Better authentication: password revolution by evolution

Daniel R. Thomas and Alastair R. Beresford

22nd Security Protocols Workshop

UNIVERSITY OF CAMBRIDGE

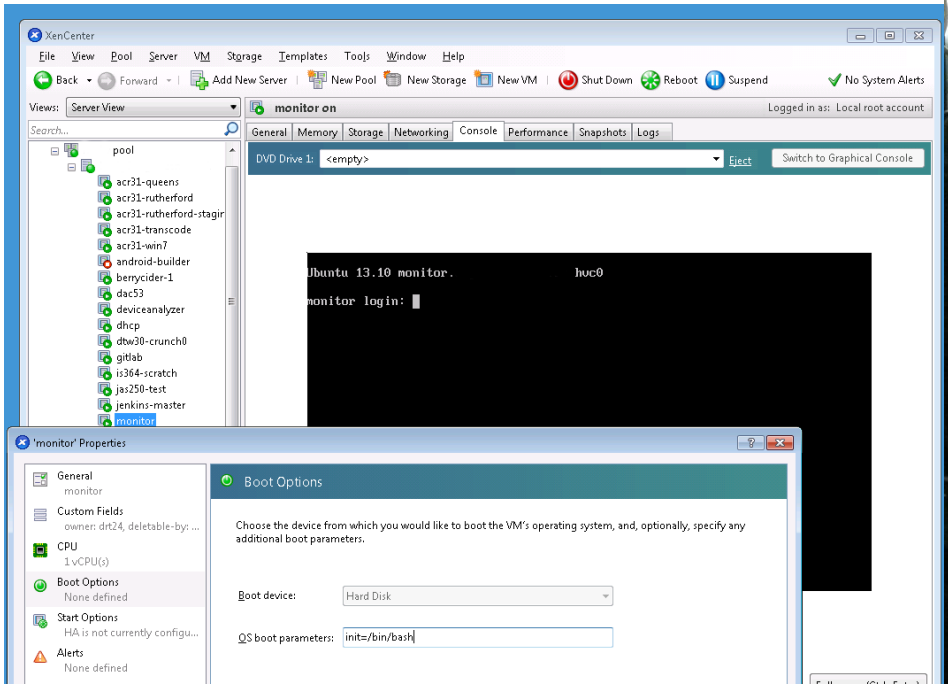
Daniel: 5017 A1EC 0B29 08E3 CF64 7CCD 5514 35D5 D749 33D9
Alastair: 9217 482D D647 8641 44BA 10D8 83F4 9FBF 1144 D9B3

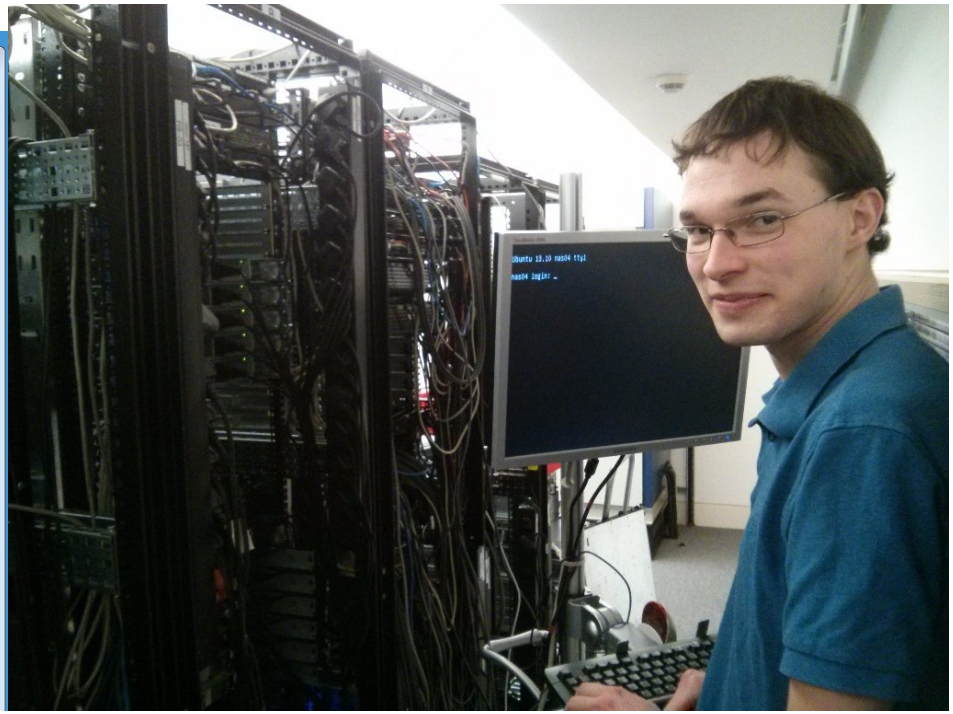
Idea: Use public key based tokens for password login

