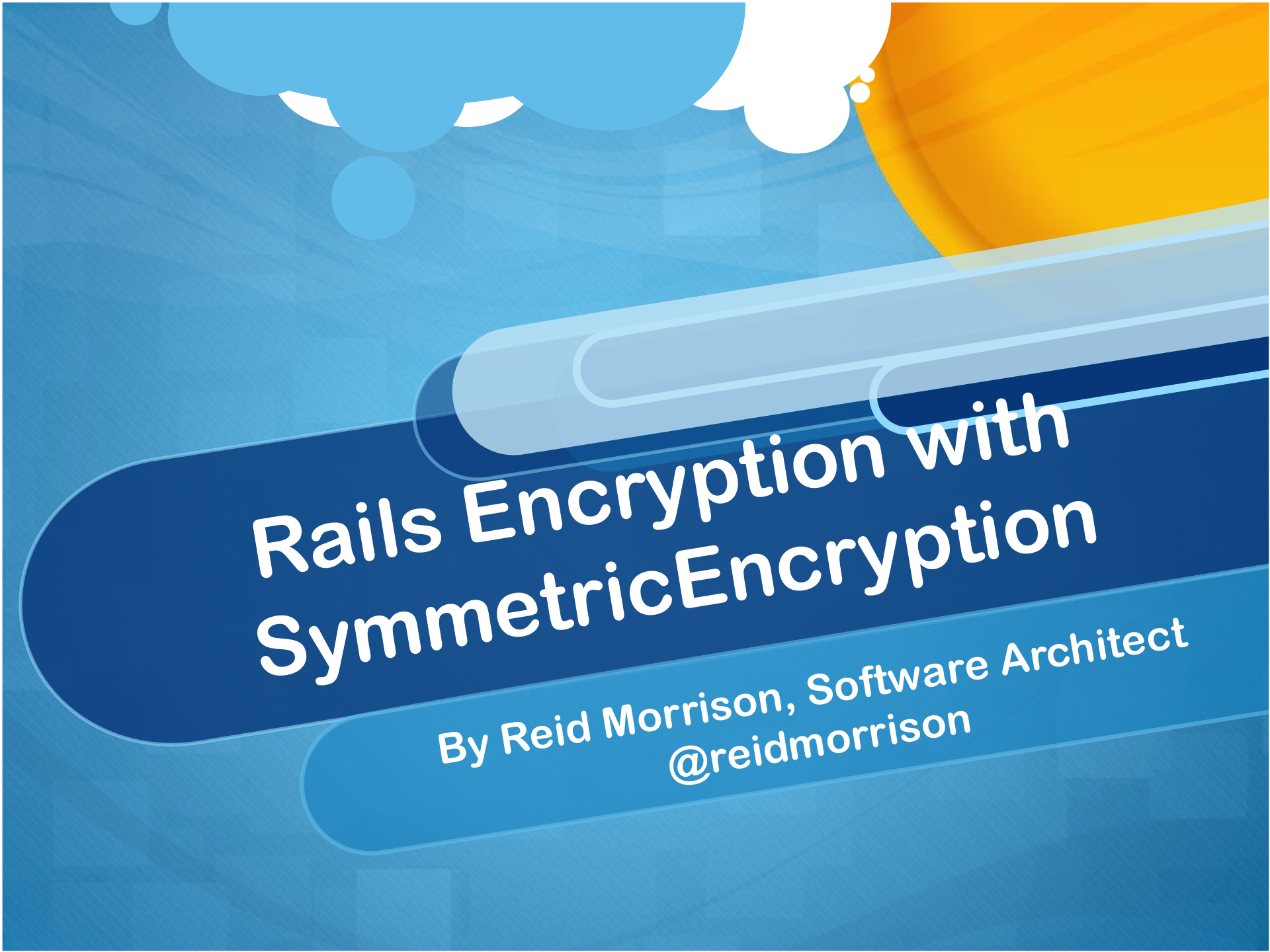




Rails Encryption with SymmetricEncryption

By Reid Morrison, Software Architect
@reidmorrison



Encrypting Sensitive Data

- Sensitive Data in the database
 - ActiveRecord attributes
 - Mongoid fields
- Passwords in configuration files
 - MySQL password (database.yml)
 - MongoDB passwords (mongoid.yml)
 - External supplier web services passwords



