RUHR-UNIVERSITÄT BOCHUM

Hack.Lu 2013 Challenges

ECKA, Geiers Lambda, Marvin is plane-Jane

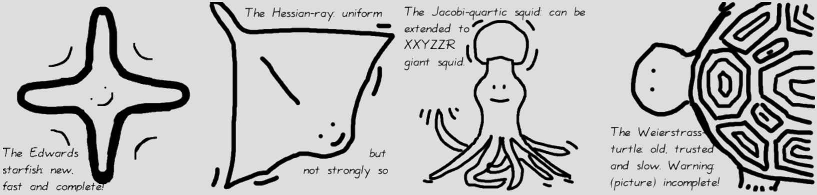
12. Februar 2014

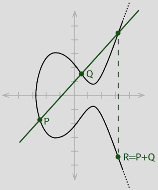
FluxFingers
Ruhr Universität Bochum

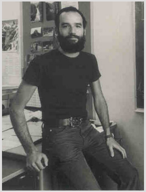
Friedrich Wiemer

Outline

ECKA


The Edwards starfish: new, fast and complete!
The Hessian-ray: uniform
but not strongly so
The Jacobi-quartic squid: can be extended to XXYZZR giant squid.
The Weierstrass turtle: old, trusted and slow. Warning: (picture) incomplete!


Q
P
R=P+Q



Story

Infos

- Service using key agreement on elliptic curves
- Combines two different ones:
  1. Exchange a point P
  2. Agree on key
  3. Send AES-ECB encrypted password

Hint: He, we have the latest news for you. The first part of their strange key agreement was designed by the famous SHA-Robot MIR!

Stats

- Category: Crypto
- Points: 100
- Solved by: 5 Teams

Service

Crypto Stuff

Involved Crypto-Stuff

- Elliptic Curve Crypto (ECC)
- Key Agreements:
  - ThreePass
  - Diffie Hellman

ECC

Elliptic Curve Cryptography

- asymmetric crypto
- uses elliptic curves over finite fields as group
- thus can replace other groups (normally $\mathbb{Z}_p^*$)
- good properties like small key size etc.

- Discrete Logarithm Problem (DLP) is hard
- algorithms like DHKE, Elgamal can be used

ThreePass

on Elliptic Curves

$$\mathscr{E}(\mathbb{Z}_p^*)$$

Alice		**Bob**
$\alpha \in_R \mathbb{Z}_p^*$ $P \in \mathscr{E}(\mathbb{Z}_p^*)$ $P_\alpha = \alpha P$		
	$\xrightarrow{P_\alpha}$	$\beta \in_R \mathbb{Z}_p^*$ $P_{\alpha\beta} = \beta P_\alpha$
$P_\beta = \alpha^{-1} \cdot P_{\alpha\beta}$	$\xleftarrow{P_{\alpha\beta}}$	
	$\xrightarrow{P_\beta}$	$P = \beta^{-1} \cdot P_\beta$

DHKE

on Elliptic Curves

$$\mathscr{E}(\mathbb{Z}_p^*),\ P \in \mathscr{E}(\mathbb{Z}_p^*)$$

Alice		**Bob**
$\alpha \in_R \mathbb{Z}_p^*$ $P_\alpha = \alpha P$		
	$\xrightarrow{P_\alpha}$	$\beta \in_R \mathbb{Z}_p^*$ $P_\beta = \beta P$
$P_{\alpha\beta} = \alpha \cdot P_\beta$	$\xleftarrow{P_\beta}$	
		$P_{\alpha\beta} = \beta \cdot P_\alpha$

Solution

Geiers Lambda

Story

Infos

Given:

- encrypted defusing-password
- haskell code for decryption
- collision for decryption password

Infos:

- decryption password consists of 8 alphanumeric chars
- defusing password contains only printable characters

Stats

- Category: Crypto
- Points: 200
- Solved by: 16 Teams

Haskell Code

Haskell Code

- used almost only lambda (anonymous) functions
- two interesting functions: HASH and DEC
- magic constant in DEC $\Rightarrow$ TEA
- HASH seems to be adler32

Haskell Code

- used almost only lambda (anonymous) functions
- two interesting functions: HASH and DEC
- magic constant in DEC $\Rightarrow$ TEA
- HASH seems to be adler32

Collision Finding:

- $\text{Byte}[n] + k$
- $\text{Byte}[n+1] - 2k$
- $\text{Byte}[n+2] + k$

Solution

Appendix

RUB

Challenges

- `https://ctf.fluxfingers.net/2013/challenges/1`
- `https://ctf.fluxfingers.net/2013/challenges/2`

Write-Ups

- `https://stratum0.org/blog/blog/2013/10/26/hack-dot-lu-2013-ecka/`
- `http://balidani.blogspot.pt/2013/10/hacklu-ctf-crypto-200-geiers-lambda.html`

Questions?

Thanks!