- One time token based authentication using public key cryptography
- Allow machines to provision themselves from public data and allow login
- Inspiration from Monkeysphere and Google Authenticator

Motivation: Passwords don't scale

- when you have many machines and don't want compromising one to compromise the others.
- per machine, per user passwords do not scale ($O(mn)$)
- Solutions for ssh but what about when networking broken?

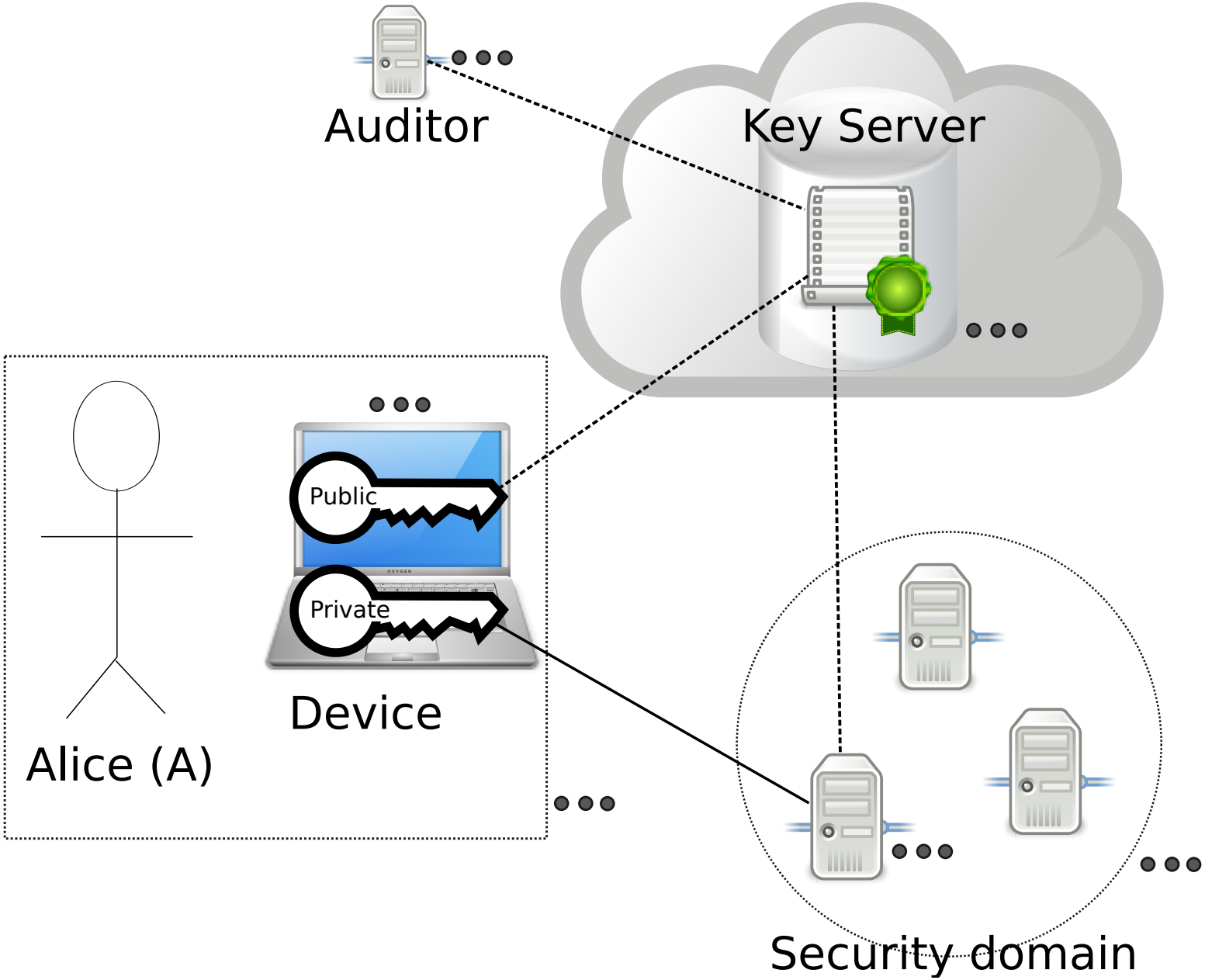




Replace passwords

- Something we could type in
- Something we don't have to remember
- Something which does not require sharing secrets
- We can use things we already have

Public keys can be distributed securely



A can authenticate by sending a token

SOTTA: Simple one time token authentication

Alice (A) can authenticate to the end point (S) in one step by sending a token:

$$A \rightarrow S : A, \{D||\lfloor t \rfloor\}_{K_A^{-1}}$$

D is the domain identifier (e.g. `dtg.cl.cam.ac.uk`) and $\lfloor t \rfloor$ is the quantised time (e.g. to the nearest minute)

Signature length is too long

Username: drt24

Password: Xq8xdTBjJHunpIC64pBm6Q94wdlZkwiyrilTJgx5b4oFEYH
ZZXIL4ouSxNW6YD3y8IsZSNDKNYKA7sYWfUi

(1 minute 20s to input password - and I got it wrong)

Signature length is too long

Need resistance to offline brute force $\Rightarrow$ 128 bits of security.

Bits	Bytes	numeric [0-9]	alphabetical [a-z]	Alphanumeric [A-Za-z0-9]	Algorithm
32	4	10	7	6	
64	8	20	14	11	
80	10	25	18	14	
128	16	39	28	22	
160	20	49	35	27	
256	32	78	55	43	Minimum
320	40	97	69	54	BSL?
512	64	155	109	86	DSA
1024	128	309	218	172	
3072	384	925	654	516	RSA

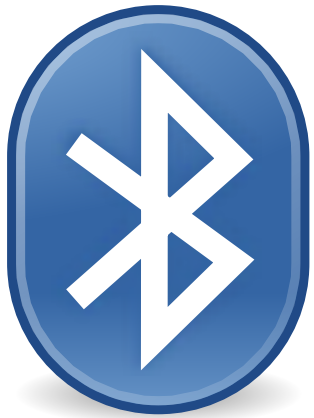
Table : Encoding sizes for different bit lengths

