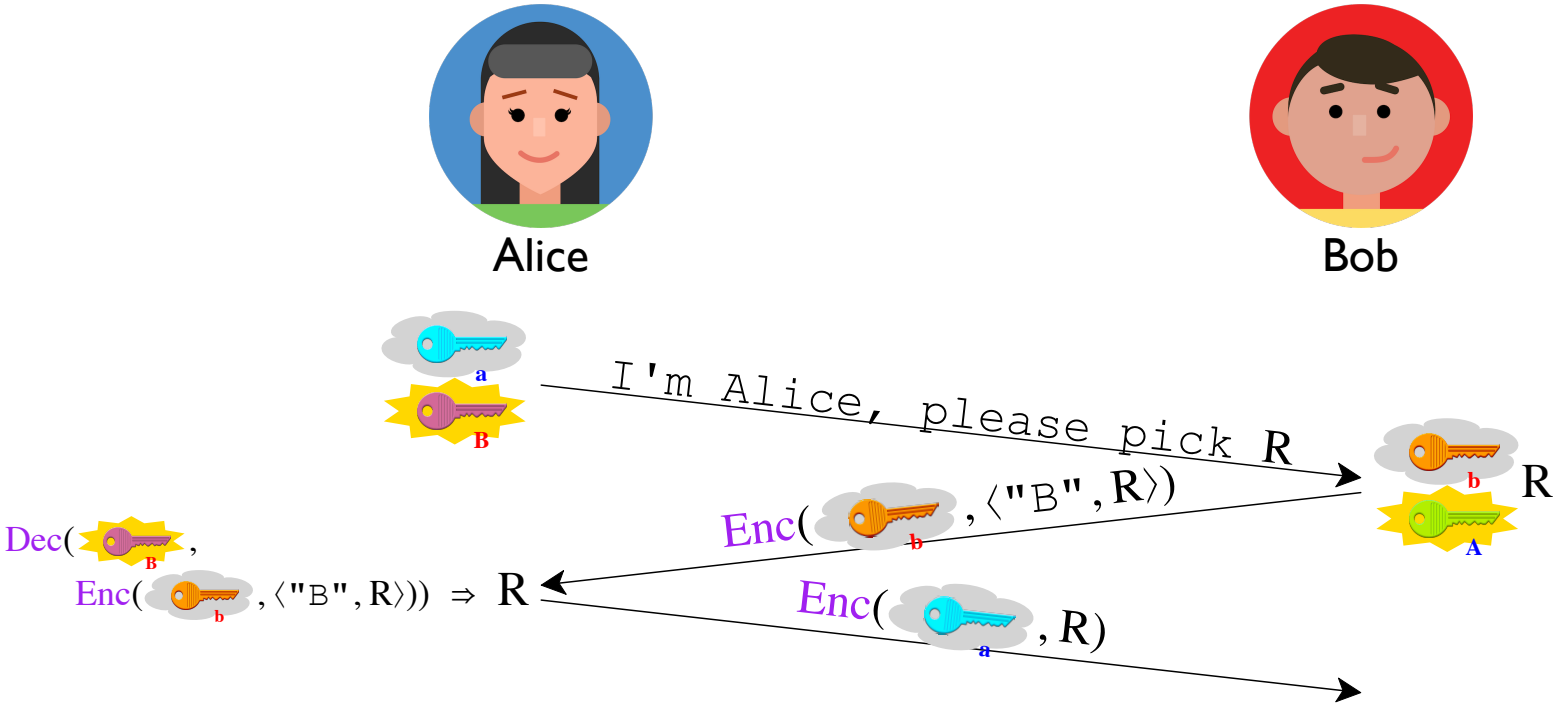
Authentication



Authentication

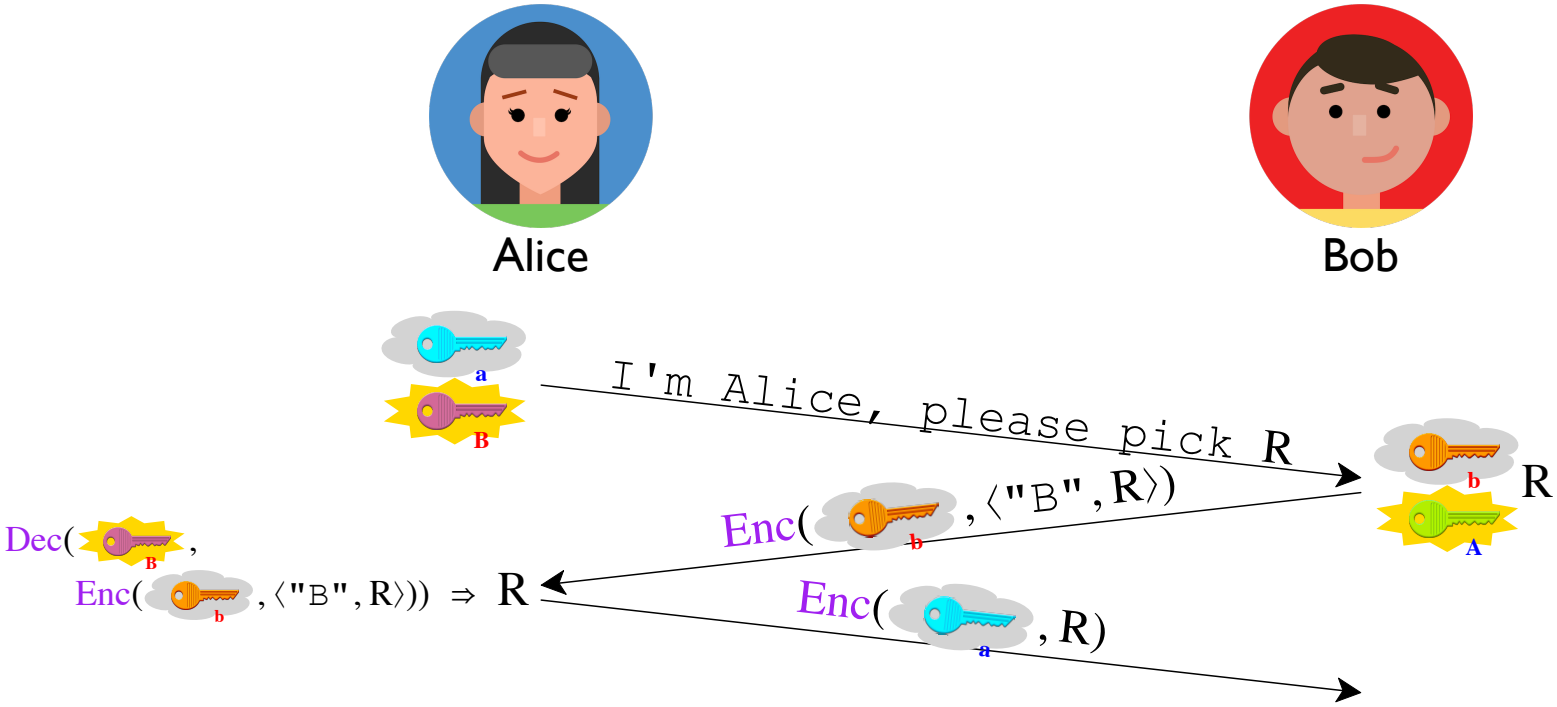


Authentication



