Introduction to Blockchain

Lecture 1: RSA, SHA and Digital Signatures

Ras Dwivedi

IIT Kanpur

May 21, 2018

Outline

Outline

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:
  - 1 on 1^{st} June

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:
  - 1 on 1^{st} June
  - Would have questions from both the section

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:
  - 1 on 1^{st} June
  - Would have questions from both the section
  - Duration: About 30 mins

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:
  - 1 on 1^{st} June
  - Would have questions from both the section
  - Duration: About 30 mins
  - Mandatory to pass the quiz

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:
  - 1 on 1^{st} June
  - Would have questions from both the section
  - Duration: About 30 mins
  - Mandatory to pass the quiz
- Assignment

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:
  - 1 on 1^{st} June
  - Would have questions from both the section
  - Duration: About 30 mins
  - Mandatory to pass the quiz
- Assignment
  - Would not be graded

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:
  - 1 on 1^{st} June
  - Would have questions from both the section
  - Duration: About 30 mins
  - Mandatory to pass the quiz
- Assignment
  - Would not be graded
  - just for practice

Course Logistic

- Week 1 (21^{st} May to 25^{th} May)
  - Blockchain
- Week 2 (28^{st} May to 1^{st} June)
  - Software Security
- Attendance: Compulsory
  - Passing this course requires satisfactory number of classes to be attended
- Quiz:
  - 1 on 1^{st} June
  - Would have questions from both the section
  - Duration: About 30 mins
  - Mandatory to pass the quiz
- Assignment
  - Would not be graded
  - just for practice

Blockchain

Blockchain

- RSA, SHA and Digital Signatures

Blockchain

- RSA, SHA and Digital Signatures
- Introduction to Cryptocurrency and Bitcoin

Blockchain

- RSA, SHA and Digital Signatures
- Introduction to Cryptocurrency and Bitcoin
- Introduction to Ethereum and Mist

Blockchain

- RSA, SHA and Digital Signatures
- Introduction to Cryptocurrency and Bitcoin
- Introduction to Ethereum and Mist
- Hands on Mist and Geth

Blockchain

- RSA, SHA and Digital Signatures
- Introduction to Cryptocurrency and Bitcoin
- Introduction to Ethereum and Mist
- Hands on Mist and Geth
- Byzantine General Problem

Blockchain

- RSA, SHA and Digital Signatures
- Introduction to Cryptocurrency and Bitcoin
- Introduction to Ethereum and Mist
- Hands on Mist and Geth
- Byzantine General Problem

Outline

Cryptography

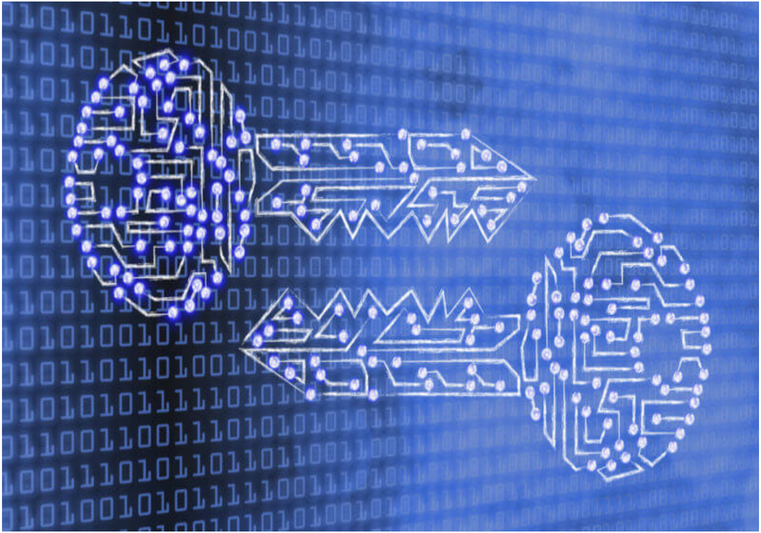

Figure:

Cesar Cipher

Figure: Cesar Cipher!!