Encryption Example

SymmetricEncryption.**encrypt**("Keep me safe")

=> **"gllubGAQqXNrpvacvfrohw==\n"**



Decryption Example

SymmetricEncryption.**decrypt**("gllubGAQqXNrpvacvfrohw==\n")

=> **"Keep me safe"**



ActiveRecord Example

```
class Person < ActiveRecord::Base
```

```
  attr_encrypted :ssn
```

```
end
```

```
person = Person.new
```

```
person.ssn = '123456789'
```

```
person.encrypted_ssn
```

```
⇒ "95kcRwKStvgkVd+LogCn4Q==\n"
```

```
# add_column :people, :encrypted_ssn, :string
```

Mongoid Example

```
class Person
  include Mongoid::Document
  field :name, :type => String
  field :encrypted_ssn, :type => String, :encrypted => true
end

person = Person.new
person.ssn = '123456789'
person.encrypted_ssn
=> "95kcRwKStvgkVd+LogCn4Q==\n"
```



config/database.yml

production:

adapter: mysql

host: db1primary

database: myapp_production

username: myapp

password: <%= SymmetricEncryption.try_decrypt

"JqLJOi6dNjWI9kX9ISL1XQ==\n" %>



PCI Compliance Requirements

- Remove Encryption key from:
 - Source Code Repository
 - Development team access
- Change encryption keys every 12 months
 - Re-encrypt existing data
 - Zero downtime
 - Encrypt with new key, decrypt with new and old keys
 - Destroy old keys after re-encryption



Options

- shuber/attr_encrypted
 - Adds encryption methods and attributes to Object
 - Already in production encrypting data
- Hours digging through github and google searches
 - Similar and different to attr_encrypted
 - None addressed PCI requirements
- Built symmetric-encryption

What symmetric-encryption does for you





File Encryption

- Large File Encryption and decryption
 - “On the fly”
 - Streaming API
 - Compression
 - Header
 - Compressed?
 - Encryption Key Version



Writing

```
SymmetricEncryption::Writer.open(  
  'filename', :compress => true) do |file|  
  file.write "Hello World\n"  
  file.write "Keep this safe and secure\n"  
end
```



Reading

```
SymmetricEncryption::Reader.open('filename') do |file|  
  file.each_line { |line| puts line }  
end
```



Features

- Not just for PCI compliance – Good practice
- Lightweight and simple to use
- Secures Passwords in configuration files
- Waterfall decryption to support older data
- Multiple Keys and versioning
- ORM: ActiveRecord & Mongoid
- Can be used standalone without Rails
- File Streaming API to encrypt files on the fly
- Rake tasks for Operations to generate keys and random passwords



Installation

- For Bundler, add to Gemfile:

```
gem 'symmetric-encryption'
```

- Remove 'attr_encrypted' if present

```
bundle install
```

- Otherwise

```
gem install symmetric-encryption
```

```
require 'symmetric-encryption'
```

- Create config file

```
config/symmetric-encryption.yml
```



Questions?

SymmetricEncryption:

github.com/ClarityServices/symmetric-encryption

Reid Morrison

@reidmorrison

reidmo@gmail.com

www.linkedin.com/in/reidmorrison



Other Gems

- **active_record_slave**
 - Replacement for read from slave
 - Supports dynamic SQL calls, AREL, etc
 - Highly performant with no overhead for calls to master/primary
- **sync_attr**
 - Thread-safe Synchronized attributes and class variables for lazy loading and/or default values
 - Don't have to stick everything into a Rails initializer
- **Jms4jruby**
 - JMS API for JRuby to talk to ActiveMQ, HornetQ, WebSphere MQ, Oracle AQ, any JMS provider.
- **hyperic-mongodb**
 - Monitoring a MongoDB sharded cluster using Hyperic HQ
- **RubyWMQ**
 - Ruby MRI gem for communicating with IBM WebSphere MQ