Alice is signing R from Bob, but trusts Bob enough

Authentication



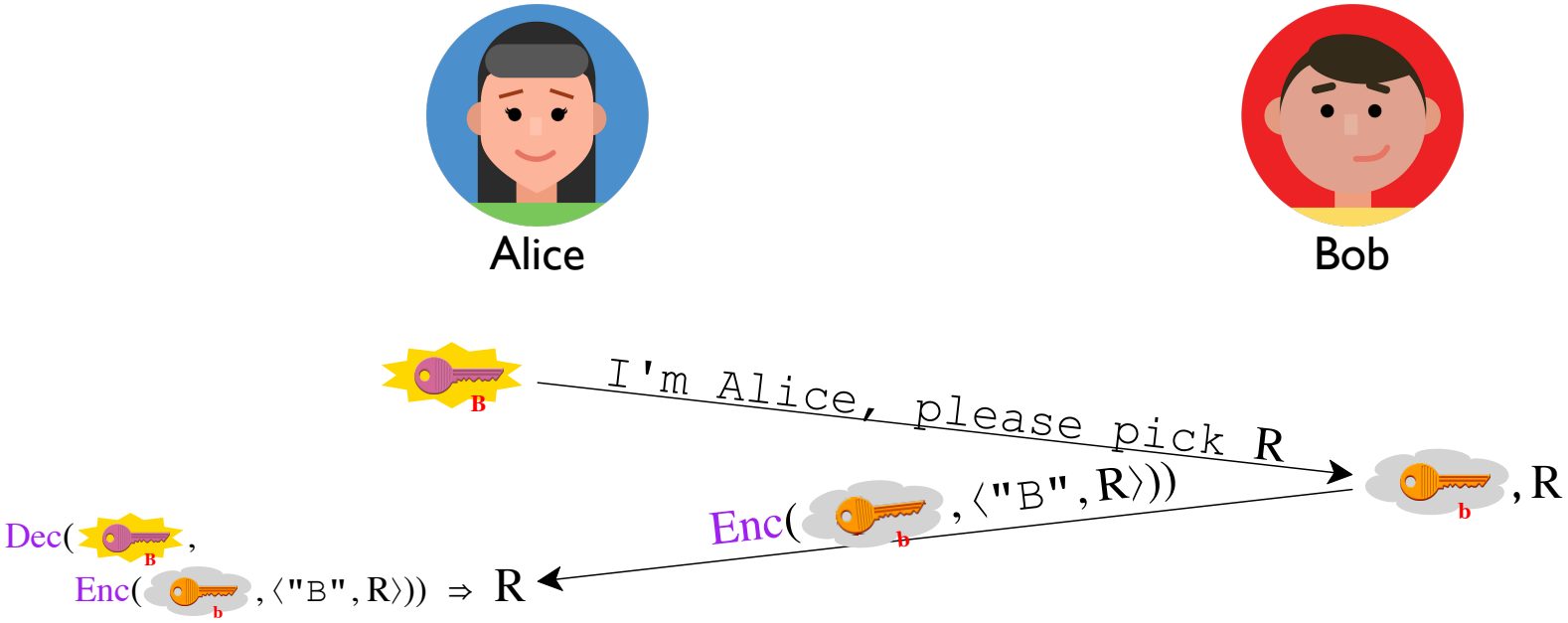
SSH often works something like this, but not web sites