RAS $->$ UDV

Symmetric key Cryptography

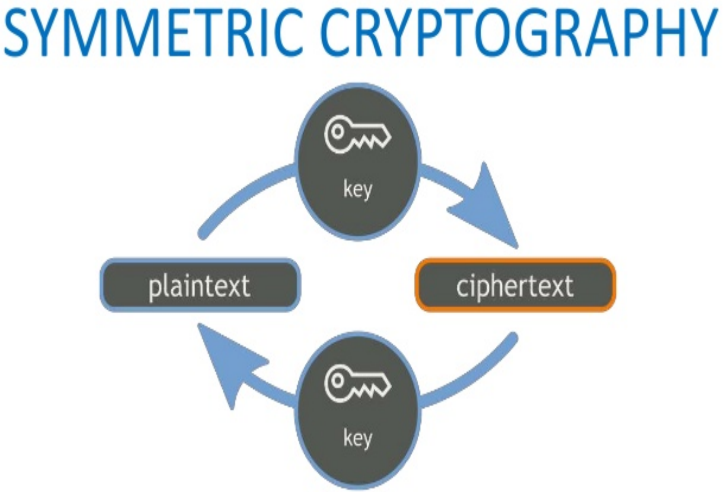

Figure:

Public key cryptography

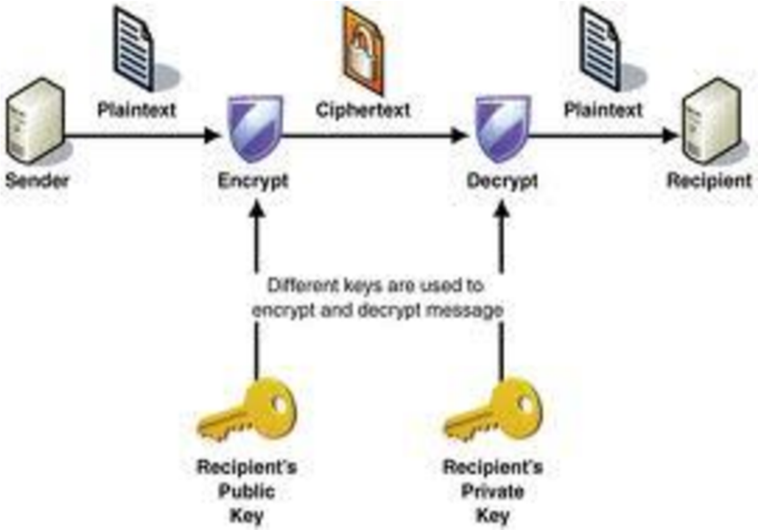

Figure:

Outline

Factoring is hard

Factoring is hard

$6 = 2 \times 3$

Factoring is hard

$6 = 2 \times 3$

Convince yourself that factoring is hard!!

Factoring is hard

$6 = 2 \times 3$

Convince yourself that factoring is hard!!

$100 =$

Factoring is hard

$6 = 2 \times 3$

Convince yourself that factoring is hard!!

$100 = 10 \times 10 = 2 \times 2 \times 5 \times 5$

Factoring is hard

$6 = 2 \times 3$

Convince yourself that factoring is hard!!

$100 = 10 \times 10 = 2 \times 2 \times 5 \times 5$

$299 =$

Factoring is hard

$6 = 2 \times 3$

Convince yourself that factoring is hard!!

$100 = 10 \times 10 = 2 \times 2 \times 5 \times 5$

$299 = 13 \times 23$

Factoring is hard

$6 = 2 \times 3$

Convince yourself that factoring is hard!!

$100 = 10 \times 10 = 2 \times 2 \times 5 \times 5$

$299 = 13 \times 23$

$437 =$

Factoring is hard

$6 = 2 \times 3$

Convince yourself that factoring is hard!!

