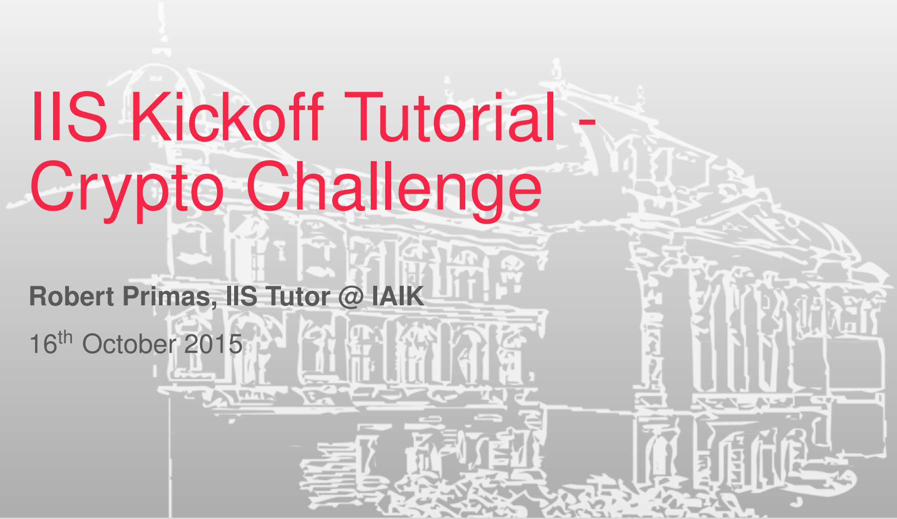


IIS Kickoff Tutorial - Crypto Challenge

Robert Primas, IIS Tutor @ IAIK

16th October 2015

Content

1. Organisatorial Information
2. Project Specification
3. Crypto Challenge
4. Submissions
5. Grading
6. Timeline

Tutorials

No further tutorials, support directly from Tutor

- Newsgroup (interesting for everyone)
- Email (small problem, interesting for your group)
  - rprimas@student.tugraz.at
  - stefan.steinegger@student.tugraz.at
- Meeting (bigger problem, interesting for your group)
  - Contact tutor for arrangement

Practicals

- IIS Practicals are also in English
  - English is appreciated
  - German is also ok (oral/written)
- Slides will be available on the wiki
- Assignment sheet is available HERE

Project Specification

2-4 Pages, answer questions about Crypto Challenge

- Format is PDF
- Hand in by 17.11 via Stics
- One specification per group

Crypto Challenge

Three challenge chapters

- Vigenère cipher
- RSA Encryption
- Hash Functions

You need to implement all tasks for a good grade

Vigenère Cipher

- Extension of Caesar cipher
- Key contains multiple letter offsets
- Weak against letter frequency analysis

$$\text{Encryption} : C_i = E_K(M_i) = (M_i + K_i) \mod 26$$

$$\text{Decryption} : M_i = D_K(C_i) = (C_i - K_i) \mod 26$$

Vigenère cipher - Submissions

For Specification

- How would you attack it?

For Final Delivery

- Implement attack on Vigenère ciphertexts provided by us

RSA Encryption

- Asymmetric encryption scheme
- Separate keys for en/decryption
- Based on one-way trapdoor functions
- Heavily used today
- Long keys necessary
- Slow

RSA Encryption

Key generation

- Choose 2 primes p, q
- Compute modulus $n = p * q$ and $\phi(n) = (p-1)(q-1)$
- Choose public exponent e coprime to $\phi(n)$ (and $e \neq \pm 1$)
- Compute private exponent $d = e^{-1} \mod \phi(n)$
- public key = (e, n)
- private key = (d, n) or (e, p, q)

RSA Encryption

Encryption

$$C = M^e \mod n$$

Decryption

$$M = C^d \mod n$$

$$(= M^{e*d} = M^{1+k\phi(n)} = M \mod n)$$

RSA Encryption - Submissions

For Specification

- Have a look at key generation ...

$$n = p * q\,, \qquad p \text{ and } q \text{ are prime}$$

... and make 3 suggestions what could go wrong here (look at different factorization algorithms)

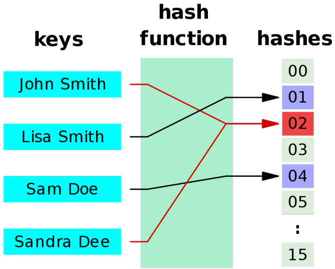
For Final Delivery

- Break RSA with short key size
- Attack multiple weak keys provided by us

Hash Functions

“A cryptographic hash function produces cryptographic checksums or fingerprints”

- Irreversible one-way function
- Created fingerprints have constant size and are “ideally unique”



Ⓒhttps://en.wikipedia.org/wiki/Hash_function

Birthday Paradox

"In a room of just 23 *people there is a* $50 - 50$ *chance of two people having the same birthday."*

- Complexity of finding n-bit hash collisions

$$\approx \sqrt{2^n} = 2^{n/2}$$

- Complexity of breaking a 160-bit hash (SHA-1)

$$\leq 2^{80}$$

Hash Functions - Submissions

For Specification

- How would you perform your attack?

For Final Delivery

- Find a “special” hash collision for a reduced version of SHA-2
- Have a look at ways to reduce the memory consumption...

Submissions

- Programming language is free of choice, should be reasonable (Matlab/Octave, Sage/Python)
- Add Readme with instructions if necessary
- We need to be able to run your code
- Write summary for each solved challenge, max. 5 pages in total

Grading

- Points are earned at the final interview for explaining your correctly implemented attacks
- All group members get same grade (except corner cases)
- All tasks need to be solved for a good grade
- All group members need to be able to explain all implemented attacks

Timeline

- 17th November
  - Signed rule confirmation email
  - Project Specification
- $\sim$ 23th November
  - Group Meetings
- $<$ 24th December - Give progress update to tutor
- 21.01 - Final deliverable
- $\sim$ 28.01 - Final interviews