Authentication

Alice

Bob

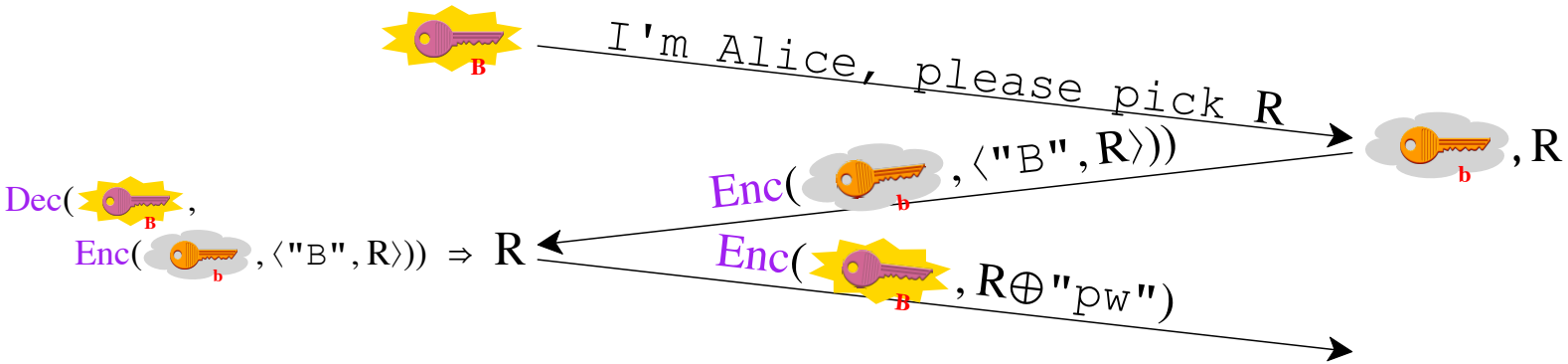
Authentication



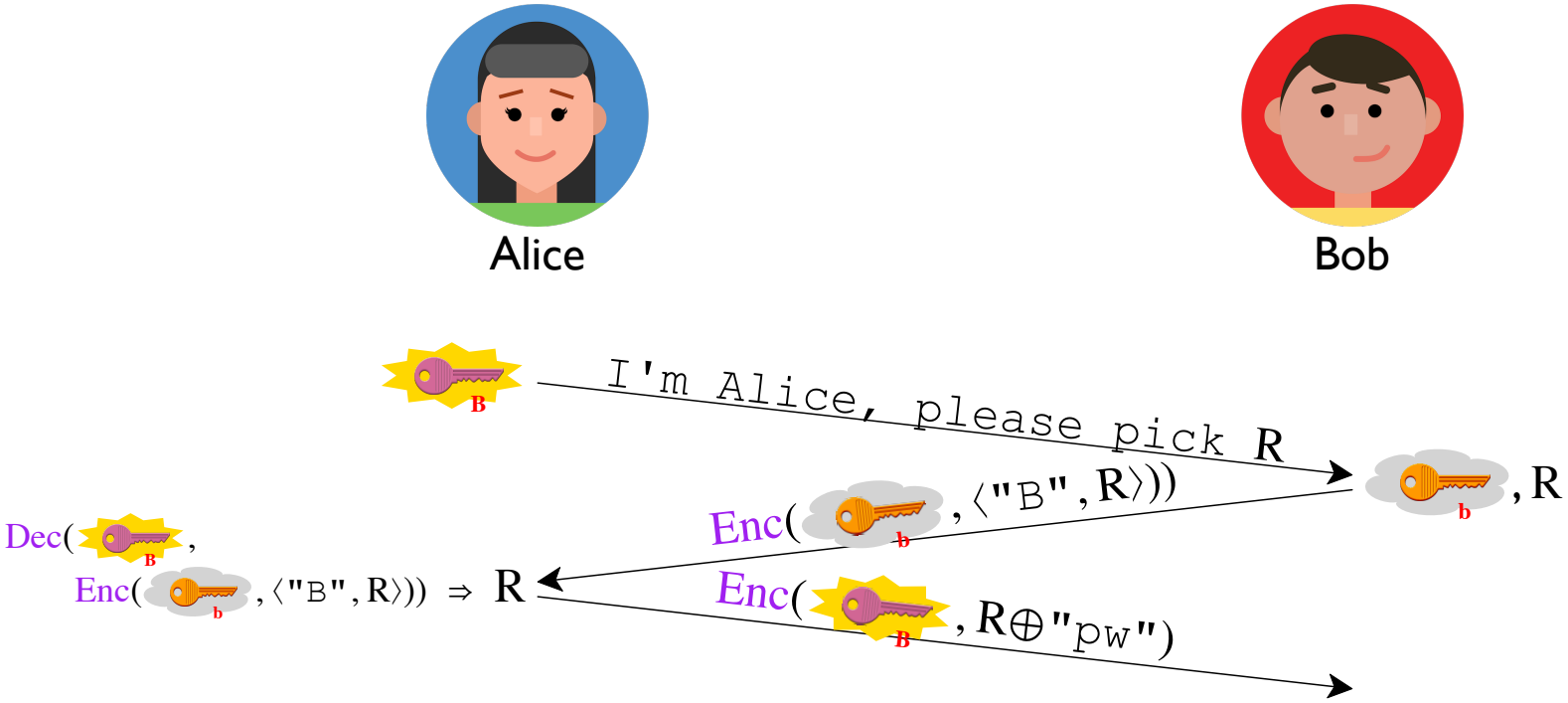
Authentication

Alice

Bob



Authentication



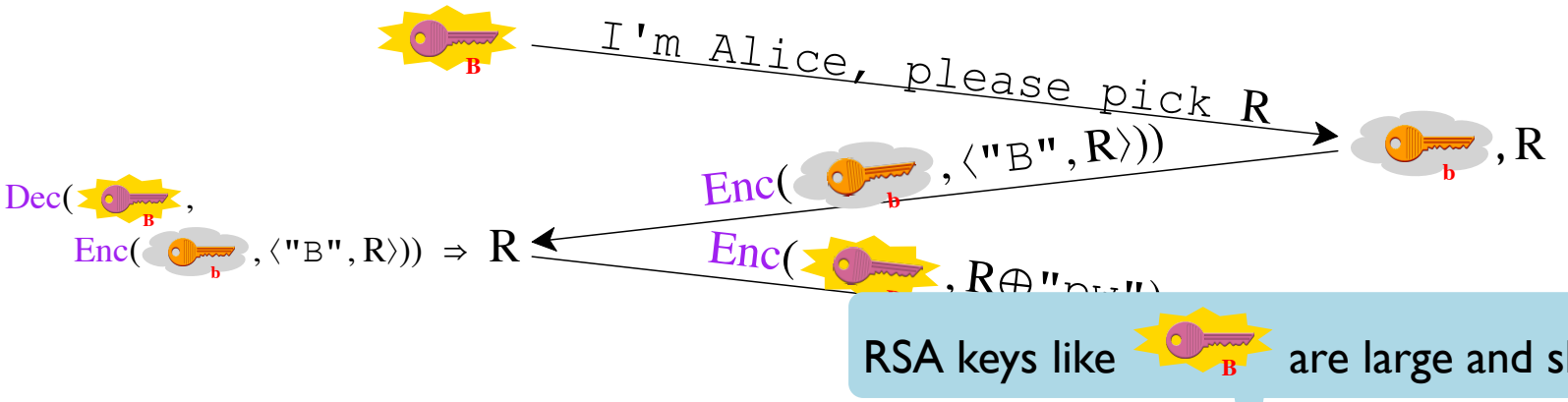
As long as Bob picks a unique random R each time, this will work ok