$100 = 10 \times 10 = 2 \times 2 \times 5 \times 5$

$299 = 13 \times 23$

$437 = 19 \times 23$

Factoring is hard

$6 = 2 \times 3$

Convince yourself that factoring is hard!!

$100 = 10 \times 10 = 2 \times 2 \times 5 \times 5$

$299 = 13 \times 23$

$437 = 19 \times 23$

$589 = 19 \times 31$

So how to use it?

Fermat's little theorem

$a^{p-1} = 1$ modulo p

Fermat's little theorem

$a^{p-1} = 1$ modulo p

p is prime

Fermat's little theorem

$a^{p-1} = 1$ modulo p
p is prime
Example
$2^4 \ \%5 =$

Fermat's little theorem

$a^{p-1} = 1$ modulo p
p is prime
Example
$2^4 \ \%5 = 16\%5 = 1$

Fermat's little theorem

$a^{p-1} = 1$ modulo p
p is prime
Example
$2^4 \ \%5 = 16\%5 = 1$
$4^{10} \ \%11$

Fermat's little theorem

$a^{p-1} = 1$ modulo p

p is prime

Example

$2^4 \ \%5 = 16\%5 = 1$

$4^{10} \ \%11 = 1048576\%11 = 1$

RSA

RSA

- proposed by Rivest,Shamir,Adleman

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$
- calculate $\phi = lcm(p - 1, q - 1)$

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose e such that $gcd(e, \phi) = 1$

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose e such that $gcd(e, \phi) = 1$
- calculate d such that $d = e^{-1} mod \phi$

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose e such that $gcd(e, \phi) = 1$
- calculate d such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose e such that $gcd(e, \phi) = 1$
- calculate d such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$
- Idea: $m^{e \times d} = m^{e^d} = m \ modulo \ n$

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose e such that $gcd(e, \phi) = 1$
- calculate d such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$
- Idea: $m^{e \times d} = m^{e^d} = m \ modulo \ n$
- **encryption:** $c = m^e \ mod \ n$

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose e such that $gcd(e, \phi) = 1$
- calculate d such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \; mod \; \phi$
- Idea: $m^{e \times d} = m^{e^d} = m \; modulo \; n$
- **encryption:** $c = m^e \; mod \; n$
- **decryption:** $p = c^d \; mod \; n$

RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number p, q
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose e such that $gcd(e, \phi) = 1$
- calculate d such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$
- Idea: $m^{e \times d} = m^{e^d} = m \ modulo \ n$
- **encryption:** $c = m^e \ mod \ n$
- **decryption:** $p = c^d \ mod \ n$

RSA: Example

- $p = 5$,

RSA: Example

- $p = 5$, $q = 7$

RSA: Example

- $p = 5$, $q = 7$ $p \times q = 35$

RSA: Example

- $p = 5$, $q = 7$ $p \times q = 35$
- $p - 1 = 4$

RSA: Example

- $p = 5$, $q = 7$ $p \times q = 35$
- $p - 1 = 4$, $q - 1 = 6$

RSA: Example

- $p = 5$, $q = 7$ $p \times q = 35$
- $p - 1 = 4$, $q - 1 = 6$, $\phi = 24$

RSA: Example

- $p = 5$, $q = 7$ $p \times q = 35$
- $p - 1 = 4$, $q - 1 = 6$, $\phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11$

RSA: Example

- $p = 5$, $q = 7$ $p \times q = 35$
- $p - 1 = 4$, $q - 1 = 6$, $\phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11$,$d =$

RSA: Example

- $p = 5$, $q = 7$ $p \times q = 35$
- $p - 1 = 4$, $q - 1 = 6$, $\phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11$,$d = 11$
  $e \times d = 121$

RSA: Example

- $p = 5$, $q = 7$ $p \times q = 35$
- $p - 1 = 4$, $q - 1 = 6$, $\phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11$,$d = 11$
  $e \times d = 121$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