The long random string can be automatically input

AOTTA: Automatic one time token authentication

Some of the time the long random string can be automatically input

- bluetooth keyboard
- Copy & Paste
- QR code
- audio networking

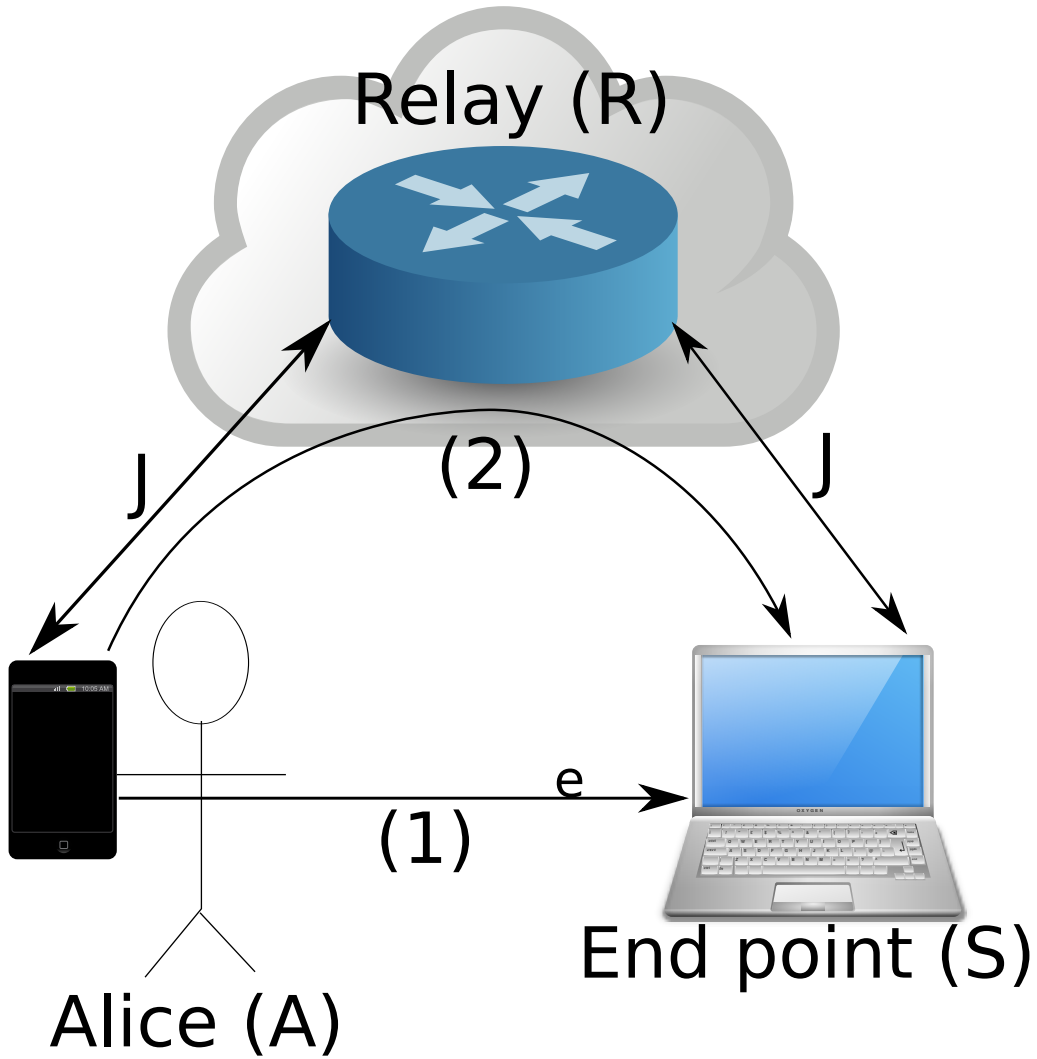



Online connectivity enables short tokens

OOTTA: Online one time token authentication

$$A \overrightarrow{e} S : s \quad (1)$$

$$A \rightarrow S : \mathrm{E}_{\mathcal{K}}(\{D||\lfloor t \rfloor\}_{K_A^{-1}}) \quad (2)$$



We can deploy this

- Can bootstrap with existing PGP / GPG infrastructure used for monkeysphere
- Authenticate to what?
- Key servers
- Relays
- Devices

Better authentication: password revolution by evolution

We can use public key cryptography to produce one time tokens allowing authentication through typing on a keyboard.

- Daniel R. Thomas drt24@cam.ac.uk
  5017 A1EC 0B29 08E3 CF64 7CCD 5514 35D5 D749 33D9
- Alastair R. Beresford arb33@cam.ac.uk
  9217 482D D647 8641 44BA 10D8 83F4 9FBF 1144 D9B3

Acknowledgements

- Useful feedback received from Andrew Rice and Robert Watson. Suggestion of BSL scheme from Markus Kuhn.
- Bluetooth logo from open icon library
- Phone picture from openclipart
- Computer laptop from open icon library
- Copy from open icon library
- Google Authenticator logo from Google.
- Router image from open clip art
- Monkeysphere logo based on wikimedia commons image
- Networked server picture from open icon library
- Paste icon from open icon library
- Green seal from open clip art
- Key diagram based on one from open clip art