... but there is still some room for improvement

Authentication

Alice

Bob



RSA keys like are large and slow

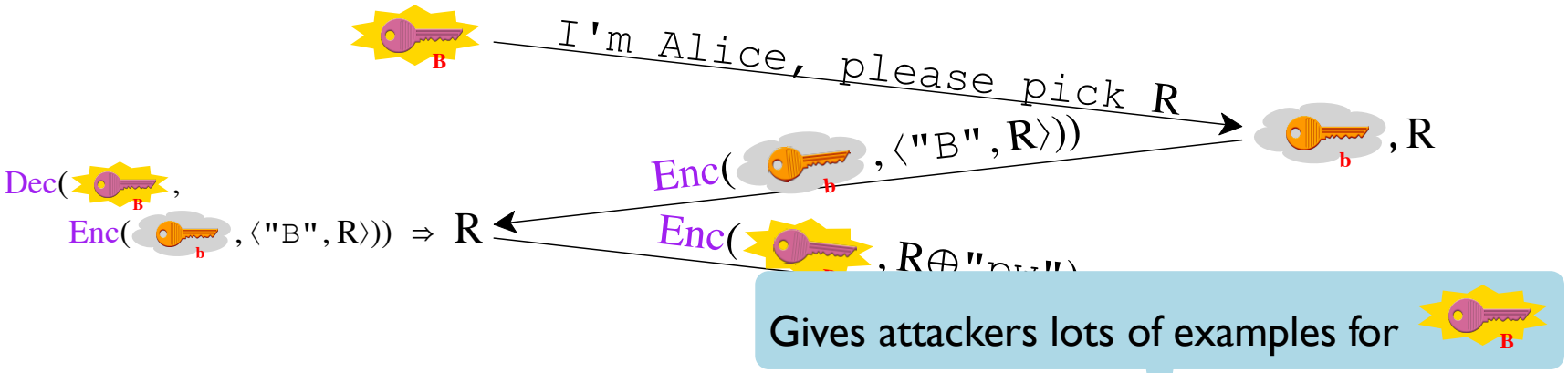
As long as Bob picks a unique random R each time this will work ok

... but there is still some room for improvement

Authentication

Alice

Bob



Gives attackers lots of examples for

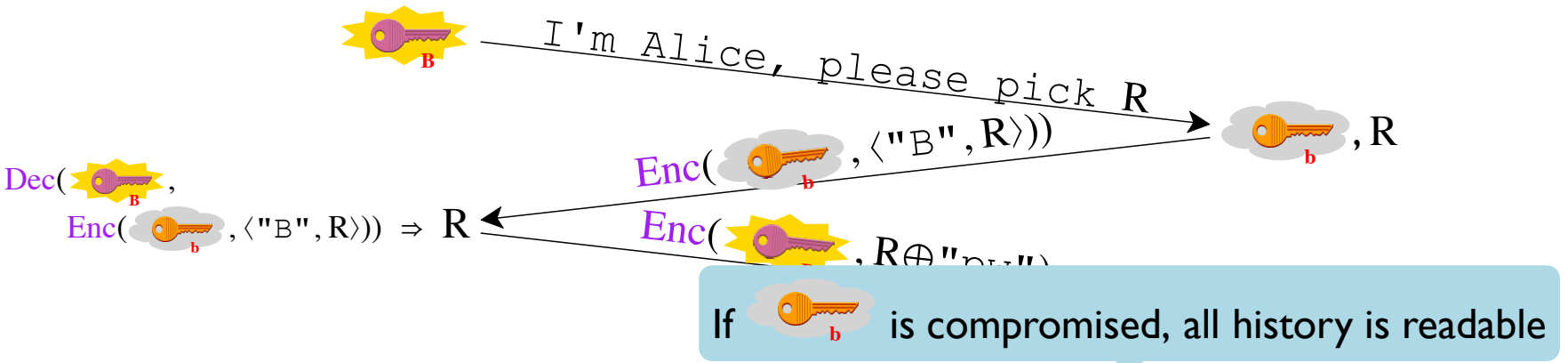
As long as Bob picks a unique random R each time this will work ok

... but there is still some room for improvement

Authentication

Alice

Bob



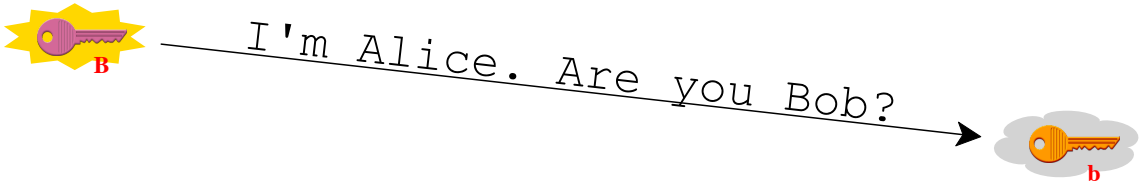
As long as Bob picks a unique random R each time this will work ok