$121\%\ 24 = 1$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

$121\%\ 24 = 1$

- $m = 2$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$
  $c = m^e\ mod\ n$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$

$c = m^e\ mod\ n$

$= 2^{11}\ mod\ 35$

$c = 2048\ mod\ 35$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$

$c = m^e\ mod\ n$

$= 2^{11}\ mod\ 35$

$c = 2048\ mod\ 35\ ;\ c = 18$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$
  $c = m^e\ mod\ n$
  $= 2^{11}\ mod\ 35$
  $c = 2048\ mod\ 35\ ;\ c = 18\ (35 \times 58 = 2030)$
  Decryption
- $d = 11$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$
  $c = m^e\ mod\ n$
  $= 2^{11}\ mod\ 35$
  $c = 2048\ mod\ 35\ ;\ c = 18\ (35 \times 58 = 2030)$
  Decryption
- $d = 11\ ,\ c = 18$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$
  $c = m^e \ mod \ n$
  $= 2^{11} \ mod\ 35$
  $c = 2048\ mod\ 35\ ;\ c = 18\ (35 \times 58 = 2030)$
  Decryption
- $d = 11\ ,\ c = 18$
- $m = c^d \ mod \ n$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$
  $c = m^e \ mod\ n$
  $= 2^{11}\ mod\ 35$
  $c = 2048\ mod\ 35\ ;\ c = 18\ (35 \times 58 = 2030)$
  Decryption
- $d = 11\ ,\ c = 18$
- $m = c^d\ mod\ n$
  $m = 18^{11}\ mod\ 35$
  $= 64268410079232\%35$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$

$c = m^e\ mod\ n$

$= 2^{11}\ mod\ 35$

$c = 2048\ mod\ 35\ ;\ c = 18\ (35 \times 58 = 2030)$

Decryption

- $d = 11\ ,\ c = 18$
- $m = c^d\ mod\ n$

$m = 18^{11}\ mod\ 35$

$= 64268410079232\%35$

RSA: Example

- $p = 5,\ q = 7\ p \times q = 35$
- $p - 1 = 4\ ,\ q - 1 = 6\ ,\ \phi = 24$
  Oops! $\phi = 12$, but 24 would still work
- $e = 11\ , d = 11$
  $e \times d = 121$
  $24 \times 5 = 120$

121% 24 =1

- $m = 2$

$c = m^e\ mod\ n$

$= 2^{11}\ mod\ 35$

$c = 2048\ mod\ 35\ ;\ c = 18\ (35 \times 58 = 2030)$

Decryption

- $d = 11\ ,\ c = 18$
- $m = c^d\ mod\ n$

$m = 18^{11}\ mod\ 35$

$= 64268410079232\%35$

RSA: Example

- $p = 7$,

RSA: Example

- $p = 7$, $q = 13$

RSA: Example

- $p = 7$, $q = 13$ $p \times q = 91$

RSA: Example

- $p = 7$, $q = 13$ $p \times q = 91$
- $p - 1 = 6$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12$

RSA: Example

- $p = 7$, $q = 13$ $p \times q = 91$
- $p - 1 = 6$, $q - 1 = 12$, $\phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ , d =$

RSA: Example

- $p = 7$, $q = 13$ $p \times q = 91$
- $p - 1 = 6$, $q - 1 = 12$, $\phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5$,$d = 29$

RSA: Example

- $p = 7$, $q = 13$ $p \times q = 91$
- $p - 1 = 6$, $q - 1 = 12$, $\phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5$,$d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ , d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ , d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15\ c = m^e\ mod\ n$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ ,d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15\ c = m^e\ mod\ n$
  $= 15^5\ mod\ 91$
  $c = 759375\ mod\ 91$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ , d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15\ c = m^e\ mod\ n$
  $= 15^5\ mod\ 91$
  $c = 759375\ mod\ 91\ ;\ c = 71$