... but there is still some room for improvement

Session Keys

Alice

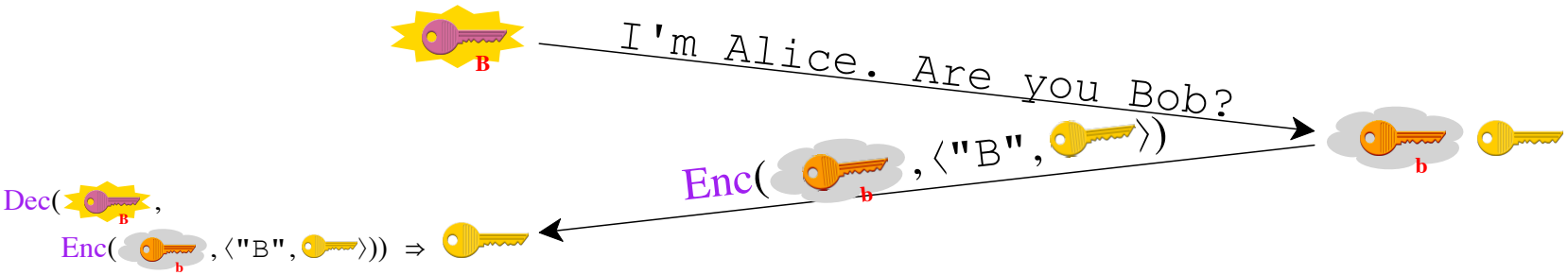
Bob



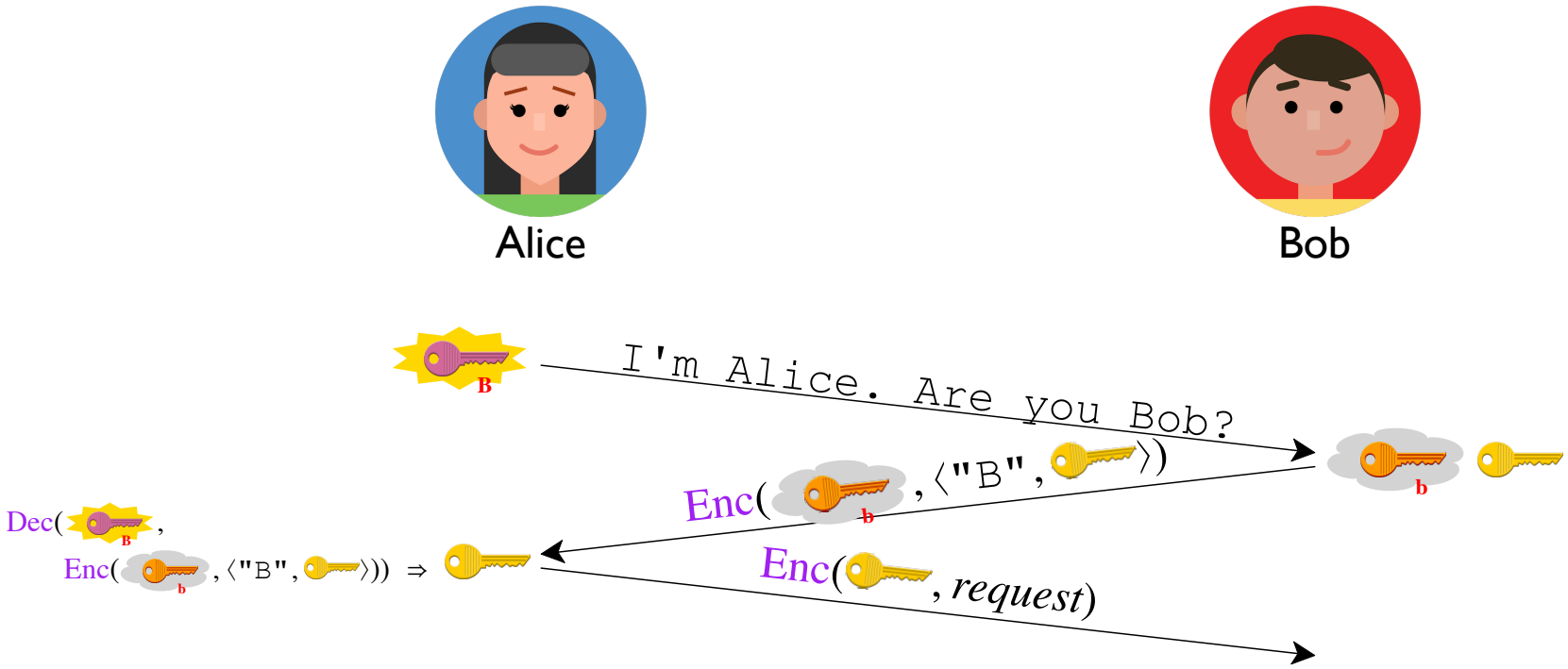
Session Keys

Alice

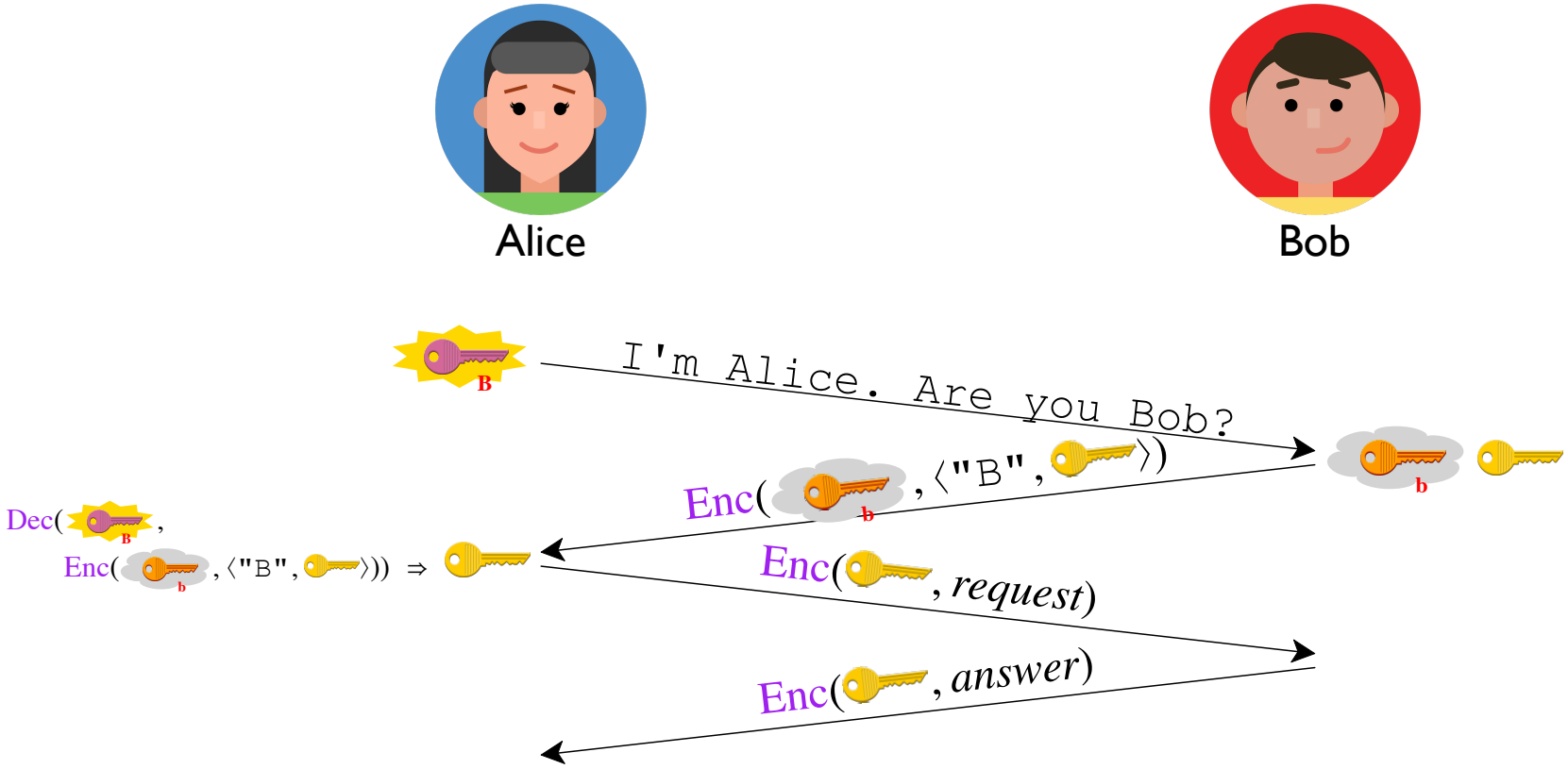
Bob



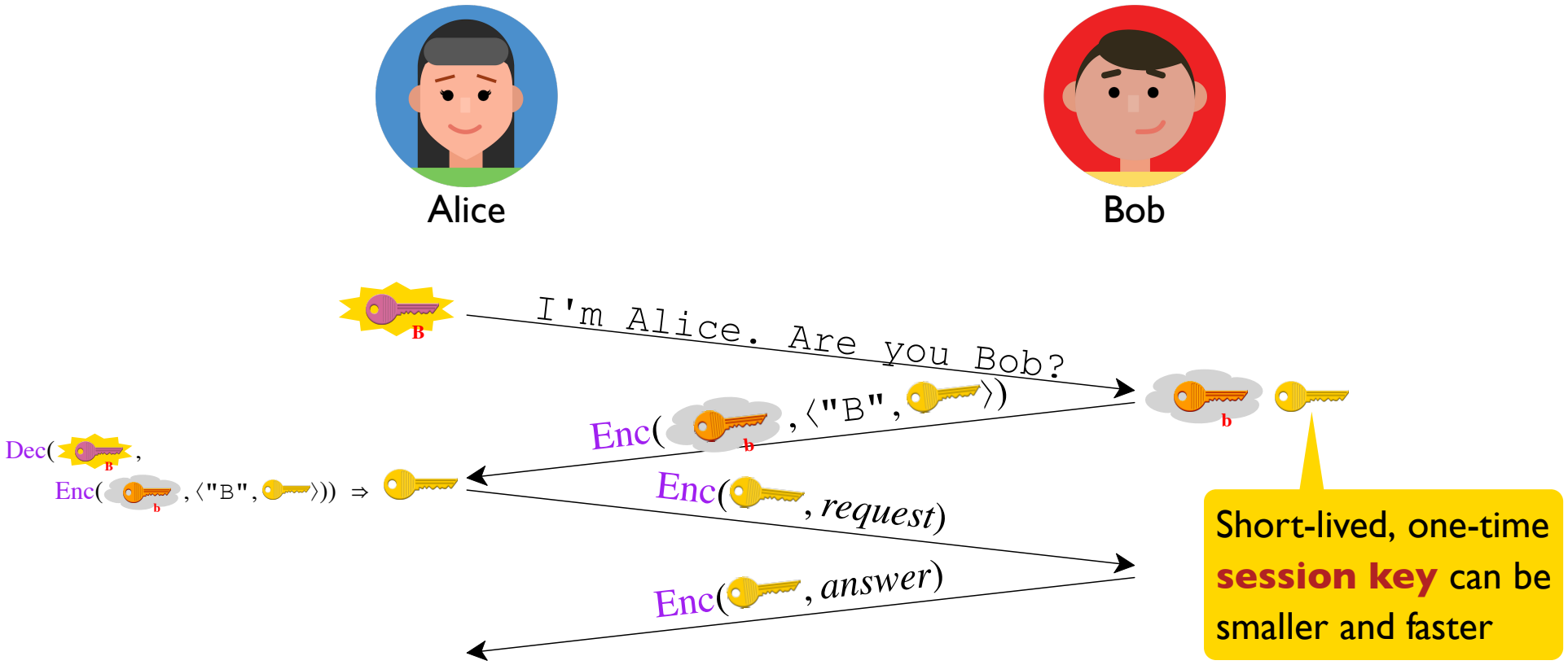