RSA: Example

- $p = 7$, $q = 13$ $p \times q = 91$
- $p - 1 = 6$, $q - 1 = 12$, $\phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5$,$d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15$ $c = m^e\ mod\ n$
  $= 15^5\ mod\ 91$
  $c = 759375\ mod\ 91$; $c = 71$ Decryption
- $d = 47$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ ,d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15\ c = m^e\ mod\ n$
  $= 15^5\ mod\ 91$
  $c = 759375\ mod\ 91\ ;\ c = 71$ Decryption
- $d = 47\ ,\ c = 71$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ , d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15\ c = m^e\ mod\ n$
  $= 15^5\ mod\ 91$
  $c = 759375\ mod\ 91\ ;\ c = 71$ Decryption
- $d = 47\ ,\ c = 71$
- $m = c^d\ mod\ n$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ , d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15\ c = m^e\ mod\ n$
  $= 15^5\ mod\ 91$
  $c = 759375\ mod\ 91\ ;\ c = 71$ Decryption
- $d = 47\ ,\ c = 71$
- $m = c^d\ mod\ n$
  $m = 71^{29}\ mod\ 91$
  $=$
  $48583870762480666770881138170405337679268897592532343 1\%91$

RSA: Example

- $p = 7,\ q = 13\ p \times q = 91$
- $p - 1 = 6\ ,\ q - 1 = 12\ ,\ \phi = 72$
  Oops! $\phi = 12$, but 72 would still work
- $e = 5\ ,d = 29$
  $72 \times 2 = 144$
  $5 \times 29 = 145$
  $(145)\%72 == 1$
- $m = 15\ c = m^e\ mod\ n$
  $= 15^5\ mod\ 91$
  $c = 759375\ mod\ 91\ ;\ c = 71$ Decryption
- $d = 47\ ,\ c = 71$
- $m = c^d\ mod\ n$
  $m = 71^{29}\ mod\ 91$
  $=$
  $48583870762480666770881138170405337679268897592532431\%91$
  $m = 15$

Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document

Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document

Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work

Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA

Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA
- for document m Alice uses $s = m^d$ as her digital signature. To verify, verifier calculates s^e and if $m = s^e \ mod \ n$, signature is genuine

Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA
- for document m Alice uses $s = m^d$ as her digital signature. To verify, verifier calculates s^e and if $m = s^e \ mod \ n$, signature is genuine
- d is called Alice's secret key and e is called Alice's Public key

Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA
- for document m Alice uses $s = m^d$ as her digital signature. To verify, verifier calculates s^e and if $m = s^e \ mod \ n$, signature is genuine
- d is called Alice's secret key and e is called Alice's Public key

Is the Scheme secure?

Is the Scheme secure?

No!

Is the Scheme secure?

No!

- given (n, e) private key of Alice cannot be calculated

Is the Scheme secure?

No!

- given (n, e) private key of Alice cannot be calculated
- given document m, $s = m^d$ could not be guessed.

Is the Scheme secure?

No!

- given (n, e) private key of Alice cannot be calculated
- given document m, $s = m^d$ could not be guessed.
- **Problem:** forging given m_1, m_2 as two document, and s_1, s_2 as their digital signature, one can find the valid signature of $m_1.m_2$ as $s_1.s_2$

Is the Scheme secure?

No!

- given (n, e) private key of Alice cannot be calculated
- given document m, $s = m^d$ could not be guessed.
- **Problem:** forging given m_1, m_2 as two document, and s_1, s_2 as their digital signature, one can find the valid signature of $m_1.m_2$ as $s_1.s_2$
- Also the length of the signature is proportional to the size of the document

Is the Scheme secure?

No!

- given (n, e) private key of Alice cannot be calculated
- given document m, $s = m^d$ could not be guessed.
- **Problem:** forging given m_1, m_2 as two document, and s_1, s_2 as their digital signature, one can find the valid signature of $m_1.m_2$ as $s_1.s_2$
- Also the length of the signature is proportional to the size of the document
- slow

Is the Scheme secure?

No!

- given (n, e) private key of Alice cannot be calculated
- given document m, $s = m^d$ could not be guessed.
- **Problem:** forging given m_1, m_2 as two document, and s_1, s_2 as their digital signature, one can find the valid signature of $m_1.m_2$ as $s_1.s_2$
- Also the length of the signature is proportional to the size of the document
- slow

What could we do now?

Outline

SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess x

SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess x
- given x_1 its is impossible to find x_2 such that $f(x_1) = f(x_2)$

SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess x
- given x_1 its is impossible to find x_2 such that $f(x_1) = f(x_2)$
- it is impossible to find x_1, x_2, such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$

SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess x
- given x_1 its is impossible to find x_2 such that $f(x_1) = f(x_2)$
- it is impossible to find x_1, x_2, such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$

Lets understand by example

Example: Hash Function

Suppose I can see Future. So I can foretell score of tomorrow's IPL's match.

Example: Hash Function

Suppose I can see Future. So I can foretell score of tomorrow's IPL's match. But If I tell score before, you can always change it and prove me wrong.

Example: Hash Function

Suppose I can see Future. So I can foretell score of tomorrow's IPL's match. But If I tell score before, you can always change it and prove me wrong. I publish Hash of tomorrow's score as: "a34728bfed78dc89..."

Example: Hash Function

Suppose I can see Future. So I can foretell score of tomorrow's IPL's match. But If I tell score before, you can always change it and prove me wrong. I publish Hash of tomorrow's score as: "a34728bfed78dc89..."

- can you tell what score I had in Mind?

Example: Hash Function

Suppose I can see Future. So I can foretell score of tomorrow's IPL's match. But If I tell score before, you can always change it and prove me wrong. I publish Hash of tomorrow's score as: "a34728bfed78dc89..."

- can you tell what score I had in Mind?
- can I later change the score I thought before?

Example: Hash Function

Suppose I can see Future. So I can foretell score of tomorrow's IPL's match. But If I tell score before, you can always change it and prove me wrong. I publish Hash of tomorrow's score as: "a34728bfed78dc89..."

- can you tell what score I had in Mind?
- can I later change the score I thought before?
- can I purposely find hash such that two score are possible for that hash ?

Example: Hash Function

Suppose I can see Future. So I can foretell score of tomorrow's IPL's match. But If I tell score before, you can always change it and prove me wrong. I publish Hash of tomorrow's score as: "a34728bfed78dc89..."

- can you tell what score I had in Mind?
- can I later change the score I thought before?
- can I purposely find hash such that two score are possible for that hash ?

SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess x
- given x_1 its is impossible to find x_2 such that $f(x_1) = f(x_2)$
- it is impossible to find x_1, x_2, such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$

SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess x
- given x_1 its is impossible to find x_2 such that $f(x_1) = f(x_2)$
- it is impossible to find x_1, x_2, such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$

Difference between 2^{nd} and 3^{rd} condition?

Merkle Demgrad Construction

Need of Padding message m?

Merkle Demgrad Construction

Need of Padding message m?

- m is prefix of $PAD(m)$

Merkle Demgrad Construction

Need of Padding message m?

- m is prefix of $PAD(m)$
- if $|m_1| = |m_2|$ then $|PAD(m_1)| = |PAD(m_2)|$

Merkle Demgrad Construction

Need of Padding message m?

- m is prefix of $PAD(m)$
- if $|m_1| = |m_2|$ then $|PAD(m_1)| = |PAD(m_2)|$
- if $|m_1| \neq |m_2|$ then the last block of $PAD(m_1) \neq PAD(m_2)$

Merkle Demgrad Construction

Need of Padding message m?

- m is prefix of $PAD(m)$
- if $|m_1| = |m_2|$ then $|PAD(m_1)| = |PAD(m_2)|$
- if $|m_1| \neq |m_2|$ then the last block of $PAD(m_1) \neq PAD(m_2)$