Session Keys



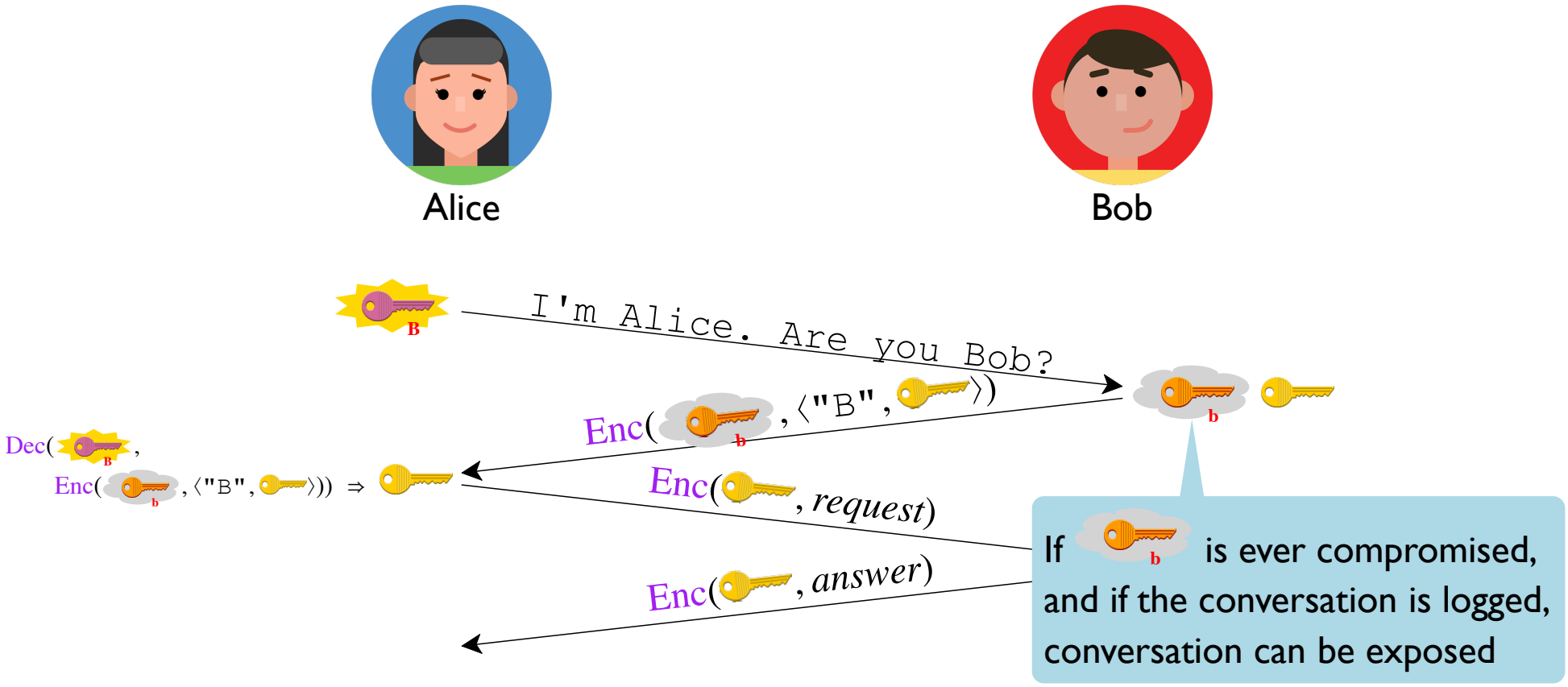
Session Keys



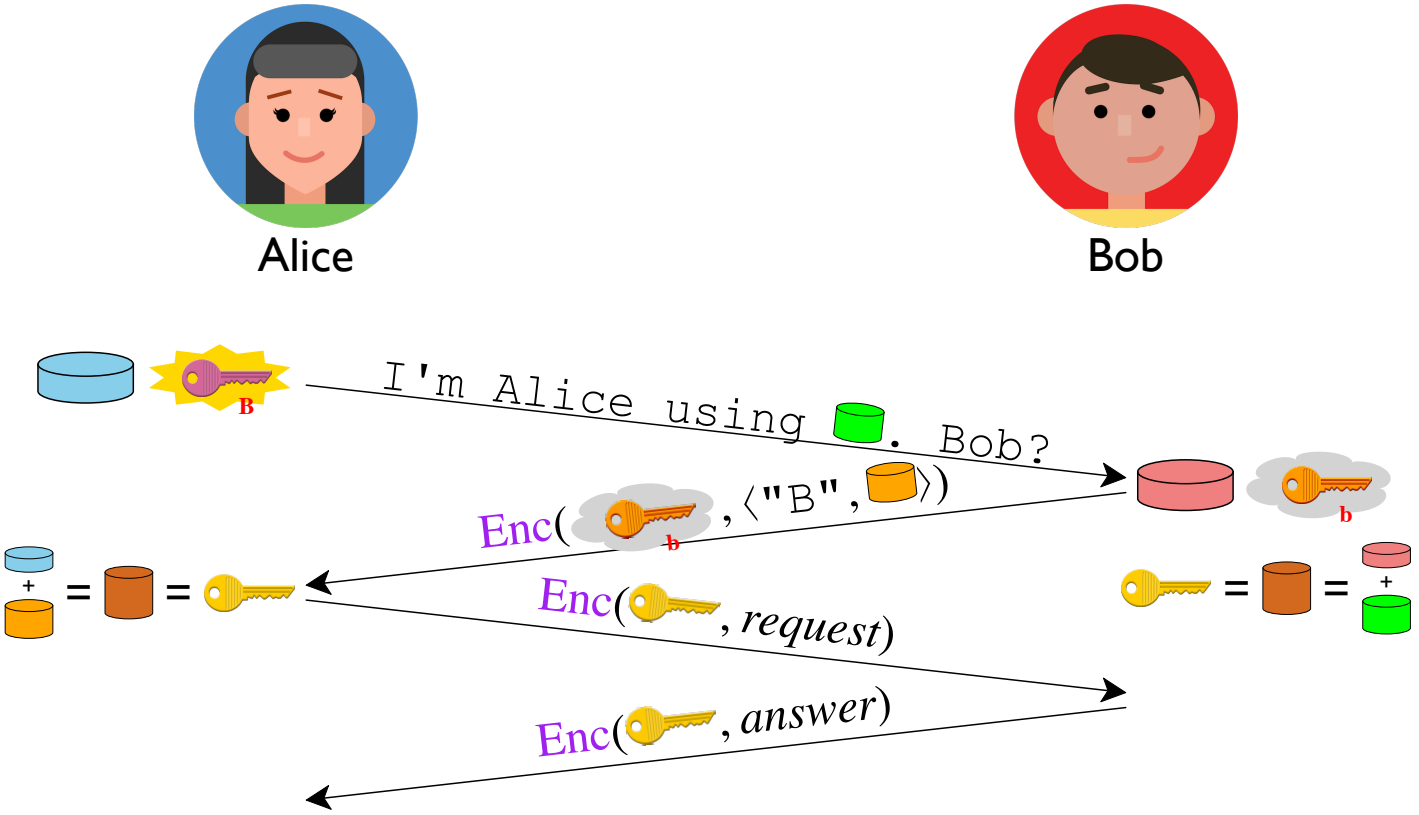
Session Keys



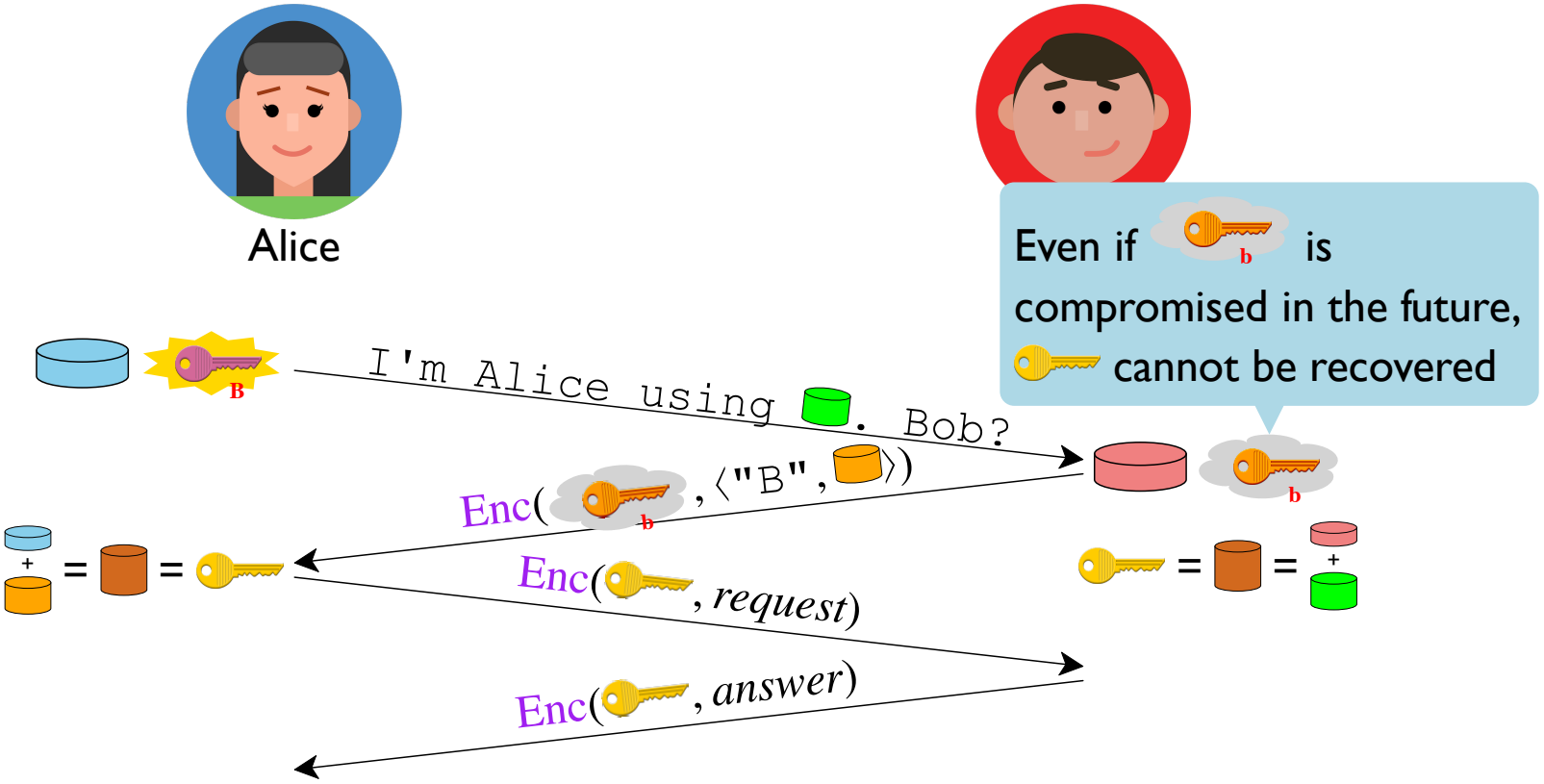
Session Keys



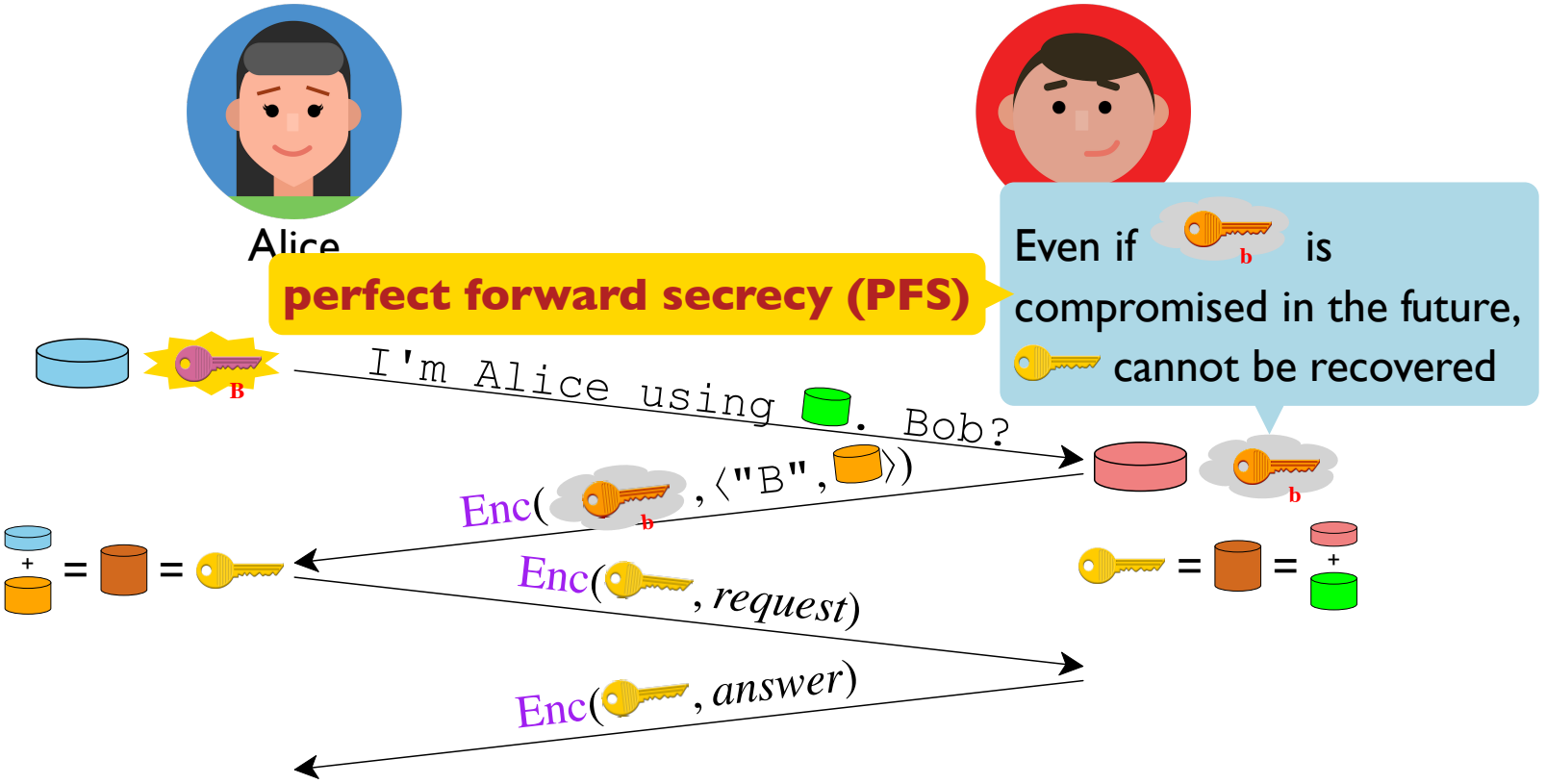
Session Keys with Perfect Forward Secrecy



Session Keys with Perfect Forward Secrecy



Session Keys with Perfect Forward Secrecy



Summary

Authentication can mean logging in with a password, but it can also mean making sure that you're logging into the right server

We need our whole cryptography toolbox to get it right
... and there are still many pitfalls

Certificates and **certificate chains** address the problem of ahead-of-time sharing

Setting up **session keys** during authenticating is a best practice